Neural Network as Complex Adaptive System and Its Emergent Benign Loss Landscape

Shuai Li*
lishuai918@gmail.com

Abstract

We formally define neural networks as *complex adaptive systems* through measure transport and statistical learning theory. For a class of loss functions including *hinge loss*, we prove that when a certain condition on *order parameters* of the system hold, the risk of the system has a loss landscape where all local minima are global minima with zero risk, and its Hessian's *eigenspectrum* (distribution of eigenvalues) is symmetric w.r.t. zero. The zero risk, eigenspectrum symmetry and the condition on order parameters are found to characterize a VGG net trained on CIFAR10 in experiments. The results logically lead to variance normalization in techniques like Batch Normalization.

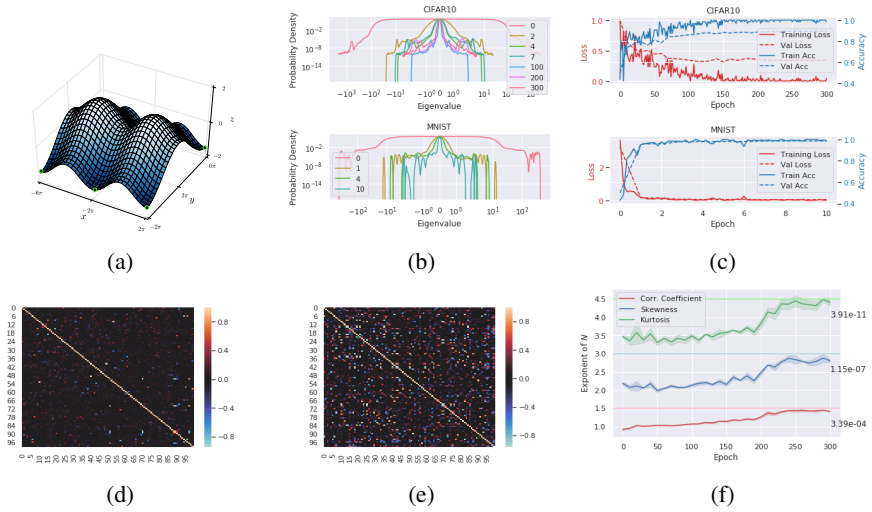
1 Introduction

Deep neural networks (NNs) are biologically inspired families of algorithms that model the perception intelligence, e.g., image recognition, in an astoundingly successful way. For example, the state-of-the-art models surpass human-level image classification performance [36]. However, the mathematical principle that underlies NNs is still not well understood, e.g., its representation power, optimization and generalization abilities. By integrating ideas from from high dimensional statistics, complex adaptive systems, and non-convex optimization, this work aims to study the optimization of NNs.

The biological NNs are commonly held as a class of complex systems known as *complex adaptive systems* [24, Chp 8]. A complex adaptive system consists of a vast number of relatively autonomous microscopic agents (such as cells, neurons, or individuals) that interact locally in a coherent coordinated way against feedback signals and induce an *emergent* macroscopic phenomenon [77] known as *self-organizing* [39]. The macroscopic emergent behavior of the system normally depends on some *order parameters*, e.g., the strength of the interaction between agents, and the behaviors would be completely different with different order parameters. Some introductory materials on more familiar water molecule complex systems are given in appendix B.3. Artificial NN has been inspired by the biological NNs; it is natural to ask the question of whether there exists a formal mathematical characterization of the biological NN adaptive system [20; 56]. We formally frame NNs as complex adaptive systems in this work.

The optimization problem of NNs has been an unsolved issue for decades. With some risk of over-generalization, existing works tend to either follow a reductionism philosophy to study parts in isolation by focusing on shallow NNs [3; 13; 14; 21; 30; 34; 52; 74; 79; 80; 81; 85; 95], or sidestep nonlinearity by studying linear NNs [35; 49; 53; 61; 62; 63; 72; 76; 93], absorbing nonlinearity into kernel space [22; 41] or unrealistic mean field conditions [15; 16; 67]. As a result, the shallow NNs results do not apply to deep NNs; linear NNs results do not apply to nonlinear NNs; results by kernel space ask the layer width to be either polynomials of training sample size (which is order-of-magnitude larger than NNs in practice) because kernel space is the space that embeds samples, or literally infinity; and mean field results are unrealistic [16]. We discuss related works in details in appendix C. Nonlinear interaction, hierarchy, and complexity are defining characteristics of the NN complex systems. By directly characterizing them, the hierarchically organized complexity and nonlinearly

*Preprint. Author information is incomplete.



(a) (b) (c)

(d) (e) (f)

Figure 1: **(a)** Illustration of saddle points and a loss landscape where local minima are global minima; the white dot in the middle, and black dots at the corners are a saddle point, and local/global minima respectively. **(b)** The numerically computed eigenspectrum of Hessian of 4-layer LeNet and 12-layer VGGNet (cross entropy loss replaced by hinge loss) being trained on modified MNIST and CIFAR10 datasets for binary classification. The numbers in the legends are epoch numbers in training. *Note that the axes are in logarithmic scale.* The eigenspectrum is computed with the Lanczos spectrum approximation algorithm [29; 55; 66] (details in appendix G). **(c)** The loss and accuracy curves of the experiments. **(d) (e)** are the correlation coefficient (CC) matrices of Hessian entries at the beginning and the end of training respectively of the VGG net. We uniformly randomly subsample a 100×100 CC submatrix from the original CC matrix for a high resolution for clarity. The diagonal line is the variance, which is 1 as the result of normalization by standard deviation. **(f)** The average order parameter, i.e., coupling set size, of all Hessian entries characterized through the average counts of non-zero statistics, i.e., CC, skewness, and excess kurtosis, in term of the exponent of Hessian dimension N during training. *The non-zero CC of each entry is equivalent to its coupling set size.* The relationship between the counts and coupling set sizes is explained in section 5.1. The transparent regions are standard deviations of the average counts of bootstrap samples that we used in the experiments. The light color straight lines and the annotated numbers in the rightmost indicate the ratio between the non-zero count and the overall number of the statistics (both zero and non-zero).

induced sparsity become the cornerstones to characterize the behaviors of NNs *as it is* used in practice without modification in this work.

Contributions and outlines. Our investigation has led to the following results.

- In section 2, we give a formal definition of neural networks as *complex adaptive systems*, and it characterizes interaction among neurons as *statistical dependency*.
- In section 3 and 4, for *large-size nonlinear deep* NNs with a class of loss functions $\mathcal{L}_0$, including the *hinge loss*, we present theorem 2 that shows under a condition on order parameters and some regularity conditions on statistical dependency, NNs have an *emergent benign loss landscape* characterized as follows: eigenvalues of the empirical risk's Hessian are distributed symmetrically w.r.t. zero, and a lower risk tends to concentrate eigenvalues more towards zero. It implies if the conditions are maintained throughout training, all local minima are global minima with zero risk. In addition, we experimentally show that the landscape above characterizes a VGG NN trained on CIFAR10 and a LeNet trained on MNIST, throughout training, as shown in fig. 1 (b)(c). Detailed analysis on the experiment results can be found in section 4.2.
- We also experimentally show that the condition on order parameters characterizes practical NNs above. Each Hessian entry is a random variable, and it could correlate with a *subset* (denoted as the *coupling set* of this

entry) of all the remaining Hessian entries. An *order parameter* is the coupling set *size*, or, the *count* of elements in the subset. The plural form, i.e., order parameters, is normally used because different entries may have different counts. Informally, the condition on order parameters is a threshold on the counts, which leads to sparse correlations between entries. We estimate the coupling set sizes and validate the correlations are indeed sparse, as shown in fig. 1 (d)(e)(f), and detailed analysis can be found in section 5.1. In section 5.2, we show that the study also logically leads to variance normalization in techniques like Batch Normalization [40].

Implications. Overall, we believe that the theoretical and experiment results in this work have initially hacked a pathway to understand and control the optimization behaviors of NNs by outlining an initial, crude, yet relatively complete picture of such behaviors of VGG and LeNet style NNs trained with hinge loss on CIFAR10 and MNIST datasets. *The discussion on normalization in section 5.2 is mostly to demonstrate such a picture.* However, many problems are still left unsolved. The underlying reason why the condition on order parameters is held has not been theoretically characterized. Our results only apply to relatively simple loss functions, e.g., hinge loss, and do not include quadratic loss, or cross entropy loss, though we hypothesize that the fundamental idea would hold for any loss functions. Training might be trapped in saddle points without reaching minima. The result does not involve generalization (different local minima are of different generalization ability, leading to different test errors), which is a challenge as significant as optimization.

Notations. To index entries of a matrix, we denote $\mathbb{J} = [N] := \{1, \ldots, N\}$, the ordered pairs of indices by $\mathbb{I} = \mathbb{J} \times \mathbb{J}$. For $\alpha \in \mathbb{I}, A \subseteq \mathbb{I}, i \in \mathbb{J}$, given a matrix $\boldsymbol{W}$, w_α, $\boldsymbol{W}_A$ denotes the entry at α, the vector that consists of entries at A, respectively. $\kappa(\alpha, \beta)$ or $\kappa(w_\alpha, w_\beta)$ denote covariance between w_α, w_β. A more complete summary is given in appendix A.

2 Neural Network as Complex Adaptive System

In this section, we reinterpretation a NN as a complex adaptive system that learns to classify samples through feedback given from training samples. The reinterpretation may mot be obviously significant on its own, but it would bring physical intuitions to the conditions that induce a benign loss landscape later. In the reinterpretation, a key difference from the commonly used definition of NNs is that NNs are formally formulated as *hierarchical measure transport maps*. The formalism is the foundation of the landscape analysis.

2.1 Measure transport

NNs are widely taken as universal function approximators, and since a probability distribution is a function, NNs can also approximate probabilities. However, underlying the widely perceived picture described, a richer structure exists: the probability distribution that a NN approximates is the probability measure transported from the data to the hidden neurons. To describe the structure, we introduce *measure transport*.

Given measurable spaces $(E, \mathcal{E})$, $(F, \mathcal{F})$, a measurable mapping $T : E \to F$, and a measure $\mu : \mathcal{E} \to [0, +\infty]$, the *pushforward probability measure* of μ by T is defined to be the measure $\nu : \mathcal{F} \to [0, +\infty]$ given by

$$\nu(B) = \mu(T^{-1}(B)), B \in \mathcal{F}$$

The mapping T is often called a *transport map*. Without further delving into the formalism of probability measure theory, given a random variable (r.v.) X with a probability measure μ, a measurable mapping T, and the r.v. $Y := T(X)$ with probability measure ν obtained by applying the measurable mapping T on X, we say the *measure transported* from X to Y is the pushforward probability measure ν of μ by T. An illustration of measure transport is given in appendix B.1.

2.2 Statistical learning

With measure transport, we are ready to introduce the learning problem of supervised NNs. Assume an instance space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the space of input data, and $\mathcal{Y}$ is the label space. $Z := (X, Y)$ are the r.v.s with an

unknown distribution μ, from which we draw samples. We use $S_m = \{z_i = (\boldsymbol{x}_i, y_i)\}_{i=1}^m$ to denote the training set of size m whose samples are drawn independently and identically distributed (i.i.d.) by sampling Z. Given a loss function l, the goal of learning is to identify a function $T : \mathcal{X} \mapsto \mathcal{Y}$ in a hypothesis space (a class $\mathcal{T}$ of functions) that minimizes the expected risk

$$R(l \circ T) = \mathbb{E}_{Z\sim\mu}\left[l\left(T(X), Y\right)\right],$$

Since μ is unknown, the observable quantity serving as the proxy to the true risk $R(f)$ is the empirical risk

$$R_m(l \circ T) = \frac{1}{m}\sum_{i=1}^{m} l\left(T(\boldsymbol{x}_i), y_i\right) \tag{1}$$

Thus, given a mapping T, e.g., a NN, **the risk function R_m measures the empirical discrepancy between the probability measure $T(X)$ transported from X and the probability measure of Y.** Thus, the learning of NNs is to learn parametric approximation of conditional measure of Y through minimizing a surrogate risk.

2.3 Neural networks as hierarchical measure transport maps

Not only the ultimate output of a NN, $T(X)$, is a r.v., so are the activation in the intermediate layers of a NN. We characterize the finer stochastic structure in NNs using Multiple Layer Perceptron (MLP) in this section.

Denote g as the activation function, $\boldsymbol{W}$ the linear transform, X the input, $\hat{H}$ the output activation, of a layer of a NN. We have

$$\begin{aligned} \hat{H} &:= g(\boldsymbol{W}X) = \text{ReLU}(\boldsymbol{W}X) \\ \text{ReLU}(\boldsymbol{W}X) &:= \boldsymbol{W}X \odot H = \text{dg}(H)\boldsymbol{W}X \\ H &:= \mathbf{1}^{\boldsymbol{W}X}, \mathbf{1}_i^{\boldsymbol{W}X} = \begin{cases} 1, \text{ if } (\boldsymbol{W}X)_i > 0 \\ 0, \text{ otherwise} \end{cases} \end{aligned} \tag{2}$$

where $\odot$ denotes Hadamard product (element-wise multiplication); ReLU denotes the Rectified Linear Unit; $\text{dg}(H)$ is the diagonal matrix whose diagonal is H. We use the widely used activation function ReLU as an example. The derivation in this paper can be trivially generalized to activation functions like Swish [71], or variants of ReLU [17; 36] etc. Generalizing to classic activation functions like Sigmoid may need more involved matrix calculus, but is fundamentally similar.

$\hat{H}, H$ are both r.v.s created by transporting measure of X through map $g(\boldsymbol{W}X), \mathbf{1}^{\boldsymbol{W}X}$ respectively, thus they are *stochastic* (which is a tautology). Repeat the process, and **we can write an MLP with a single output neuron as the following transport map**

$$T(X;\boldsymbol{\theta}) = X^T \overrightarrow{\prod}_{i=1}^{L-1} \boldsymbol{W}_i \text{dg}(H_i)\boldsymbol{\alpha} \tag{3}$$

where $\boldsymbol{\alpha}$ is the weights of the last layer of a NN, and is a vector; i is the layer index; $\text{dg}(H_i)$ is the diagonal matrix whose diagonal is H_i; $\boldsymbol{\theta}$ is the parameters of T, a.k.a. $\{\boldsymbol{W_i}\}_{i=1,\dots,L-1}, \boldsymbol{\alpha}$; L is the layer number; $\overrightarrow{\prod}_{i=1}^{L-1} \boldsymbol{W}_i \text{dg}(H_i)$ denotes $\boldsymbol{W}_1 \text{dg}(H_1) \dots \boldsymbol{W}_n \text{dg}(H_n)$.

2.4 Neural networks as complex adaptive systems

We are ready to describe NNs as complex adaptive systems by mapping its constitution to the formal object defined previously as in table 1.

A NN $T(X;\boldsymbol{\theta})$ is a probability estimation *complexy adaptive system* that is composed by a vast number of *microscopic units* (the thresholding units $\text{ReLU}(\boldsymbol{W}X)_i$, where i indexes a single neuron) that self-organize *hierarchically*. The weights $\{\boldsymbol{W}\}_{i=1,\dots,L-1} \cup \{\boldsymbol{\alpha}\}$ are its *state*. The state of the system adapts to minimize the empirical discrepancy $R_m(l \circ T)$ between the estimated probabilities of Y and its true probabilities according to the *feedback* given by the *feedback source* S_m.

Table 1: The constitution of neural networks as complex adaptive systems.

Constitution	**Formal Objects**		
Microscopic unit	Thresholding neuron:	$\text{ReLU}(\boldsymbol{W}X)_i$	(eq. (2))
Organizing principle	Hierarchy:	$\overrightarrow{\prod}_{i=1}^{L-1}\boldsymbol{W}_i\text{dg}(H_i)$	(eq. (3))
State of the system	Weights:	$\{\boldsymbol{W}\}_{i=1,\ldots,L-1}\cup\{\boldsymbol{\alpha}\}$	
Feedback	Empirical risk:	$R_m(l\circ T)$	(eq. (1))
Feedback source	Training samples:	S_m	
Interaction between Units	Statistical dependency	κ	
Order parameter	Coupling set size:	$\lvert\mathcal{N}\rvert$	(cond. 3.1)
Emergent behavior	Symmetric eigenspectrum		(theorem 2)

A novel insight in the reinterpretation is how the *interaction* between units is characterized. Such characterization is vital because it is the interaction that leads to emergent behaviors of complex systems. The characterization would decide whether it is obvious or elusive to identify the corresponding emergent behavior. We identify the *statistical dependency*, e.g., covariance, denoted as κ, as such characterization, which helps us characterize the phenomena in the following sections, i.e., the *order parameters* $|\mathcal{N}|$ (defined in section 3.4) and the *emergent symmetric eigenspectrum* (distribution of eigenvalues) of NN Hessian. The key message of the upcoming sections is: if a NN is controlled to be in a disordered state, where correlations between neurons are sparse, the sparse correlations would allow neurons to adapt freely and collectively to minimize the empirical risk.

3 Statistical dependency as interaction and order parameter as coupling set size

We are to study the loss landscape of NNs. To study the loss landscape, we need to study the Hessian of the risk. The hierarchical organizing principle of NNs results in the fact that the entries of Hessian can be interpreted as *how the weight change would be propagated to the risk change by activation paths that connect neurons from lower layers to higher layers*. The phenomenon has appealing intuitive interpretation: it resonates with the conjecture that the patterns in NNs are coded hierarchically, where neurons in lower layers encode low-level patterns, e.g., textures, neurons in higher layers encode semantic patterns, e.g., objects, and the *activation paths* connecting neurons in the lower to higher layers encode the *composition relationship* between these patterns. The *interaction* among the microscopic neuron units is characterized as *statistical dependency* between entries. The *order parameter* that induces a benign loss landscape is the count of Hessian entries that each Hessian entry is statistically correlated with, or as later defined, the size of the *coupling set* of each entry.

3.1 The restricted loss function class $\mathcal{L}_0$

Our goal is to investigate the fundamental principle that makes $R_m(T)$ in eq. (1) tractable when T is a NN. To do this, we study the risk landscape by studying the Hessian of $R_m(T)$ of a particular class of loss functions. To motivate the class of losses, we calculate the Hessian of eq. (1) when T is given by eq. (3). The choice of a NN with a single output neuron simplifies the problem considerably: it reduces a weighted summation of a great number of matrices to a single matrix, though we do not want to digress too much to elaborate the simplification. The calculation gives

$$\frac{d^2}{d\boldsymbol{\theta}^2}l(T(\boldsymbol{x};\boldsymbol{\theta}),y)=l''(T(\boldsymbol{x};\boldsymbol{\theta}),y)\frac{d}{d\boldsymbol{\theta}}T(\boldsymbol{x};\boldsymbol{\theta})\frac{d}{d\boldsymbol{\theta}}T(\boldsymbol{x};\boldsymbol{\theta})^T+l'(T(\boldsymbol{x};\boldsymbol{\theta}),y)\frac{d^2}{d\boldsymbol{\theta}}T(\boldsymbol{x};\boldsymbol{\theta}) \tag{4}$$

Note that $\boldsymbol{x}, y$ are realizations sampled from X, Y now.

To further simplify the problem, we restrict the class of loss functions to class $\mathcal{L}_0$.

Definition 3.1 (Function class $\mathcal{L}_0$)**.** *We study the class $\mathcal{L}_0$ of functions l and the class of NNs T, such that for $l \in \mathcal{L}_0$, it satisfies:*
1. *$l : \mathbb{R} \to \mathbb{R}^+$, when y is taken as a constant;*
2. *l is convex;*
3. *the second order subderivatives $\frac{d^2}{d\boldsymbol{x}^2} l$ is zero;*
4. *$\min_x l(\boldsymbol{x}, y) = 0$, while for T it satisfies: $\dim(T(\boldsymbol{x}; \boldsymbol{\theta})) = 1$.*

The restriction allows us to study the most critical aspect of the risk function of supervised NNs by making the first term on the right side of the equation in eq. (4) zero, while the second term a single matrix. The class of l includes important loss functions like the **hinge loss** $\max(0, 1 - \hat{H}_L Y)$, and the absolute loss $|\hat{H}_L - Y|$, which were first studied in Choromanska et al. [15].

3.2 Hessian of neural networks with loss functions in the function class $\mathcal{L}_0$

Denote $\boldsymbol{H}_{pq}$ as in eq. (5) (the equation should be read vertically). The operands of the Kronecker product are shortened as $\tilde{\boldsymbol{\alpha}}_q, \tilde{\boldsymbol{W}}_{p\sim(q-1)}, \tilde{\boldsymbol{x}}_{p-1}^T$ on the right most side in eq. (5), which corresponds to the physical meaning that is explained in section 3.3 later and illustrated in fig. 2a. We would have the Hessian of l (eq. (4)) as in eq. (6).

The calculation is deferred to appendix D.2.

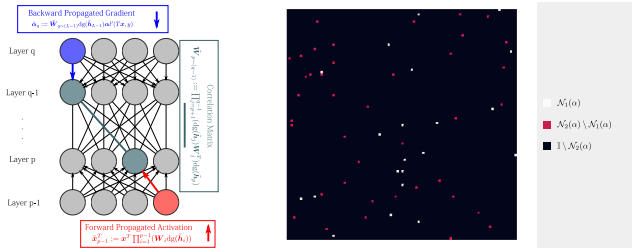
$$\begin{aligned} \boldsymbol{H}_{pq} =& l'(T\boldsymbol{x}, y)\mathrm{dg}(\tilde{\boldsymbol{h}}_q)\overrightarrow{\Pi}_{k=q+1}^{L-1}(\boldsymbol{W}_k\mathrm{dg}(\tilde{\boldsymbol{h}}_k))\boldsymbol{\alpha} && = \tilde{\boldsymbol{\alpha}}_q \\ & \otimes \overleftarrow{\Pi}_{j=p+1}^{q-1}(\mathrm{dg}(\tilde{\boldsymbol{h}}_j)\boldsymbol{W}_j^T)\mathrm{dg}(\tilde{\boldsymbol{h}}_p) && \otimes \tilde{\boldsymbol{W}}_{p\sim(q-1)}^T \\ & \otimes \boldsymbol{x}^T\overrightarrow{\Pi}_{i=1}^{p-1}(\boldsymbol{W}_i\mathrm{dg}(\tilde{\boldsymbol{h}}_i)) && \otimes \tilde{\boldsymbol{x}}_{p-1}^T \end{aligned} \tag{5}$$

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{H}_{12}^T & \dots & \boldsymbol{H}_{1L}^T \\ \boldsymbol{H}_{12} & \boldsymbol{0} & \dots & \boldsymbol{H}_{2L}^T \\ \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{H}_{1L} & \boldsymbol{H}_{2L} & \dots & \boldsymbol{0} \end{bmatrix} \tag{6}$$

Intuitive understanding of eq. (5) is provided in section 3.3 shortly. For now, we just need to focus on the form of eq. (6), and ignore the complicated details, e.g., Kronecker products. Notice that $\boldsymbol{x}$ and $\tilde{\boldsymbol{h}}$ (subscripts omitted) are both realizations of r.v.s X and H. Thus, **Hessian $\boldsymbol{H}$ is a matrix obtained by applying a complicated transport map on X. The matrix is a symmetric random matrix**, the randomness of which comes from data. It implies the Hessian $\boldsymbol{H}$ of $R_m(T)$ is a random matrix created by summing random matrices, each of which is a gradient l' multiplies the Hessian of T at example z_i. Its eigenspectrum can be studied through random matrix theory [82]. Thus, we can have a complete characterization of the landscape of $R_m(T)$ of eq. (1).

3.3 Entries of NN Hessian as the influence of activation paths on risk

The Hessian present in eq. (5) is rather complicated. But it has a rather intuitive physical meaning. **The entry of the Hessian characterizes how the weight change would be propagated to the risk change by activation paths that connect neurons from lower layers to higher layers.** As illustrated in fig. 2a, when an example $\boldsymbol{x}$ is propagated through a NN, and the gradient calculated is propagated back afterwards, for neurons in layer $p-1$, we have *activation* $\tilde{\boldsymbol{x}}_{p-1}^T$, shown as the red neuron in fig. 2a; for neurons in layer q, we have *gradient* $\tilde{\boldsymbol{\alpha}}_q$, shown as the blue neuron in fig. 2a; for neurons in layers $p, q-1$ respectively, if we treat everything between $\mathrm{dg}(\tilde{\boldsymbol{h}}_{q-1})$ and $\mathrm{dg}(\tilde{\boldsymbol{h}}_p)$ in $\tilde{\boldsymbol{W}}_{p\sim(q-1)}$ as a matrix whose entries are all one, the expectation of $\tilde{\boldsymbol{W}}_{p\sim(q-1)}$ is the *covariance matrix* between $\tilde{\boldsymbol{h}}_{q-1}$ and $\tilde{\boldsymbol{h}}_p$, i.e., $\mathbb{E}_{z\sim\mu^z}[\tilde{\boldsymbol{W}}_{p\sim(q-1)}]$. It could be understood as the strength of correlations between the neurons at layer p and $q-1$ — are these neurons tend to be activated simultaneously, or otherwise? — shown as the green neurons in fig. 2a. Informally, if we connect these neurons, we would obtain *activation paths* in the NN from layer $p-1$ to q, shown as the green lines connecting neurons in fig. 2a. The picture is clearer if we write $\boldsymbol{H}_{pq}$ in the case where operands in the Kronecker product only have one matrix respectively, i.e., $\partial^2 l(T\boldsymbol{x}, y)/(\partial \mathrm{vec}\boldsymbol{W}_p \partial \mathrm{vec}\boldsymbol{W}_q) = \boldsymbol{H}_{pq} = l'(\cdot)\mathrm{dg}(\tilde{\boldsymbol{h}}_q)\boldsymbol{W}_k\boldsymbol{\alpha} \otimes \mathrm{dg}\tilde{\boldsymbol{h}}_j\boldsymbol{W}_j^T\mathrm{dg}(\tilde{\boldsymbol{h}}_p) \otimes \boldsymbol{x}^T\boldsymbol{W}_i\mathrm{dg}(\tilde{\boldsymbol{h}}_i)$. Its entries



(a) (b)

Figure 2: **(a)**: Illustration of entries of the Hessian of NNs. **(b)** The illustration of the size of the coupling set defined in cond. 3.1. The image represents the square matrix $\boldsymbol{W}$ defined in eq. (7) with its index set $\mathbb{I}$ color coded. The condition roughly states that for the two set $\mathcal{N}_1$ (color coded by white pixels) and $\mathbb{I} \setminus \mathcal{N}_2$ (color coded by black pixels) of entries of $\boldsymbol{W}$, the correlations between $\boldsymbol{W}_{\mathcal{N}_1}$ and $\boldsymbol{W}_{\mathbb{I}\setminus\mathcal{N}_2}$ varnish. Meanwhile, strong correlations are allowed in $\mathcal{N}_2$ (same as $\mathcal{N}$). Informally, each coupling set is a set of entries where the patterns coded are correlated, e.g., legs, and arms of persons. Quantitatively, suppose that we have a 10-layer NN with 100 neurons in each layer, then the Hessian $\boldsymbol{H}$ would be of dimension $N = 10^9$ (due to Kronecker product), then $N^{1/2}$ is $10^{4.5}$. That is, that the coupling set could have almost $10^{4.5}$ entries. **That means for each entry, it could have almost $10^{4.5}$ number of correlated entries among only 10^3 neurons.**

are given below, where the layer index p, q, i, j, k is moved up as upper indexes for clarity.

$$\partial^2 l(T\boldsymbol{x}, y)/(\partial w^p_{ac}\partial w^q_{df}) = \sum_b l'(\cdot)h^q_a w^k_{ab}\alpha_b\cdot(h^j_c w^i_{cd}h^p_d)\cdot\sum_e x_e w^i_{ef}h^i_f = \sum_{b,e} l'(\cdot)\alpha_b w^k_{ab}h^q_a\cdot h^j_c w^i_{cd}h^p_d\cdot w^i_{ef}h^i_f\cdot x_e$$

It is a summation of multiplication of the weights, the hidden variables H of each layer (with corresponding the weights of the layers that are differentiated out) and the data X, i.e., activation paths that connect data to hidden variables in each layer in the network from bottom to top through weights. Change in $w^p_{ac}w^q_{df}$ would be propagated to l by all the paths involved in $\partial^2 l(T\boldsymbol{x}, y)/(\partial w^p_{ac}\partial w^q_{df})$, weighted by their respective strength/values.

3.4 Coupling set size as the order parameter that controls emergent benign loss landscape

Notice that the adaptive goal of a NN is to minimize the empirical expectation of risk eq. (1). *If in a large enough NN, the risk changes induced by weights do not contradict each other statistically, some weights could always be found to be adjusted to minimize the risk, thus the expected risk would decrease.* This is the observation that leads to a formal characterization of the emergent benign loss landscape of NNs built by a vast amount of relatively autonomous neurons (capable of making non-contradicting changes). It leads to the order parameter that characterizes the statistical correlations between NN Hessian entries. And if the number is low, the risk can be further minimized, as characterized in the following.

Let $\boldsymbol{A} := \mathbb{E}[\boldsymbol{H}], \boldsymbol{W}/N := \boldsymbol{H} - \boldsymbol{A}$ (previews of definitions in eq. (7) presented later). **Note that $\boldsymbol{W}$ is reused.** The statistical dependency is the *covariance* of entries of $\boldsymbol{W}$. The order parameter is the count of entries in $\boldsymbol{W}_{\mathbb{I}\setminus\{\alpha\}}$ that each entry $w_\alpha, \alpha \in \mathbb{I}$ is correlated with, i.e., the *size* $|\mathcal{N}|$ *of the coupling set* defined below. In addition to some regularity conditions that would be specified in the next section, if $|\mathcal{N}| < N^{1/2}$, where N is the dimension of Hessian $\boldsymbol{H}$, a NN would have the benign loss landscape. And formally, the condition is given below.

Condition 3.1 (Diversity)**.** *There exists $\mu > 0$ such that the following holds: for every $\alpha \in \mathbb{I}$, there exists two sets $\mathcal{N}_k = \mathcal{N}_k(\alpha), k = 1, 2$ such that $\alpha \in \mathcal{N}_1 \subset \mathcal{N}_2 = \mathcal{N} \subset \mathbb{I}, |\mathcal{N}| \leq N^{1/2-\mu}$ and $\forall\beta \in \mathcal{N}_1$ and $\forall\gamma \in \mathbb{I} \setminus \mathcal{N}$, the following holds*

$$\kappa(w_\beta, w_\gamma) = 0$$

We call the set $\mathcal{N}$ of α the **coupling set** *of α.*

Cond. 3.1 divides entries of $\boldsymbol{W}$ into two sets, and states that the covariances between $\boldsymbol{W}_{\mathbb{I}\setminus\mathcal{N}}$ and $\boldsymbol{W}_{\mathcal{N}_1}$ varnish. We note that the condition is a simplified one under an extra condition to introduce the core idea, and to serve as a pedagogical example for the actual condition (cond. D.5) used in the proof that involves statistics of all orders instead of just covariances. The extra condition is described in the next section. To give an intuitive feeling about the size of $\mathcal{N}$, we illustrate it with fig. 2b.

4 Emergent benign loss landscape: theorem and proof sketch

We have introduced the key order parameter, i.e., the size of coupling sets, that controls the emergent behavior of NNs. It is the most critical characteristics of the NN complex adaptive system. But some issues are still left not characterized. When a NN should be considered complex enough to manifest such a behavior? What if a particular covariance goes to infinity? These regularity conditions are also important to NN systems. In this section, we describe these conditions in section 4.1, present the emergent landscape as theorem 1 in section 4.2, and sketch its proof in section 4.3.

4.1 Regularity conditions for the emergent benign loss landscape

We introduce the conditions under a simpler case where the sample size m is large in eq. (1). $\boldsymbol{H}$ is a random matrix, since it is a function of a random variable, i.e., the input sample. The conditions characterize the statistical properties of entries of $\boldsymbol{H}$. Let

$$\boldsymbol{A} := \mathbb{E}[\boldsymbol{H}], \frac{1}{\sqrt{N}}\boldsymbol{W} := \boldsymbol{H} - \boldsymbol{A}, \mathcal{S}[\boldsymbol{R}] := \frac{1}{N}\mathbb{E}_{\boldsymbol{W}}[\boldsymbol{W}\boldsymbol{R}\boldsymbol{W}] \tag{7}$$

where the expectation in $\mathcal{S}$ is taken w.r.t. $\boldsymbol{W}$ while keeping $\boldsymbol{R}$ fixed — it is a linear operator on the space of matrices. **Note that $\boldsymbol{W}$ is reused.** We state the conditions as follows.

Condition 4.1 (Large Sample Size)**.** *$m \to \infty$, where m is the sample size in the empirical risk.*

This additional large-sample-size requirement (cond. 4.1) greatly simplifies conditions, and enables us to describe the conditions with more familiar concepts to the community. The theorem is proved *without* the further requirement. Cond. 4.4 3.1 in this section are simplified version of cond. D.2 D.5 under the further condition cond. 4.1. Why cond. 4.1 simplifies conditions can be found in appendix D.4. The theorem is more generally held under cond. 4.2 4.3 D.2 4.5 D.5. The conditions are from Erdos et al. [25], the work that proposes a key technique we use. **It suffices to understand for now that when the Hessian of a NN satisfies cond. 4.1 4.2 4.3 4.4 4.5 3.1, we have theorem 1.** The key idea is not lost under this setting.

Condition 4.2 (Complexity)**.** *The dimension N of the Hessian $\boldsymbol{H}$ is sufficiently large in the following sense. This is an informal version of a non-asymptotic type condition, where a large N gives a small error bound, and the full exposure is deferred to theorem 2 in appendix D.3. Given the Hessian $\boldsymbol{H}$, the resolvent matrix $\boldsymbol{G}(z)$ (c.f. definition D.2) of $\boldsymbol{H}$ uniquely determines the probability distribution of eigenvalues of $\boldsymbol{H}$ at $\Im z$ (c.f. lemma D.1), where z is a complex number, and $\Im z$ denotes the imaginary part of z. $\boldsymbol{M}(z)$ is a matrix obtained later that has a symmetric eigenspectrum. The closeness of $\boldsymbol{G}$ and $\boldsymbol{M}$ is given by a high-probability bound as follows. Under cond. 4.1 4.3 4.4 4.5 3.1, for any $\epsilon > 0$, for all $D > 0$, we have*

$$P(|trace\left(\boldsymbol{B}(\boldsymbol{G}(z) - \boldsymbol{M}(z))\right)| \leq ||\boldsymbol{B}||N^{\epsilon}/\left((1+|z|)^2 N\right)) \geq 1 - CN^{-D} \tag{8}$$

where $\boldsymbol{B}$ is an arbitrary deterministic matrix, $C > 0$ is a constant depending on D, ϵ and constants in cond. 4.4 3.1, $\epsilon > 0$ can be chosen sufficiently small.

Cond. 4.2 states that the size of the NN is measured in a hierarchical way that involves both depth and width of the NN, mirroring the organizing principle of the system. Compared with existing works that characterize the NN size solely with layer width, and require astronomical [22], or literally infinite [41] width, the large N limits of the Hessian can be achieved by adding more layers. Assuming each layer has n neurons,

and there are L layers, N would be n^4L, which is a huge number even for small n, c.f. the example in fig. 2b. Technically, the bound roughly states that the probability that the eigenspectrum of the Hessian $\boldsymbol{H}$ of NNs is symmetric grows at a rate of $1 - CN^{-D}$ w.r.t. N (note that $C > 0, D > 0$ are constants). As in generalization bounds in statistical learning theory, the exact value of the bound is not critical, while the parameters that influence the rate of the bound are more relevant. The bound implies that the benign loss landscape manifests at large N. As the experiments in section 4.2 shows, the behaviors characterized by the bound, i.e., symmetric eigenspectrum, manifest at finite N of practical NNs that are considered small, e.g., LeNet.

Condition 4.3 (Zero Mean)**.** $\boldsymbol{A} = \boldsymbol{0}$

Cond. 4.3 states that the mean strength of the entries are zero (note the strength could be negative). The condition is not required in a fundamental way, more discussion can be found in remark D.4.

Condition 4.4 (Boundedness)**.** *1)* $\exists \mu, \sigma \in \mathbb{R}, \forall \alpha \in \mathbb{I}, \mathbb{E}[|w_\alpha|] \leq \mu, \mathbb{E}[|w_\alpha|^2] \leq \sigma$. *2) For any* $\epsilon > 0, \exists C_1, C_2(\epsilon) \in \mathbb{R}$*, where* $C_2(\epsilon)$ *depends on* ϵ*,* $|||\kappa|||_2^{iso} \leq C_1, |||\kappa|||_2 := |||\kappa|||_2^{av} + |||\kappa|||_2^{iso}, |||\kappa|||_2 \leq C_2(\epsilon)N^\epsilon$

Cond. 4.4 are regularity conditions that require values to be bounded. Such assumptions are required for rigor to prevent singularities that are involved with the infinity. Specifically, it states that 1) the mean and variance of w_α are bounded; 2) covariances between $\boldsymbol{W}_\mathbb{I}$, i.e., all entries in $\boldsymbol{W}$, are bounded. The bound $|||\kappa|||_2 \leq C_2(\epsilon)N^\epsilon$ asks for a non-uniform bound w.r.t. N: since $|||\kappa|||_2$ is required to be smaller than $C(\epsilon)N^\epsilon$, its upper bound can grow with N. $||||^{\text{av}}$ and $||||^{\text{iso}}$ are norms that characterize the largest weighted average of all the covariances. Their exact forms are of minor interests here, and can be found in appendix D.1.1. **In any NNs of finite size, these norms are always finite, and thus the conditions hold.** The relationship between the constants C_1, C_2 and the high probability bound in the complexity cond. 4.2 is given in appendix D.7 for interested readers.

4.2 Theorem: the emergent benign loss landscape

We defer cond. 4.5 to section 4.3, for that it needs more context, and present the theorem first in this section.

Theorem 1. *Let* $R_m(T)$ *be the risk function (defined at eq. (1)) of a NN* T *having only one output neuron (defined at eq. (3)) with a loss function* l *of class* $\mathcal{L}_0$ *(defined in section 3.1). If the Hessian* $\boldsymbol{H} \in \mathbb{R}^{N \times N}$ *(c.f. eq. (6)) of* $R_m(T)$ *satisfies conditions 4.2 4.3 D.2 4.5 D.5, then for* $R_m(T)$*,*

a) all local minima are global minima with zero risk.

b) all non-zero-risk stationary points are saddle points with half of the non-zero eigenvalues negative.

c) A constant $\lambda_0 \in \mathbb{R}$ *exists, such that* $||\boldsymbol{H}||_2$ *is upper bounded by* $\mathbb{E}_m[l'(T(X), Y)]\lambda_0$*, where* $\mathbb{E}_m[l'(T(X), Y)]$ *is the empirical expectation of derivative* l' *of* l*. It implies the eigenspectrum of* $\boldsymbol{H}$ *is increasingly concentrated around zero as more examples have the zero loss (classified correctly in term of the surrogate loss) — because for a well-designed loss function, when the loss* $l(\boldsymbol{x}, y)$ *of an example is zero (property 4 of the loss function class* $\mathcal{L}_0$*), its derivative at* $\boldsymbol{x}$ *is zero, i.e.,* $l'(\boldsymbol{x}, y) = 0$*. Thus, the regions around the minima are flat basins.*

We explain the theorem with experiments that show the characterized loss landscape is presented in practical NNs. As well known, given a risk function, the landscape of the function is characterized by the *eigenspectrum (the distribution of eigenvalues)* of its Hessian: when all eigenvalues are positive, we are at a local minimum (black dots in fig. 1a); when all eigenvalues are negative, we are at a local maximum; otherwise, *we are at a saddle point, implying we have extra directions to further minimize the risk (the white dot in fig. 1a).* We train a LeNet [50] and a VGGNet [78] (the cross entropy loss replaced by the hinge loss) on the MNIST and CIFAR10 datasets modified for binary classification respectively, shown in fig. 1. Details of the experiments can be found in appendix G. The theorem states that the eigenspectrum of the Hessian is symmetric w.r.t. zero; that is, half of the non-zero eigenvalues of the Hessian are negative. The symmetry is shown in fig. 1b. Thus, every non-zero loss stationary point in the loss landscape can be further minimized. Yet, if all stationary points are saddle points, how can local minima be found? Theorem 1 c) states that as the risk decreases, the Hessian eigenspectrum would increasingly concentrate towards zero, and the loss landscape would become increasingly flat. As shown in fig. 1b, the eigenspectrums of the NN's Hessian during training concentrate increasingly around zero. The minima are achieved when all samples $\boldsymbol{x}$ are classified correctly — the risk is zero; in this case, the Hessian becomes a zero matrix, and all eigenvalues are zero. The training loss and accuracy curves are shown in fig. 1c.

4.3 Proof sketch: the mathematical principle behind the emergent benign loss landscape

4.3.1 Random matrices with correlations

To begin with, we introduce the *concentration of measure* phenomenon that makes the eigenspectrum of the random matrix symmetrically distributed w.r.t. zero. A slightly more detailed introduction can be found in appendix B.2. Given an eigenvector $\boldsymbol{t}$ of $\boldsymbol{W}$, we know that when we multiply it with $\boldsymbol{W}$, we have $(\boldsymbol{W}\boldsymbol{t})_i = \sum_k w_{ik}t_k$. Note that if $\{w_{ik}\}_{k=1,\dots,N}$ are i.i.d. r.v.s with zero means and symmetric distributions, $\{w_{ik}t_k\}_{k=1,\dots,N}$ tend to cancel each other out and spread around 0 with similar chances of being positive or negative, resulting in a close-to-zero symmetric eigenvalue distribution. The cancellation holds when $\{w_{ik}t_k\}_{k=1,\dots,N}$ are only weakly/sparsely correlated. The conditions introduced previously are to characterize such a phenomenon. **Under cond. D.1 D.2 4.5 D.3, Erdos et al. [25] obtain the eigenspectrum of a symmetric random matrix by solving a matrix equation**, termed *Matrix Dyson Equation (MDE)*

$$\boldsymbol{I} + (z - \boldsymbol{A} + \mathcal{S}[\boldsymbol{G}])\boldsymbol{G} = \boldsymbol{0}, \Im\boldsymbol{G} \succ \boldsymbol{0}, \Im z > 0, \tag{9}$$

where $\Im\boldsymbol{G}$ denotes the imaginary part of $\boldsymbol{G}$, $\Im\boldsymbol{G} \succ 0$ means $\Im\boldsymbol{G}$ is positive definite, and the rest of the symbols are defined at eq. (7). The details on its derivation and its relationship to eigenspectrum are explained in appendix D.3. Lastly, we describe the last cond. 4.5.

Condition 4.5 (Comparability)**.** $\exists 0 < c < C, \forall \boldsymbol{T} \succ \boldsymbol{0}, c\, N^{-1} tr\, \boldsymbol{T} \preceq \mathcal{S}[\boldsymbol{T}] \preceq CN^{-1} tr\, \boldsymbol{T}$.

Cond. 4.5 safeguards eq. (9) to be meaningful by preventing $\mathcal{S}[\boldsymbol{G}]$ from exploding $\boldsymbol{G}$ to infinite ($\preceq CN^{-1}\text{tr}\,\boldsymbol{T}$), or degenerate $\boldsymbol{G}$ to zero ($c\, N^{-1}\text{tr}\,\boldsymbol{T} \preceq$). This is also a regularity condition similar with that of cond. 4.4. More intuitions of the condition can be found in appendix D.1.2

4.3.2 Proof sketch of theorem 1: landscape analysis of NNs

The main content of the paper is actually the proof sketch. With all the key ideas introduced, we connect these ideas to present the proof sketch of theorem 1.

1. First, we show that NNs are hierarchical measure transport maps in section 2, so the Hessian $\boldsymbol{H}$ of a NN is a random matrix.
2. Second, in section 3.2, for a restricted class $\mathcal{L}_0$ of loss functions, by calculating the Hessian of any function in $\mathcal{L}_0$, we show that it is a real symmetric matrix. The actual calculation of the Hessian is carried out in appendix D.2.
3. Third, if the Hessian satisfies the sparse correlations characterized in cond. 4.1 4.2 4.3 4.4 4.5 3.1, as a result of the concentration of measure phenomenon (introduced in section 4.3.1), the eigenspectrum of $\boldsymbol{H}$ can be obtained by solving the Matrix Dyson Equation. MDE is formally introduced in appendix D.3.
4. Lastly, we can proceed to analyze the eigenspectrum of Hessian of NNs, which is carried out in appendix D.6. Though the eigenspectrum cannot be obtained in closed-form as semi-circle law in the case of Wigner matrix, we still can analyze it behaviors through MDE. In **theorem 3**, we prove the solutions to MDE is symmetric w.r.t. zero under the conditions of theorem 1: that is, given any positive eigenvalue λ, $-\lambda$ is also an eigenvalue. Since there are always non-zero eigenvalues by theorem 2, we always have negative eigenvalues. We break down the analysis of Hessian $\boldsymbol{H}$ into two cases: 1) for all training examples, at least one example (x, y) has non-zero loss value; 2) and all training examples are classified properly with zero loss values. In case 1), by theorem 3, $\boldsymbol{H}$ is always indefinite, and thus corresponds to a saddle point. In case 2), we are at global optima, where all losses are zero. Lastly, theorem 1.c is proved, which is straightforward.

5 The Neural Network complex adaptive system in practice

It seems that hierarchically organized complexity has made possible a fabulous nonlinear system to minimize empirical risk, and NNs are computational implementations of the system. In section 4.2, we have shown in

fig. 1a that the loss landscape in practical NNs behaves as characterized in theorem 1. As explained in section 3.4, the coupling set size is the order parameter that dictates the macroscopic behavior of the system. In section 5.1, we study the correlations between Hessian entries and estimate the coupling set size, and show that the condition on the order parameter also characterizes these NNs. Furthermore, we show in section 5.2 the theoretical and experiment results naturally leads to cornerstone techniques in NN optimization, i.e, normalization of variance in online normalization such as Batch Normalization (BN) [40], or normalization at initialization [31].

5.1 Order parameter in practical NNs

Counts of non-zero statistics and coupling set size. We compute non-zero statistics of Hessian entries to study the correlations between Hessian entries and estimate coupling set sizes. As visualized in fig. 2b previously, for each Hessian entry α, it has a coupling set $\mathcal{N}(\alpha)$, where the covariances $\kappa(\beta, \gamma), \beta \in \mathcal{N}_1, \gamma \in \mathbb{I} \setminus \mathcal{N}(\alpha)$ between w_β and entries outside the coupling set w_γ is zero. To estimate the coupling set sizes, we compute the count of non-zero normalized covariances, i.e., Pearson correlation coefficients (CC), for each entry, and average them. Cond. D.5 implies that the non-zero count of normalized $\kappa(\beta, \gamma), \beta, \gamma \in \mathcal{N}(\alpha)$ would be around the coupling set size $|\mathcal{N}|$.

Meanwhile, we also compute higher order statistics. As mentioned in section 4.1, cond. 3.1 is simplified to help understanding. The full condition cond. D.5 involves statistics of all orders instead of only the second, and cond. 4.1 is not necessary in the proof. Consequently, we also compute the non-zero higher order statistics, i.e., the skewness and excess kurtosis, to study the higher order correlations between coupling sets according to cond. D.5. We believe the statistics up to four have validated our theoretical results, and further higher statistics are too computational expensive to pursue. In this case, the statistics involve $k = 3, 4$ sets. The idea is that suppose k Hessian entries are picked. If one of them is outside the coupling sets of the rest, then the k-way statistics is zero. More specifically, for skewness, the condition corresponds to the phenomenon that for any Hessian entry α, β, given three Hessian entries $\alpha_1, \beta_1, \gamma_1$, if at least one entry, e.g., $\gamma_1 \in \mathbb{I} \setminus (\mathcal{N}(\alpha) \cup \mathcal{N}(\beta)), \alpha_1 \in \mathcal{N}(\alpha), \beta_1 \in \mathcal{N}(\beta)$, does not come from the coupling set $\mathcal{N}(\alpha), \mathcal{N}(\beta)$ of α, β, then the skewness $\kappa(\alpha_1, \beta_1, \gamma_1) = 0$. *That implies that given a specific α, $|\mathcal{N}|^2$ non-zero skewness are allowed.* The same logic applies to kurtoses, and we could have $|\mathcal{N}|^3$ non-zero kurtoses. We count by bootstrap methods and the details can be found in appendix E.

Sparse correlations between Hessian entries. The results are shown in fig. 1 (d)(e)(f). They show that cond. D.5 characterizes the behaviors of NNs well qualitatively. We can see at the start of the training (fig. 1d), the correlation coefficient matrix is very sparse in the sense that a vast majority of the entries are close to zero. As training progresses, the correlations increase (fig. 1f). At the end of the training (fig. 1e), more strong correlations emerge as the statistical information in data is learned, but the matrix is still sparse in the sense that most of the entries are still close to zero. Quantitatively, as shown by the light red line in fig. 1f at the end of the training, for a given Hessian entry w_α, of overall $8.68 * 10^{12}$ possible coefficients between w_α and $\boldsymbol{W}_{\mathbb{I}\setminus\{\alpha\}}$, w_α only correlates $3.39 * 10^{-4}$ of them. It suggests that neurons are relatively autonomous to adapt to minimize the risk without interfering the adaptation of other neurons, as characterized in section 3.4. We might see this is a blessing of both *high dimensionality* and *hierarchy*, or in other words, *complexity* and *organizing principle*, which allow the coexistence of *an extraordinary absolute coupling set size* and *a tiny relative coupling set size* compared with the overall possible correlations. Meanwhile, fig. 1f also shows that the average non-zero counts of CCs, skewness, kurtoses are roughly $N^{3/2}, N^3, N^{9/2}$ respectively at the end of the training. It implies the coupling set size $|\mathcal{N}|$ could be as large as roughly $N^{3/2}$ instead of $N^{1/2}$ in cond. 3.1. It suggests that the condition for theorem 1 can be further improved on the upper bound of the coupling set size to allow more correlations, which is an ongoing theme in the random matrix community in the past 60 years since Wigner [89] (1957) initialized the field, and the technique [Erdos et al. [25] (2019)] that we build on is the state-of-the-art result.

5.2 Normalization in NNs

The theoretical and experiment results so far have shown that all stationary points are saddle points except the zero risk global minima. It implies that if the following extra conditions are met, risk minimization by gradient descent can reach global minima. 1) The step size (learning rate) is chosen properly so that the risk

decreases at each step. 2) Saddle points are escaped. Previously, NNs' optimization behaviors somehow have a mysterious aura, and techniques that enable successful optimization are not fully understood. Among them, variance normalization [31; 36; 40] in the intermediate layers is one of the most vital techniques. The absence of local minima implies the success of variance normalization has a simple explanation: *it enables the decrease of risk at each step using normally used step size, e.g.,* $0.5 \sim 0.01$ *by avoiding exponential growth of gradients and keeping the norm of gradients in a suitable scale.* Thus, if the risk keeps decreasing at each step and saddle points are avoided, the global minima could be reached eventually. We explain why variance normalization is a logic alternative of the previous results in this section.

Small step size in steepest gradient descent. To begin with, we briefly describe the mechanism of steepest gradient descent methods. By Taylor series, the risk $R_m(l \circ T(X; \boldsymbol{\theta} + \delta\boldsymbol{\theta}))$ around point $\boldsymbol{\theta}$ in the parameter space can be approximated as follows.

$$R_m(l \circ T(X; \boldsymbol{\theta} + \delta\boldsymbol{\theta})) = R_m(l \circ T(X; \boldsymbol{\theta})) + \nabla_{\boldsymbol{\theta}} R_m^T \delta\boldsymbol{\theta}$$

A step of a fixed norm minimizes the risk the most in the direction of the gradient when only the first order information is considered. Thus, steepest gradient descent modifies $\boldsymbol{\theta}$ in the opposite direction of the gradient. The update equation of step size η for gradient descent is given as follows on the left, and the approximated change on the right.

$$\boldsymbol{\theta'} = \boldsymbol{\theta} - \eta \nabla_\theta R_m, \qquad \Delta R_m \approx -\eta ||\nabla_{\boldsymbol{\theta}} R_m||_2^2 \tag{10}$$

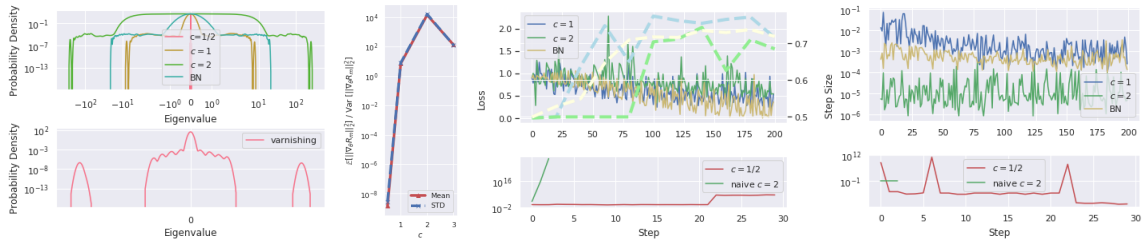
Since the change is only approximated, it is not valid for points far away from $\boldsymbol{\theta}$. *It implies that the step size* η *should be chosen small enough so that* $-\eta||\nabla_{\boldsymbol{\theta}} R_m||_2^2$ *approximates* ΔR_m *well to actually minimize the risk.* We show that such a principle explains the variance normalization, and the step size choice of NN training in practice.

Variance normalization at initialization. We first study the behaviors of NNs at initialization without BN. Suppose that $\forall l \in [L-1], \boldsymbol{W}_l \in \mathbb{R}^{n \times n}, n \in \mathbb{N}$, and $w_{ij}^l \sim \mathcal{N}(0, c/n)$. We study three classic initialization schemes that have factor c as $1/2, 1, 2$ respectively, which map to the conventional varnishing, stable, and exploded gradient cases respectively. The latter two corresponds to Xavier init [31] and MSRA init [36] respectively.

Previous experiment results have shown that the NN risk has a symmetric eigenspectrum at initialization. The symmetry is mainly a result of the small coupling set size characterized by cond. 3.1. Thus, it implies that the magnitude of the Hessian entries does not influence the symmetry because if two r.v.s do not correlate with one another, scaling the two r.v.s up wouldn't induce a correlation either. Formally, we show in corollary F.1 and lemma F.2 in appendix F.2 that multiplying c by a factor d would scale all Hessian eigenvalues potentially exponentially by $d^{(L-2)/2}$. It suggests that the symmetry exists at initialization regardless of the value of c, and consequently NNs with conventional initialization schemes would have symmetric eigenspectrums with similar shapes but exponentially scaled ranges. We validate that this is true in experiments, as shown in fig. 3a.

It suggests that by random initialization, the NNs are not at local minima, thus the problem is converted to find the proper step size that makes the risk decrease. As shown in eq. (10), the step size should scale the gradient norm $||\nabla_{\boldsymbol{\theta}} R_m||_2^2$ down to make ΔR_m small relatively to the current risk R_m. We undertake a mean-field analysis on the gradient norm with some common independence assumptions assumed previously in network initialization analysis, e.g., MSRA init [36], and find that the expected gradient norm grows at rate of $\Theta(Ln^2(c/2)^L)$ (lemma F.1 in appendix F.1) — note c is the variance used in initializing weights. We validate that the exponential behavior holds in experiments. As shown in fig. 3b, as c increases, the expected gradient norm grows exponentially. Thus, small differences in c would induce exponentially large changes in the resulted gradient norms.

Since risk approximation by gradient is only valid locally, step size η needs to be chosen properly so that ΔR_m is a fraction of current risk R_m. With such hugely different gradient norms, a normally naive 0.1 step size would make no progress in the $c = 1/2$ case, and lead to divergent training in the $c = 2$ case. The divergence



(a) (b) (c) (d)

Figure 3: Empirical study of NNs with different initialization schemes. (a) The eigenspectrum of a VGG net with the same configuration as that in fig. 1 but with BN removed and the initialization scheme changed. The bottom panel is a zoom-in of the eigenspectrum of the network initialized with varnishing gradients, since it has degenerated into a line in the panel above. (b) The plot of the means and standard deviations (STDs) of gradient norms against c. Note that the two lines are almost identical, and the standard deviation line overlays on the mean line. We do not plot the STDs as error bars since we have used a logarithmic scale, and error bars in this case are not intuitive. (c) Top panel: the training loss curve for 200 steps of NNs whose risk can be minimized through our dynamic step size scheme. The dashed lines are the test accuracy curves, and the right y-axis shows the accuracy. Bottom panel: the training loss curves for 2 steps of $c = 2$ case with constant step size 0.1, and for 30 steps of $c = 1/2$ case with dynamic step sizes respectively. (d) The corresponding step size curves of fig. 3c during training.

can be seen in the bottom panel in fig. 3c, where two steps would lead to a risk beyond 10^{16}. By dynamically choosing a proper step size $1/||\nabla_{\boldsymbol{\theta}} R_m||_2^2 * 0.01$ for each NN, as shown in fig. 3d, we manage to minimize the risk for both the $c = 1, 2$ cases, as shown in the top panel in fig. 3c. The dynamic learning rate scheme is not immediately obvious without the knowledge that NNs are not trapped in local minima. As can be seen in the green line in fig. 3d, the step size is around 10^{-5}, and is not the usual initial learning rate choice in practice. However, to make progress in the $c = 1/2$ case requires exponentially large step sizes, which leads to rather unpredictable behaviors, likely because the descended points are in a region far away from current point in the parameter space, and a working step size scheme with only local information is perhaps not viable to decrease the risk. The dynamic scheme leads to violently different step sizes as shown in red line in the bottom panel in fig. 3d, and the risk initially makes no progress and jumps to an exponentially large scale after 20 steps, as shown in the red line in the bottom panel in fig. 3c.

Online variance normalization. The above analysis shows that the gradient norm is an essential quantity to make NNs optimizable using normal practice. Instead computing step size according to the gradient norm, one logical alternative is simply to force the gradient norm to be in a good scale, which leads to Batch Normalization [40]. BN amounts to multiplying a weight matrix $\boldsymbol{W}_l$ with a diagonal matrix $\boldsymbol{\Sigma}_l$, i.e., $\tilde{\boldsymbol{x}}_{l-1}^T \boldsymbol{W}_l \boldsymbol{\Sigma}_l$, where $1/\Sigma_{ii}^l$ is the standard deviations of $\tilde{\boldsymbol{x}}_l$. At initialization, it amounts to initialize $w_{ij}^l, i, j \in [n]$ with $\mathcal{N}(0, \Sigma_{jj}^l c/n)$ respectively. More specifically, NN with BN has gradient as below (the calculation of gradient can be found in appendix D.2):

$$\partial l(T\boldsymbol{x}, y)/\partial \text{vec}\boldsymbol{W}_p = l'(T\boldsymbol{x}, y)\boldsymbol{\alpha}^T \overleftarrow{\prod}_{j=p+1}^{L-1} (\text{dg}(\tilde{\boldsymbol{h}}_j)\boldsymbol{\Sigma}_j \boldsymbol{W}_j^T)\text{dg}(\tilde{\boldsymbol{h}}_p) \otimes \boldsymbol{x}^T \overrightarrow{\prod}_{i=1}^{p-1} (\boldsymbol{W}_i \boldsymbol{\Sigma}_i \text{dg}(\tilde{\boldsymbol{h}}_i)),$$

where $\overleftarrow{\prod}_{j=p+1}^{L-1} (\text{dg}(\tilde{\boldsymbol{h}}_j)\boldsymbol{\Sigma}_j \boldsymbol{W}_j^T)$ denotes $\text{dg}(\tilde{\boldsymbol{h}}_{L-1})\boldsymbol{\Sigma}_{L-1}\boldsymbol{W}_{L-1}^T \dots \text{dg}(\tilde{\boldsymbol{h}}_{p+1})\boldsymbol{\Sigma}_{p+1}\boldsymbol{W}_{p+1}$. It prevents explosion by maintaining the rightmost term to have unit variance in each dimension, and alleviate the exponential behavior in gradient norms. By applying BN to the $c = 2$ case previously, the gradient norm is scaled down from $\sim 10^4$ to $\sim 10^2$, and it enables NNs to be trained in ~ 100 times larger step sizes, as shown in the yellow lines in fig. 3c and fig. 3d.

References

[1] Johannes Alt, Laszlo Erdos, and Torben Krüger. The Dyson equation with linear self-energy: spectral bands, edges and cusps. Technical report, 2018. URL http://arxiv.org/abs/1804.07752.

[2] Amari. Information geometry of the {EM} and {EM} algorithm for neural networks. *Neural Networks*, 8 (9):1379–1408, 1995.

[3] Alexandr Andoni, Rina Panigrahy, Gregory Valiant, and Li Zhang. Learning polynomials with neural networks. In *ICML*, 2014.

[4] Ankit B. Patel, Tan Nguyen, and Richard G. Baraniuk. A Probabilistic Framework for Deep Learning. In *NIPS*, 2016.

[5] Fabio Anselmi, Joel Z. Leibo, Lorenzo Rosasco, Jim Mutch, Andrea Tacchetti, and Tomaso Poggio. Unsupervised learning of invariant representations. *Theoretical Computer Science*, 633:112–121, jun 2015.

[6] Fabio Anselmi, Lorenzo Rosasco, and Tomaso Poggio. On Invariance and Selectivity in Representation Learning. *Information and Inference, Special Issue: Deep Learning*, may 2016.

[7] Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable Bounds for Learning Some Deep Representations. In *ICML*, 2014.

[8] Sanjeev Arora, Zhiyuan Li, and Kaifeng Lyu. Theoretical Analysis of Auto Rate-Tuning by Batch Normalization. In *ICLR*, 2019.

[9] M. Baity-Jesi, L. Sagun, M. Geiger, S. Spigler, G. Ben Arous, C. Cammarota, Y. LeCun, M. Wyart, and G. Biroli. Comparing Dynamics: Deep Neural Networks versus Glassy Systems. In *ICML*, 2018.

[10] Andrew J Ballard, Ritankar Das, Stefano Martiniani, Dhagash Mehta, Levent Sagun, Jacob D Stevenson, and David J Wales. Perspective: Energy Landscapes for Machine Learning. *Phys. Chem. Chem. Phys.*, 19 (20):12585–12603, 2017.

[11] S Becker and G E Hinton. Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 355(6356):161–3, jan 1992. ISSN 0028-0836. doi: 10.1038/355161a0.

[12] Johan Bjorck, Carla Gomes, Bart Selman, and Kilian Q. Weinberger. Understanding Batch Normalization. In *NeurIPS*, 2018.

[13] Alon Brutzkus and Amir Globerson. Globally Optimal Gradient Descent for a ConvNet with Gaussian Inputs. In *ICML*, 2017.

[14] Lenaic Chizat and Francis Bach. On the Global Convergence of Gradient Descent for Over-parameterized Models using Optimal Transport. In *NIPS*, 2018.

[15] Anna Choromanska, Mikael Henaff, and Michael Mathieu. The Loss Surfaces of Multilayer Networks. In *AISTATS*, 2015.

[16] Anna Choromanska, Yann LeCun, and Gérard Ben Arous. Open Problem: The landscape of the loss surfaces of multilayer networks. In *COLT*, 2015.

[17] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). In *ICLR*, 2015.

[18] Amit Daniely, Roy Frostig, and Yoram Singer. Toward Deeper Understanding of Neural Networks: The Power of Initialization and a Dual View on Expressivity. In *NIPS*, 2016.

[19] I. Daubechies, R. DeVore, S. Foucart, B. Hanin, and G. Petrova. Nonlinear Approximation and (Deep) ReLU Networks. Technical Report Id, 2019. URL http://arxiv.org/abs/1905.02199.

[20] Bruno Del Papa, Viola Priesemann, and Jochen Triesch. Criticality meets learning: Criticality signatures in a self-organizing recurrent neural network. *PLoS ONE*, 12(5):1–21, 2017. ISSN 19326203. doi: 10.1371/journal.pone.0178683.

[21] Simon S. Du and Jason D. Lee. On the Power of Over-parametrization in Neural Networks with Quadratic Activation. In *ICML*, 2018.

[22] Simon S. Du, Jason D. Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient Descent Finds Global Minima of Deep Neural Networks. In *ICML*, 2019.

[23] N. Dunford and J.T. Schwartz. *Linear operators. Part 1: General theory*. Pure and Applied Mathematics. Interscience Publishers, 1957.

[24] P. Erdi. *Complexity Explained*. Springer complexity. Springer Berlin Heidelberg, 2007. ISBN 9783540357780.

[25] Laszlo Erdos, Torben Kruger, and Dominik Schroder. Random Matrices with Slow Correlation Decay. *Forum of Mathematics, Sigma*, 7(8), 2019. doi: doi:10.1017/fms.2019.2.

[26] Ling Feng and Choy Heng Lai. Optimal Machine Intelligence Near the Edge of Chaos. Technical report, 2019. URL http://arxiv.org/abs/1909.05176.

[27] Kunihiko Fukushima. Cognitron: A self-organizing multilayered neural network. *Biological Cybernetics*, 20(3-4):121–136, 1975. ISSN 0340-1200. doi: 10.1007/BF00342633.

[28] Kunihiko Fukushima. Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biological Cybernetics*, 36(4):193–202, 1980.

[29] Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An Investigation into Neural Net Optimization via Hessian Eigenvalue Density. In *ICML*, 2019.

[30] Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Limitations of Lazy Training of Two-layers Neural Networks. Technical report, 2019. URL http://arxiv.org/abs/1906.08899.

[31] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[32] C. Gros. *Complex and Adaptive Dynamical Systems: A Primer*. Springer International Publishing, 2015. ISBN 9783319162652.

[33] Uffe Haagerup and Steen Thorbjørnsen. A new application of random matrices:. *Annals of Mathematics*, 162(2):711–775, 2005. ISSN 0003-486X. doi: 10.4007/annals.2005.162.711.

[34] Benjamin D. Haeffele and René Vidal. Global optimality in neural network training. In *CVPR*, 2017.

[35] Moritz Hardt and Tengyu Ma. Identity Matters in Deep LearningIdentity Matters in Deep Learning. In *ICLR*, 2017.

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *ICCV*, 2015.

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, pages 770–778, 2016.

[38] J. William Helton, Reza Rashidi Far, and Roland Speicher. Operator-valued semicircular elements: Solving a quadratic matrix equation with positivity constraints. *International Mathematics Research Notices*, 2007: 1–14, 2007. ISSN 10737928. doi: 10.1093/imrn/rnm086.

[39] Francis Heylighen. Challenge Propagation: Towards a theory of distributed intelligence and the global brain. *Spanda Journal*, 5(2):51–63, 2014. ISSN 2210-2175.

[40] Christian Ioffe, Sergey and Szegedy. Batch Normalization : Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*, 2015.

[41] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In *NIPS*, 2018.

[42] L.P. Kadanoff. *Statistical Physics: Statics, Dynamics and Renormalization*. Statistical Physics: Statics, Dynamics and Renormalization. World Scientific, 2000. ISBN 9789810237646.

[43] Kenji Kawaguchi. Deep Learning without Poor Local Minima. In *NIPS*, 2016.

[44] Kenji Kawaguchi and Jiaoyang Huang. Gradient Descent Finds Global Minima for Generalizable Deep Neural Networks of Practical Sizes. In *Annual Allerton Conference on Communication, Control, and Computing*, 2019.

[45] A. Klenke. *Probability Theory: A Comprehensive Course*. World Publishing Corporation, 2012. ISBN 9787510044113.

[46] Kazuyuki Koizumi, Takuma Sumikawa, and Tatjana Pavlenko. Measures of multivariate skewness and kurtosis in high-dimensional framework. *SUT Journal of Mathematics*, 50(2):483–511, 2014. ISSN 09165746.

[47] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, 2009.

[48] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012.

[49] Thomas Laurent and James von Brecht. Deep linear neural networks with arbitrary loss: All local minima are global. In *ICML*, 2018.

[50] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. ISSN 00189219. doi: 10.1109/5.726791.

[51] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep Neural Networks as Gaussian Processes. In *ICLR*, 2018.

[52] Yuanzhi Li and Yang Yuan. Convergence Analysis of Two-layer Neural Networks with ReLU Activation. In *NIPS*, 2017.

[53] Shiyu Liang, Ruoyu Sun, Yixuan Li, and R Srikant. Understanding the Loss Surface of Neural Networks for Binary Classification. In *ICML*, 2018.

[54] Henry W. Lin and Max Tegmark. Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168(6):1223–1247, aug 2017.

[55] Lin Lin, Yousef Saad, and Chao Yang. Approximating spectral densities of large matrices. *SIAM review*, 58 (1):34–65, 2016.

[56] A. Londei. Artificial Neural Networks and Complexity: An Overview. *Archeologia e Calcolatori*, 6 (Supplemento):113–130, 2014.

[57] J.R. Magnus and H. Neudecker. *Matrix differential calculus with applications in statistics and econometrics: 3rd. ed*. John Wiley & Sons, Limited, 2007.

[58] K. V. Mardia. Measures of multivariate skewness and kurtosis with applications. *Biometrika*, 57(3): 519–530, 1970. ISSN 00063444. doi: 10.1093/biomet/57.3.519.

[59] P. McCullagh. *Tensor methods in statistics*. Monographs on statistics and applied probability. Chapman and Hall, 1987. ISBN 9780412274800.

[60] Pankaj Mehta and David J. Schwab. An exact mapping between the Variational Renormalization Group and Deep Learning. Technical report, oct 2014. URL http://arxiv.org/abs/1410.3831.

[61] Quynh Nguyen and Matthias Hein. The loss surface of deep and wide neural networks. In *ICML*, 2017.

[62] Quynh Nguyen and Matthias Hein. Optimization Landscape and Expressivity of Deep CNNs. In *ICML*, 2018.

[63] Quynh Nguyen, Mahesh Chandra Mukkamala, and Matthias Hein. On the loss landscape of a class of deep neural networks with no bad local valleys. In *ICLR*, 2019.

[64] Gregoire Nicolis and Catherine Nicolis. *Foundations of Complex Systems: Emergence, Information and Prediction*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2nd edition, 2012. ISBN 9789814366601, 9814366609.

[65] Amilcar Oliveira and Antonio Seijas-Macias. An Approach to Distribution of the Product of Two Normal Variables. *Discussiones Mathematicae Probability and Statistics*, 32(1-2):87, 2012. ISSN 1509-9423. doi: 10.7151/dmps.1146.

[66] Vardan Papyan. The Full Spectrum of Deep Net Hessians At Scale: Dynamics with Sample Size. Technical report, 2018. URL http://arxiv.org/abs/1811.07062.

[67] Jeffrey Pennington and Yasaman Bahri. Geometry of Neural Network Loss Surfaces via Random Matrix Theory. In *ICML*, 2017.

[68] Jeffrey Pennington, Samuel S. Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. In *NIPS*, 2017.

[69] F W Pfeiffer. Automatic differentiation in prose. In *ICLR Workshop*, 2017.

[70] Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *NIPS*, jun 2016.

[71] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for Activation Functions. Technical report, oct 2017. URL http://arxiv.org/abs/1710.05941.

[72] Itay Safran and Ohad Shamir. On the Quality of the Initial Basin in Overspecified Neural Networks. In *ICML*, 2016.

[73] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How Does Batch Normalization Help Optimization? In *NeurIPS*, may 2018.

[74] Hanie Sedghi and Anima Anandkumar. Provable Methods for Training Neural Networks with Sparse Connectivity. In *ICLR Workshop*, 2014.

[75] Ravid Shwartz-Ziv and Naftali Tishby. Opening the Black Box of Deep Neural Networks via Information. Technical report, mar 2017. URL http://arxiv.org/abs/1703.00810.

[76] N Siddharth, Brooks Paige, Alban Desmaison, Frank Wood, Philip Torr, and Noah D Goodman. Learning deep models: Critical points and local openness. In *ICLR Workshop*, 2018.

[77] Herbert A. Simon. The Architecture of Complexity. *Proceedings of the American Philosophical Society*, 106(6):467–482, 1962. ISSN 1475-9551. doi: 10.1080/14759550302804.

[78] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015.

[79] Mahdi Soltanolkotabi. Learning ReLUs via Gradient Descent. In *NIPS*, 2017.

[80] Mahdi Soltanolkotabi, Adel Javanmard, and Jason D Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 4(8), 2018. doi: 10.1109/TIT.2018.2854560.

[81] Daniel Soudry and Elad Hoffer. Exponentially vanishing sub - optimal local minima in multilayer neural networks. In *ICLR Worksop*, 2018.

[82] T. Tao. *Topics in Random Matrix Theory*. Graduate studies in mathematics. American Mathematical Soc., 2012. ISBN 9780821885079.

[83] Alberto Testolin, Michele Piccolini, and Samir Suweis. Deep learning systems as complex networks. *Journal of Complex Networks*, 2019. doi: 10.1093/comnet/cnz018.

[84] Yuandong Tian, Tina Jiang, Qucheng Gong, and Ari Morcos. Luck Matters: Understanding Training Dynamics of Deep ReLU Networks. Technical report, 2019. URL http://arxiv.org/abs/1905.13405.

[85] Luca Venturi, Afonso S. Bandeira, and Joan Bruna. Spurious Valleys in Two-layer Neural Network Optimization Landscapes. Technical report, 2018. URL http://arxiv.org/abs/1802.06384.

[86] D. Wales, R. Saykally, University of Cambridge, A. Zewail, and D. King. *Energy Landscapes: Applications to Clusters, Biomolecules and Glasses*. Cambridge Molecular Science. Cambridge University Press, 2003. ISBN 9780521814157.

[87] Robert Ware and Frank Lad. Approximating the distribution for sums of products of normal variables. *Research-Paper 2003–15. Department of Mathematics and Statistics*, 2003.

[88] J. Weng, N. Ahuja, and T.S. Huang. Cresceptron: a self-organizing neural network which grows adaptively. *IJCNN*, 1992. doi: 10.1109/IJCNN.1992.287150.

[89] E. P. Wigner. Statistical properties of real symmetric matrices with many dimensions.pdf. In *Canadian Mathematical Congress Proceedings*, 1957.

[90] Stephen Wolfram. *A New Kind of Science*. Wolfram Media, 2002. ISBN 1579550088.

[91] Sho Yaida. Non-Gaussian processes and neural networks at finite widths. Technical report, 2019. URL http://arxiv.org/abs/1910.00019.

[92] Greg Yang, Jeffrey Pennington, Vinay Rao, Jascha Sohl-Dickstein, and Samuel S. Schoenholz. A Mean Field Theory of Batch Normalization. In *ICLR*, 2019.

[93] Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. Global optimality conditions for deep neural networks. In *ICLR*, 2018.

[94] Sergey Zagoruyko. 92.45% on cifar-10 in torch, 2015. URL http://torch.ch/blog/2015/07/30/cifar.html.

[95] Kai Zhong, Zhao Song, Prateek Jain, Peter L. Bartlett, and Inderjit S. Dhillon. Recovery Guarantees for One-hidden-layer Neural Networks. In *ICML*, 2017.

Appendices

A Notation

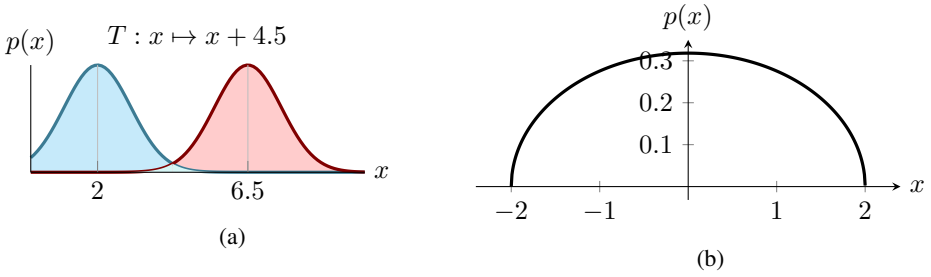
Notations. All scalar functions are denoted as normal letters, e.g., f; bold, lowercase letters denote vectors, e.g., $\boldsymbol{x}$; bold, uppercase letters denote matrices, or random matrices, e.g., $\boldsymbol{W}$; normal, uppercase letters denote random elements/variables, e.g., H. r.v. and r.v.s are short for random variable, random variables respectively. To index entries of a matrix, we denote $\mathbb{J} = [N] := \{1, \dots, N\}$, the ordered pairs of indices by $\mathbb{I} = \mathbb{J} \times \mathbb{J}$. For $\alpha \in \mathbb{I}, A \subseteq \mathbb{I}, i \in \mathbb{J}$, given a matrix $\boldsymbol{W}$, $w_\alpha, \boldsymbol{W}_A, \boldsymbol{W}_{i:}, \boldsymbol{W}_{:i}$ denote the entry at α, the vector that consists of entries at A, the ith row, ith column of $\boldsymbol{W}$ respectively. Occasionally, $\boldsymbol{W}_{ij}$ also denotes w_{ij} when the matrix and its entries are defined and used ad hoc, and should be self-clear in the context. Given two matrices $\boldsymbol{A}, \boldsymbol{B}$, the curly inequality $\preceq$ between matrices, i.e., $\boldsymbol{A} \preceq \boldsymbol{B}$, means $\boldsymbol{B} - \boldsymbol{A}$ is a positive definite matrix. Similar statements apply between a matrix and a vector, and a matrix and a scalar. $\succeq$ is defined similarly. $:=$ is the "define" symbol, where $x := y$ defines a new symbol x by equating it with y. tr denotes matrix trace. $\mathrm{dg}(\boldsymbol{h})$ denotes the diagonal matrix whose diagonal is the vector $\boldsymbol{h}$. $\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \boldsymbol{x}^T y, \langle \boldsymbol{A}^T \boldsymbol{B} \rangle := \mathrm{tr}\,(\boldsymbol{A}^T \boldsymbol{B})$. $\kappa(\cdot, \cdot)$ denotes covariance, e.g., $\kappa(w_\alpha, w_\beta)$ as the covariance between $w_\alpha, w_\beta, \alpha, \beta \in \mathbb{I}$. or covariance between functions f, g of entries in $\boldsymbol{W}$, i.e., $\kappa(f(\boldsymbol{W}_A), \kappa(w_\alpha, w_\beta)$. κ also denotes cumulants. $||\cdot||$ denotes 2-norm if not specified otherwise. $\Re$ and $\Im$ take the real and imaginary part of its operand (a complex number) respectively. $\overrightarrow{\Pi}_{i=1}^n \boldsymbol{W}_i, \overleftarrow{\Pi}_{i=1}^n \boldsymbol{W}_i, i < n, i, n \in \mathbb{N}$ denote $\boldsymbol{W}_1 \dots \boldsymbol{W}_n, \boldsymbol{W}_n \dots \boldsymbol{W}_1$ respectively. If matrices are indexed, we move the index up when indexing its entries, i.e., w_{ij}^1 denotes the ijth entry of $\boldsymbol{W}_1$. All the remaining symbols are defined when needed, and should be self-clear in the context.

B Background materials

We summarize some introductory materials for measure transportation theory, concentration of measure phenomenon, and complex systems in this section.

B.1 Measure transport

An illustration of measure transport defined in section 2.1 is given in fig. 4a.



(a) (b)

Figure 4: **(a)**: Measure Transport Illustration. The blue shaded area on the left is the probability measure of a random variable X taking values in $[-\infty, +\infty]$. It is a normal distribution centering at 2. A transport map $T : x \mapsto x + 4.5$ transports the measure of X to the measure of a new random variable $Y := X + 4.5$. The pushforward measure is the red shaded area on the right, centering at 6.5. That is, the measure transported from X to Y by T. **(b)**: eigenspectrum of Wigner matrix. The x-axis is the numerical value of eigenvalues, and the y-axis is the probability of obtaining an eigenvalue of a specific numeric value if we randomly sample an eigenvalue of all the eigenvalues of $\boldsymbol{A}_N$.

B.2 Concentration of measure phenomenon

The core idea of concentration of measure has all been contained in the classic normal distribution. Given an ensemble of i.i.d. r.v.s centering at 0, since the r.v.s are independent with each other, and have an equal probability being positive, or negative, they tend to cancel each other out when averaging their values, symmetrically spread around 0, and approach a normal distribution. Thus, sampling many samples from the average of the ensemble, 95% of them would fall in the range between $[+2\sigma, 2\sigma]$, where σ is the standard deviation of the ensemble.

What we need is the concentration of measure in the space of matrices. For $1 \leq i < j < \infty$, let a_{ij} be i.i.d. r.v.s. with mean 0 and variance 1, and $a_{ij} = a_{ji}$. Then $\boldsymbol{A}_N = [a_{ij}]_{i,j=1}^N$ is a matrix with random entries. It is the classic random matrix named *Wigner matrix*. Given an eigenvector $\boldsymbol{t}$ of $\boldsymbol{A}_N$, we know that when we multiply it with $\boldsymbol{A}$, we have $(\boldsymbol{At})_i = \sum_k a_{ik}t_k$. Note that when $\{a_{ik}t_k\}_{k=1,\dots,N}$ are i.i.d. r.v.s, $a_{ik}t_k$ also tend to cancel each other out and spread symmetrically around 0, resulting in a close-to-zero eigenvalue. Thus, when the dimension of the matrix N is large enough, we would have the eigenspectrum of the distribution shown in fig. 4b. It is termed *semi-circle law*. The eigenspectrum is obtained by solving a *self-consistent* equation as following

$$1 + (z + m)m = 0, \Im m > 0, \Im z > 0$$

where z is a constant complex number, m is the unknown complex number to solve. Then, we apply *inverse stieltjes transform* (c.f. lemma D.1) to get the spectrum.

Note that Wigner matrix has very strong requirements on its entries: independent identically distributed. Previous works of Choromanska et al. [15] and Pennington and Bahri [67] are partly based on Wigner matrix, thus fall short on realisticity. However, notice that the key phenomenon we need is just the concentration of measure, a.k.a., $\{a_{ik}t_k\}_{k=1,\dots,N}$ tend to cancel out and symmetrically spread around 0. Thus, the conditions we need are just they are of sparse correlations with each other to obtain a semi-circle-law-like eigenspectrum.

B.3 Complex systems

A complex system is a system consisting of many microscopic units where the interaction among units is of equal importance as properties of individual units. The interaction induces a macroscopic phenomenon that is known as the *emergency*, or "the whole is more than sum of the parts" [77]. Emergency exists everywhere in our daily lives. The average microscopic kinetic energy of water molecules induces the emergent macroscopic phenomenon that we measure as temperature. Under constant atmospheric pressure, it is also the *order parameter* of the water molecule system that determines how the system is macroscopically perceived. When the temperature becomes high, and the cohesive forces that attract molecules to each other are weaker than the kinetic energy, the macroscopically perceived molecule system that we call water would undertake a phase transition at this *critical* temperature to another macroscopically perceived form, i.e., vapor.

C Related works

C.1 The complexity approach to optimization problem of NNs

Despite significant progress in recent years, it is still safe to say that a theoretical characterization has somehow lagged behind the application-oriented progress in Deep Learning. Many factors contribute to the current state of affairs. Mathematics progresses by making intelligent definitions as much as proving hard theorems. In our opinion, a major factor might be the confusion on the question: *what is a NN, and what conceptual framework is needed to approach it?* Machinery from many existing ideas has been applied to understand NNs, e.g, renormalization group [54; 60], spin glass system [15], mean field analysis [30; 67; 68; 70; 92], information theory [2; 75], theoretical computer science [7], probabilistic graphical model [4], Gaussian process [51; 91], group theory [5; 6], kernel space [18; 41], and approximation theory [19] etc. However, the mathematical principle that underlies NNs is still not well understood, e.g., its representation power, optimization and generalization abilities.

Existing works. The idea of interpreting NNs as complex systems is not entirely novel. However, similar with the confusion discussed previously, diverse phenomena are complex systems, and it is not clear what are the defining characteristics that define NNs as complex systems. In the early days of NNs in the 1980s, when the complexity science was being born, the pioneering works in the fields have been taken self-organizing as the defining characteristics of NNs, e.g., Neocognition [27] [28], Weng et al. [88] and Becker and Hinton [11]. Recently, ideas from complex systems in physics have been applied to study NNs. The technique of Renormalization group (RG), which is developed mostly to characterize the macroscopic behaviors of inorganic complex systems with criticality behaviors, has been proposed to characterize NNs [54; 60], but it meets obstacles because of the lack of symmetry structure in NNs Mehta and Schwab [60]. On the contrary, the "limited structure" taken as a problem in [60] turns out to be the key to the optimization/adaptability of NNs in this work. The optimization results from the Spin Glass complex system [15] have also been tried on NNs, and unrealistic results are obtained [16] because the NN system is fit to be more like Spin Glass systems than NNs. Pennington et al. [68] and Feng and Lai [26] point out that the evolving/training of NNs operates at *critical* regimens at the *edge of chaos*. Criticality is a characteristic phenomenon of many complex systems in physics, where a slight change in the order parameters would lead to a drastic phase transition of the system, as in the water-vapor transition. However, it is unclear what macroscopic emergent behavior such criticality in NNs leads to previously. Maintaining NNs at criticality, either through orthogonal initialization [68], or normalization as discussed in section 5.2, is only part of the learning that helps local gradient descent that minimizes the empirical risk. Thus, the characterization is incomplete and might rely too much on analogy to existing systems in physics. Self-organizing is might be a more defining characteristics, and the criticality is a part of the self-organized learnability. Efforts through empirical studies by analogizing NNs to existing systems have been made from network science [83] and spin glass systems [9]. Compared with the concern that the lack of lattice-like structure in neural networks makes them not amenable to the analysis by renormalization group [60], it is exactly this kind of disorder, or we might say adaptability, that enables the learning of macroscopic patterns through a complex system of simple thresholding neurons. We need to be cautious that when one has a hammer, everything might look like a nail, and it is important to analyze the NN complex system from the first principle.

Perspective of this work. This paper aims to give a formalism that defines NNs as complex adaptive systems with powerful self-organizing adaptability (learning capacity): with conditions on the order parameters, given feedback signals (back-propagated gradients of risk/loss function) from the external source (training data), it has the emergent behavior where the neurons cooperatively perfectly classify training data by their labels (or technically, a risk landscape where all local minima are global minima with zero risks). Furthermore, under this context and terminology of complex system, this work shows that normalization [40] techniques in NNs put a system in a critical state where it might traverse on the loss landscape in a most efficient way.

Benefits of complexity thinking. There might be doubt about why the language of complex system is needed, thus, we would like to try to show the edge of such a conceptual framework by showing that how it helps us to identify possible objects to study the hard nonconvex optimization problem of NNs.

The optimization problem of NNs has been an unsolved issue for decades that attracts many attempts to understand it. At the risk of over-generalization, the principle of existing works follows the idea of reductionism (Nicolis and Nicolis [64], Chapter 6.8) — by examining individual parts of the system, the understanding of the whole is achievable. However, complexity thinking is a paradigm shift in science [90], which points out that the organizing principles of the system is as important as the properties of the individual units in the system. In such a system, nonlinearity, hierarchy (the organizing principle) and complexity (the number of units in the system) are vital traits, and if removed, the emergent behaviors of the system would likely be different. Existing works tend to sidestep by avoiding hierarchy by studying shallow NNs [3; 13; 14; 21; 30; 34; 52; 74; 79; 80; 81; 85; 95]; or by avoiding nonlinearity by studying linear NNs or deep NNs with linear algebraic based conditions [35; 49; 53; 61; 62; 63; 72; 76; 93]; or by subduing nonlinearity to linearity of Kernel Space through kernel trick [22; 41]; or by isolating the nonlinearity through mean field analysis and treating them as independent random variables with input data [15; 16; 67]. As a result, the shallow NN results do not apply to deep NNs; linear NN results do not apply to nonlinear NNs; results by kernel space ask the layer width to be either polynomials of

training sample size (which is orders of magnitude larger NNs in practice) because kernel space is the space that embeds samples, or literally infinity; and mean field results are unrealistic [16]. If we are to understand NNs, nonlinearity and complexity probably might be the object to study, not to avoid. As shown in this work, the macroscopic benign loss landscape behavior is a result of the hierarchically organized complexity that asks the network to be large in a way that depends on both depth and width of the network. Though we have not yet elaborated on the role of nonlinearity theoretically, the sparsity inducing behavior of ReLU likely plays a crucial role in inducing the sparse correlations observed in section 5.1. By directly characterizing them, the hierarchically organized complexity and nonlinearly induced sparsity become the cornerstones to characterize the behaviors of NNs *as it is* used in practice without modification in this work.

Besides the hindsight on the benefits of complexity thinking, it also helps before we obtain the results in this work. Complexity thinking prompts a direct characterization of the nonlinearity and complexity built through hierarchy, and reveals us a roadmap to understand the optimization behaviors of NNs in a systematical way. To understand how a complex system evolves, researchers have studied the potential energy landscape of the system, e.g., that of protein, glasses [86], evolution (Gros [32] Chapter 6, Nicolis and Nicolis [64] Chapter 3.8), where emergent properties are predicted from local minima and the transition states that connect them [10]. In these problems, a full characterization of the landscape could inform researchers the system is structure-seeking that equilibrates rapidly, or has competing structures or morphologies, which may be manifested as phase transitions and even glassy phenomenology [10]. Thus, NNs as an artificial model of a complex system in nature, it might be likely that non-convexity, nonlinearity and complexity are inherent characteristics of its optimization/learning. The intuition prompts us to obtain a full characterization of the loss landscape of a fully functional NN, and study the transition in the loss landscape in this work.

C.2 Non-convex optimization in NNs

The optimization problem of NNs has been an unsolved issue for decades that attracts many attempts to understand it. Due to the sheer amount of works, we focus on the works that attack the problem in its full complexity; that is, NNs with arbitrary input data distributions, nonlinear activation functions and number of layers. We note there are emergent positive works done on shallow NNs: Andoni et al. [3]; Brutzkus and Globerson [13]; Chizat and Bach [14]; Du and Lee [21]; Haeffele and Vidal [34]; Li and Yuan [52]; Sedghi and Anandkumar [74]; Soltanolkotabi [79]; Soltanolkotabi et al. [80]; Soudry and Hoffer [81]; Tian et al. [84]; Venturi et al. [85]; Zhong et al. [95]. Roughly, three approaches have been taken in analyzing the optimization of NNs: the linear algebra perspective, the mean field theory using random matrix theory, and kernel space methods.

Linear algebraic approach to optimization. The linear algebra approach, as the name suggests, shies away from the nonlinear nature of the problem. Kawaguchi [43] proves all local minima of a deep linear NN are global minima when some rank conditions of the weight matrices are held. Improvements have been continuously made, but the best result [44] so far still relies on a condition that requires the penultimate layer to have a width at least as large as the training sample size. This is because the analysis depends on a linear structure characterized by rank. Nguyen and Hein [61, 62] prove that if in a certain layer of a NN, it has more neurons than training samples, which makes it possible that the feature maps of all samples are linearly independent, then the network can reach zero training errors. A few works in the linear-algebraic direction [35; 49; 53; 63; 72; 76; 93] improve upon the ideas of the two previous works, or prove similar results under different settings, but essentially use the same linear algebraic approach. It is likely we do not need so many linearly independent intermediate features, which would lead to poor generalization. Thus, to understand the optimization behavior of NNs, we might need to step out of the comfort zone of linearity.

Mean field approach to optimization. The mean field theory approach using the tools of random matrix theory can attack the optimization problem of NNs in its full complexity, though existing works tend to be confused on the source of randomness. Choromanska et al. [15] tries to get rid of the activation function by assuming that its value is independent of its input, which is unrealistic [16]. Nevertheless, it is an important effort, and the first paper to attack deep NNs in its full complexity. After Choromanska et al. [15], which approaches by

analogizing with spin glass systems. Pennington and Bahri [67] study NNs from mean field theory from the first principle rather than by analog. They assume data, weights and errors are of i.i.d. Gaussian distributions, which are mean-field assumptions and unrealistic, and proceeds to analyze the Hessian of the loss function of NNs. Due to limitations of their assumptions, they can only analyze a NN with one hidden layer.

Kernel space approach to optimization. The above approaches work in the parameter space, Jacot et al. [41] approaches from the dual space by studying a kernel named Neural Tangent Kernel. It shows that if the width of each layer of a NN goes to infinite literally, the training reaches global minima. The proof relies heavily on the delicate Gaussian structure in the last layer created by the infinity, and does not easily generalize to cases where the infinity does not hold.

The work by Du et al. [22] lies in a mix between the three approaches. It improves on the i.i.d. assumptions by utilizing the fact that when the number of parameter is over-parameterizingly large, a gram matrix induced by NNs is close to the random matrix at initialization, thus it enables NNs to converge to zero risk. However, the convergence requires that the training data are linearly independent, as done in the works in the linear algebraic approach. As stated, it again sidesteps the randomness in NNs and relies on random matrix at initialization, similarly as having been done by Choromanska et al. [16] and Pennington and Bahri [67], while we directly analyze the Hessian matrix at any time during training. As one of the consequences, we obtain results on NNs, e.g., LeNet, that are not necessarily over-parameterized, while in Du et al. [22], it requires the width of each layer to grow quartically w.r.t. the training sample size.

The requirement on the size of the NNs. Lastly, since all works that give results on deep nonlinear neural networks rely on assumptions that the network size should be large, we compare the requirements of different works. In our results, the dimension N of Hessian is only required to be sufficiently large, instead of literally being infinity. This is a non-asymptotic result that is different from existing results: N does not depend on the training set size as in Du et al. [22]; Nguyen and Hein [61, 62], and thus N does not need to grow polynomially or exponentially with them; the result holds non-asymptotically, instead of asymptotically as in Neural Tangent Kernel (NTK) [41], whose results cannot hold if layer width $n \not\to \infty$. For NNs, N is normally very large. In the toy example we discussed in fig. 2, N is 10^9, which makes $\boldsymbol{H}$ a *very* large random matrix. The experiment in section 1 shows that the N of a relatively small LeNet is large enough. Furthermore, our results states that the size of the system is measured in a hierarchical way that involves both depth and width of NNs, mirroring the organizing principle of the system. The large N can be achieved by adding more layers, instead of just enlarging the width.

C.3 Normalization in NNs

The result on normalization is mostly to demonstrate the picture of optimization behaviors outlined by this work, and is not a novelty in its own right. Many existing works have been working on understanding normalization in NNs, and normalization could be motivated in a variety of ways. Batch normalization has been found to smooth the loss landscape and make gradient more predictable [73] (the same results can be obtained by lemma F.1, corollary F.1 and lemma F.2 as well). It also enables larger learning rates while preventing divergent training, and improves generalization [12] — the larger learning rate result is also reached in section 5.2. It provides scale invariant parameters that ensures the convergence to stationary points with arbitrary learning rates under some smooth assumptions [8].

D Full proof of theorem 1

We present the full proof of theorem 1 in this section, the sketch of which has been outlined in section 4.3.2. We first summarize the intuitions behind the remaining conditions that have not been explained fully in the main content in appendix D.1 before we delve into the proof. In appendix D.2, we derive the Hessian of NNs (introductory materials are given in section 3.1-3.3). In appendix D.3, we introduce the random matrix tool we used, i.e., Matrix Dyson Equation (MDE), to study the Hessian (introductory materials are given in

section 4.3.1 and appendix B.2). In appendix D.4, we note the relationship between simplified conditions in the main content and the more general conditions presented in appendix D.3. The conditions in appendix D.3 is not computable, thus we convert them to computable conditions in appendix D.5. The computable conditions are for the experiments, and are not actually used in the proof. They are sufficient conditions for the non-computable ones. Then, in appendix D.6, we use MDE to study the loss landscape of NNs, and prove theorem 1 and results that lead to it. The sketch of appendix D.6 is present in section 4.3.2 previously. Lastly, we provide some extra notes on how the constants in the boundedness condition (cond. D.2 and its simplification cond. 4.4) are related to the high probability bound (cond. 4.2) that establishes the closeness of the Hessian eigenspectrum to a symmetric eigenspectrum.

D.1 Further details and intuitions of boundedness and comparability conditions

We summarize the intuitions behind the remaining conditions that have not been explained fully in the main content here before we delve into the proof.

D.1.1 Technical covariance norms in cond. 4.4

We explain the technicality of $|||\kappa|||_2^{\text{av}}$ to illustrate how statistical dependency is bounded as follows. av stands for average, while iso stands for isotropic. $\forall \alpha, \beta \in \mathbb{I}$, denote $\alpha := (a_1, a_2), \beta := (b_1, b_2)$, and let $\boldsymbol{K}^{\text{av}}$ be the matrix, of which $\boldsymbol{K}^{\text{av}}_{a_1 N + a_2, b_1 N + b_2} := |\kappa(\alpha, \beta)|$; note that $\boldsymbol{K}^{\text{av}}$ is a $N^2 \times N^2$ matrix. $|||\kappa|||_2^{\text{av}}$ is defined as $||\boldsymbol{K}^{\text{av}}||$, the operator norm of $\boldsymbol{K}^{\text{av}}$. Note that each row of $\boldsymbol{K}^{\text{av}}$ is the covariances of w_α with all entries (including w_α itself) of $\boldsymbol{W}$. $||\boldsymbol{K}^{\text{av}}||$ characterizes a collective behavior of all the covariances. To understand the characterization, recall the definition of $||\cdot||$ on matrix as follows.

$$||\boldsymbol{K}^{\text{av}}|| = \max_{||\boldsymbol{x}||\leq 1} \boldsymbol{x}^T \boldsymbol{K}^{\text{av}} \boldsymbol{x} = \max_{||\boldsymbol{x}||\leq 1} \sum_{a_1=1, b_1=1, a_2=1, b_2=1}^{a_1=N, b_1=N, a_2=N, b_2=N} x_{a_1 N + b1} \, |\kappa(a_1 b_1, a_2 b_2)| \, x_{a_2 N + b2}.$$

Thus, it is the largest possible weighted average of all the absolute covariances. Cond. 4.4 puts a cap on the largest weighted average. $|||\kappa|||_2^{\text{iso}}$ defines a similar quantity with $|||\kappa|||_2^{\text{av}}$. We defer its definition to definition D.7, so not to clutter the message with technical definitions too much at the beginning.

D.1.2 Intuitions behind the comparability condition

Cond. 4.5 could be interpreted as a certain mean field condition that makes the covariances that do exist between entries in $\boldsymbol{W}$ comparable. To see how, we write $\mathcal{S}[\boldsymbol{T}]_{pq} = 1/N \sum_{i,j=1}^{N} t_{ij} \mathbb{E}[w_{pi} w_{jq}]$. Let $s_{pq} := \mathcal{S}[\boldsymbol{T}]_{pq}$. Thus, for a given $\boldsymbol{T}$, s_{pq} is a weighted summation of a set of covariances between $\boldsymbol{W}_{:p}$ and $\boldsymbol{W}_{:q}$. Note that the weights t_{ij} are the same for all p, q. Similarly with the explanation of cond. 4.4, $c\, N^{-1}\text{tr}\, \boldsymbol{T} \preceq \mathcal{S}[\boldsymbol{T}] \preceq CN^{-1}\text{tr}\, \boldsymbol{T}$ asks the smallest and the largest weighted average of entries of $\mathcal{S}[\boldsymbol{T}]$ to be lower and upper bounded by $c\, N^{-1}\text{tr}\, \boldsymbol{T}, CN^{-1}\text{tr}\, \boldsymbol{T}$ respectively. If the largest weighted average is up to C/c times larger than the smallest, it implies s_{pq} are comparable; that is, different sets of covariances involved in s_{pq} are comparable in term of the summation arbitrated by any $\boldsymbol{T} \succeq \boldsymbol{0}$. To see an example, let T be a diagonal matrix with a single non-zero entry at (i, i), we have $\mathcal{S}[\boldsymbol{T}]_{pq} = (1/N) t_{ii} \mathbb{E}[w_{pi} w_{iq}]$. Further letting $\mathbb{E}[w_{pi} w_{iq}] = 0, p \neq q$, $\mathcal{S}[\boldsymbol{T}]$ is thus a diagonal matrix with diagonal elements $s_{pp} = (t_{ii}/N)\mathbb{E}[w_{pi}^2]$ (recall $\boldsymbol{W}$ is symmetric). The condition asks the variance $\mathbb{E}[w_{pi}^2], p = 1 \ldots N$ to be comparable (in the band $[c, C]$).

D.2 Hessian of NN is inherently a huge random matrix

As explained, to study the landscape of the loss function, we study the eigenvalue distribution of its Hessian $\boldsymbol{H}$ at the critical points. First, we derive the Hessian $\boldsymbol{H}$ of loss function of class $\mathcal{L}_0$ composed upon NNs with a single output. For a review of matrix calculus, the reader may refer to Magnus and Neudecker [57].

The first partial differential of l defined at eq. (4) w.r.t. $\boldsymbol{W}_p$ is

$$\begin{aligned}\partial l(T\boldsymbol{x}, y) = l'(T\boldsymbol{x}, y)\boldsymbol{\alpha}^T \overleftarrow{\prod}_{j=p+1}^{L-1}(\mathrm{dg}(\tilde{\boldsymbol{h}}_j)\boldsymbol{W}_j^T)\mathrm{dg}(\tilde{\boldsymbol{h}}_p) \\ \otimes \boldsymbol{x}^T \overrightarrow{\prod}_{i=1}^{p-1}(\boldsymbol{W}_i \mathrm{dg}(\tilde{\boldsymbol{h}}_i))\partial \mathrm{vec}\boldsymbol{W}_p\end{aligned} \tag{11}$$

where $\otimes$ denotes Kronecker product, and $\{\tilde{\boldsymbol{h}}_i\}_{i=1,\ldots,L-1}$ are realization of r.v.s H defined at eq. (2). Note that for clarity of presentation, we use the partial differential the same way as differential is defined and used in Magnus and Neudecker [57], i.e., ∂l is a number instead of an infinitely small quantities, though in the book, partial differential is not defined explicitly.

Since $\boldsymbol{H}$ is symmetric, we only need to compute the block matrices by taking partial differential w.r.t. $\boldsymbol{W}_q$, where $q > p$ — taking partial differential w.r.t. $\boldsymbol{W}_p$ again gives zero matrix.

$$\begin{aligned}&\partial^2 l(T\boldsymbol{x}, y) \\ =&l'(T\boldsymbol{x}, y)[(\boldsymbol{\alpha}^T \overleftarrow{\prod}_{k=q+1}^{L-1}(\mathrm{dg}(\tilde{\boldsymbol{h}}_k)\boldsymbol{W}_k^T)\mathrm{dg}(\tilde{\boldsymbol{h}}_q) \\ \otimes&\mathrm{dg}(\tilde{\boldsymbol{h}}_p)\overrightarrow{\prod}_{j=p+1}^{q-1}(\boldsymbol{W}_j \mathrm{dg}(\tilde{\boldsymbol{h}}_j))\partial \mathrm{vec}\boldsymbol{W}_q)^T \\ \otimes&\boldsymbol{x}^T \overrightarrow{\prod}_{i=1}^{p-1}(\boldsymbol{W}_i \mathrm{dg}(\tilde{\boldsymbol{h}}_i))]\partial \mathrm{vec}\boldsymbol{W}_p\end{aligned}$$

$$\begin{aligned}=&l'(T\boldsymbol{x}, y) \\ &(\partial \mathrm{vec}\boldsymbol{W}_q)^T[\mathrm{dg}(\tilde{\boldsymbol{h}}_q)\overrightarrow{\prod}_{k=q+1}^{L-1}(\boldsymbol{W}_k \mathrm{dg}(\tilde{\boldsymbol{h}}_k))\boldsymbol{\alpha} \\ \otimes&\overleftarrow{\prod}_{j=p+1}^{q-1}(\mathrm{dg}(\tilde{\boldsymbol{h}}_j)\boldsymbol{W}_j^T)\mathrm{dg}(\tilde{\boldsymbol{h}}_p) \\ \otimes&\boldsymbol{x}^T \overrightarrow{\prod}_{i=1}^{p-1}(\boldsymbol{W}_i \mathrm{dg}(\tilde{\boldsymbol{h}}_i))]\partial \mathrm{vec}\boldsymbol{W}_p\end{aligned}$$

Thus, $\boldsymbol{H}$ is an ensemble of real symmetric random matrix with correlated entries. Denote

$$\begin{aligned}\boldsymbol{H}_{pq} =&l'(T\boldsymbol{x}, y)\mathrm{dg}(\tilde{\boldsymbol{h}}_q)\overrightarrow{\prod}_{k=q+1}^{L-1}(\boldsymbol{W}_k \mathrm{dg}(\tilde{\boldsymbol{h}}_k))\boldsymbol{\alpha} \\ &\otimes \overleftarrow{\prod}_{j=p+1}^{q-1}(\mathrm{dg}(\tilde{\boldsymbol{h}}_j)\boldsymbol{W}_j^T)\mathrm{dg}(\tilde{\boldsymbol{h}}_p) \\ &\otimes \boldsymbol{x}^T \overrightarrow{\prod}_{i=1}^{p-1}(\boldsymbol{W}_i \mathrm{dg}(\tilde{\boldsymbol{h}}_i))\end{aligned} \tag{12}$$

We have the Hessian of l as

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{H}_{12}^T & \ldots & \boldsymbol{H}_{1L}^T \\ \boldsymbol{H}_{12} & \boldsymbol{0} & \ldots & \boldsymbol{H}_{2L}^T \\ \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{H}_{1L} & \boldsymbol{H}_{2L} & \ldots & \boldsymbol{0} \end{bmatrix} \tag{13}$$

Equation (12) could be reformulated, so it may become easier to comprehend, as we explain in section 3.3. We rewrite eq. (12) in the following form on the right, where on the left are shorthanded symbols defined for the

reformulation (the equation should be read vertically)

$$
\begin{aligned}
\tilde{\boldsymbol{W}}_{q\sim(L-1)} &:= \mathrm{dg}(\tilde{\boldsymbol{h}}_q)\overrightarrow{\prod}_{k=q+1}^{L-2}(\boldsymbol{W}_k\mathrm{dg}(\tilde{\boldsymbol{h}}_k))\boldsymbol{W}_{L-1} \\
\tilde{\boldsymbol{\alpha}}_q &:= \tilde{\boldsymbol{W}}_{q\sim(L-1)}\mathrm{dg}(\tilde{\boldsymbol{h}}_{L-1})\boldsymbol{\alpha}l'(T\boldsymbol{x},y) \\
\tilde{\boldsymbol{W}}^T_{p\sim(q-1)} &:= \overleftarrow{\prod}_{j=p+1}^{q-1}(\mathrm{dg}(\tilde{\boldsymbol{h}}_j)\boldsymbol{W}_j^T)\mathrm{dg}(\tilde{\boldsymbol{h}}_p) \\
\tilde{\boldsymbol{W}}_{1\sim(p-1)} &:= \overrightarrow{\prod}_{i=1}^{p-1}(\boldsymbol{W}_i\mathrm{dg}(\tilde{\boldsymbol{h}}_i)) \\
\tilde{\boldsymbol{x}}^T_{p-1} &:= \boldsymbol{x}^T\tilde{\boldsymbol{W}}_{1\sim(p-1)}
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{H}_{pq} = l'(T\boldsymbol{x},y)\tilde{\boldsymbol{W}}_{q\sim(L-1)}\mathrm{dg}(\tilde{\boldsymbol{h}}_{L-1})\boldsymbol{\alpha} \quad &= \tilde{\boldsymbol{\alpha}}_q && \text{(14a)} \\
\otimes\ \tilde{\boldsymbol{W}}^T_{p\sim(q-1)} \quad &\otimes\ \tilde{\boldsymbol{W}}^T_{p\sim(q-1)} && \text{(14b)} \\
\otimes\ \boldsymbol{x}^T\tilde{\boldsymbol{W}}_{1\sim(p-1)} \quad &\otimes\ \tilde{\boldsymbol{x}}^T_{p-1} && \text{(14c)}
\end{aligned}
$$

Through such rewriting, entries of Hessian can correspond to components in activation paths, as discussed in section 3. The connection is illustrated in fig. 2a. Drawn in blue, eq. (14a) is a vector $\tilde{\boldsymbol{\alpha}}_q$ created by back propagating the vector $\mathrm{dg}(\tilde{\boldsymbol{h}}_{L-1})\boldsymbol{\alpha}l'(T\boldsymbol{x},y)$ to layer q by left multiplying $\tilde{\boldsymbol{W}}_{q\sim(L-1)}$— note that it is the *back propagated gradient*. Drawn in green, if we treat everything between $\mathrm{dg}(\tilde{\boldsymbol{h}}_{q-1})$ and $\mathrm{dg}(\tilde{\boldsymbol{h}}_p)$ in $\tilde{\boldsymbol{W}}_{p\sim(q-1)}$ as a matrix whose entries are all one, expectation of eq. (14b) is the *covariance matrix* between $\tilde{\boldsymbol{h}}_{q-1}$ and $\tilde{\boldsymbol{h}}_p$, i.e., $\mathbb{E}_{z\sim\mu^z}[\tilde{\boldsymbol{W}}_{p\sim(q-1)}]$. It could be understood as the strength of correlations between the neurons at layer p and $q-1$. Drawn in red, eq. (14c) is the *forward propagated activation* at layer $p-1$.

Recall that the objective function is the empirical risk function $R_m(T)$ at eq. (1). With the Hessian of loss function at an example $\boldsymbol{x}$ is calculated. Given a set of i.i.d. training samples $(X_i,Y_i)_{i=1,\dots,m}$, $R_m(T)$ is a summation of the i.i.d. random matrices. Formally, reusing the notation to denote $\boldsymbol{H}$ the Hessian of $R_m(T)$ and $\boldsymbol{H}_i$ the Hessian of $l(T(X_i;\boldsymbol{\theta}),Y_i)$, we have

$$
\boldsymbol{H} = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{H}_i \tag{15}
$$

D.3 Eigenvalue distribution of symmetry random matrix with slow correlation decay

In this section, we show how to obtain the eigenspectrum of $\boldsymbol{H}$ through random matrix theory (RMT) [82]. RMT has been born out of the study on the nuclei of heavy atoms, where the spacings between lines in the spectrum of a heavy atom nucleus is postulated to be the same with spacings between eigenvalues of a random matrix [89]. In a certain way, it is one of the backbone math of complex systems, where the collective behaviors of sophisticated subunits can be analyzed stochastically when deterministic or analytic analysis is intractable.

D.3.1 Preliminary definitions

The following definitions can be found in Tao [82] unless otherwise noted.

The eigenspectrum is studied as empirical spectral distribution (ESD) in RMT, define as

Definition D.1 (Empirical Spectral Distribution)**.** *Given a $N\times N$ random matrix $\boldsymbol{H}$, its* **empirical spectral distribution** $\mu_{\boldsymbol{H}}$ *is*

$$
\mu_{\boldsymbol{H}} = \frac{1}{N}\sum_{i=1}^{N}\delta_{\lambda_i}
$$

where $\{\lambda_i\}_{i=1,\dots,N}$ are all eigenvalues of $\boldsymbol{H}$ and δ is the delta function.

Given a hermitian matrix $\boldsymbol{H}$, its ESD $\mu_{\boldsymbol{H}}(\lambda)$ can be studied via its resolvent $\boldsymbol{G}$.

Definition D.2 (Resolvent)**.** *Let $\boldsymbol{H}$ be a normal matrix, and $z \in \mathbb{H}$ a spectral parameter. The* **resolvent $\boldsymbol{G}$** *of $\boldsymbol{H}$ at z is defined as*

$$\boldsymbol{G} = \boldsymbol{G}(z) = \frac{1}{\boldsymbol{H} - z}$$

where

$$\mathbb{H} := \{z \in \mathbb{C} : \Im z > 0\}$$

$\mathbb{C}$ denotes the complex field, and $\Im$ is the function gets imaginary part of a complex number z.

$\boldsymbol{G}$ compactly summarizes the spectral information of $\boldsymbol{H}$ around z, which is normally analyzed by *functional calculus* on operators, and defined through Cauchy integral formula on operators as follows.

Definition D.3 (Functions of Operators)**.**

$$f(T) = \frac{1}{2\pi i} \int_C \frac{f(\lambda)}{\lambda - T} d\lambda \tag{16}$$

where f is an analytic scalar function and C is an appropriately chosen contour in $\mathbb{C}$.

The formula can be defined on a range of linear operators [23] (Recall that a linear operator is a linear mapping whose domain and codomain are defined on the same field). Since the most complex case involved here will be a normal matrix, we stop at stating that the formula holds true when T is a normal matrix.

Resolvent $\boldsymbol{G}$ is related to eigenspectrum of $\boldsymbol{H}$ through stieltjes transform of $\mu_{\boldsymbol{H}}(\lambda)$.

Definition D.4 (Stieltjes Transform)**.** *Let μ be a Borel probability measure on $\mathbb{R}$. Its* **Stieltjes transform** *at a spectral parameter $z \in \mathbb{H}$ is defined as*

$$m_\mu(z) = \int_{\mathbb{R}} \frac{d\mu(x)}{x - z}$$

With some efforts, it can be seen that the normalized trace of $\boldsymbol{G}$ is stieltjes transform of eigenspectrum of $\boldsymbol{H}$

$$m_{\mu_{\boldsymbol{H}}}(z) = \frac{1}{N} \text{tr } G$$

For a proof, the reader may refer to proposition 2.1 in Alt et al. [1]. $\mu_{\boldsymbol{H}}$ can be recovered from $m_{\mu_{\boldsymbol{H}}}$ through the inverse formula of Stieltjes-Perron.

Lemma D.1 (Inverse Stieljies Transform)**.** *Suppose that μ is a probability measure on $\mathbb{R}$ and let m_μ be its Stieltjes transform. Then for any $a < b$, we have*

$$\mu((a,b)) + \frac{1}{2}[\mu(\{a\}) + \mu(\{b\})] = \lim_{\Im z \to 0} \frac{1}{\pi} \int_b^a \Im m_\mu(z) d\Re z$$

where $\Re$ is the function that gets the real part of z. The proof can be found at Tao [82] p. 144.

D.3.2 Self-consistent Matrix Dyson Equation

Consequently, the problem converts to obtain $\boldsymbol{G}$ if we want to obtain $\mu_{\boldsymbol{H}}$. A recent advance in the RMT community has enabled the analysis of ESD of symmetric random matrix with correlations [25] by extending ideas from the mean field theory [42], which we introduce in the following.

The resolvent $\boldsymbol{G}$ holds an identity by definition

$$\boldsymbol{H}\boldsymbol{G} = \boldsymbol{I} + z\boldsymbol{G} \tag{17}$$

Note that in the above equation $\boldsymbol{G}$ is a function $\boldsymbol{G}(\boldsymbol{H})$ of $\boldsymbol{H}$. When the average fluctuation of entries of $\boldsymbol{G}$ w.r.t. to its mean is small as N grows large, eq. (17) can be turned into a solvable equation regarding $\boldsymbol{G}$ instead of merely a definition. Formally, it is achieved by taking the expectation of eq. (17)

$$\mathbb{E}[\boldsymbol{H}\boldsymbol{G}] = \boldsymbol{I} + z\mathbb{E}[\boldsymbol{G}] \tag{18}$$

When fluctuation higher than the second order are negligible, we can obtain a class of random matrices whose ESD can be obtained by solving a truncated cumulant expansion of eq. (18). With the above approach, using sophisticated multivariate cumulant expansion, Erdos et al. [25] proves $\boldsymbol{G}$ can be obtained as the unique solution to *Matrix Dyson Equation (MDE)* below

$$\boldsymbol{I} + (z - \boldsymbol{A} + \mathcal{S}[\boldsymbol{G}])\boldsymbol{G} = \boldsymbol{0}, \Im\boldsymbol{G} \succ \boldsymbol{0}, \Im z > 0 \tag{19}$$

where $\Im\boldsymbol{G} \succ 0$ means $\Im\boldsymbol{G}$ is positive definite,

$$\boldsymbol{A} := \mathbb{E}[\boldsymbol{H}], \frac{1}{\sqrt{N}}\boldsymbol{W} := \boldsymbol{H} - \boldsymbol{A}, \mathcal{S}[\boldsymbol{R}] := \frac{1}{N}\mathbb{E}_{\boldsymbol{W}}[\boldsymbol{W}\boldsymbol{R}\boldsymbol{W}] \tag{20}$$

$\mathcal{S}$ is a linear map on the space of $N \times N$ matrices and $\boldsymbol{W}$ is a random matrix with zero expectation. The expectation in $\mathcal{S}$ is taken w.r.t. $\boldsymbol{W}$, while taking $\boldsymbol{R}$ as a deterministic matrix.

The hard work done in Erdos et al. [25] is to find the conditions that the fluctuation defined by

$$\boldsymbol{D} := \boldsymbol{I} + (z - \boldsymbol{A} + \mathcal{S}[\boldsymbol{G}])\boldsymbol{G} \tag{21}$$

is small by expectation, i.e., $\mathbb{E}[\boldsymbol{D}] = \boldsymbol{0}$ so that eq. (19) holds. This is a concentration of measure phenomenon where the concentration can be sufficiently characterized by up-to-second-order moments/cumulants. We describe their results formally in the following, which begins with some more definitions, adopted from Erdos et al. [25].

D.3.3 Cumulant norms

Definition D.5 (Cumulant)**. Cumulants** *of $\kappa_{\boldsymbol{m}}$ of a random vector $\boldsymbol{w} = (w_1, \ldots, w_n)$ are defined as the coefficients of log-characteristic function*

$$\log \mathbb{E}e^{i\boldsymbol{t}^T\boldsymbol{w}} = \sum_{\boldsymbol{m}} \kappa_{\boldsymbol{m}} \frac{i\boldsymbol{t}^{\boldsymbol{m}}}{\boldsymbol{m}!}$$

where $\sum_{\boldsymbol{m}}$ is the sum over all n-dimensional multi-indices $\boldsymbol{m} := (m_1, \ldots, m_n)$, and $\boldsymbol{t}^{\boldsymbol{m}}$ denotes $\prod_{i=1}^n t_{m_i}$.

To recall, a multi-indices is

Definition D.6 (Multi-index)**.** *a n-dimensional multi-index is a n-tuple*

$$\boldsymbol{m} = (m_1, \ldots, m_n)$$

of non-negative integers. Note that $|\boldsymbol{m}| = \sum_{i=1}^n m_i$, and $\boldsymbol{m}! = \prod_{i=1}^n m_i!$.

Similar with the more familiar concept *moment*, cumulant is also a measure of statistical properties of r.v.s. Particularly, the k-order cumulant κ characterizes the k-way correlations of a set of r.v.s. We have used the same symbols κ for covariance previously. It is because **the first order cumulant is just mean, while the second order cumulant is covariance.** Compared with moments, cumulants have many properties that are conducive to theoretical analysis. For example, for a Gaussian distribution, all cumulants beyond the second order ones are zero, while the statement is not true for moments.

The key insight of the work is to identify the conditions where a matrix entry $w_\alpha, \alpha \in \mathbb{I}$ is only strongly correlated with a minority of $\boldsymbol{W}_{\mathbb{I}\setminus\{\alpha\}}$, and higher order cumulants tend to be weak and not influential in large N limit. Thus, a proper formulation of the correlation strength is needed, and is defined as the cumulant norms on entries of $\boldsymbol{W}$ in the following. Given k entries $\boldsymbol{W}_{\boldsymbol{\alpha}}$ at $\boldsymbol{\alpha} = \{\alpha_i\}_{i=1,\ldots,k}, \alpha_i \in \mathbb{I}$ of matrix $\boldsymbol{W}$, where duplication

is allowed, denote $\kappa(\alpha_1,\ldots,\alpha_k)=\kappa(w_{\alpha_1},\ldots,w_{\alpha_k})$. The cumulant norms defined in Erdos et al. [25] are summarized below. We do not want to explain the cumulant norms beyond what would be said, considering it is too technically involved and rather a distraction. **For interested readers, we suggest reading the paper [25]. In order to understand the key idea of the proof, it suffices to only understand that they characterize the strength of correlations in a certain way.** How they relate to NNs are explained in section 3, where we explain the condition in detail up to second order cumulants. Understanding higher order conditions requires a high level of mathematical sophistication given they are fresh results in the random matrix community, if readers are not familiar with the progress already.

Definition D.7 (Cumulant Norms)**.**

$$
\begin{aligned}
|||\kappa||| &:= |||\kappa|||_{\leq R} := \max_{2\leq k\leq R} |||\kappa|||_k,\\
|||\kappa|||_k &:= |||\kappa|||_k^{av} + |||\kappa|||_k^{iso}\\
|||\kappa|||_2^{av} &:= ||\,|\kappa(*,*)|\,||,\\
|||\kappa|||_k^{av} &:= N^{-2}\sum_{\alpha_1,\ldots,\alpha_k}|\kappa(\alpha_1,\ldots,\alpha_k)|, k\geq 4\\
|||\kappa|||_3^{av} &:= ||\sum_{\alpha_1}|\kappa(\alpha_1,*,*)|\,||+\\
&\quad \inf_{\kappa=\kappa_{dd}+\kappa_{dc}+\kappa_{cd}+\kappa_{cc}}(|||\kappa_{dd}|||_{dd}+|||\kappa_{dc}|||_{dc}\\
&\quad +|||\kappa_{cd}|||_{cd}+|||\kappa_{cc}|||_{cc})\\
|||\kappa|||_{cc} &= |||\kappa|||_{dd}\\
&:= N^{-1}\sqrt{\sum_{b_2,a_3}(\sum_{a_2,b_3}\sum_{\alpha_1}|\kappa(\alpha_1,a_2b_2,a_3b_3)|)^2}\\
|||\kappa|||_{cd} &:= N^{-1}\sqrt{\sum_{b_3,a_1}(\sum_{a_3,b_1}\sum_{\alpha_2}|\kappa(a_1b_1,\alpha_2,a_3b_3)|)^2},\\
|||\kappa|||_{dc} &:= N^{-1}\sqrt{\sum_{b_1,a_2}(\sum_{a_1,b_2}\sum_{\alpha_3}|\kappa(a_1b_1,a_2b_2,\alpha_3)|)^2}\\
|||\kappa|||_2^{iso} &:= \inf_{\kappa=\kappa_d+\kappa_c}(|||\kappa_d|||_d+|||\kappa_c|||_c),\\
|||\kappa|||_d &:= \sup_{||\boldsymbol{x}||\leq\mathbf{1}}||\,||\,\kappa(\boldsymbol{x}*,\cdot*)||\,||,\\
|||\kappa|||_c &:= \sup_{||\boldsymbol{x}||\leq\mathbf{1}}||\,||\kappa(\boldsymbol{x}*,*\cdot)||\,||\\
|||\kappa|||_k^{iso} &:= ||\sum_{\alpha_1,\ldots,\alpha_{k-2}}|\kappa(\alpha_1,\ldots,\alpha_{k-2},*,*)|\,||, k\geq 3
\end{aligned}
\tag{22a}\tag{22b}\tag{22c}
$$

where in eq. (22b)*, the infimum is taken over all decomposition of* κ *in four symmetric components* $\kappa_{dd},\kappa_{dc},\kappa_{cd},\kappa_{cc}$*; in eq.* (22c) *the infimum is taken over all decomposition of* κ *into the sum of symmetric* κ_c *and* κ_d*. The norms defined in eq.* (22a) *and eq.* (22c) *need some explanation on the notation. If in place of an index* $\alpha\in\mathbb{J}$*, we write a dot* $(\cdot)$ *in a scalar quantity then we consider the quantity as a vector indexed by the coordinate at the place of the dot. For example,* $\kappa(a_1\cdot,a_2b_2)$ *is a vector, the* i*-th entry of which is* $\kappa(a_1i,a_2b_2)$ *and therefore the inner norms in eq.* (22c) *indicate vector norms. In contrast, the outer norms indicate the operator norm of the matrix indexed by star* $(*)$*. More specifically,* $||\boldsymbol{A}(*,*)||$ *refers to the operator norm of the matrix with matrix elements* $\boldsymbol{A}_{ij}$*. Thus* $||\,||\kappa(\boldsymbol{x}*,*\cdot)||\,||$ *is the operator norm* $||\boldsymbol{A}||$ *of the matrix* $\boldsymbol{A}$ *with matrix elements* $\boldsymbol{A}_{ij}=||\kappa(\boldsymbol{x}i,j\cdot)||$*.* $\kappa(\boldsymbol{x}b_1,a_2b_2)$ *denotes* $\sum_{a_1}\kappa(a_1b_1,a_2b_2)x_{a_1}$*, where* $\boldsymbol{x}$ *is a vector.*

Before proceeding further, we expand the remark 2.8 in Erdos et al. [25] to explain the symmetric decomposition in the above definition.

Example 1 (Symmetry decomposition)**.** *The norms essentially define different ways that the maximal weighted average value of cumulants could be characterized. To use the decomposition of the form $\kappa = k_d + k_c$ as an example, the calculation in Erdos et al. [25] that uses norm $||\kappa_c||_c, ||\kappa_d||_d$ resembles the following form.*

$$\boldsymbol{K}_{kl}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{ij} x_i \kappa(kl, ij) y_j$$

where , and $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ and $\boldsymbol{K}$ is a matrix function in $\mathbb{R}^{n\times n} \mapsto \mathbb{R}^{n\times n}$. Two ways exist to estimate the functional norm of $\boldsymbol{K}$ used in Erdos et al. [25]: first assimilate $\boldsymbol{y}$ by treating $||\kappa(\boldsymbol{y}*, *\cdot)||$ as an entry of a matrix, and calculating its spectral norm; or, first assimilate $\boldsymbol{x}$ by treating $||\kappa(\boldsymbol{x}*, *\cdot)||$ as an entry of a matrix, and calculating its spectral norm. To achieve the sharpest bound, its infinum is taken.

To see how the two norms corresponds to the symmetry structure in $\boldsymbol{W}$ in eq. (19), we look at the Wigner matrix, in which we have

$$\begin{aligned}\kappa(a_1 b_1, a_2 b_2) &= \delta_{a_1,a_2}\delta_{b_1,b_2} + \delta_{a_1,b_2}\delta_{b_1,a_2} \\ &=: \kappa_d(a_1 b_1, a_2 b_2) + \kappa_c(a_1 b_1, a_2 b_2).\end{aligned}$$

The cumulant naturally splits into a direct and a cross part κ_d and κ_c. Let $\boldsymbol{K}^d$ denote the matrix induced in the norm $|||\kappa_d|||$, then $\boldsymbol{K}^d_{ij} = ||\kappa_d(\boldsymbol{x}i, \cdot j)||$, which only involves the cumulant between the ith column $\boldsymbol{W}_{:,i}$ and the jth column $\boldsymbol{W}_{:,j}$. Note that due to symmetry $\boldsymbol{W}_{:,j}$ is the same as $\boldsymbol{W}_{j,:}$. Thus, to characterize/bound the cumulant between $\boldsymbol{W}_{:,i}$ and $\boldsymbol{W}_{:,j}$, the norm defined should also consider $\boldsymbol{W}_{:,j}$ as well. By decomposing κ into κ_d, κ_c, the cumulant involving $\boldsymbol{W}_{j,:}$ is taken care in $\boldsymbol{K}^c$, where similarly, $\boldsymbol{K}^c_{ij}$ only involves with the ith column $\boldsymbol{W}_{:,i}$ and jth row $\boldsymbol{W}_{j,:}$. Thus, by decomposing κ into κ_d, κ_c, the cumulant arisen due to symmetry can be tracked separately. Such symmetry decomposition is used to separately estimate the cumulants involved in the multivariate cumulant expansion in the paper, which results in "$\kappa_c(\alpha_1, \alpha_2)$ forces $\alpha_1 = (a_1, b_1)$ to be close to $\alpha_2 = (a_2, b_2)$, whereas $\kappa_d(\alpha_1, \alpha_2)$ forces (a_1, b_1) to be close to (b_2, a_2)" [25]. It characterizes the collective behaviors of cumulants similarly with $||\kappa||_2^{av}$, which is explained in cond. 4.4, though differently.

D.3.4 Eigenspectrum (density function of eigenvalues) of random matrices with slow decay

Equipped with the cumulant norms, we can state the conditions that make MDE valid. The following boundedness conditions are higher order correlation version of the simplified one stated in cond. 4.4.

Condition D.1 (Bounded Mean)**.** $\exists C \in \mathbb{R}, \forall N \in \mathbb{N}, ||\boldsymbol{A}|| \leq C$*, where $||\boldsymbol{A}||$ denotes the operator norm.*

Condition D.2 (Boundedness)**.** *1)* $\forall q \in \mathbb{N}, \exists \mu_q \in \mathbb{R}, \forall \alpha \in \mathbb{I}, \mathbb{E}[|\boldsymbol{W}_\alpha|^q] \leq \mu_q$*. 2)* $\forall R \in \mathbb{N}, \epsilon > 0, \exists C_1, C_2(\epsilon) \in \mathbb{R}$*, where $C_2(\epsilon)$ depends on ϵ, we have $|||\kappa|||_2^{iso} \leq C_1, |||\kappa||| \leq C_2(\epsilon) N^\epsilon$;*

Condition D.3. *There exists $\mu > 0$ such that the following holds: for every $\alpha \in \mathbb{I}$ and $q, R \in \mathbb{N}$, there exists a sequence of nested sets $\mathcal{N}_k = \mathcal{N}_k(\alpha)$ such that $\alpha \in \mathcal{N}_1 \subset \mathcal{N}_2 \subset \cdots \subset \mathcal{N}_R = \mathcal{N} \subset \mathbb{I}, |\mathcal{N}| \leq N^{1/2-\mu}$ and*

$$\kappa(f(\boldsymbol{W}_{\mathbb{I}\setminus\cup_j \mathcal{N}_{n_j+1}(\alpha_j)}), g_1(\boldsymbol{W}_{\mathcal{N}_{n_1}(\alpha_1)\setminus\cup_{j\neq 1}\mathcal{N}(\alpha_j)}), \ldots, g_q(\boldsymbol{W}_{\mathcal{N}_{n_q}(\alpha_q)\setminus\cup_{j\neq q}\mathcal{N}(\alpha_j)})) \leq N^{-3q}||f||_{q+1}\prod_{j=1}^{q}||g_j||_{q+1}$$

, for any $n_1, \ldots, n_q < R$, $\alpha_1, \ldots, \alpha_q \in \mathbb{I}$ and non-constant real analytic functions $f, g_1, \ldots, g_q$, where $||||_p$ is the L^p norm on function space. We call the set $\mathcal{N}$ of α the **coupling set** *of α.*

Remark D.1. *In Erdos et al. [25], the functions $f, g_1, \ldots, g_q$ in cond. D.3 are assumed to be functions without any qualifiers. We change it to non-constant analytic functions for further usage. In the proof of theorem 2, the functions are only required to be analytic, and obviously not constant functions, thus even if the conditions are changed, the conclusion still holds.*

The diversity condition requires that a matrix entry w_α only couples with a minority of the overall entries, and for the rest of the entries, the coupling strength does not exceed a certain value $N^{-3q}||f||_{q+1}\prod_{j=1}^{q}||g_j||_{q+1}$

characterized by cumulants. For example, suppose $q = 1$, given a entry α_1, the condition essentially states that the entries in the coupling set $\mathcal{N}(\alpha_1)$ is not strongly coupled with the resting of the population $\boldsymbol{W}_{\mathbb{I}\setminus\mathcal{N}_{n_1+1}(\alpha_1)}$. The explanation goes similar as q grows, of which the coupling strengh is characterized by higher order cumulants: if we randomly pick q entries from the random matrix $\boldsymbol{H}$, the correlations of their coupling sets, characterized by cumulants κ, is less than a certain value. While boundedness conditions state the expectation of $\boldsymbol{H}$ is bounded, moments are finite, cumulants are bounded for the entries that do strongly couples. Explanation in more details in the second cumulant case can be found in section 3.4 and section 4.1 previously.

When the conditions D.1 D.2 D.3 4.5 are satisfied, we have the resolvent $\boldsymbol{G}$ of a random matrix $\boldsymbol{H}$ close to the solution to the MDE probabilistically with some regularity properties as the following, adopted from Erdos et al. [25] theorem 2.2, Helton et al. [38] theorem 2.1, and Alt et al. [1] theorem 2.5.

Theorem 2. *Assume the conditions D.1 D.2 D.3 4.5 are satisfied for a given symmetric random matrix $\boldsymbol{H}$. Let $\boldsymbol{M}$ be the solution to the Matrix Dyson Equation eq. (19), and ρ the density function (measure) recovered from normalized trace $\frac{1}{N}tr\ \boldsymbol{M}$ through Stieljies inverse lemma D.1. We have*

a) The MDE has a unique solution $\boldsymbol{M} = \boldsymbol{M}(z)$ for all $z \in \mathbb{H}$.

b) suppρ is a finite union of closed intervals with nonempty interior. Moreover, the nonempty interiors are called the **bulk** *of the eigenvalue density function ρ.*

c) The resolvent $\boldsymbol{G}$ of $\boldsymbol{H}$ converges to $\boldsymbol{M}$ as $N \to \infty$. The convergence rate is given by the probability bound present as follows. For any $\gamma, \epsilon > 0$, there exists $\delta > 0$, such that for all $D > 0$, give any $z \in \mathbb{D}_\gamma^\delta$, we have

$$P(|\langle \boldsymbol{B}, (\boldsymbol{G}(z) - \boldsymbol{M}(z))\rangle| \leq ||\boldsymbol{B}|| \frac{N^\epsilon}{(1+|z|)^2 N}) \geq 1 - CN^{-D}$$

where $\boldsymbol{B}$ is an arbitrary deterministic matrix, N is the dimension of $\boldsymbol{H}$, $C > 0$ is a constant depending on D, ϵ, γ and constants in cond. D.1 D.2 D.3. ϵ can be chosen small, so $||\boldsymbol{x}||||\boldsymbol{y}||N^\epsilon/\sqrt{N\Im z}$ goes zero as N grows. The region $\mathbb{D}_\gamma^\delta$ is roughly the region in the complex plane around the intervals that are inside the support of μ. Formally, it is defined as

$$\mathbb{D}_\gamma^\delta := \{z \in \mathbb{H} \mid |z| \leq N^{C_0}, \Im z \geq N^{-1+\gamma}, \mu_{\boldsymbol{H}}(x) + dist(x, supp\mu_{\boldsymbol{H}}) \geq N^{-\delta}\},$$

where $\mathbb{H}$ is the complex domain, C_0 is a constant larger than 100, $\mu_{\boldsymbol{H}}$ is the empirical spectral distribution of $\boldsymbol{H}$ (c.f. definition D.1), and supp$(\mu_{\boldsymbol{H}})$ denotes the support of $\mu_{\boldsymbol{H}}$. Roughly, $\mathbb{D}_\gamma^\delta$ characterizes the region inside the support of $\mu_{\boldsymbol{H}}$, which is normally denoted as the bulk of the eigenspectrum.

D.4 Condition simplification in Gaussian ensembles

We explain why cond. 4.4 in section 4.1 and cond. 3.1 in section 3.4 are simplified the version of cond. D.2 in appendix D.3.4 and D.5 in appendix D.5.3 respectively.

Recall that the objective function is the empirical risk function $R_m(T)$ at eq. (1). Given a set of i.i.d. training samples $(X_i, Y_i)_{i=1,\ldots,m}$, $R_m(T)$ is a summation of the i.i.d. random matrices. Formally, reusing the notation to denote $\boldsymbol{H}$ the Hessian of $R_m(T)$ and $\boldsymbol{H}_i$ the Hessian of $l(T(X_i;\boldsymbol{\theta}), Y_i)$, we have

$$\boldsymbol{H} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{H}_i \tag{23}$$

By the multivariate central limiting theorem (Klenke [45] theorem 15.57), $\boldsymbol{H}$ will converge to a Gaussian ensemble, i.e., a random matrix of which the distribution of entries is a Gaussian process (GP), asymptotically as $m \to \infty$. This is why we need cond. 4.1.

When $\boldsymbol{H}$ is a Gaussian ensemble, all cumulants of order higher than the second vanishes, thus the diversity condition is solely about the second order cumulants, and the case when $q = 1$. So are higher cumulants in the boundedness condition. Thus, it suffices to have cond. 4.4 3.1 only involve cumulants up to second order. Some extra notes are needed for cond. D.5. In cond. 3.1, we only need the $q = 1$ is due to the how multivariate cumulant expansion is used in the proof in [25], which is rather involved and cannot be explained without delving into the proof. We suggest the interested readers to read the paper [25], since it is not possible to explain a close-to-100-page proof here.

D.5 The computable form of the diversity condition

This section can be safely skipped if the readers are not interested in how the computable conditions are derived from the cond. D.3 in appendix D.3.4 for the experiments in section 5.1, and are only interested in the proof of theorem 1.

In this section, we show that cond. D.3 can be implied from a computable condition: given k Hessian entries $\{w_{\alpha_i}\}_{i=1,\dots,k}$, if for any k-way statistics/cumulants $\kappa(\beta_1,\dots,\beta_k)$, where $\forall i \in [k-1], \beta_i \in \mathcal{N}(\alpha_i)$ (or verbosely, $k-1$ of them are from the coupling sets of $\{w_{\alpha_i}\}_{i=1,\dots,k}$), if $\beta_k \in \mathbb{I} \setminus \cup_{i=1}^{k}\mathcal{N}(\alpha_i)$, then $\kappa(\beta_1,\dots,\beta_k)=0$. Note that the order of β_k is exchangeable, thus we can assume the last one is not from the coupling sets without loss of generality. We state the above computable condition formally in appendix D.5.3. Before that, we first explain how the computable form of the simplified diversity condition cond. 3.1 in section 3.4 is derived from a simplified version of cond. D.3 in appendix D.5.2.

D.5.1 Preliminaries

To begin with, we introduce the definition of multi-set, following the notation in Erdos et al. [25].

Definition D.8 (Multiset)**.** *A multiset is an unordered set with possible multiple appearances of the same element. We put a underline under a normal letter, e.g., $\underline{\beta}$, to denote it is a multiset.*

Notation. Notice that a normal set can also be taken as a multiset, so given a multiset $\underline{\beta}$, and a set $\mathcal{B}$, $\underline{\beta} \subset \mathcal{B}$ means the underlying set of $\underline{\beta}$, i.e., the unique elements of $\underline{\beta}$ are from $\mathcal{B}$. We also follow our convention that $\boldsymbol{W}_{\underline{\beta}}$ denotes the vector consists of entries at $\underline{\beta}$. We also use a shorthand to denote cumulant between multisets. Given a multiset $\underline{\beta}$, $\kappa(\underline{\beta})$ denotes $\kappa(\beta_1,\dots,\beta_p), \beta_i \in \underline{\beta}, i=1,\dots,p, p\in\mathbb{N}$. Given two multisets $\underline{\beta},\underline{\gamma}$, $\kappa(\underline{\beta},\underline{\gamma})$ denotes $\kappa(\beta_1,\dots,\beta_p,\gamma_1,\dots,\gamma_q), \beta_i \in \underline{\beta}, i=1,\dots,p, \gamma_j \in \underline{\beta}, j=1,\dots,q, p,q\in\mathbb{N}$.

D.5.2 Derivation of condition 3.1

We first state the simpler version of cond. D.3 that leads to cond. D.5 under the cond. 4.1.

Condition D.4. *There exists $\mu > 0$ such that the following holds: for every $\alpha \in \mathbb{I}$, there exists two sets $\mathcal{N}_k = \mathcal{N}_k(\alpha), k=1,2$ such that $\alpha \in \mathcal{N}_1 \subset \mathcal{N}_2 = \mathcal{N} \subset \mathbb{I}, |\mathcal{N}| \le N^{1/2-\mu}$ and*

$$\kappa(f(\boldsymbol{W}_{\mathbb{I}\setminus\mathcal{N}}), g(\boldsymbol{W}_{\mathcal{N}_1})) \le N^{-3}||f||_3||g||_3,$$

for any non-constant real analytic functions f,g, where $||||_3$ is the L^3 norm on function space. We call the set $\mathcal{N}$ of α the **coupling set** *of α.*

Under the large-sample-size setting, cond. D.4 can be further simplified to only involve covariances of Hessian entries. The simplification is intuitive since for Gaussian distributions, zero covariance is equivalent to independence. Thus, covariance contains all the information of statistical dependency characterized by cond. D.4. As mentioned in appendix D.3, the second order cumulant is covariance. Since $f, \{g_i\}_{i=1,\dots,q}$ are analytic, $\kappa(f(\cdot), g_1(\cdot))$ can be decomposed into a summation of generalized cumulants (McCullagh [59] Chapter 3) that only involve two Hessian entries. Generalized cumulant can simply be understood as cumulants of function of r.v.s here. Note that by the well known property of cumulant that if all cumulants between two r.v.s are zero, such two r.v.s are statistically independent. Thus, the varnishing of the second order cumulants equates with statistical independence. Consequently, any r.v.s obtained by applying functions on these two r.v.s respectively are statistically independent. So any generalized cumulants between these two r.v.s are zero, which implies the decay of $\kappa(f(\cdot), g_1(\cdot))$. We formalize the phenomenon in the following.

Corollary D.1. *If cond. 4.1, cond. 4.3 and cond. 3.1 hold, then cond. D.4 holds.*

Proof of corollary D.1. Note that f,g are analytic functions, thus $\kappa(f(\boldsymbol{W}_{\mathbb{I}\setminus\mathcal{N}}), g(\boldsymbol{W}_{\mathcal{N}_1}))$ in cond. 3.1 can be decomposed into generalized cumulants (McCullagh [59] Chapter 3). By cond. 4.1, as explained in appendix D.4, Hessian entries are of Gaussian distribution, and thus all cumulants of higher order than the second are zero.

Only the second order cumulants are left. Since the lemma requires $\forall \beta \in \mathbb{I} \setminus \mathcal{N}, \forall \gamma \in \mathcal{N}_1, \kappa(w_\beta, w_\gamma) = 0$, they varnishes as well. Thus, the $\kappa(f(\boldsymbol{W}_{\mathbb{I}\setminus\mathcal{N}}), g(\boldsymbol{W}_{\mathcal{N}_1}))$ is zero, and cond. 3.1 holds. We formalize the statements as follows.

For any analytic function $f(\boldsymbol{w})$, where $\boldsymbol{w}$ is a vector (c.f. definition D.5), it can be written as a polynomial series expanded at zero as

$$f(\boldsymbol{w}) = \sum_{\boldsymbol{m}} \frac{1}{\boldsymbol{m}!} \frac{\partial^{|\boldsymbol{m}|} f(\boldsymbol{0})}{\partial^{\boldsymbol{m}} \boldsymbol{w}} \boldsymbol{w}^{\boldsymbol{m}}$$

where the sum $\sum_{\boldsymbol{m}}$ is the sum over all n-dimensional multi-indices $\boldsymbol{m} := (m_1, \cdots, m_n)$ (c.f. definition D.6), and $\partial^{\boldsymbol{m}} \boldsymbol{w}$ denotes $\prod_{i=1}^{n} \partial w_{m_i}$. Thus, $\kappa(f(\boldsymbol{W}_{\mathbb{I}\setminus\mathcal{N}}), g(\boldsymbol{W}_{\mathcal{N}_1}))$ can be decomposed as

$$\kappa(f(\boldsymbol{W}_{\mathbb{I}\setminus\mathcal{N}}), g(\boldsymbol{W}_{\mathcal{N}_1})) = \kappa(\sum_{\boldsymbol{m}} \frac{1}{\boldsymbol{m}!} \frac{\partial^{|\boldsymbol{m}|} f(\boldsymbol{0})}{\partial^{\boldsymbol{m}} \boldsymbol{W}_{\mathbb{I}\setminus\mathcal{N}}} \boldsymbol{W}_{\mathbb{I}\setminus\mathcal{N}}^{\boldsymbol{m}}, \sum_{\boldsymbol{m}'} \frac{1}{\boldsymbol{m}'!} \frac{\partial^{|\boldsymbol{m}'|} g(\boldsymbol{0})}{\partial^{\boldsymbol{m}'} \boldsymbol{W}_{\mathcal{N}_1}} \boldsymbol{W}_{\mathcal{N}_1}^{\boldsymbol{m}'})$$

In this case, the n for the multi-index $\boldsymbol{m}$ is $|\mathbb{I} \setminus \mathcal{N}|$ and n for the multi-index $\boldsymbol{m}'$ is $|\mathcal{N}_1|$.

Notice that covariance κ is bilinear, and since f, g are not constant functions, at least one random variable from each of the sets $\mathbb{I} \setminus \mathcal{N}, \mathcal{N}_1$ would be left. Thus, the above equation would be of the summation of many terms as the following

$$\sum_{\underline{\beta} \subseteq \mathcal{N}_1 \underline{\gamma} \subseteq \mathbb{I}\setminus\mathcal{N}} c_{\underline{\beta\gamma}} \kappa \left(g^{\underline{\beta}}(\boldsymbol{W}_{\underline{\beta}}), f^{\underline{\gamma}}(\boldsymbol{W}_{\underline{\gamma}}) \right),$$

where $g^{\underline{\beta}}(\boldsymbol{W}_{\underline{\beta}}), f^{\underline{\gamma}}(\boldsymbol{W}_{\underline{\gamma}})$ are monomials whose factors are from the multisets $\underline{\beta}, \underline{\gamma}$ respectively, and $c_{\underline{\beta\gamma}}$ are constants indexed by $\underline{\beta\gamma}$.

By cond. 4.1, cumulants that are of third order or higher varnish, and by cond. 4.3, the means (first order cumulants) varnish as well. Thus, only pairwise cumulants of the second order of $\kappa(g^{\underline{\beta}}(\boldsymbol{W}_{\underline{\beta}}), f^{\underline{\gamma}}(\boldsymbol{W}_{\underline{\gamma}}))$ are left. Thus, $\kappa(f(\boldsymbol{W}_{\mathbb{I}\setminus\mathcal{N}}), g(\boldsymbol{W}_{\mathcal{N}_1}))$ would be of the form of the summation as follows

$$\sum_{\beta \in \mathbb{I}\setminus\mathcal{N}, \gamma \in \mathcal{N}_1} c_{\beta\gamma} \kappa(w_\beta, w_\gamma).$$

where again $c_{\beta\gamma}$ are constants.

Since $\forall \beta \in \mathbb{I} \setminus \mathcal{N}, \forall \gamma \in \mathcal{N}_1, \kappa(w_\beta, w_\gamma) = 0$ (that is, the pairwise second order cumulants varnish), the above summation is zero. Thus, cond. 3.1 holds. □

D.5.3 Computable form of condition D.5

Condition D.5 (Diversity)**.** *There exists $\mu > 0$ such that the following holds: for every $\alpha \in \mathbb{I}$ and $q, R \in \mathbb{N}$, there exists a sequence of nested sets $\mathcal{N}_k = \mathcal{N}_k(\alpha)$ such that $\alpha \in \mathcal{N}_1 \subset \mathcal{N}_2 \subset \cdots \subset \mathcal{N}_R = \mathcal{N} \subset \mathbb{I}, |\mathcal{N}| \leq N^{1/2-\mu}$, and for any $n_1, \ldots, n_q < R$, $\alpha_1, \ldots, \alpha_q \in \mathbb{I}$, $\forall \underline{\beta} \subseteq \cup_{j=1}^{q} \mathcal{N}_{n_j}(\alpha_j)$ and $\forall \underline{\gamma} \subseteq \mathbb{I} \setminus \cup_j \mathcal{N}_{n_j+1}(\alpha_j)$, the following holds*

$$\kappa(\underline{\beta}, \underline{\gamma}) = 0.$$

Lemma D.2. *If cond. D.5 holds, then cond. D.3 holds.*

Proof of lemma D.2. The logic is the same as the proof of the simpler corollary D.1, except that higher order terms are involved. Previously in the proof of corollary D.1, we only need to calculate the second order cumulant as follows

$$\kappa(f(\boldsymbol{W}_{\mathbb{I}\setminus\mathcal{N}}), g(\boldsymbol{W}_{\mathcal{N}_1})).$$

Now, we need to deal with the cumulant of all orders as follows

$$\kappa(f(\boldsymbol{W}_{\mathbb{I}\setminus\cup_j \mathcal{N}_{n_j+1}(\alpha_j)}), g_1(\boldsymbol{W}_{\mathcal{N}_{n_1}(\alpha_1)\setminus\cup_{j\neq 1}\mathcal{N}(\alpha_j)}), \ldots, g_q(\boldsymbol{W}_{\mathcal{N}_{n_q}(\alpha_q)\setminus\cup_{j\neq q}\mathcal{N}(\alpha_j)})).$$

Since all $f, g_i, i = 1, \ldots, q$ are analytic, it can be written as

$$\kappa\Big(\sum_{\boldsymbol{m}_f} \frac{1}{\boldsymbol{m}_f!} \frac{\partial^{|\boldsymbol{m}_f|} f(\boldsymbol{0})}{\partial^{\boldsymbol{m}_f} \boldsymbol{W}_{\mathbb{I}\setminus\cup_j \mathcal{N}_{n_j+1}(\alpha_j)}} \boldsymbol{W}^{\boldsymbol{m}_f}_{\mathbb{I}\setminus\cup_j \mathcal{N}_{n_j+1}(\alpha_j)},$$
$$\sum_{\boldsymbol{m}_{g_1}} \frac{1}{\boldsymbol{m}_{g_1}!} \frac{\partial^{|\boldsymbol{m}_{g_1}|} g(\boldsymbol{0})}{\partial^{\boldsymbol{m}_{g_1}} \boldsymbol{W}_{\mathcal{N}_{n_1}(\alpha_1)\setminus\cup_{j\neq 1}\mathcal{N}(\alpha_j)}} \boldsymbol{W}^{\boldsymbol{m}_{g_1}}_{\mathcal{N}_{n_1}(\alpha_1)\setminus\cup_{j\neq 1}\mathcal{N}(\alpha_j)},$$
$$\ldots,$$
$$\sum_{\boldsymbol{m}_{g_2}} \frac{1}{\boldsymbol{m}_{g_2}!} \frac{\partial^{|\boldsymbol{m}_{g_2}|} g(\boldsymbol{0})}{\partial^{\boldsymbol{m}_{g_2}} \boldsymbol{W}_{\mathcal{N}_{n_q}(\alpha_q)\setminus\cup_{j\neq q}\mathcal{N}(\alpha_j)}} \boldsymbol{W}^{\boldsymbol{m}_{g_2}}_{\mathcal{N}_{n_q}(\alpha_q)\setminus\cup_{j\neq q}\mathcal{N}(\alpha_j)}\Big)$$

The multilinearity of cumulant and the non-constant of f implies that the above equation would be of the summation of many terms as follows

$$\sum_{\underline{\beta}_i \subseteq \mathcal{N}_{n_i}(\alpha_i)\setminus\cup_{j\neq 1}\mathcal{N}(\alpha_j), i=1,\ldots,p, \underline{\gamma}\subseteq \mathbb{I}\setminus\cup_j \mathcal{N}_{n_j+1}(\alpha_j)} c_{\underline{\beta}_1\ldots\underline{\beta}_q\underline{\gamma}} \kappa\left(g_1^{\underline{\beta}_1}(\boldsymbol{W}_{\underline{\beta}_1}), \ldots, g_p^{\underline{\beta}_p}(\boldsymbol{W}_{\underline{\beta}_p}), f^{\underline{\gamma}}(\boldsymbol{W}_{\underline{\gamma}})\right),$$

where $g_i^{\underline{\beta}_i}(\boldsymbol{W}_{\underline{\beta}_i}), i = 1, \ldots, p, f^{\underline{\gamma}}(\boldsymbol{W}_{\underline{\gamma}})$ are monomials whose factors are from the multisets $\underline{\beta}_i, i = 1, \ldots, p, \underline{\gamma}$ respectively, and $c_{\underline{\beta}_1\ldots\underline{\beta}_q\underline{\gamma}}$ are constants indexed by $\underline{\beta}_1 \ldots \underline{\beta}_q\underline{\gamma}$.

Each $\kappa(g_1^{\underline{\beta}_1}(\boldsymbol{W}_{\underline{\beta}_1}), \ldots, g_p^{\underline{\beta}_p}(\boldsymbol{W}_{\underline{\beta}_p}), f^{\underline{\gamma}}(\boldsymbol{W}_{\underline{\gamma}})$ is a generalized cumulant (McCullagh [59] Chapter 3), and would be decomposed into summation of ordinary cumulants of the form $\kappa(\underline{\beta}, \underline{\gamma})$, where $\underline{\beta} \subseteq \cup_{j=1}^q \mathcal{N}_{n_j}(\alpha_j)$ and $\underline{\gamma} \subseteq \mathbb{I} \setminus \cup_j \mathcal{N}_{n_j+1}(\alpha_j)$. Thus, if the ordinary cumulants are zero, so are any of their combinations.

Thus, cond. D.5 holds. □

D.6 NN loss landscape: all local minima are global minima with zero losses

We have obtained the operator equation to describe the eigenspectrum of the Hessian of NNs and explained its conditions. With one further condition, we show in this section that for NNs with objective function belonging to the function class $\mathcal{L}_0$, all local minima are global minima with zero loss values.

We outline the strategy first. Since MDE is a nonlinear operator equation, it is not possible to obtain a close form analytic solution. The only way to get its solution is an iterative algorithm [38], which is not an easy task given the millions of parameters of a NN — remembering that we are dealing with large N limit — though it can serve as an exploratory tool. However, we do are able to get qualitative results by directly analyzing the equation. Our goal is to show all critical points are saddle points, except for the ones has zero loss values, which are global minima. To prove it, we prove that at the points where $R_m(T) \neq 0$, the eigenspectrum $\mu_{\boldsymbol{H}}$ of the Hessian $\boldsymbol{H}$ is symmetric w.r.t. the y-axis, which implies that as long as non-zero eigenvalues exist, half of them will be negative. To prove it, we prove the stieltjes transform $m_{\mu_{\boldsymbol{H}}}(z)$ of $\mu_{\boldsymbol{H}}$ satisfies $\Im m_{\mu_{\boldsymbol{H}}}(-z^*) = \Im m_{\mu_{\boldsymbol{H}}}(z)$, where z^* denote the complex conjugate of z. In the following, we present the proof formally.

Lemma D.3. *Let $\boldsymbol{M}(z), \boldsymbol{M}''(-z^*)$ be the unique solution to the MDE at spectral parameter $z, -z^*$ defined at eq. (19) respectively, and $\boldsymbol{A} = \boldsymbol{0}$. We have*

$$\boldsymbol{M}' = -\boldsymbol{M}^*$$

where $*$ means taking conjugate transpose.

Proof. First, we rewrite the MDE. Note that $\mathcal{S}[\boldsymbol{G}]$ is positivity preserving, i.e., $\forall \boldsymbol{G} \succ \boldsymbol{0}, \mathcal{S}[\boldsymbol{G}] \succ \boldsymbol{0}$ by condition 4.5. In addition, we have $\Im z > 0$, thus $\Im(z + \mathcal{S}[\boldsymbol{G}]) \succ \boldsymbol{0}$. Then, by Haagerup and Thorbjørnsen [33] lemma 3.1.(ii), we have $z + \mathcal{S}[\boldsymbol{G}]$ invertable. Thus, we can rewrite the MDE into the following form

$$\boldsymbol{G} = -(z + \mathcal{S}[\boldsymbol{G}])^{-1} \tag{24}$$

Suppose $\boldsymbol{M}$ is a solution to the MDE at spectral parameter z. The key to the proof is the fact that $\mathcal{S}[\boldsymbol{G}]$ is linear and commutes with taking conjugate, thus by replacing $\boldsymbol{M}$ with $-\boldsymbol{M}^*$, and z with $-z^*$, we would get the same equation. We show it formally in the following.

First, note that $\mathcal{S}[\boldsymbol{M}]$ is a linear map of $\boldsymbol{M}$, so the we have

$$\mathcal{S}[-\boldsymbol{M}] = -\mathcal{S}[\boldsymbol{M}]$$

Also, $\mathcal{S}[\boldsymbol{M}]$ commutes with $*$, for the fact

$$\mathcal{S}[\boldsymbol{M}^*] = \mathbb{E}[\boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W}] = \mathbb{E}[(\boldsymbol{W}\boldsymbol{M}\boldsymbol{W})^*] = \mathbb{E}[\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}]^* = \mathcal{S}[\boldsymbol{M}]^*$$

Furthermore, $*$ commutes with taking inverse, for the fact

$$\begin{aligned} & \boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{I} \\ \Longrightarrow \quad & (\boldsymbol{A}\boldsymbol{A}^{-1})^* = \boldsymbol{I} \\ \Longrightarrow \quad & \boldsymbol{A}^{-1*}\boldsymbol{A}^* = \boldsymbol{I} \\ \Longrightarrow \quad & \boldsymbol{A}^{-1*} = \boldsymbol{A}^{*-1} \end{aligned}$$

With the commutativity results, we do the proof. The solution $\boldsymbol{M}$ satisfies the equation

$$\boldsymbol{M} = -(z + \mathcal{S}[\boldsymbol{M}])^{-1}$$

Replacing $\boldsymbol{M}$ with $-\boldsymbol{M}^*$, z with $-z^*$, we have

$$\begin{aligned} & -\boldsymbol{M}^* = -(-z^* + \mathcal{S}[-\boldsymbol{M}^*])^{-1} \\ \Longrightarrow \quad & \boldsymbol{M}^* = -(z^* + \mathcal{S}[\boldsymbol{M}^*])^{-1} \\ \Longrightarrow \quad & \boldsymbol{M}^* = -(z^* + \mathcal{S}[\boldsymbol{M}]^*)^{-1} \\ \Longrightarrow \quad & \boldsymbol{M}^* = -(z + \mathcal{S}[\boldsymbol{M}])^{-1*} \\ \Longrightarrow \quad & \boldsymbol{M} = -(z + \mathcal{S}[\boldsymbol{M}])^{-1} \end{aligned}$$

After the replacement, we actually get the same equation. Thus, $-\boldsymbol{M}^*, -z^*$ also satisfy eq. (24). Since the pair also satisfies the constrains $\Im \boldsymbol{M} \succ 0, \Im z > 0$, and by theorem 2, the solution is unique, we proved the solution $\boldsymbol{M}'$ at the spectral parameter $-z^*$ is $-\boldsymbol{M}^*$.

□

Theorem 3. *Let $\boldsymbol{H}$ be a real symmetric random matrix satisfies cond. 4.2 D.2 D.5 4.5, in addition to the condition that $\boldsymbol{A} = \boldsymbol{0}$. Let the ESD of $\boldsymbol{H}$ be $\mu_{\boldsymbol{H}}$. Then, $\mu_{\boldsymbol{H}}$ is symmetric w.r.t. to y-axis. In other words, half of the non-zero eigenvalues are negative. Furthermore, non-zero eigenvalues always exist, implying $\boldsymbol{H}$ will always have negative eigenvalues.*

Proof. By theorem 2, the resolvent $\boldsymbol{G}$ of $\boldsymbol{H}$ is given by the unique solution to eq. (19) at spectral parameter z. Let the solution to eq. (19) at spectral parameter $z, -z^*$ be $\boldsymbol{M}, \boldsymbol{M}'$, By lemma D.3, we have the solutions satisfies

$$\boldsymbol{M}' = -\boldsymbol{M}^*$$

By lemma D.1, the ESD of $\boldsymbol{H}$ at $\Re z$ is given at

$$\mu_{\boldsymbol{H}}(\Re z) = \lim_{\Im z \to 0} \frac{1}{\pi} \Im m_{\mu_{\boldsymbol{H}}}(z)$$

Since $m_{\mu_{\boldsymbol{H}}}(z) = \frac{1}{N} \operatorname{tr} M$, we have

$$\mu_{\boldsymbol{H}}(\Re z) = \lim_{\Im z \to 0} \frac{1}{\pi} \frac{1}{N} \Im \operatorname{tr} M$$

Similarly,

$$\mu_{\boldsymbol{H}}(\Re(-z^*)) = \lim_{\Im(-z^*)\to 0} \frac{1}{\pi}\frac{1}{N}\Im\text{tr}\ M'$$

Note that

$$\begin{aligned} & \mu_{\boldsymbol{H}}(\Re(-z^*)) = \lim_{\Im(-z^*)\to 0} \frac{1}{\pi}\frac{1}{N}\Im\text{tr}\ M' \\ \implies \quad & \mu_{\boldsymbol{H}}(\Re(-z^*)) = \lim_{\Im(-z^*)\to 0} \frac{1}{\pi}\frac{1}{N}\Im\text{tr}\ (-M^*) \\ \implies \quad & \mu_{\boldsymbol{H}}(-\Re z) = \lim_{\Im z\to 0} \frac{1}{\pi}\frac{1}{N}\Im\text{tr}\ M \end{aligned}$$

Thus, $\mu_{\boldsymbol{H}}(\lambda), \lambda \in \mathbb{R}$ is symmetric w.r.t. y-axis. It follows that for all non-zero eigenvalues, half of them are negative.

By theorem 2.b, there are always bulks in supp$\mu_{\boldsymbol{H}}$, thus there are always non-zero eigenvalues. Since half of the non-zero eigenvalues are negative, it follows $\boldsymbol{H}$ always has negative eigenvalues. □

Theorem (Theorem 1 restated)**.** *Let $R_m(T)$ be the risk function (defined at eq. (1)) of a NN T having only one output neuron (defined at eq. (3)) with a loss function l of class $\mathcal{L}_0$ (defined in section 3.1). If the Hessian $\boldsymbol{H} \in \mathbb{R}^{N\times N}$ (c.f. eq. (6)) of $R_m(T)$ satisfies conditions 4.2 4.3 D.2 4.5 D.3, then for $R_m(T)$,*

a) all local minima are global minima with zero risk.

b) all non-zero-risk stationary points are saddle points with half of the non-zero eigenvalues negative.

c) A constant $\lambda_0 \in \mathbb{R}$ exists, such that $||\boldsymbol{H}||_2$ is upper bounded by $\mathbb{E}_m[l'(T(X),Y)]\lambda_0$, where $\mathbb{E}_m[l'(T(X),Y)]$ is the empirical expectation of derivative l' of l. It implies the eigenspectrum of $\boldsymbol{H}$ is increasingly concentrated around zero as more examples have the zero loss (classified correctly in term of the surrogate loss) — because for a well-designed loss function, when the loss $l(\boldsymbol{x}, y)$ of an example is zero (property 4 of the loss function class $\mathcal{L}_0$), its derivative at $\boldsymbol{x}$ is zero, i.e., $l'(\boldsymbol{x}, y) = 0$. Thus, the regions around the minima are flat basins.

Remark D.2. *The restatement is slightly different from the one give in section 4.2. That is we do not explicitly assume cond. D.5 in the theorem but assume cond. D.3 instead, since it is assumed mostly to justify the empirical study in section 5.1, and is a sufficient condition for cond. D.3. The proof is actually proved with cond. D.3.*

Remark D.3. *A note on the generality of the theorem. First, though the network T is restricted to be of an MLP defined in eq. (3), it is chosen this way to communicate the core idea better. The results can trivially generalize to arbitrary feed forward type network architecture, e.g., Convolutional Neural Network [48], Residual Network [37].*

Remark D.4. *It is not necessary for the condition $\boldsymbol{A} = \boldsymbol{0}$ to be held for the theorem to hold, or more specifically, for $\boldsymbol{H}$ to have negative eigenvalues at its critical points. By proposition 2.1 in Alt et al. [1]*

$$supp\mu_{\boldsymbol{H}} \subset Spec\boldsymbol{A} + [-2||\mathcal{S}||^{1/2}, 2||\mathcal{S}||^{1/2}]$$

where Spec$\boldsymbol{A}$ denotes the support of the spectrum of the expectation matrix $\boldsymbol{A}$ and $||\mathcal{S}||$ denotes the norm induced by the operator norm. Thus, it is possible for the spectrum $\mu_{\boldsymbol{H}}$ of $\boldsymbol{H}$ to lie at the left side of the y-axis, as long as the spectrum of $\boldsymbol{A}$ is not too way off from the origin. However, existing characterizations on supp$\mu_{\boldsymbol{H}}$ based on bound are too inexact to make sure the existence of support on the left of the y-axis. To get rid of the zero expectation condition, more works are needed to obtain a better characterization, and could be a direction for future work.

Proof of theorem 1. The majority of the proof has been dispersed earlier in the paper. The proof here mostly is to collect them into one piece.

First, we prove (a)(b) of the theorem. The Hessian $\boldsymbol{H}$ of the risk function eq. (1), can be decomposed into a summation of Hessians of loss functions of each training sample, which is described in eq. (15). For each Hessian in the decomposition, it is computed in eq. (13), and it has been shown that $\boldsymbol{H}$ is a random matrix in appendix D.2.

The analysis of the random matrix $\boldsymbol{H}$ needs to break down into two cases: 1) for all training examples, at least one example (x, y) has non-zero loss value; 2) and all training samples are classified properly with zero loss values.

We first analyze case 1), since the loss l belongs to function class $\mathcal{L}_0$, l is convex and is valued zero at its minimum. When $l(x, y) \neq 0$, we have $l'(x, y) \neq 0$, thus $\boldsymbol{H}$ is a random matrix — not a zero matrix. The analysis of this type of random matrix is undertaken in appendix D.3. For a NN satisfies cond. 4.3 D.2 D.5 4.5, and by theorem 2, the eigenspectrum $\mu_{\boldsymbol{H}}$ of $\boldsymbol{H}$ is given by the MDE defined at eq. (19).

By theorem 3, $\mu_{\boldsymbol{H}}$ is symmetric w.r.t. y-axis, and half of its non-zero eigenvalues are negative. Thus, for all critical points of $R_m(T)$, it will have half of its non-zero eigenvalues negative. It implies all critical points are saddle points.

Now we turn to the case 2). In this case, all training samples are properly classified with zero loss value. Considering the lower bound of l is zero, we have reached the global minima. Also, since all critical points in case 1) are saddle points, local minima can only be reached in case 2), implying all local minima are global minima. Thus, the first part of the theorem is proved.

Now we prove (c). Note that the minima is reached for the fact that we have reached the situation where the Hessian $\boldsymbol{H}$ has degenerated into a zero matrix. Thus, each local minimum is not a traditional critical point, but a flat region, where in a local region around the minima in the parameter space, all the eigenvalues are increasingly close to zero as the parameters of the NN approach the parameters at the infimum. We show it formally in the following.

Writing a block $\boldsymbol{H}_{pq}$ (defined at eq. (12)) in the Hessian $\boldsymbol{H}_i$ of one example (defined at eq. (13)) in the form of

$$\boldsymbol{H}_{pq} = l'(T\boldsymbol{x}, y)\tilde{\boldsymbol{H}}_{pq}$$

where i is the index of the training examples, defined at eq. (1). Then, putting together $\tilde{\boldsymbol{H}}_{pq}$ together to form $\tilde{\boldsymbol{H}}_i$, $\boldsymbol{H}_i$ is rewritten in the form of

$$\boldsymbol{H}_i = l'(T\boldsymbol{x}_i, y_i)\tilde{\boldsymbol{H}}_i$$

Then the Hessian $\boldsymbol{H}$ (defined at eq. (15)) of the risk function (defined eq. (1)) can be rewritten in the form of

$$\boldsymbol{H} = \frac{1}{m}\sum_{i=1}^{m} l'(T\boldsymbol{x}_i, y_i)\tilde{\boldsymbol{H}}_i$$

Taking the operator norm on the both sides

$$||\boldsymbol{H}||_2 = ||\frac{1}{m}\sum_{i=1}^{m} l'(T\boldsymbol{x}_i, y_i)\tilde{\boldsymbol{H}}_i||_2 \leq \frac{1}{m}\sum_{i=1}^{m} |l'(T\boldsymbol{x}_i, y_i)|\, ||\tilde{\boldsymbol{H}}_i||_2$$

Denote $\max_i\{||\tilde{\boldsymbol{H}}_i||_2\}$ as λ_0, we have

$$\begin{aligned} ||\boldsymbol{H}||_2 &\leq \frac{1}{m}\sum_{i=1}^{m} |l'(T\boldsymbol{x}_i, y_i)|\lambda_0 \\ &= \mathbb{E}_m[l'(TX, Y)]\lambda_0 \end{aligned}$$

The above inequality shows that, as the risk decreases, more and more samples will have zero loss value, consequently $l' = 0$, thus $\mathbb{E}_m[l']$ will be increasingly small, thus the operator norm of $\boldsymbol{H}$. At the minima where all $l' = 0$, the Hessian degenerates to a zero matrix. □

D.7 Relationship between cumulant/covariance bounds and the resolvent probability bound

We explain the relationship between the constants in the boundedness cond. D.2 (cond. 4.4) and the probability bound in cond. 4.2 (theorem 2) in this section.

First, we outline the logic how the probability bound is proved in Erdos et al. [25] just enough to explain the relationship.

Recall that theorem 2 is established on the observation that if the higher order fluctuations of $\boldsymbol{HG}$ in eq. (17) are negligible, $\boldsymbol{D}$ (eq. (21)) would concentrate on the zero matrix, and the resolvent of Hessian $\boldsymbol{H}$ can be obtained by the Matrix Dyson Equation eq. (19). The concentration of $\boldsymbol{D}$ is characterized through the varnishing of the high moments bound of $\boldsymbol{D}$, which are defined as follows.

$$||X||_p := (\mathbb{E}[|X|^p])^{1/p}, ||\boldsymbol{A}||_p := \sup_{||\boldsymbol{x}||,||\boldsymbol{y}||\leq 1} ||\langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y}\rangle = (\sup_{||\boldsymbol{x}||,||\boldsymbol{y}||\leq 1} \mathbb{E}[|\langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y}\rangle|^p])^{1/p}.$$

To begin with, we reproduce theorem 3.1 of Erdos et al. [25] as follows.

Theorem 4 (Theorem 3.1 in Erdos et al. [25])**.** *Under cond. D.1 D.2 D.5, there exist constant C_*, C such that for any $p \geq 1, \epsilon > 0$, z with $\Im z \geq N^{-1}$, $\boldsymbol{B} \in \mathbb{C}^{N\times N}$, it holds that*

$$\frac{||\langle \boldsymbol{B}, \boldsymbol{D}\rangle||_p}{||\boldsymbol{B}||} \leq C(1 + |||\kappa|||_2^{iso} + |||\kappa|||^{av})N^{\epsilon}\frac{1}{N\Im z} \tag{25}$$

$$\cdot ||\Im \boldsymbol{G}||_q(1 + (1 + |z|)||\boldsymbol{G}||_q)^{C_*/\mu}(1 + \frac{1 + |z|||\boldsymbol{G}||_q}{N^{\mu}})^{C_* p/\mu} \tag{26}$$

where $q = C_* p^4/\mu\epsilon, R = 4p/\mu$.

The bound shows that $\boldsymbol{D}$ (defined in eq. (21)) by expectation, high-moment norms of all orders varnish w.r.t. N. The norms are in proportion to the ratio between cumulant/covariance norms and N, i.e., the right hand side of eq. (25) — the terms in eq. (26) are multiplication factors and would later turn out to be in the same functional form of eq. (25) (see proof of theorem 2.1, 2.2 in Erdos et al. [25]), so it is safe to ignore for now. By lemma 4.4 in Erdos et al. [25], the moment bounds can be converted to probability bounds. The intuition is that if the expectation of the high moment norms of $\boldsymbol{D}$ concentrate towards zero, with high probability, $\boldsymbol{D}$ is a zero matrix as well. The result of such conversion is given as the high probability bound in theorem 2.c.

Thus, *if the cumulant/covariance norms $|||\kappa|||_2^{iso} + |||\kappa|||^{av}$ are bounded, the conversion to probability bounds holds*. This leads to the boundedness requirement of cumulant/covariance norms in cond. 4.4 (cond. D.2). Larger constants simply lead to a looser bound, and the qualitatively, the behaviors characterized in theorem 2 still hold. And as the bounds in the statistical learning theory, the exact value of the bound is not critical. The more critical one is the parameters identified by the bound that induce the qualitative behaviors.

Some note might be needed for the N^{ϵ} term, since it exists in both the moment bound alongside with $|||\kappa|||^{\text{av}}$ and in cond. 4.4 (cond. D.2). $|||\kappa|||^{\text{av}}$ is a norm whose upper bound has a term similar with N^{ϵ} (which is estimated by combinatorial counting on graph), and the N^{ϵ} in $|||\kappa|||^{\text{av}} \leq C_2(\epsilon)N^{\epsilon}$ is absorbed in the N^{ϵ} in the probability bound since ϵ is allowed to be arbitrary.

E Non-zero cumulants computation through bootstrapping

To begin with, since this section intended to be readable without the knowledge of cumulants, we note that cumulants in this section can be interchanged with covariances, or statistics, without worrying that something might be missing.

As a result of random fluctuations, even for random variables that are independent with each other, their cross-cumulants will not be exactly zero, but of some small values fluctuating around zero. Thus, when estimating the coupling set size of a particular Hessian entry w_α, i.e., the set where w_α correlates with, we need to weed out those fluctuations. This leads us to standard techniques in statistics, i.e, the hypothesis testing and bootstrap.

Outline. We provide the outline of this section. **1)** in appendix E.1, we describe the statistics/cumulants that we are interested and are computed in the empirical study. **2)** we describe the algorithms that compute the non-zero count of cumulants in appendix E.2 and E.3. Overall, two core steps are involved in the computation. First, we need to have a criterion to determine when a covariance/cumulant is large enough to not to be taken as random fluctuations. Second, we need to compute the number of covariances/cumulants that exceed the threshold. The first step is achieved through *percentile bootstrapping* to compute a confidence interval that for cumulants are outside the confidence interval, they are considered to be cross-cumulants of correlated r.v.s. The second step is achieved by computing the sample mean and standard errors of the number of cumulants that cross the threshold of bootstrap samples. We describe the overall procedure, which is the step two, in appendix E.2. Then step one is described in appendix E.3. **3)** we describe the actual experiments in appendix E.4 and E.5.

We also note that the bootstrapping and hypothesis testing techniques used in this section might be subject to refinements, but they have shown at least qualitatively the sparse correlation phenomenon in the Hessian of NNs, as shown in section 5.1. Thus, they are considered to suffice in this iteration of study.

E.1 Statistics/cumulants in the empirical study

We describe the statistics/cumulants we study in experiments in this section. We compute the cumulants of Hessian entries $w_\alpha, \alpha \in \mathbb{I}$ that are up to the fourth order. We want to rule out the influence of the scale of w_α, thus we compute the normalized cumulants by dividing the cumulants by their standard deviation. This is a standard practice in statistics, so widely used that they have dedicated names for them. They are *Pearson correlation coefficients*, *skewness*, and *excess kurtosis*, corresponding to normalized second cumulant, third cumulant and fourth cumulant. We briefly describe them in this section since their high dimensional definitions may not be widely familiar.

Let $\boldsymbol{W}$ be a random matrix of dimension N, and $\{\boldsymbol{W}_i\}_{i=1,\dots,n}$ be a sample of size n. Let $\sigma(\alpha), \alpha \in \mathbb{I}$ denotes the standard deviation of w_α

Normalized second cumulant, or Pearson correlation coefficients. The second cumulant is the covariance, as mentioned previously, which might be the most standard definition in statistics. It is written as $\kappa_2(\alpha, \beta), \alpha, \beta \in \mathbb{I}$, and calculated as

$$\kappa_2(\alpha, \beta) = \frac{1}{n} \sum_{i=1}^{n} w_\alpha^i w_\beta^i.$$

To get a scale invariant measurement, we normalize it by standard deviation, and obtain the normalized second cumulant, or Pearson correlation coefficients.

$$\frac{\kappa_2(\alpha, \beta)}{\sigma(\alpha)\sigma(\beta)} = \frac{1}{n} \sum_{i=1}^{n} \frac{w_\alpha^i w_\beta^i}{\sigma(\alpha)\sigma(\beta)}.$$

Normalized third cumulant, or skewness. The third cumulant is write as $\kappa_3(\alpha, \beta, \gamma), \alpha, \beta, \gamma \in \mathbb{I}$ and calculated as

$$\kappa_3(\alpha, \beta, \gamma) = \frac{1}{n} \sum_{i=1}^{n} w_\alpha^i w_\beta^i w_\gamma^i.$$

Its scale invariant version is called skewness, and is also a standard definition to measure the asymmetry of a probability distribution. Again, it is obtained by normalizing the third cumulant (or third moment, since they are the same).

$$\frac{\kappa_3(\alpha, \beta, \gamma)}{\sigma(\alpha)\sigma(\beta)\sigma(\gamma)} = \frac{1}{n} \sum_{i=1}^{n} \frac{w_\alpha^i w_\beta^i w^i(\gamma)}{\sigma(\alpha)\sigma(\beta)\sigma(\gamma)}.$$

Normalized fourth cumulant, or excess kurtosis. The fourth cumulant is write as $\kappa_4(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathbb{I}$ and calculated as

$$\kappa_4(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = \frac{1}{n}\sum_{i=1}^{n} w^i_{\alpha_1} w^i_{\alpha_2} w^i_{\alpha_3} w^i_{\alpha_4} - \kappa_2(\alpha_1, \alpha_2)\kappa_2(\alpha_3, \alpha_4) - \kappa_2(\alpha_1, \alpha_3)\kappa_2(\alpha_2, \alpha_4) - \kappa_2(\alpha_1, \alpha_4)\kappa_2(\alpha_2, \alpha_3)$$

Its scale invariant version is called excess kurtosis, though it might mostly used in one dimensional setting, and is also a standard definition to measure extreme outlines of a probability distribution. Again, it is obtained by normalizing as follows

$$\frac{\kappa_4(\alpha_1, \alpha_2, \alpha_3, \alpha_4)}{\sigma(\alpha_1)\sigma(\alpha_2)\sigma(\alpha_3)\sigma(\alpha_4)}.$$

We also note that high dimensional setting, to characterize the overall behavior of the distribution, varied definitions of kurtosis have been proposed e.g., Koizumi et al. [46]. Considering that we only concern with individual cumulants between r.v.s here, we have adopted the classic definition of normalized kurtosis by Mardia [58].

E.2 Bootstrap estimation of non-zero cumulants

Notice that the random matrix under study is very large. For the Hessian $\boldsymbol{H}$ of the VGG network that we investigate experimentally, it is roughly of the dimension $10^7 \times 10^7$. It is both computationally intractable in term of time and space to compute all the non-zero cumulants directly. Thus, we use bootstrapping methods to estimate desired statistics. We use the bootstrap estimation of the correlation coefficients as an example to describe the procedure. The estimation of skewness and excess kurtosis is similar, except that in step 2 in algorithm 1, instead of counting non-zero correlation coefficients, we count the non-zero skewness, or excess kurtosis.

To being with, we describe the bootstrapping procedure to estimate the sample mean and standard errors of the non-zero count of correlation coefficients. Recall that $\boldsymbol{H}$ denotes Hessian and $\boldsymbol{H}$ is of dimension $N \times N$.

Algorithm 1: Bootstrap estimation of non-zero cumulants

1. Draw B independent bootstrap sample $\boldsymbol{H}_b$ of dimension $N_b \times N_b$ and of size n by sampling submatrices from $\boldsymbol{H}$, where $n < N$ (actually, $n \ll N$ due to computational constraints). More specifically, Each $\boldsymbol{H}_b$ is a submatrix sampled from the Hessian $\boldsymbol{H}$ by randomly taking $\boldsymbol{H}_{\boldsymbol{i}\boldsymbol{j}^T}$ $\boldsymbol{i}, \boldsymbol{j} \in [\mathbb{N}]^n$ ($\boldsymbol{i}, \boldsymbol{j}$ are random vectors whose elements are natural numbers less than or equal to N).
2. Compute the correlation coefficients matrix of vec$\boldsymbol{H}_b$.
3. Estimate the count s_0 of non-zero correlation coefficients for each bootstrap sample. This is done through bootstrap hypothesis testing described in appendix E.3.
4. Compute the mean of the count by s_0/N_b^2.
5. Estimate the standard error of the sample mean of s_0/N_b^2.

Some issues might need some explanations. 1) We have not described how to count the number of non-zero coefficients in algorithm 1, and it is done intentionally. The counting involves bootstrap hypothesis testing, and is described in appendix E.3. 2) Normally in bootstrapping methods, B is asked to be of order $10^2 \sim 10^3$. However, we use $B = 3$ in the experiments. This is because as a random matrix, an example of $\boldsymbol{H}_b$ is an ensemble of 10^8 of r.v.s itself, and its behavior is statistical on its own. Thus, even with a small B value, the resulted count is stable. We also note that it is also very computationally expensive to run the experiments. The standard errors can be seen in fig. 6a. 3) The sample size n here is the number of batches in an epoch, and further explanation can be found in appendix E.4.

E.3 Bootstrap hypothesis testing on correlations

We describe how the counting in algorithm 1 is done in this section. We take the value of cumulants as the test statistics. Instead of identifying a confidence interval corresponding to a specific P value, we identify a

confidence interval by maintaining that the coupling set size of the assumed distribution of the null hypothesis should be at least two magnitude smaller than the expected given by cond. D.5. We explain the strategy using correlation coefficients as follows.

We use the hypothesis testing on the correlation coefficient as an example to describe the null hypothesis. We use a null hypothesis that characterizes the lack of correlations between r.v.s. That is, if the null hypothesis holds, no correlations are considered to exist between the r.v.s in the test. We first state our null hypothesis.

Null hypothesis. Let X, Y be two r.v.s, and $\{x_i\}_{i=1,\dots,n}, \{y_i\}_{i=1,\dots,n}$ be a sample of X, Y respectively. The correlation coefficient $\kappa_2(X,Y)/(\sigma(X)\sigma(Y))$ of X, Y equates to the correlation coefficient of two r.v.s that are of independent normal distributions.

To obtain the probability distribution involved in such a null hypothesis is difficult. More specifically, the sample covariance is $\frac{1}{n}\sum_{i=1}^{n} X_i Y_i$, where $X_i, Y_i, i = 1, \dots, n$ are i.i.d. copies of X, Y. Thus, when X, Y are of independent normal distributions, the distribution of the sample covariance is a sum of the product of two normal distributions. The calculation of such a distribution still remains an open problem that is being tackled with varied approaches, e.g., Oliveira and Seijas-Macias [65] and Ware and Lad [87]. We use bootstrapping method to avoid actually calculating the distribution. We use the percentile bootstrap to identify a threshold that would give an adequate P value to the above null hypothesis. To proceed, we briefly explain the procedure of percentile bootstrapping.

Percentile bootstrap. The method constructs a two-sided equal-tailed $1-\alpha$ confidence interval for an estimate $\hat{\theta}$ from an empirical distribution by following the below steps.

Algorithm 2: Percentile bootstrapping

1. Draw B independent bootstrap samples z_b of size N from the distribution.
2. Estimate the parameter θ for each bootstrap sample: $\hat{\theta}_b$.
3. Order the bootstrap replications of $\hat{\theta}$ such that $\hat{\theta}_1 \leq \dots \leq \hat{\theta}_B$. The upper and lower bound confidence bounds are the $B(1-\alpha/2)$th and the $B(B\alpha/2)$th ordered elements, respectively. The estimated $(1-\alpha)$ confidence interval of $\hat{\theta}$ is $[\hat{\theta}_{B\alpha/2}, \hat{\theta}_{B(1-\alpha/2)}]$.

Monte Carlo percentile bootstrap. Notice that in algorithm 1 for a Hessian $\boldsymbol{H}$ of dimension $N \times N$, even if we bootstrap sample only a $N_b \times N_b$ submatrix of it, there would be $N_b^2(N_b^2-1)$ number of correlation coefficients, which would be a rather large number. In the experiment, it is of the magnitude of 10^8. Thus, the normal α at the magnitude of 1% is not small enough, since it would pass 10^6 number of correlation coefficients as non-zero even if entries of $\boldsymbol{H}$ are of Gaussian distribution. Thus, to prevent random fluctuations from interfering the count of non-zero coefficients, we work backward to choose a α through *Monte Carlo simulation* such that in the simulation, $\alpha N_b^2(N_b^2-1)$ would be two orders of magnitude smaller than the estimate number, i.e., the coupling set size $N_b^{1/2}$ in the case of non-zero correlation coefficients.

To choose $\alpha N_b^2(N_b^2-1)$, $N_b^2(N_b^2-1)$ number of correlation coefficients of independent normally distributed r.v.s are need to be sampled. To get such a sample, notice that given a random vector $\boldsymbol{w}$ of the multivariate Gaussian distribution with identity covariance matrix, i.e. $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, for $i, j \in [n], i \neq j$, we have $\kappa(w_i, w_j) = 0$, so are any of its higher order cross-cumulants. Thus, we use such a Gaussian distribution to generate desired samples.

The procedure to identify a confidence interval through Monte Carlo percentile bootstrapping is given as follows in algorithm 3, and see algorithm 1 for missing notations. The procedure identifies a threshold of the test statistics. And in this case, we do not need to know the exact α of the confidence interval, while at the same time, we are ensured that any correlation coefficients that pass that threshold should be considered to have come from two correlated r.v.s. Also note that we directly use the κ_2 to denote correlation coefficient since the samples are of unit variance.

Algorithm 3: Monte Carlo percentile bootstrapping

1. Draw B independent bootstrap sample $\boldsymbol{H}_0$ of dimension $N_b \times N_b$ and of size n, where $\mathrm{vec}\boldsymbol{H}_0$ is of multivariate normal distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.
2. Compute the correlation coefficients matrix of $\mathrm{vec}\boldsymbol{H}_0$.
3. Identify a threshold θ of the test statistics, i.e., correlation coefficients, such that the count $s_0 := \sum_{\alpha,\beta\in[N_b]\times[N_b],\alpha\neq\beta} \mathbf{1}_{|\kappa_2(h^0_\alpha, h^0_\beta)|>\theta}$ is least two orders of magnitudes smaller than $N_b^2 N_b^{1/2}$. Notice that $N_b^{1/2}$ is the upper bound of coupling set size in cond. D.5, and since each Hessian entry can have up to $N_b^{1/2}$ correlated entries, overall there are $N_b^2 N_b^{1/2}$ of them.
4. Estimate the standard error of s_0.

Algorithms for higher order statistics. The procedure to identify the thresholds for higher order statistics are similar, and we describe the difference in the following.

Strategy for skewness. In step 3 of algorithm 3, we identify a threshold θ of the test statistics, i.e., skewness, such that the count $s_0 := \sum_{\alpha,\beta,\gamma\in[N_b]\times[N_b]\times[N_b]} \mathbf{1}_{|\kappa_3(h^0_\alpha, h^0_\beta, h^0_\gamma)|>\theta}$ is least two magnitudes smaller than $N_b^2 N_b^1$.

Strategy for excess kurtosis. In step 3 of algorithm 3, we identify a threshold θ of the test statistics, i.e., excess kurtosis, such that the count $s_0 := \sum_{\alpha_1,\alpha_2,\alpha_3,\alpha_4\in[N_b]\times[N_b]\times[N_b]\times[N_b]} \mathbf{1}_{|\kappa_4(h^0_{\alpha_1}, h^0_{\alpha_2}, h^0_{\alpha_3}, h^0_{\alpha_4})|>\theta}$ is least two magnitudes smaller than $N_b^2 N_b^{3/2}$.

E.4 Bootstrap hypothesis testing on correlations in practice

In this section, we describe the obtainment of the thresholds of test statistics, i.e., correlation coefficients, skewness, excess kurtosis. We use algorithm 3 described earlier to compute the thresholds for test statistics.

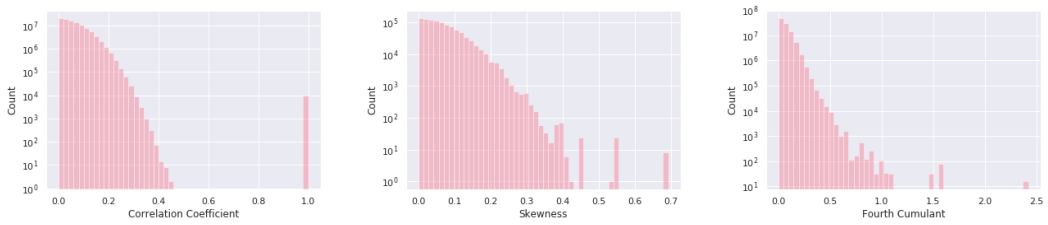
Experiment settings. First, we describe the parameters used in bootstrap sampling algorithms.

We choose the dimension N_b of the sampled submatrix $\boldsymbol{H}_b$ to be 100 when counting the non-zero correlation coefficients. We find the standard errors of the count mean is small enough to have a stable estimation of the exponent of N_b, as shown in fig. 6a, where the standard errors are shown as the transparent region around the curve. It is remarkable that such small N_b could provide such a stable estimation, considering it samples from $\sim 10^7$ possible dimensions. We also note that it is computationally expensive to compute the Hessian, and magnitude larger N_b would require a massively distributed parallel implementation and we believe the current small scale experiment has fulfilled our goal in this iteration of study.

We choose the dimension of N_b of $\boldsymbol{H}_b$ to be 10 in the experiments for skewness and excess kurtosis. This is because the number of cumulants grows by N_b^2, and we need to reduce the computational time and memory requirement in the experiments on them. Also note that despite such a small number, there are 10^6 individual skewness values and 10^8 individual kurtosis values. The results are also stable, as shown in standard errors of the counts in fig. 6a.

The sample size n in algorithm 1 is $157 = \lceil 10000/64 \rceil$, thus so is the n in algorithm 3. This is because the random matrix in question is the empirical mean of Hessian of individual examples, c.f. eq. (15). Thus, each batch of data contributes to an example of a sample. The dataset has 10000 examples, and the batch size is 64. Overall, there are 157 examples in a sample.

Threshold for correlation coefficients. To identify a threshold for correlation coefficients, we sample one $\boldsymbol{H}_0$ described in algorithm 1, and plot its the distribution, as shown in fig. 5a. $N_b^2 N_b^{1/2}$ in this case is 10^5, which makes the acceptable threshold $0.3 \sim 0.4$, and we pick 0.4 as the candidate threshold. Then we run algorithm 3 with $B = 5$ and obtain the mean and standard errors of s_0 in table 2.



(a) (b) (c)

Figure 5: Histograms of correlation coefficients, skewness, and excess kurtoses (fourth cumulants) generated in algorithm 3.

Table 2: Thresholds of test statistics that if exceeded, the null hypotheses that they are zero are rejected, and the statistics are considered non-zero. The non-zero counts are the numbers of non-zero test statistics under null hypotheses.

Test Statistics	Correlation Coefficients	Skewness	Excess Kurtosis
Non-zero Count s_0	16.00 ± 6.69	67.80 ± 73.74	2.20 ± 2.32
Threshold	0.4	0.4	1

Threshold for skewness. To identify a threshold for correlation coefficients, we sample one $\boldsymbol{H}_0$ described in algorithm 1, and plot its the distribution, as shown in fig. 5b. $N_b^2 N_b^1$ in this case is 10^3, which makes the acceptable threshold around 0.4, and we pick 0.4 as the candidate threshold. Then we run algorithm 3 with $B = 5$ and obtain the mean and standard errors of s_0 in table 2.

Threshold for skewness. To identify a threshold for correlation coefficients, we sample one $\boldsymbol{H}_0$ described in algorithm 1, and plot its the distribution, as shown in fig. 5c. $N_b^2 N_b^{3/2}$ in this case is $10^3\sqrt{10}$, which makes the acceptable threshold around 1, and we pick 1 as the candidate threshold. Then we run algorithm 3 with $B = 5$ and obtain the mean and standard errors of s_0 in table 2.

E.5 Bootstrap estimation of non-zero cumulants in practice

We describe the obtainment of the non-zero cumulants of Hessian throughout NN training in this section. We refer readers to appendix G.3 for the details of the dataset and the neural network in experiments.

We run algorithm 1 at the beginning of the NN training, and every 10 epoch with $B = 3$. The value of N_b and sample size n are described in appendix E.4 since they are the same with those of algorithm 3. The results are shown in fig. 6a.

Note that the coupling set size is given in term of $N^{1/2}$, thus, we would like to know the non-zero count in term of exponent of N. Since we have sample N_b^2 entries from N^2 possible entries of the Hessian through bootstrapping, the exponents of N_b in the experiment is an estimate of those of the population. We convert the actual counts to exponents of N_b, and the result is shown in fig. 1f. And fig. 6b is the one, i.e., fig. 1f, we provide in section 5.1. For analysis on the results, please also refer to section 5.1.

F Potentially exponential behaviors of NNs

We describe the potential exponential scaling behaviors of gradient norms (appendix F.1) and Hessian eigenvalues (appendix F.2) in NNs in this section.

(a) (b)

Figure 6: The average coupling set size of all Hessian entries characterized through the counts of non-zero statistics between Hessian entries, i.e., CC, skewness, and excess kurtosis during training. The non-zero CC of each entry is equivalent to its coupling set size. The relationship between the counts and coupling set sizes is explained in section 5.1. The transparent regions are standard deviations of the mean counts of the $B = 3$ bootstrap samples. Figure (a) gives the count in absolute value, and (b) in term of exponent of Hessian dimension N. The light color straight lines and the annotated numbers in the rightmost of figure (b) indicate the ratio between the non-zero count and the overall number of the statistics (both zero and non-zero).

F.1 Potential exponential scaling of expected square of NN gradient norm

We describe the mean field analysis of the expectation of square of gradient norm in this section.

Notation. We begin with notations. We move the layer index up when an individual component of a matrix or a vector is denoted. For weight matrix $\boldsymbol{W}_l$ at layer l, we denote $w_{ij}^l, i, j \in [n]$ its entry at ith row and jth column. To avoid confusion, we do not use upper case for them, though they are random variables in the analysis. So is the input random variable X, where we denote its ith component x_i. We denote $\boldsymbol{y}_l := \overleftarrow{\prod}_{i=1}^{l} \mathrm{dg}(\boldsymbol{h}_l)\boldsymbol{W}_l^T \boldsymbol{x}$, and $y_j^l, j \in [n]$ its jth component. Note that it uses the same alphabet as that of label defined in section 2.2. We only use it same way in this section, thus there should be no confusion. Such definition is preferred because the intermediate layer output could be seen as a form of estimated intermediate labels. When it is used along with $\delta \boldsymbol{y}_l$ defined in the following, the derivation is more intuitive. Also, note that $\boldsymbol{y}_l$ is the same as $\tilde{\boldsymbol{x}}_l$, in case there is any confusion. We denote δy_i^l the ith component of the back propagated gradient at layer l — that is the ith component of $\delta \boldsymbol{y}_l := \boldsymbol{\alpha}^T \overleftarrow{\prod}_{j=l+1}^{L-1} (\mathrm{dg}(\tilde{\boldsymbol{h}}_j)\boldsymbol{W}_j^T)$, the term on the left of the Kronecker product of eq. (11) with $\mathrm{dg}(\tilde{\boldsymbol{h}}_i)$ removed.

Assumptions. We assume the commonly assumed assumptions in variance analysis of NN at initialization [36]: for $i, j \in [n]$, x_i is independent with x_j; weights are initialized with i.i.d. normal distribution of zero mean and variance c/n, where c is a constant; For $i, j \in [n], i \neq j$, δy_i^l is independent with δy_j^l; For $i, j \in [n]$, δy_j^l is independent with y_i^l. Furthermore, since we only aim to demonstrate the exponential change behavior, we assume all layers all have the same width n and x_i is of zero mean. The further zero mean assumption is not necessary, and is only assumed to convert $\mathbb{E}[x_i^2]$ to Var $[x_i]$ to make the relation uniformly about variance. Throughout this section, we treat the last layer parameter $\boldsymbol{\alpha}$ as a one column matrix, so we do not need to reserve a special notation for it.

Lemma F.1. *Given a NN $T(X; \boldsymbol{\theta})$, if the above assumption hold at initialization, and individual weight parameter are initialized with i.i.d. normal distribution $\mathcal{N}(0, c/n)$, then the expected square of gradient norm $\mathbb{E}[(\nabla_{\boldsymbol{\theta}} l((T(X), y); \boldsymbol{\theta}))^2]$ grows at rate of $\Theta(Ln^2(c/2)^L)$, where L is the number of layers, and n is the layer width.*

Proof. **Forward propagated activation variance.** The recursive relation between y_i^l and y_i^{l-1} is given by

$$y_i^l = \sum_{j=1}^{n} h_i^l w_{ij}^l y_j^{l-1}$$

The recursive relation between variance of y_i^l and y_i^{l-1} is given by

$$\begin{aligned}\text{Var}\left[y_i^l\right] &= \frac{1}{2}\sum_{j=1}^{n}\text{Var}\left[w_{ij}^l\right]\text{Var}\left[y_j^{l-1}\right] \\ &= \frac{c}{2}\text{Var}\left[y_j^{l-1}\right]\end{aligned}$$

Thus, for $i, j \in [n]$ and a L layer NN, the variance at layer L would be

$$\text{Var}\left[y_i^L\right] = (\frac{c}{2})^L \text{Var}\left[x_j\right]$$

Backward propagated gradient variance. Similarly, the variance of gradient has a recursive relation between $\delta y_i^l, \delta y^{l+1}$ as follows.

$$\delta y_i^l = \sum_{j=1}^{n} w_{ji}^l h_j^l \delta y_j^{l+1}$$

The recursive relation between variance is given by

$$\begin{aligned}\text{Var}\left[\delta y_i^l\right] &= \frac{1}{2}\sum_{j=1}^{n}\text{Var}\left[w_{ij}^l\right]\text{Var}\left[\delta y_j^{l+1}\right] \\ &= \frac{c}{2}\text{Var}\left[y_j^{l+1}\right]\end{aligned}$$

Thus, for $i, j \in [n]$ and a layer L NN, the variance of gradient at layer L would be

$$\text{Var}\left[\delta y_i^1\right] = (\frac{c}{2})^L \text{Var}\left[\delta y_j^L\right],$$

where δy_i^L is the gradient at the top layer calculated from the loss function w.r.t. the top layer activation.

Gradient norm variance. Recall in eq. (11), the gradient is given by

$$\partial l(T\boldsymbol{x}, y)/\partial \text{vec}\boldsymbol{W}_p = l'(T\boldsymbol{x}, y)\boldsymbol{\alpha}^T \overleftarrow{\Pi}_{j=p+1}^{L-1}(\text{dg}(\tilde{\boldsymbol{h}}_j)\boldsymbol{W}_j^T)\text{dg}(\tilde{\boldsymbol{h}}_p) \otimes \boldsymbol{x}^T \overrightarrow{\Pi}_{i=1}^{p-1}(\boldsymbol{W}_i \text{dg}(\tilde{\boldsymbol{h}}_i)).$$

Notice that the term on the left of the Kronecker product is the back propagated gradient at layer p, i.e., $\delta \boldsymbol{y}_{p+1} \odot \boldsymbol{h}_p($, where $\odot$ denote Hadamard product that multiplies vectors element-wise), and the term on the right is the forward propagated activation at layer $p-1$ as well, i.e., $\boldsymbol{y}_{p-1}$. Thus, if we assume that $i, j \in [n], \delta y_i^{p+1}, y_j^p, h_j^p$ are independent with each other, we can reach derivation as follows.

$$\begin{aligned}\text{Var}\left[\partial l(T\boldsymbol{x}, y)/\partial w_{ij}^p\right] &= \text{Var}\left[\delta y_j^{p+1} h_j^p y_i^{p-1}\right] \\ &= \mathbb{E}[(\delta y_j^{p+1} h_j^p y_i^{p-1})^2] - \mathbb{E}^2[(\delta y_j^{p+1} h_j^p y_i^{p-1})] \\ &= \mathbb{E}[(\delta y_j^{p+1} h_j^p y_i^{p-1})^2] \\ &= \text{Var}\left[\delta y_j^{p+1}\right]\mathbb{E}[(h_j^p)^2]\text{Var}\left[y_i^{p-1}\right] \\ &= \mathbb{E}^2[h_j^p](\frac{c}{2})^{L-p-1}\text{Var}\left[\delta y_j^L\right](\frac{c}{2})^{p-1}\text{Var}\left[y_i^1\right] \\ &= \mathbb{E}^2[h_j^p](\frac{c}{2})^{L-2}\text{Var}\left[\delta y_j^L\right]\text{Var}\left[x_i\right]\end{aligned}$$

The above derivation holds for any layer p, thus we reach the conclusion that the variance of components of gradient grows at a rate of $\Theta((c/2)^L)$.

Expected gradient norm.

We shorten the gradient $\partial l(T\boldsymbol{x}, y)/\partial w_{ij}^p$ as ∇_{ij}^p hereafter. Notice that in the above derivation, the gradient ∇_{ij}^p has zero mean, thus we have

$$\mathbb{E}[(\nabla_{ij}^p)^2] = \text{Var}\,[\nabla_{ij}^p]$$

Consequently, the expectation of square of the gradient norm is given by

$$\begin{aligned}
\mathbb{E}[||\nabla||_2^2] &= \sum_p^L \sum_{i,j=1}^n \mathbb{E}[(\nabla_{ij}^p)^2] \\
&= \sum_p^L \sum_{i,j=1}^n \text{Var}\,[\nabla_{ij}^p] \\
&= \sum_p^L \sum_{i,j=1}^n \mathbb{E}^2[h_j^p](\frac{c}{2})^{L-2}\text{Var}\,[\delta y_j^L]\text{Var}\,[y_i^1] \\
&= Ln^2(\frac{c}{2})^{L-2}\mathbb{E}^2[h_j^p]\text{Var}\,[\delta y_j^L]\text{Var}\,[x_i]
\end{aligned}$$

Thus, the gradient norm square grow at a rate of $\Theta(Ln^2(c/2)^L)$. We remark on the polynomial factors briefly. Since the norm is a summation along all its components, it is intuitive that the number of components in the gradient vector contribute to the count factor Ln^2.

□

F.2 Potential exponential scaling on Hessian eigenvalues at initialization

In this section, we show that the factor used in initialization induces a potentially exponential scaling on the eigenvalues of NN Hessian. More specifically, suppose that at initialization $\forall l \in [L-1], \boldsymbol{W}_l \in \mathbb{R}^{n\times n}$, and $w_{ij}^l \sim \mathcal{N}(0, c/n)$. Then, the NN Hessian $\boldsymbol{H}$ satisfies the following property.

Corollary F.1. *Assuming the same assumptions of theorem 2, the following statement holds.*

$$\forall c \in \mathbb{R}, \boldsymbol{H}(c) = c^{(L-2)/2}\boldsymbol{H}(1)$$

where $\boldsymbol{H}(c)$ denotes the Hessian matrix obtained at c — for example, $\boldsymbol{H}(1)$ denote the Hessian obtained at $c = 1$.

Note that $\boldsymbol{H}$ can be decomposed as the summation of its eigenvectors as

$$\boldsymbol{H} = \sum_{i=1}^N \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^T.$$

It implies that change in c is scaled exponentially in $\lambda_i, i = 1, \ldots, N$. More formally, this is stated in the following lemma.

Lemma F.2. *Let $\boldsymbol{H}$ be a random matrix that satisfies Matrix Dyson Equation, and $\mu_{\boldsymbol{H}}$ be the eigenvalue density function of $\boldsymbol{H}$. Denote $\mu_{c\boldsymbol{H}}$ the eigenvalue density function of $c\boldsymbol{H}$. Then we have*

$$\mu_{c\boldsymbol{H}}(z) = \mu_{\boldsymbol{H}}(cz).$$

Outline. We first build the intuition of the above phenomenon in appendix F.2.1 through mean field analysis that shows variance of individual Hessian entries are changed exponentially w.r.t. the change in c. The mean field analysis assumes much weaker independent assumptions, but are also easier to understand, thus we feel that including it is good to warm up the readers. In Matrix Dyson Equation, the variances and covariances largely decide the resulting eigenspectrum. After the mean field analysis appendix F.2.2, under the conditions we assume in this work, which are found to characterize practical NNs, we prove corollary F.1 and lemma F.2 that formally characterizes that the change in c is potentially exponentially reflected in eigenvalues.

F.2.1 Mean field analysis of potentially exponential scaling

The mean field analysis of variance of Hessian entries are similar with the mean field analysis of expected gradient norm in appendix F.1. Also refer to notations in appendix F.1 for notations in this section. In addition the assumptions made in appendix F.1, we make the following further assumptions: the hidden variable $h_l, l = 1, \dots, L-1$ is of an independent Bernoulli distribution with variance $1/2$.

Recall that in eq. (5), the block Hessian between layer p, q is given as

$$\boldsymbol{H}_{pq} = l'(T\boldsymbol{x}, y)\mathrm{dg}(\tilde{\boldsymbol{h}}_q)\overrightarrow{\prod}_{k=q+1}^{L-1}(\boldsymbol{W}_k\mathrm{dg}(\tilde{\boldsymbol{h}}_k))\boldsymbol{\alpha} \otimes \overleftarrow{\prod}_{j=p+1}^{q-1}(\mathrm{dg}(\tilde{\boldsymbol{h}}_j)\boldsymbol{W}_j^T)\mathrm{dg}(\tilde{\boldsymbol{h}}_p) \otimes \boldsymbol{x}^T\overrightarrow{\prod}_{i=1}^{p-1}(\boldsymbol{W}_i\mathrm{dg}(\tilde{\boldsymbol{h}}_i)).$$

The leftmost term is the gradient, and the rightmost term is the activation, which we have calculated their variance in appendix F.1. It reminds to calculate the variance of the term in the middle.

Similarly, we first calculate the between-layer recursive relation. For two layers, the middle term is $\tilde{\boldsymbol{W}}_{l-1\sim l+1} := \mathrm{dg}(\boldsymbol{h}_{l-1})\boldsymbol{W}_l\mathrm{dg}(\boldsymbol{h}_l)\boldsymbol{W}_{l+1}\mathrm{dg}(\boldsymbol{h}_{l+1})$. Note that the shorthand notation $\tilde{\boldsymbol{W}}_{l-1\sim l+1}$ follows eq. (5). It induces the recursive relation as follows.

$$\tilde{w}_{ij}^{l-1\sim l+1} = h_j^{l+1}\sum_{k=1}^{n} w_{kj}^{l+1}\tilde{w}_{ik}^{l-1\sim l}$$

The recursive variance relationship is

$$\begin{aligned}
\mathrm{Var}\,[\tilde{w}_{ij}^{l-1\sim l+1}] &= \mathbb{E}[(h_j^{l+1}\sum_{k=1}^{n} w_{kj}^{l+1}\tilde{w}_{ik}^{l-1\sim l})^2] \\
&= \mathbb{E}[(h_j^{l+1})^2]\sum_{k=1}^{n}\mathrm{Var}\,[(w_{kj}^{l+1})^2]\mathrm{Var}\,[(\tilde{w}_{ik}^{l-1\sim l})^2] \\
&= \frac{1}{2}\sum_{k=1}^{n}\frac{c}{n}\mathrm{Var}\,[\tilde{w}_{ik}^{l-1\sim l}] \\
&= \frac{c}{2}\mathrm{Var}\,[\tilde{w}_{ik}^{l-1\sim l}]
\end{aligned}$$

Thus, for $p, q \in [n], p < q-1$,

$$\mathrm{Var}\,[\tilde{w}_{ij}^{p\sim(q-1)}] = (\frac{c}{2})^{q-p-2}\mathrm{Var}\,[\tilde{w}_{ik}^{l-1\sim l}] = (\frac{c}{2})^{q-p-2}\mathbb{E}[(h_k^{p+1})^2]\mathbb{E}[(h_i^p)^2]\mathbb{E}[(w_{ik}^{p+1})^2].$$

Consequently, the middle term also grows at a rate of $\Theta((c/2)^{q-p})$.

Similarly, for the block Hessian matrix $\boldsymbol{H}_{pq}$, its entries $h_{\alpha,\beta}^{pq}, \alpha, \beta \in [n^2] \times [n^2], \alpha := in + j, \beta := kn + l$

have the variance as follows

$$\begin{aligned}
\text{Var}\left[\boldsymbol{H}_{\alpha\beta}^{pq}\right] &= \text{Var}\left[\tilde{\alpha}_i^q \tilde{w}_{jk}^{p\sim(q-1)} \tilde{x}_l^{p-1}\right] \\
&= \text{Var}\left[\delta y_i^q\right]\mathbb{E}[(h_i^q)^2]\text{Var}\left[\tilde{w}_{jk}^{p\sim(q-1)}\right]\text{Var}\left[y_l^{p-1}\right] \\
&= \left((\frac{c}{2})^{L-q}\mathbb{E}[(h_a^q)^2]\text{Var}\left[\delta y_a^L\right]\right)\left((\frac{c}{2})^{q-p-2}\mathbb{E}[(h_c^{p+1})^2]\mathbb{E}[(h_b^p)^2]\mathbb{E}[(w_{bc}^{p+1})^2]\right)\left((\frac{c}{2})^{p-1}\text{Var}\left[x_d\right]\right) \\
&= \left((\frac{c}{2})^{L-q}\frac{1}{2}\text{Var}\left[\delta y_a^L\right]\right)\left((\frac{c}{2})^{q-p-1}\frac{1}{2}\frac{1}{n}\right)\left((\frac{c}{2})^{p-1}\text{Var}\left[x_d\right]\right) \\
&= (\frac{c}{2})^{L-2}(\frac{1}{2})^2\frac{1}{n}\text{Var}\left[\delta y_a^L\right]\text{Var}\left[x_d\right]
\end{aligned}$$

where $a, b, c, d \in [n]$ are arbitrary. From the above equation, we can see that the Hessian entry also grows at the rate of $(c/2)^L$. Note that to distinguish hidden variable with the Hessian entries, we use the bold upper case for the Hessian entry $\boldsymbol{H}_{\alpha\beta}^{pq}$ above instead lower case.

F.2.2 Matrix Dyson Equation analysis of potentially exponential scaling

The Matrix Dyson Equation eq. (9) allows us to analyze the eigenspectrum in stronger assumptions than the previous mean field analysis.

Notice that in the previous mean-field analysis, the variable that induces the potentially exponential behavior is the variance of the distribution of entries in the weight matrices. To recall, for $l \in [L]$, the entries of $\boldsymbol{W}_l$ are initialized with zero mean, c/n variance normal distribution. Thus, the Hessian $\boldsymbol{H}$ can be viewed as a random matrix generated by the mapping that maps c to a random matrix, rewritten as

$$\boldsymbol{H}(c) : c \mapsto \boldsymbol{H},$$

where we use the $\boldsymbol{H}$ itself to represent the mapping.

The mapping satisfies the following property that the change in c exponentially amplifies the eigenvalues of $\boldsymbol{H}$, formally stated as follows.

Corollary F.2.

$$\forall c \in \mathbb{R}, \boldsymbol{H}(c) = c^{(L-2)/2}\boldsymbol{H}_1$$

where $\boldsymbol{H}_1$ denotes the Hessian matrix obtained when $c = 1$.

It is the corollary of the following proposition.

Proposition F.1.

$$\forall c, d \in \mathbb{R}, \boldsymbol{H}(c) = \boldsymbol{H} \implies \boldsymbol{H}(dc) = d^{(L-2)/2}\boldsymbol{H}$$

Proof. Suppose that $f(c)$ emits the following Hessian matrix. As always, we write it in block matrix form.

$$\boldsymbol{H}(c)_{pq} = l'(T\boldsymbol{x}, y)\text{dg}(\tilde{\boldsymbol{h}}_q)\overrightarrow{\prod}_{k=q+1}^{L-1}(\boldsymbol{W}_k\text{dg}(\tilde{\boldsymbol{h}}_k))\boldsymbol{\alpha}\otimes\overleftarrow{\prod}_{j=p+1}^{q-1}(\text{dg}(\tilde{\boldsymbol{h}}_j)\boldsymbol{W}_j^T)\text{dg}(\tilde{\boldsymbol{h}}_p)\otimes\boldsymbol{x}^T\overrightarrow{\prod}_{i=1}^{p-1}(\boldsymbol{W}_i\text{dg}(\tilde{\boldsymbol{h}}_i)).$$

In $\boldsymbol{W}_l, l \in [L]$, w_{ij}^l is of zero mean and c/n variance — again, we treat $\boldsymbol{W}_L := \boldsymbol{\alpha}$ as a one column matrix. Then if $\boldsymbol{w}_{ij}^l$ is initialized with zero mean and cd/n variance required by $f(cd)$, then the new Hessian matrix can be written as

$$\begin{aligned}
\boldsymbol{H}(d)_{pq} =& l'(T\boldsymbol{x}, y)\text{dg}(\tilde{\boldsymbol{h}}_q)\overrightarrow{\prod}_{k=q+1}^{L-1}(\sqrt{d}\boldsymbol{W}_k\text{dg}(\tilde{\boldsymbol{h}}_k))\sqrt{d}\boldsymbol{\alpha} \\
&\otimes\overleftarrow{\prod}_{j=p+1}^{q-1}(\text{dg}(\tilde{\boldsymbol{h}}_j)\sqrt{d}\boldsymbol{W}_j^T)\text{dg}(\tilde{\boldsymbol{h}}_p) \\
&\otimes\boldsymbol{x}^T\overrightarrow{\prod}_{i=1}^{p-1}(\sqrt{d}\boldsymbol{W}_i\text{dg}(\tilde{\boldsymbol{h}}_i)).
\end{aligned}$$

Take the $\sqrt{d}$ out, we have

$$\boldsymbol{H}(d)_{pq} = d^{(L-2)/2}\boldsymbol{H}(c)_{pq}$$

□

Combined with the MDE equation, the proposition leads to the following lemma F.2, the proof of which is the given as follows.

Proof of lemma F.2. Since $\boldsymbol{H}$ satisfies the MDE, we have

$$\boldsymbol{I} + (z - \boldsymbol{A} + \mathbb{E}_{\boldsymbol{W}}[\boldsymbol{W}\boldsymbol{G}\boldsymbol{W}])\boldsymbol{G} = \boldsymbol{0}, \Im \boldsymbol{G} \succ \boldsymbol{0}, \Im z > 0.$$

Again, note that $\boldsymbol{W}$ without a subscript or a superscript is the normalized Hessian defined in 7, not the weight matrix $\boldsymbol{W}_l$. Substitute $d^{L/2}\boldsymbol{H}$ into the left part of the equation, we have

$$\boldsymbol{I} + (z - d^{L/2}\boldsymbol{A} + \mathbb{E}_{\boldsymbol{W}}[d^{L/2}\boldsymbol{W}\boldsymbol{G}d^{L/2}\boldsymbol{W}])\boldsymbol{G} = \boldsymbol{I} + z\boldsymbol{G} - d^{L/2}\boldsymbol{A}\boldsymbol{G} + \mathbb{E}_{\boldsymbol{W}}[d^{L/2}\boldsymbol{W}\boldsymbol{G}d^{L/2}\boldsymbol{W}\boldsymbol{G}].$$

Notice that if we absorb the $d^{L/2}$ term into $\boldsymbol{G}$, and replace z with $d^{L/2}z$, we find that

$$\boldsymbol{I} + (z - \boldsymbol{A} + \mathbb{E}_{\boldsymbol{W}}[\boldsymbol{W}d^{L/2}\boldsymbol{G}\boldsymbol{W}])d^{L/2}\boldsymbol{G} = \boldsymbol{0}$$

also holds. Replace $d^{L/2}\boldsymbol{G}$ with $\boldsymbol{G}'$, and $d^{L/2}z$ with a new variance z' we have

$$\boldsymbol{I} + (z' - \boldsymbol{A} + \mathbb{E}_{\boldsymbol{W}}[\boldsymbol{W}\boldsymbol{G}'\boldsymbol{W}])d^{L/2}\boldsymbol{G}' = \boldsymbol{0}$$

Since by theorem 2, the solution to MDE is unique, we have

$$\boldsymbol{G}' = \boldsymbol{G}.$$

It implies $\boldsymbol{G}'(z') = d^{L/2}\boldsymbol{G}(z') = \boldsymbol{G}(z)$, and thus we have

$$\boldsymbol{G}'(d^{L/2}z) = \boldsymbol{G}(z).$$

Correspondingly, the Stieltjes transform of the eigenspectrum density is

$$m_{\mu_{d^{L/2}\boldsymbol{H}}}(z) = \frac{1}{N}\text{tr}\, G'(z) = \frac{1}{N}\text{tr}\, G(d^{-L/2}z).$$

By lemma D.1, the eigenspectrum density $\mu_{d^{L/2}\boldsymbol{H}}$ of the $d^{L/2}\boldsymbol{H}$ is

$$\mu_{d^{L/2}\boldsymbol{H}}(z) = \mu_{\boldsymbol{H}}(d^{L/2}z).$$

□

The theorem suggests that if the constant factor c is scaled with by d, the all eigenvalues of $\boldsymbol{H}$ are scaled by $d^{(L-2)/2}$.

G Experiment details

We describe the details of the experiments in section 4.2 in this section. All experiments are run with PyTorch [69].

G.1 Algorithm details of the computation of eigenspectrum

The eigenspectrum is computed Lanczos spectrum approximation algorithms [29; 55; 66].

Lanczos Spectrum Approximation Algorithm. We use the Lanczos approximation algorithm in Papyan [66], which is proposed by Lin et al. [55], and implemented contemporarily by Ghorbani et al. [29]. The eigenspectum shown in fig. 1b is computed on 200 training samples. Similar to what has been done in Papyan [66], when computing the extreme eigenvalues to normalize the eigenspectrum to $[-1, 1]$, we run 128 iterations with $\kappa = 0.05$ (the iteration number M_0 and margin percentage κ in `Normalization` algorithm, i.e., Algorithm 4, in [66] respectively); when approximating the eigenspectrum of Hessian, i.e., the `LanczosApproxSpec` algorithm (Algorithm 3), we use iteration number $M = 128$, the number of points $K = 1024$, and the number of starting random vector $n_{\text{vec}} = 1$.

G.2 LeNet on MNIST

We use a classical LeNet type NN [50] on the MNIST dataset. The LeNet has 4 layer and 1232768 parameters.

NN Architecture
Conv $5 \times 5 \times 32$
ReLU
MaxPool 2×2
Conv $5 \times 5 \times 64$
ReLU
MaxPool 2×2
Fully Connected 512
ReLU
Fully Connected 1
Hinge Loss

Table 3: LeNet Network Architecture. On the right of "Conv" is kernel height $\times$ kernel width $\times$ the output channel number. On the right of "MaxPool" is the kernel size. On the right of "FullyConnected" is output channel number. Each convolution layer uses strides 1, while pooling layer uses strides the same as the kernel size.

Model & Training. No simplification is done on the LeNet type NN. We only replace the cross entropy loss with hinge loss, so that the network can be used for binary classification. The NN has architecture as in table 3. The NN is trained with Stochastic Gradient Descent with momentum 0.9. The NN is trained for 10 epochs. The initial learning rate is 0.01, and decayed at 6th epoch to 0.001. The network is initialized with a very old initialization method that initializes each weight by sampling from the uniform distribution with standard deviation $\frac{1}{\sqrt{n}}$, where n is the number of input channels. No explicit regularization is used, e.g., weight decay [48].

Dataset. We train the NN on the classic MNIST data. Since we need to do binary classification, we take label 0 as the positive class, and the rest as the negative class. To create a balanced dataset, we use all images that are digital 0, and an equal number of samples of the negative class, which are randomly sampled digital from $1 - 9$.

The training set consists of about 12000 examples, and the validation set consists of about 2000 samples. The samples are preprocessed to be zero-centered and of unit variance. No data augmentation is used.

G.3 VGG on CIFAR10

The VGG network has 12 layers and 8684684 parameters.

Model. We use a VGG type NN [78] on the CIFAR10 dataset [47]. The network has 12 layers. The first 10 layers are convolution layers with 3×3 kernel size and the ReLU activation function, the last two layers are fully connected layers. The output channel number of the 12 layers are $[64, 64, 128, 128, 256, 256, 256, 512, 512, 512, 512, 1]$. Pooling layers with kernel size $[2, 2]$ are applied to convolution layer $2, 4, 7, 10$. Dropout layers are applied to convolution layer $1, 3, 5, 6, 8, 9, 10, 11$. The specification is adopted from the [94]. We use a version of Batch Normalization that only subtracts mean and divides standard deviation of the current batch; that is, we do not use the extra γ, β affine transformation after normalization, and we do not maintain the global statistics of mean and standard deviation. This is because as explained in section 5.2, our theoretical results imply that the BN improves optimization by scaling the gradient norm, and the affine transformation, and the global statistics are not critical for benign loss landscape that make NNs reach zero risk global minima. However, we also note that the global statistics and the affine transformation could have regularization effect that improves generalization, but such effects are not our concern in this work, which focuses on optimization.

Training. No simplification is done on the VGG type NN. We only replace the cross entropy loss with hinge loss, so that the network can be used for binary classification. The NN is trained with Stochastic Gradient Descent with momentum 0.9. The NN is trained for 300 epochs. The initial learning rate is 0.05, and decayed at 200th epoch to 0.005, and at 250th epoch to 0.0005. The weights of the network are initialized with i.i.d. normal distribution with standard deviation of 0.0001 No explicit regularization is used, e.g., weight decay [48].

Dataset. We train the NN on the CIFAR10 data. Since we need to do binary classification, we take label 0 as the positive class, and the rest as the negative class. To create a balanced dataset, we use all images that are of label 0, and an equal number of samples of the negative class, which are randomly sampled from the rest of the classes. The training set consists of 10000 examples, and the validation set consists of 2000 examples. The samples are preprocessed to be zero-centered and of unit variance. During training, we zero-pad 4 pixels along each image side, and sample a 32×32 region cropped from the padded image or its horizontal flip; during testing, we use the original non-padded image.