Author:
Baskerville, Nick P

Title:
Random matrix theory and the loss surfaces of neural networks

University of Bristol

Random matrix theory and the loss surfaces of neural networks

Nicholas P. Baskerville
MMath MA (Cantab), CMath MIMA

May, 2023

A dissertation submitted to the University of Bristol in accordance with the requirements for award of the degree of Doctor of Philosophy in the Faculty of Science, School of Mathematics.

Abstract

Neural network models are one of the most successful approaches to machine learning, enjoying an enormous amount of development and research over recent years and finding concrete real-world applications in almost any conceivable area of science, engineering and modern life in general. The theoretical understanding of neural networks trails significantly behind their practical success and the engineering heuristics that have grown up around them. Random matrix theory provides a rich framework of tools with which aspects of neural network phenomenology can be explored theoretically. In this thesis, we establish significant extensions of prior work using random matrix theory to understand and describe the loss surfaces of large neural networks, particularly generalising to different architectures. Informed by the historical applications of random matrix theory in physics and elsewhere, we establish the presence of local random matrix universality in real neural networks and then utilise this as a modeling assumption to derive powerful and novel results about the Hessians of neural network loss surfaces and their spectra. In addition to these major contributions, we make use of random matrix models for neural network loss surfaces to shed light on modern neural network training approaches and even to derive a novel and effective variant of a popular optimisation algorithm.

Overall, this thesis provides important contributions to cement the place of random matrix theory in the theoretical study of modern neural networks, reveals some of the limits of existing approaches and begins the study of an entirely new role for random matrix theory in the theory of deep learning with important experimental discoveries and novel theoretical results based on local random matrix universality.

Dedication and acknowledgements

I dedicate this thesis to Charlotte, without whose love, encouragement and support it would not have been possible. Doing a PhD essentially as a hobby was a bit deranged and I had no right to expect you to put up with the silly hours nor my doing maths on the beach or other such nonsense, but you always did. As romantic gestures go, 200-odd pages of maths is a poor substitute for a symphony, but for what it's worth, it is yours.

My parents have never failed to support my endeavours, academic or otherwise, and have ever been a source of steadfast encouragement, for which I am truly grateful. This thesis is really the conclusion of something we began together in the Cambridge branch of *Eat* over ten years ago.

I thank Jon Keating, Francesco Mezzadri and Joseph Najnudel for the years of enjoyable collaboration and for taking the punt on me and my project in the first place. Their advice and ideas were integral to the success of this research. They provided just the right amount of guidance to keep projects moving, while allowing me to develop my own ideas and approaches.

I thank Diego Granziol for several years of enjoyable collaboration. Much of the second half of this thesis was influenced by our discussions and is, I am sure, much the better for it.

Thanks are due to the RMT group in Bristol for making me feel part of things, despite only being in the office once a week. I gratefully acknowledge financial support for tuition fees from the School of Mathematics in the University of Bristol.

Finally, I thank my former team at GCHQ and particularly Tom L. His initial support and practical help were instrumental in getting my PhD off the ground and I shall always be grateful for the three years during which he gladly encouraged me as I juggled PhD work and our day-jobs. I also gratefully acknowledge the three years of funding for tuition fees and travel expenses that came from AR/RIF/STR and the DRE scholarship scheme.

Author's declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

Nicholas P. Baskerville
Saturday 20th May, 2023, Bristol

Table of Contents

Page

List of Tables

List of Figures

Abbreviations

The following abbreviations are used throughout this thesis.

NN	neural network
ANN	artificial neural network
DNN	deep neural network
SGD	stochastic gradient descent
MLP	multi-layer perceptron
CNN	convolutional neural network
RNN	recurrent neural network
GAN	generative adversarial network
RMT	random matrix theory
GOE	Gaussian orthogonal ensemble
LSD	limiting spectral density
ESD	empirical spectral density
NNSD	nearest neighbour spacing distribution
WLOG	without loss of generality
a.s.	almost surely

Notation

The following notation will be used consistently throughout this thesis unless stated otherwise.

δ_x	A Dirac δ-function centred at the point x
$\hat{\mu}_N$	The empirical spectral measure of an $N \times N$ matrix
ρ_{SC}	The semi-circle density
g_μ	The Stieljtes transform of the measure μ
R_μ	The R-transform of the measure μ
$\mu \boxplus \nu$	Additive free convolution of measure μ and ν
I_N	The $N \times N$ identity matrix
$O(N)$	The orthogonal group on $N \times N$ matrices
$\Delta(\boldsymbol{x})$	The Vandermonde determinant over N symbols $\{x_1, \ldots, x_N\}$
$\mathcal{N}(\mu, \Sigma)$	A Gaussian random variable with mean μ and covariance Σ
$\Re z$	The real part of $z \in \mathbb{C}$
$\Im z$	The imaginary part of $z \in \mathbb{C}$
$i(X)$	The index of an Hermitian matrix X
μ_{Haar}	The Haar measure on $O(N)$
$\mathcal{O}(\cdot)$	Asymptotic "big-o" notation. $f(x) = \mathcal{O}(g(x))$ if $\exists$ some constant $c > 0$ such that $\lvert f(x) \rvert \leqslant c \lvert g(x) \rvert$ for all large enough x.
$o(\cdot)$	Asymptotic "little-o" notation. $f(x) = o(g(x))$ if $f(x)/g(x) \to 0$ as $x \to \infty$.
$f \sim g$	Asymptotic equivalence. $f \sim g$ if $\lvert f(x)/g(x) \rvert \to 1$ as $x \to \infty$.
$[N]$	The set of integers from 1 to N: $\{1, 2, \ldots, N\}$.

Chapter 1

Introduction

In this chapter we introduce the central objects of study for this thesis, namely deep neural networks and their loss surfaces. Deep neural networks are an important sub-field of machine learning, so we begin with some introductory material and context for machine learning. We make no attempts to be exhaustive, but aim to provide a self-contained introduction, accessible for any mathematically literate reader, to the key ideas from machine learning relevant to our investigations. We will provide a rather more detailed introduction to deep neural networks specifically, again aiming to be accessible to any mathematical reader. The reader familiar with machine learning and deep neural networks may well safely skip these introductory sections, however they do establish some conventions and points of view, which may be more or less familiar depending on the reader's background. Following these broad introductory sections, we will sharpen the focus to provide a summary of the prior literature on deep neural network loss surfaces, particular focusing on the mathematical works upon which this thesis is built. We will also take this opportunity to draw out and summarise the existing connections between deep neural network loss surfaces and random matrix theory, but an introduction to random matrix theory itself is postponed until the next chapter. We conclude this introductory chapter with a summary of the new results which make up the principal intellectual contribution of this thesis and a literature review of related work.

1.1 Machine learning

Machine learning encompasses to a great variety of areas of study and practical application in computer science, statistics, data science, engineering, economics, genomics etc. See, for example, Chapter 5 of [LBH15] for a high-level summary of many applications. One could summarise the essential aspects of machine learning as: *data* and a *model*. Data could refer to traditional tabular numeric values (e.g. stock market indices or weather readings), natural language, digital imagery, digital voice recordings, internet search engine logs etc. All of these fields (and many more besides) make use of data of one form or another. Researchers and practitioners typically wish to use data they have acquired to address questions such as:

1. Do these data support a particular hypothesis?
2. What underlying structure or dynamics are suggested by the observations in these data?
3. Can one use past data to predict future events?
4. Can one algorithmically find certain interesting subsets of a dataset?

None of these questions are unique to the field of machine learning. Indeed, many such questions have been asked by statisticians and physical and biological scientists for centuries. The lines between machine learning and other, as it were, traditional statistical or mathematical modeling techniques are not entirely clear. Generally speaking, a machine learning approach to a problem is driven more by the data than any particular model. Motivated by intuition, prior observations or theoretical work, a physicist would traditionally start by proposing a model for the physical system under consideration and then obtain predictions to be tested theoretically. The physical model may well contain a number of parameters, such as physical constants, which should be estimated from data, however these parameters are typically few in number and possess meaningful physical interpretations. The physicist's model is as much a tool for making useful predictions about the world as it is a tool with which the underlying physical reality may be studied. A physicist may be able to improve the *predictive* power of their model, say, by introducing more parameters that can be tuned to the available data, but doing so would compromise its physical foundations and degrade its *explanatory* power. To the machine learning practitioner, there is no tension here: data is king and, crudely speaking, a model that better fits and predicts the data is a superior model.

The preceding description certainly does not precisely define machine learning and there are doubtless examples of machine learning applications that lie outside of what we have presented, however our focus is exclusively on neural networks which, as we shall see, fall well within the boundary of machine learning as we have presented it. In the following subsections, we will outline sub-fields within machine learning. Such is the success of deep neural networks in modern machine learning, they are to be found in use in all of these sub-fields and, in many cases, they are the best available approach.

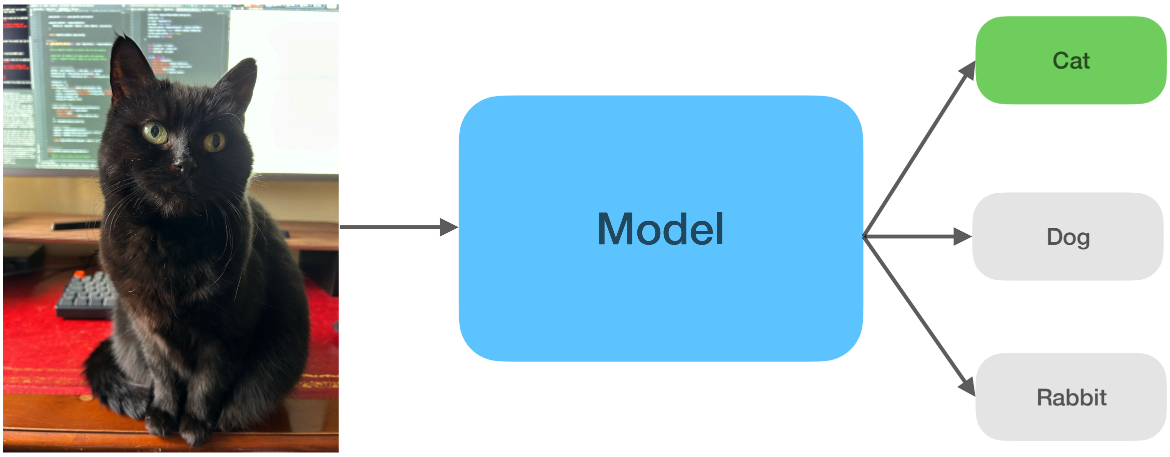
1.1.1 Supervised learning

A very common problem in machine learning is that of constructing a model from a *labeled dataset*. Consider a dataset of the form $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$, where $\boldsymbol{x}_i$ are the data points and the y_i are the *labels*. The $\boldsymbol{x}_i$ may have come from any source and may or may not have a natural numerical representation as column vectors in some $\mathbb{R}^d$, however we assume that a representation of that form has been found.

In some cases, the $\boldsymbol{x}_i$ may have genuine geometrical meaning, while in other cases they may simply be numerical values stacked into vectors. The labels can be categorical, in which case the problem is called *classification*, or continuous, in which case the problem is *regression*. Here are two specific examples:

1. $\boldsymbol{x}_i = (\#\text{bedrooms}_i, \#\text{bathrooms}_i, \text{floor area}_i, \text{latitude}_i, \text{longitude}_i)$ and $y_i = \text{market value (£)}$ for a set of houses in the UK.

2. $\boldsymbol{x}_i = (\text{pixel}_{i1}, \ldots, \text{pixel}_{id})$ and $y_i \in \{(0,0,1),(0,1,0),(1,0,0)\}$ for a set of images categorised into three disjoint classes: cat, dog and rabbit.



A model in this context is a function f that is a good approximation to $\boldsymbol{x}_i \mapsto y_i$. f should not simply be a memorisation of the pairs $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, since applications typically require f to be useful on other datasets $\{(\hat{\boldsymbol{x}}_i, \hat{y}_i)\}_{i=1}^M$ generated from the same underlying distribution as $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, or else to reveal something of the underlying distribution. f can be deterministic or stochastic and must be computable by some algorithm, preferably quite efficiently, though this is not a universal necessity. To be more precise, let us introduce a data generating distribution $\mathbb{P}_{\text{data}}$ supported on $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is in the majority of cases some $\mathbb{R}^d$ or a subset thereof. $\mathcal{Y}$ may be a subset of some $\mathbb{R}^c$ in the regression case, or a countable or even finite set in the classification case. A single sample $(\boldsymbol{x}, y)$ from $\mathbb{P}_{\text{data}}$ is a single data point and its corresponding label, while a dataset $\mathcal{D}$ is some finite sample from $\mathbb{P}_{\text{data}}$ (usually taken to be sampled i.i.d.). Let $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ be two separate finite datasets sampled from $\mathbb{P}_{\text{data}}$. Supervised learning consists of using the *training set* $\mathcal{D}_{\text{train}}$ to construct a model $f : \mathcal{X} \to \mathcal{Y}$ such that $(\boldsymbol{x}, f(\boldsymbol{x}))$ is close in distribution, in some sense, to $\mathbb{P}_{\text{data}}$, and practically this is measured using the *test set* $\mathcal{D}_{\text{test}}$.

No modern summary of supervised learning would be complete without mentioning *semi-supervised learning*. Within the context of this thesis, the distinction between supervised and semi-supervised learning is not of much importance; the difference lies in how the labels are obtained. Standard supervised learning datasets are often constructed by expending human effort to assign labels to data points. For example, people may be paid to label images with which they are presented as containing some objects of interest. In some cases, labels can be obtained systematically without any human labeling, for instance in the example of house prices above, the data already exist in some database (though of course human effort was almost certainly required at some point to generate the data and input them to the database). In semi-supervised learning, labels are derived directly from the data points in some algorithmic manner. A quite natural example is that of time series, where a model may be constructed to predict, say, the temperature in Bristol tomorrow given the observed temperate today and for every day in the previous week. Thus the $\mathcal{X}$ is 7 dimensional (one dimension for each day), and $\mathcal{Y}$ is one dimensional (the temperature tomorrow). Given a dataset of historical temperatures in Bristol, simply a univariate time series T_i where i indexes the day, one can automatically construct a labelled dataset: $\boldsymbol{x}_i = ((T_{i-7}, \dots, T_{i-1}), T_i)$, for all i for which the indices are valid. Any supervised learning method can then be applied to the resulting labelled data set to produce a model capable of predicting tomorrow's temperature. Again, from the perspective of this thesis, semi-supervised learning is indistinguishable from supervised learning, so we will not discuss it further.

1.1.2 Unsupervised learning

Unsupervised learning considers the case where one only has data points $\boldsymbol{x}$ and no labels y. Returning again to the house prices example, given only a dataset of data points $\boldsymbol{x}$ containing key parameters about houses, but no labels giving their market value, what can one learn about houses in UK? For example, one might imagine that using only the key parameters contained in $\boldsymbol{x}$ from a large dataset of houses, one could discover useful structure about broad categories of houses. One common strategy that is particularly relevant in the context of deep learning is *embedding*. Given only a data set of data points $\{\boldsymbol{x}_i\}_{i=1}^N$, an embedding model is some map $f : \mathbb{R}^d \to \mathbb{R}^e$ where typically $e < d$. Whatever the meaning or structure of the native data points $\boldsymbol{x}_i \in \mathbb{R}^d$, the embedding model f will usually be constructed so that the embeddings $\{f(\boldsymbol{x}_i)\}_{i=1}^N$ have some useful geometrical meaning. The canonical example of embedding models are word embedding models, for example see [Mik+13; PSM14; Boj+17], where the data sets are just large collections of natural language, and the embedding models aim to represent words in some Euclidean space such that the geometry of Euclidean space has semantic meaning.

1.1.3 Generative modelling

Consider a dataset $\{\boldsymbol{x}_i\}_{i=1}^N$ sampled from some underlying distribution $\mathbb{P}$. We wish to construct an approximating distribution $\tilde{\mathbb{P}}$ from which samples can be easily drawn. In this case, $\tilde{\mathbb{P}}$ would be the model. A very elementary example of a generative modeling problem would be heights of people in some population, say x_i = height of person i. In this case, we expect $\mathbb{P}$ to be Gaussian and so $\tilde{\mathbb{P}}$ can be obtained simply by estimating the mean and variance. We can extend to produce a less trivial example, where the population is a co-educational school. Rather than fitting a single Gaussian to the whole population, it would clearly be sensible to split into boys and girls and by year groups, and fit a Gaussian to each. Sampling a height from the population then consists of sampling boy/girl from a Bernoulli random variable, sampling year group from a Categorical random variable, and then sampling the height from a Gaussian. Clearly, even in the still rather modest example, the problem of appropriately estimating all of the Gaussian means and variances and the Bernoulli and categorical probabilities is much harder than estimating a single Gaussian, but the model is more expressive and will likely better represent the data. A much more complicated and modern example, is $\boldsymbol{x}_i = (\text{pixel}_{i1}, \dots, \text{pixel}_{in})$ for some set of images of faces. Constructing an adequate parametric model is likely to be infeasible in this case, with the overwhelmingly most successful modern approach being *generative adversarial models* (GANs) [Goo+14b] (see below).

1.1.4 Loss surfaces and the training of machine learning models

As some of the above examples have already hinted, constructing a machine learning model has two distinct stages: model design and model training. In the height example above, model design is simply the choice to use a Gaussian distribution and model training is just estimating the mean and variance, e.g. by taking the sample mean and the unbiased estimate of the population variance. Increasing in complexity, let us consider a linear regression model $f(\boldsymbol{x}) = W\boldsymbol{x} + \boldsymbol{b}$, where the matrix W and the vector $\boldsymbol{b}$ contain the parameters of the model. Here model design is the choice of the form of f, namely as a linear map, while model training consists of choosing W and $\boldsymbol{b}$ to obtain f that best fits the data out of all possible models of the same linear form. It happens that the linear regression models, like Gaussian models, are one of the few model types for which optimal parameters can be computed exactly and in closed-form.

Let us discuss how more general machine learning models are constructed and trained. We will describe the supervised case, for the sake definiteness, but much of what we say applies, *mutatis mutandis*, to unsupervised and generative modeling. Consider again a dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$ where $\boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}^c$, for some positive integers d, c. Denote again by $\mathbb{P}$ the underlying distribution from which the pairs $(\boldsymbol{x}_i, y_i)$ are sampled; all expectations below are taken with respect to $\mathbb{P}$. We fix some *loss function*[1]

$$\begin{aligned} \mathcal{L}\colon \mathbb{R}^c \times \mathbb{R}^c &\to \mathbb{R} \\ y, \hat{y} &\mapsto \mathcal{L}(y, \hat{y}) \end{aligned}$$

which is some typically simple function chosen to measure the performance of a model. Typically there is some quantity of practical interest that one wishes to optimise a model with respect to, for example classification accuracy or mean-squared-error. $\mathcal{L}$ will either be directly the quantity of interest (e.g. mean-squared-error) or will be chosen to correlate with the quantity of interest (e.g. mutual entropy in the case of accuracy). We can now state the central aim of machine learning as an

[1] Also known as an *objective function*, or simply 'loss' or 'objective'.

optimisation problem:

$$\mathrm{argmin}_{f\in\mathscr{F}}\mathbb{E}\mathcal{L}(y, f(\boldsymbol{x})) \tag{1.1}$$

where $\mathscr{F}$ is some class of functions. Of course, in any non-trivial case, one does not have access to $\mathbb{P}$ but only the finite sample $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$. The training set is used to optimise the function f, while the test set is reserved for estimating $\mathbb{E}\mathcal{L}(y, f(\boldsymbol{x}))$ so that the quality of the training procedure can be measured. Here are some examples of loss functions:

1. L_2: $\mathcal{L}(y, \hat{y}) = (y-\hat{y})^2$.

2. L_1: $\mathcal{L}(y, \hat{y}) = |y-\hat{y}|$.

3. Cross-entropy: $\mathcal{L}(y, \hat{y}) = -\sum_j y_j \log \hat{y}_j$.

The set of functions $\mathscr{F}$ can be defined in a variety of ways, but will always have some set of parameters which are tuned to minimise the training loss

$$\sum_i \mathcal{L}(y_i, f(\boldsymbol{x}_i)). \tag{1.2}$$

Here are some examples of $\mathscr{F}$:

1. Linear regression: $\mathscr{F} = \{f(\boldsymbol{x}) = W\boldsymbol{x} + \boldsymbol{b} \; : \; W \in \mathbb{R}^{c\times d}, \boldsymbol{b} \in \mathbb{R}^c\}$. Parameters are $\boldsymbol{w}$ (regression coefficients) and $\boldsymbol{b}$ (bias). $\mathscr{F}$ is isomorphic to $\mathbb{R}^{dc+c}$ as a vector space.

2. Gaussian process regression: $\mathscr{F}$ consists of the posteriors given the data $\{(\boldsymbol{x}_i y_i)\}_i$ and a prior with mean function $m\colon \mathbb{R}^d \to \mathbb{R}$ and covariance function $k\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. m and k may be simple functions possessing of a small number of *hyper-parameters*. $\mathscr{F}$ is infinite-dimensional, though it is possible to consider the posterior for a fixed data set and then there are simply the prior hyperparameters to tune, giving again a space isomorphic to some $\mathbb{R}^K$.

3. Neural networks, a full discussion of which is given below in Section 1.2.

Henceforth, we shall consider only finite-dimensional $\mathscr{F}$ isomorphic to some $\mathbb{R}^N$ and we assume a given parametrisation of $\mathscr{F}$ with some vector parameter denoted by $\boldsymbol{w}$. For $\boldsymbol{w} \in \mathbb{R}^N$, $f_{\boldsymbol{w}} \in \mathscr{F}$ denote the member of $\mathscr{F}$ corresponding the vector of parameters $\boldsymbol{w}$.

Having defined $\mathscr{F}$ and $\mathcal{L}$, we obtain the notion of the *loss surface*

$$\{\mathbb{E}\mathcal{L}(y, f(\boldsymbol{x})) \; : \; f \in \mathscr{F}\}. \tag{1.3}$$

Finding the global minimum, or some sufficiently good local minimum or saddle point, on the loss surface is a matter of tuning a finite number of parameters. As mentioned above, there are some special cases for which the globally optimal parameters can be computed in closed-form. For linear regression with L_2 loss, one can straightforwardly compute derivatives and solve $\partial\mathcal{L}/\partial\boldsymbol{w} = 0$ to find a unique global minimum. In almost all cases, however, no such exact solution will be possible and one must resort of approximate algorithmic approaches. A simple approach which nevertheless turns out to be extremely powerful and the basis of much of modern machine learning is *gradient descent*. Suppose that one can compute the gradient

$$\frac{\partial}{\partial \boldsymbol{w}} \mathcal{L}(y, f_{\boldsymbol{w}}(\boldsymbol{x})), \tag{1.4}$$

where this may be exact or in some cases approximate. Defining a small *learning rate* $\eta > 0$, a natural way to slightly improve the parameters is

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \sum_i \frac{\partial}{\partial \boldsymbol{w}} \mathcal{L}(y_i, f_{\boldsymbol{w}}(\boldsymbol{x}_i)) \tag{1.5}$$

where $\boldsymbol{w}_t$ our the current parameter estimates and $\boldsymbol{w}_{t+1}$ are the updated parameters. One could imagine repeatedly iterating to update the parameters and obtaining optimal, or at least sufficiently performant, parameters. $\sum_i$ can refer to a sum over the whole training set, some subset, or a a single item. In the the first case, the described algorithm is precisely gradient descent, whereas in the latter two cases, if the subset is randomly sampled, the algorithm is stochastic gradient descent, since at each iteration a noisy estimate of the gradient is computed.

1.2 Neural networks

In this thesis, a *neural network* shall refer exclusively to a particular type of machine learning model that was originally coined as *artificial neural network* (ANN) [JMM96] to draw distinction between the machine learning models and the biological systems by which they are inspired. For our purposes, and typically for the purposes of modern machine learning, any historical connection with biological neural networks is of limited value (despite being historically important) and so we adopt the common terminology of merely *neural network*, with the 'artificial' being implicit. The distinction between neural networks and *deep neural networks* is important, practically and theoretically, and will be made clear in the following discussion.

Conceptually, neural networks are non-linear functions from $\mathbb{R}^d$ to $\mathbb{R}^c$ parameterised by some $\boldsymbol{w} \in \mathbb{R}^N$ and formed as the composition of simple affine-linear maps and simple pointwise non-linearities in a layered structure. Being composed of simple, modular components, neural networks provide an elegant and efficient way of effectively arbitrarily scaling the capacity of models. Heuristically, the number of parameters N of a parameterised model determines its capacity to learn patterns in data: the larger N is, the more complicated and diverse the patterns that can be learned. Naturally one then wishes to define models with many parameters and easily scale up the number of parameters to obtain better results on complicated datasets. With traditional statistical models, the parameters typically have some interpretation, being attached to some distribution for example, and so substantially increasing the number of parameters will typically require complete redesign of the model. Even with non-neural machine learning models, it is typically not possible to arbitrarily scale the number of model parameters, as they are typically constrained by the design of the model and/or the data. For example, a linear regression model has no freedom: the number of parameters is determined entirely by the data dimensionality. Neural networks immediately solve this issue, essentially providing a simple recipe for constructing arbitrarily large models of some fixed type. Neural networks are defined by their *architecture* and their parameters. The architecture is the specification of how the parameters $\boldsymbol{w}$ are used to define a function $f_{\boldsymbol{w}}$. There are many different architectures in the machine learning literature and in practical use [LBH15], however there are a small number of standard types of architecture that cover the vast majority of architectures - we shall describe a few of the most significant types below. Finally, we note that it is near-ubiquitous in the machine learning literature to use the term neural network to refer to specific architectures with arbitrary parameters (which are families of functions) *and* specific architectures with specific parameters (which are bona fide functions).

Multi-layer perceptrons (MLPs) The simplest and oldest type of neural network is the MLP[2] [GD98; Mur91]. Let $L > 0$ be an integer, and let $n_0, n_1, \ldots, n_L > 0$ be integers, with $n_0 = d$ and $n_L = c$. Define matrices $W^{(i)} \in \mathbb{R}^{n_{i-1} \times n_i}$ and vectors $\boldsymbol{b}^{(i)} \in \mathbb{R}^{n_i}$; these are the *weights* and *biases* respectively. Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a non-linear function[3] - the *activation function*. Theoretically, σ is often assumed to be differentiable, though this assumption is not required by some of our results. In all practical cases, σ will be twice-differentiable except possibly at a finite set of points at which it is merely continuous. We shall use this latter, weaker, condition, with the convention that, whenever expressions involving derivatives of σ are encountered, they implicitly exclude the finite set of points at which the derivative does not exists. This convention mirrors what is seen in practice, where $\sigma'(x_*) = \lim_{x \to x_*^-} \sigma'(x)$ for any non-differentiable point x_*. An MLP with L layers is now defined as

$$f_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{z}^{(L)}, \quad \boldsymbol{z}^{(l)} = W^{(l)} \sigma(\boldsymbol{z}^{(l-1)}) + \boldsymbol{b}^{(l)}, \ l = 1, \ldots, L, \quad \boldsymbol{z}^{(0)} = \boldsymbol{x}, \tag{1.6}$$

where $\sigma(\boldsymbol{x})$ for vector $\boldsymbol{x}$ is defined as the vector with components $\sigma(x_i)$, i.e. σ is applied element-wise. There may optionally be another non-linearity applied to $\boldsymbol{z}^{(L)}$, which may be different from σ, but we will not need to consider that case here. Note that if $L > 1$, all layers apart from the final layer are called *hidden layers*. *Deep neural networks* are usually defined to be networks with at least one hidden layer, though most of the major practical successes of neural networks comes from models with tens, or even hundreds, or hidden layers. Machine learning using deep neural network is commonly referred to as *deep learning*.

Convolutional neural networks (CNNs) MLPs are a very general form of neural network that can be applied to data of any structure, given some strategy for converting each data point to a single vector representation. If the data are not naturally represented as vectors, forcing them into such

[2] MLPs are also commonly called fully-connected networks.

[3] Note that the definition of any neural network works if σ is linear, but this case is not generally interesting (as it results in linear neural networks), so we exclude it by definition.

a representation so that an MLP can be used is likely to be sub-optimal. The classical motivating example is that of image data, where each data point is an image and so naturally represented as a rank 3 array of pixels: (width, height, channels). By flattening the pixel arrays in vectors and applying an MLP, we would almost certainly be making the learning problem more difficult than it really is. For example, if the network's only objective is to detect cats in images, a picture of a cat located in the top left of the image should appear the same to the network as a picture of a cat in the bottom right of the image, but an MLP presented with flattened vectors must learn separately to identify cats in all possible locations. CNNs are the standard solution to this kind of problem, particularly for image data [LeC+89; LB+95], but also for other data types such as time series and even natural language text [DG14]. We can write a basic CNN as:

$$f_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{z}^{(L)}, \quad \boldsymbol{z}^{(l)} = g(\sigma(\boldsymbol{z}^{(l-1)}); W^{(l)}, \boldsymbol{b}^{(l)}), \; l = 1, \ldots, L, \quad \boldsymbol{z}^{(0)} = \boldsymbol{x}, \tag{1.7}$$

where $g(\cdot; W, \boldsymbol{b})$ is an affine-linear function with respect to its input and also its parameters $W, \boldsymbol{b}$, and the shape of the weights and biases are entirely general. This definition is clearly a strict generalisation of the MLP, which is given by $g(\boldsymbol{x}; W, \boldsymbol{b}) = W\boldsymbol{x} + \boldsymbol{b}$. CNNs take g to be a *convolution* operation. Let $W \in \mathbb{R}^{2k+1 \times 2k+1 \times c_1 \times c_2}$ be a *kernel* and let $\boldsymbol{x} \in \mathbb{R}^{h \times l \times c_1}$, then

$$g(\boldsymbol{x}; W_{ijk}) = \sum_{p=i-k}^{i+k} \sum_{q=j-k}^{j+k} \sum_{r=1}^{c_1} W_{pqrk} x_{pqr}. \tag{1.8}$$

g can be similarly defined to include biases, care must be taken with the definition at the edges (e.g. when $i - k < 0$) and the first two indices of W needn't have odd dimension, but for our purposes there is no need to consider these details. Here $2k+1$ is the *filter size*, c_1 is the number of input channels and c_2 the number of output channels. c_1, c_2 are the analogue of the input and output size of each layer of an MLP. Typically, in the first layer of a CNN, k is much less than h and l, so that the number of parameters in W is much less than the number of parameters in the a corresponding weight matrix of an MLP: $(2k+1)^2 c_1 c_2$ compared to hlc_1c_2.

Note also that the convolutional structure of (1.8) reuses entries of W in multiple location on the input $\boldsymbol{x}$. As well as reducing the number of parameters compared to equivalent MLPs, CNNs also restrict to functions which are translation invariant in the desired sense motivated by the above example of cat detection in images. Finally, note that CNNs are special case of MLPs; the operation defined in (1.8) is affine-linear and so for any index flattening transformation $\phi(\boldsymbol{x})$ there exists a matrix $\hat{W}$ such that $g(\boldsymbol{x}; W) = \hat{W}\phi(\boldsymbol{x})$. Nevertheless, CNNs are preferred to MLPs on any data for which the convolution operation is appropriate, as they provide a beneficial *inductive bias*, essentially encouraging the optimisation procedure (recall (1.5)) to find superior local optima than would be found for an MLP.

Sequential modelling architectures CNNs are well-adapted to image data and, loosely speaking, data which can reasonably be represented as images (e.g. spectrograms [Bad+17]). CNNs have also been successfully applied to natural language data [DG14], however there are a few other architecture types designed for natural language data and other sequential data. In particular, *recurrent neural networks* (RNNs) [MJ01] and later variants such as long short-term memory (LSTM) [HS97b] networks and gated recurrent units (GRUs) [Chu+14] have architectures designed to respect the time-ordering of the data (e.g. the order of words in a sentence) while possessing the appropriate time re-parametrisation invariance. More recently, transformer models [Dev+19; Bro+20] have been proposed and enjoyed considerably practical success over RNN and CNN architectures.

Architecture combinations The different architecture types outlined above need not be used in isolation, but can be combined. For instance, it is standard practice to construct architectures as a concatenation of a CNN and an MLP, with the MLP acting on the flattened output of the CNN[4]. RNNs and transformer architectures are usually built as extensions of MLPs, though there are also convolutional examples (see e.g. convolutional RNNs).

Generative adversarial networks (GANs) MLPs, CNNs and the various sequential modelling architectures are the most common neural network architecture types in practical use and, between them, provide the basis of the vast majority of applications of deep learning to supervised, unsupervised and semi-supervised learning problems. GANs [Goo+14b] are the canonical basic approach to generative modelling using neural networks. GANs are composed of two neural networks: *generator* (G) and *discriminator* (D). G is a map $\mathbb{R}^m \to \mathbb{R}^d$ and D is a map $\mathbb{R}^d \to \mathbb{R}$. G's purpose is to generate synthetic data samples by transforming random input noise, while D's is to distinguish between real data samples and those generated by G. Given some probability distribution $\mathbb{P}_{data}$ on some $\mathbb{R}^d$, GANs have the following minimax training objective

$$\min_{\boldsymbol{w}_G}\max_{\boldsymbol{w}_D}\left\{\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_{data}}\log D(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(0,\sigma_z^2)}\log(1-D(G(\boldsymbol{z})))\right\}, \tag{1.9}$$

where $\boldsymbol{w}_D, \boldsymbol{w}_G$ are the parameters of the discriminator and generator respectively. Given a well optimised generator model, one can sample approximately from the data distribution by sampling latent vectors in the space $\mathbb{R}^l$ and passing them through the generator.

Training neural networks By defining neural network architecture suitable for some data and by varying the number of layers, or the size of the layers (i.e. the size dimensions of the weights), one can specify very large families of parameterised non-linear functions with essentially arbitrary expressivity and complexity. Indeed, there are many results beginning with shallow networks [Bar93; Cyb89; HSW89] that establish neural networks as universal function approximators within certain classes of functions and considerable amounts of more recent work that establish the representational power of deep networks [Dau+22; Tel15; PV18; Lu+17; LL20]. Therefore, given any data, any learning task defined on that data and any theoretically possible level of performance, one can be quite sure of constructing an neural network architecture, and hence a family of parametrised

[4] Historically, such concatenations of CNNs and MLPs were the standard approach, so are universally referred to simply as CNNs and networks with only convolutional layers are often called *fully-convolutional networks*.

functions, such that there exist some parameter values giving the specified level of performance at the task on the data. If neural networks are to be practically useful, however, there must exist some feasible algorithm to find such parameter values. Feasible here has at least two meanings:

- computationally feasible, i.e. the algorithm must terminate in a reasonable time using a reasonable amount of computational resource;
- the algorithm must be general-purpose, i.e. one requires algorithms that apply to a wide variety of datasets and architectures - it would be infeasible if a bespoke algorithm were required for every (dataset, architecture) combination.

We have already seen how the layered structure of neural network, building complicated functions from the composition of simple primitives, makes feasible the specification of models with arbitrary capacity and complexity, the layered structure is also essential for feasible training. In particular, despite their potentially enormous size and considerable complexity, most neural networks are efficient to evaluate, as the vast majority of the computational work in their evaluation comprises linear algebraic operations which have been well-optimised for many computational architectures [LBH15; Pas+17; Aba+16; Ber+15]. Moreover, the layered structure makes possible the efficient and automatic computation of derivatives of neural networks with respect to their parameters. Indeed, consider the form an MLP in (1.6). Differentiating $f_{\boldsymbol{w}}(\boldsymbol{x})$ with respect to any of the $W^{(l)}$ is a mathematically simple matter: one simply applies the chain rule. Let us define $\boldsymbol{y}^{(l)} = \sigma(\boldsymbol{z}^{(l)})$, so $\boldsymbol{z}^{(l)} = W^{(l)}\boldsymbol{y}^{(l-1)} + \boldsymbol{b}^{(l)}$. Then

$$\frac{\partial \boldsymbol{z}^{(l)}}{\partial \boldsymbol{y}^{(l-1)}} = W^{(l)}, \quad \frac{\partial \boldsymbol{y}^{(l)}}{\partial \boldsymbol{z}^{(l)}} = \sigma'(\boldsymbol{z}^{(l)}), \quad \frac{\partial \boldsymbol{z}^{(l)}}{\partial W^{(l)}} = \boldsymbol{y}^{(l)}, \tag{1.10}$$

so observe that, if σ' is known in closed-form and an implementation provided, a computer can implement the chain rule to automatically compute exact derivatives of $f_{\boldsymbol{w}}$. If derivatives of $\mathcal{L}$ are also implemented, then the full derivatives $\partial_{\boldsymbol{w}}\mathcal{L}(y, f_{\boldsymbol{w}}(\boldsymbol{x}))$ can be computed for any $\boldsymbol{x}, y$ and at any $\boldsymbol{w}$. Moreover, all these gradient computations also benefit from the optimised implementations of linear algebraic primitives. In the machine learning literature, computing $f_{\boldsymbol{x}}(\boldsymbol{x})$ is called a *forward pass* and computing $\partial_{\boldsymbol{w}} f_{\boldsymbol{w}}(\boldsymbol{x})$ is called a *backward pass*. Since neural networks allow for efficient automatic computation of loss gradients $\partial_{\boldsymbol{w}}\mathcal{L}(y, f_{\boldsymbol{w}}(\boldsymbol{x}))$, the simplest algorithm one could imagine to optimise the parameters $\boldsymbol{w}$ for a dataset is stochastic gradient descent (1.5). So far it is clear that using SGD in combination with neural network backward pass represents a feasible optimisation algorithm for general neural networks and it quite feasible to perform hundreds of thousands of steps of SGD in an acceptable time-frame, though obviously this varies with model and dataset size, as do the requirements on the computational hardware. However this discussion does not address the quality of the optimisation. That is to say, we have described a procedure for neural network optimisation that is general-purpose, feasible to implement and apply to any architecture and dataset, and simple computational experiments would be sufficient to determine how many SGD steps can be performed per second for a given model and given hardware. For this procedure to be of value, however, it must, with sufficient probability, find parameter values $\boldsymbol{w}$ that give sufficiently good performance of the neural network on the defined task. While SGD is an intuitive and appealing algorithm, the cases for which it can be proven to find, say, global minima are narrow [PJ92; VPF21] and certainly cannot be expected to generically apply to deep neural networks. Indeed, a priori, for large neural networks with many parameters, one should expect there to be a great many saddle points and local optima of the loss surface around which SGD could get stuck. Algorithmic innovations can somewhat mitigate the problem of saddle points, such as endowing the gradient descent trajectory with momentum [Nes13] or adjusting the learning rates in different directions on the loss surface [DHS11; KB14], and these techniques can greatly improve practical performance of neural networks [Bot12]. In very high dimensions the intuition of such techniques does not necessarily apply and if there are a great many local minima, then we should expect SGD to converge to, at best, some local minimum determined by the random initialisation of $\boldsymbol{w}$. In general, there is no reason to expect that these local minima will provide network performance anywhere near the global optimum, or even useful performance at all. In bold defiance of these arguments, neural networks continue to have substantial success when applied to an increasingly long list of machine learning problems: computer vision, speech processing, natural language processing, reinforcement learning, media generation etc. We refer the interested reader to the excellent website [cod20] where they will find links to published literature detailing the success of neural networks in all fields of machine learning. Networks are trained using stochastic gradient-based optimisation methods on very high-dimensional, strongly non-convex surfaces for which no formal convergence or performance guarantees exist and yet excellent practical performance is routinely obtained with little concern for whether the optimisation problem has been solved. Extremely over-parametrised models can be trained with large numbers of passes through the data without overfitting. Models with equivalent training performance can have radically different generalisation performance depending on complicated interactions between design choices such as learning rate size (and scheduling) and weight-decay [LH18].

1.3 Structure of neural network loss surfaces

One strand of theoretical work focuses on studying properties of the loss surfaces of large neural networks and the behaviour of gradient descent algorithms on those surfaces. Much of the content of this thesis sits within this line of research. [Sag+14] presented experimental results pointing to a similarity between the loss surfaces of multi-layer networks and spherical multi-spin glasses [MPV87]. [Cho+15] built on this work by presenting modeling assumptions under which the training loss of multi-layer perceptron neural networks with `ReLU` activations can be shown to be equivalent to a spherical multi-spin glass (with network weights corresponding to spin states). The authors then applied spin glass results of [AAC13] to obtain precise asymptotic results about the complexity[5] of the training loss surfaces. Crudely, the implication of this work is that the unreasonable efficacy of

[5] *Complexity* will be given a formal definition in Chapter 2.

gradient descent on the high-dimensional and strongly non-convex loss surfaces of neural network models can in part be explained by favourable properties of their geometry that emerge in high dimensions. Relationships between simpler neural networks and spin glasses have been known since [KS87; GD88; EV01] and, more generally, connections between spin glass theory and computer science were studied in [Nis01] in the context of signal processing (image reconstruction, error correcting codes).

More recent work has dispensed with deriving explicit links between neural networks and spin glasses, instead taking spin glass like objects as a tractable playground for gradient descent in complex high-dimensional environments. In particular, [Bai+19] compare empirically the dynamics of state-of-the-art deep neural networks and glassy systems, while [Man+19b; Ros+19; Aro+19; Man+19a] study random tensor models containing some 'spike' to represent other features of machine learning problems (some 'true signal' to be recovered) and perform explicit complexity calculations as well as gradient descent dynamical calculations revealing phase transitions and landscape trivialisation. [MBB20] simplify the model in favour of explicitly retaining the activation function non-linearity and performing complexity calculations à la [AAC13; FW07; Fyo04] for a single neuron. [PB17] study the loss surface of random single hidden layer neural networks by applying the generalised Gauss-Newton matrix decomposition to their Hessians and modelling the two components as freely-additive random matrices from certain ensembles. [PW17; BP19] consider the loss surfaces of single layer networks by computing the spectrum of the Gram matrix of network outputs. These works demonstrate the value of studying simplified, randomised neural networks for understanding networks used in practice. The situation at present is far from clear. The spin glass correspondence and consequent implications for gradient descent based learning from [Cho+15; Sag+14] are tantalising, however there are significant challenges. Even if the mean asymptotic properties of deep neural network loss surfaces were very well described by corresponding multi-spin glass models, the question would still remain whether these properties are in fact relevant to gradient-based algorithms running for sub-exponential time, with some evidence that the answer is negative [Bai+19; Man+19a; FFR19]. Another challenge comes from recent experimental studies of deep neural network Hessians [Pap18; GKX19; Gra20a; Gra+19b] which reveal spectra with several large outliers and considerable rank degeneracy, deviating significantly from the Gaussian Orthogonal Ensemble semi-circle law implied by a spin glass model. Bearing all this in mind, there is a long and illustrious history in the physics community of fruitfully studying quite unrealistic simplified models of complicated physical systems and still obtaining valuable insights into aspects of the true systems.

Several of the assumptions used in [Cho+15] to obtain a precise spherical multi-spin glass expression are undesirable, as outlined clearly in [CLA15]. Assuming i.i.d. Gaussian data and random labels is clearly a going to greatly simplify the problem, however it is also the case that many of the properties of deep neural networks during training are not specific to any particular dataset, and there may well be phases of training to which such assumptions are more applicable than one might first expect. Gaussian and independence assumptions are commonplace when one is seeking to analyse theoretically very complicated systems, so while they are strong, they are not unusual and it is not unreasonable to expect some important characteristics of real networks to persist. By contrast, the restriction of the arguments in [Cho+15] to exclusively `ReLU` activations seems innocuous, but we argue quite the opposite is true. There are deep mathematical reasons why Gaussian and independence assumptions are required to make progress in the derivation in [Cho+15], while the restriction to `ReLU` activations appears to be an obscure peculiarity of the calculations. The `ReLU` is certainly a very common choice in practice, but it is by no means the only valid choice, nor always the best; see e.g. leaky `ReLU` in state-of-the-art image generation [KLA19] and GELU in state-of-the-art language models [Dev+18]. It would not be at all surprising if a spin glass correspondence along the lines of [Cho+15] were impossible without Gaussian and/or independence assumption on the data, however it would be extremely concerning if such a correspondence specifically required `ReLU` activations. If the conclusions drawn in [Cho+15] about deep neural networks from this correspondence are at all relevant in practice, then they must apply equally to all activation functions used in practice. On the other hand, if the conclusions were *precisely* the same for all reasonable activation functions, it would reveal a limitation of the multi-spin glass correspondence, since activation function choice can have significant implications for training neural networks in practice.

1.4 Contributions of this thesis

In Figure 1.1 below we give a diagram that outlines the contributions of this thesis and their position within the literature. Rounded purple boxes denote antecedents and influences of our contributions from the literature. The references given in these boxes are not intended to be exhaustive but simply indicators. Rectangular orange boxes denote our contributions, where we display both the published papers and the corresponding Chapter in this thesis. We expand further on the context of this thesis and its contributions in the following subsections. Chapters 3 and 4 form the first major contribution and are discussed in section 1.4.1. Chapters 7 and 8 form a distinct major contribution but are nevertheless related to the the earlier chapters, as indicated in the diagram. Chapters 5 and 6 are distinct contributions that are certainly connected to the major parts of the thesis, but are more peripheral in their contribution; they are discussed in section 1.4.3 and 1.4.4 respectively.

1.4.1 Generalisation of spin glass models for neural networks loss surfaces

The first major contribution of this thesis is a significant generalisation of the understanding of spin glass models for neural network loss surfaces. Beginning with Chapter 3, we return to the modeling assumptions and methodology of [Cho+15] and extend their results to multi-layer perceptron models with any activation function. We demonstrate that the general activation function has the effect of modifying the exact multi-spin glass by the addition of a new deterministic term in the Hamiltonian. We then extend the results of [AAC13] to this new high-dimensional random function.

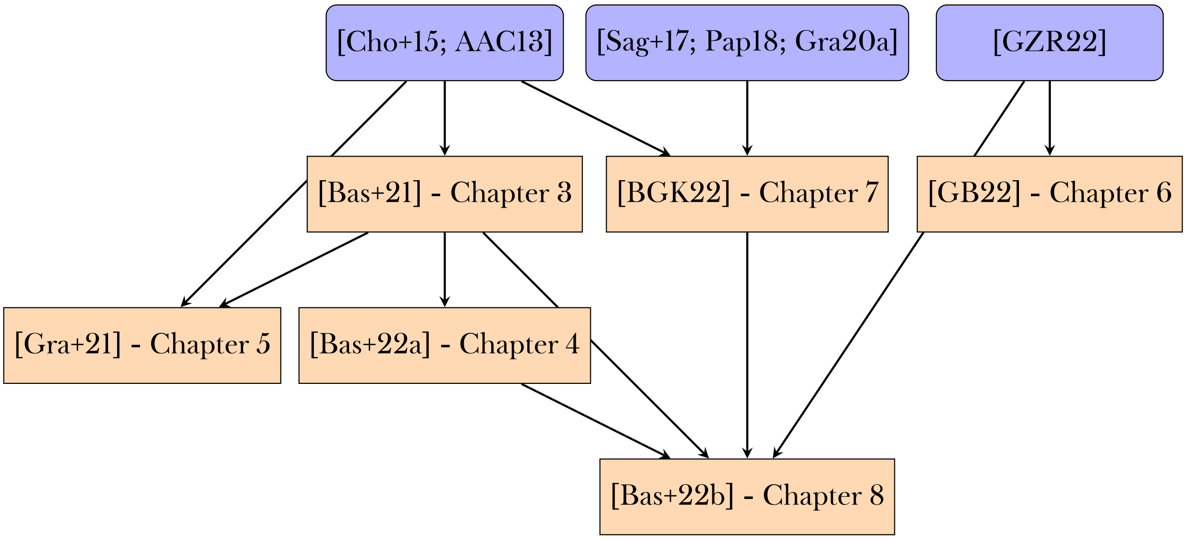


Figure 1.1: Schematic of the contributions of this thesis

At the level of the logarithmic asymptotic complexity of the loss surface, we obtain precisely the same results as [Cho+15], however the presence of a general activation function is felt in the sharp asymptotic complexity. On the one hand, our results strengthen the case for [Cho+15] by showing that their derivation is not just an accident in the case of `ReLU` networks. On the other hand, we have shown that this line of reasoning about neural networks is insensitive to an important design feature of real networks that can have significant impacts on training in practice, namely the choice of activation function. The main calculation for our result uses a Kac-Rice formula to compute landscape complexity of the modified multi-spin glass model we encounter. Kac-Rice formulae have a long history in the Physics literature [BM80; BM81] and more specifically to perform complexity calculations [Fyo04; Fyo05; AAC13]. Complexity calculations in spiked matrix and tensor models in [Ros+19; Aro+19] have addressed spin glass objects with specific rank-1 deterministic additive terms, however those calculations do not extend to the case encountered here since those deterministic terms create a single distinguished direction — parallel to the gradient of that term everywhere on the sphere — which is critical to their analysis; our extra deterministic term creates no such single distinguished direction. We chart a different course using supersymmetric methods in Random Matrix Theory. Supersymmetric methods have been used before in spin glass models and complexity calculations [CGG99; Ann+03; Cri+03; Fyo04], often using the replica trick. We show how the full logarithmic complexity results of [AAC13] can be obtained using a supersymmetric approach quite different to the approach used in that and similar works. By moving to this approach, we can make progress despite the presence of the extra deterministic term in the multi-spin glass. Our approach to the supersymmetric calculations most closely follows [FN15; Noc16], but several steps require approximations due to the extra term. Some of our intermediate results in the supersymmetric and RMT calculations are stronger than required here, but may well be useful in future calculations, e.g. spiked spherical multi-spin glass models with any fixed number of spikes. Finally, our approach computes the total complexity summed over critical points of any index and then uses large deviations principles to obtain the complexity with specified index. This is the reverse order of the approach taken in [AAC13] and may be more widely useful when working with perturbations of matrices with known large deviations principles.

Motivated by our results in Chapter 3, we ask if it is possible to further extend the spin glass modeling approach to capture yet further peculiarities and details of modern neural networks. We seek, in particular, a model that is capable of revealing the influence of architectural details *at leading order* in the annealed complexity, unlike the relatively weak effect of the activation function seen in Chapter 3. Modern deep learning contains a very large variety of different design choices in network architecture, such as convolutional networks for image and text data (among others) [Goo+16; Con+17], recurrent networks for sequence data [HS97b] and self-attention transformer networks for natural language [Dev+19; Rad+18]. Given the ubiquity of convolutional networks, one might seek to study those, presumably requiring consideration of local correlations in data. One could imagine some study of architectural quirks such as residual connections [He+16], and batch-norm has been considered to some extent by [PW17]. In Chapter 4, we propose a novel model for *generative adversarial networks* (GANs) [Goo+14a] as two interacting spherical spin glasses. GANs have been the focus of intense research and development in recent years, with a large number of variants being proposed [RMC15; Zha+18b; LT16; KLA20; MO14; ACB17; Zhu+17] and rapid progress particularly in the field of image generation. From the perspective of optimisation, GANs have much in common with other deep neural networks, being complicated high-dimensional functions optimised using local gradient-based methods such as stochastic gradient descent and variants. On the other hand, the adversarial training objective of GANs, with two deep networks competing, is clearly an important distinguishing feature, and GANs are known to be more challenging to train than single deep networks. Our objective is to capture the essential adversarial aspect of GANs in a tractable model of high-dimensional random complexity which, though being a significant simplification, has established connections to neural networks and high dimensional statistics.

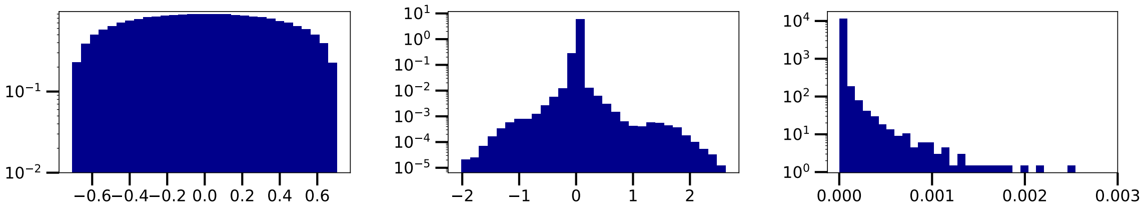
Our model is inspired by [Cho+15; Ros+19; Man+19b; Aro+19] with spherical multi-spin glasses being used in place of deep neural networks. We thus provide a complicated, random, high-dimensional model with the essential feature of GANs clearly reflected in its construction. By employing standard Kac-Rice complexity calculations [Fyo04; FW07; AAC13] we are able to reduce the loss landscape complexity calculation to a random matrix theoretic calculation. We then employ various Random Matrix Theory techniques as in [Bas+21] to obtain rigorous, explicit leading order asymptotic results. Our calculations rely on the supersymmetric method in Random Matrix Theory, in particular the approach to calculating limiting spectral densities follows [Ver04] and the calculation also follows [GW90; Guh91] in important ways. The greater complexity of the random matrix spectra encountered present some challenges over previous such calculations, which we overcome with a combination of analytical and numerical approaches. Using our complexity results, we are able to draw qualitative implications about GAN loss surfaces analogous to those of [Cho+15] and also investigate the effect of a few key design parameters included in the GAN. We compare the effect of these parameters on our spin glass model and also on the results of experiments training real GANs. Our calculations include some novel details, in particular, we use precise sub-leading terms for a limiting spectral density obtained from supersymmetric methods to prove a required concentration result to justify the use of the Coulomb gas approximation. We note that our complexity results could be also be obtained in principle using the methods developed in [ABM21a], however our work was completed several months before this pre-print appeared. Our approach for computing the limiting spectral density may nevertheless be the simplest and would be used as input to the results of [ABM21a].

The role that statistical physics models such as spherical multi-spin glasses are to ultimately play in the theory of deep learning is not yet clear, with arguments both for and against their usefulness and applicability. Before our contributions, the major result was [Cho+15] which, though influential, has received considerable criticism and could have reasonably been considered a parochial curiosity, rather than profound insight into neural network loss surfaces. Our work in Chapter 3 considerably weakens the case against [Cho+15], and our work in Chapter 4 clearly demonstrates the potential of spin glass models (and statistical physics based models in general) to capture and explain phenomena in deep neural networks. Indeed, to the best of our knowledge, Chapter 4 provides the first attempt to model an important architectural feature of modern deep neural networks within the framework of spin glass models. Our analysis reveals potential explanations for observed properties of GANs and demonstrates that it may be possible to inform practical hyperparameter choices using models such as ours. Much of the advancement in practical deep learning has come from innovation in network architecture, so if deep learning theory based on simplified physics models like spin-glasses is to keep pace with practical advances in the field, then it will be necessary to account for architectural details within such models; our work is a first step in that direction.

1.4.2 Discovery of RMT universality in loss surfaces and consequences for loss surface models

The other major contribution of this thesis is the instigation of the study of the role of random matrix theory statistics in deep learning at the local (i.e. microscopic) scale and the building of a strong case that the results which characterise the first half of the thesis, and other RMT-based results from the literature besides, can be expected to be much more general in applicability than their very restrictive modeling assumptions would suggest.

An important and fundamental problem with Chapters 3 and 4 and related work in the literature is that typically the average spectral density of the Hessian of neural networks does not match that of the associated canonical random matrix ensembles that results from the modeling assumptions and are crucial in the technicalities of the calculations. This is illustrated in Figure 1.2. Put simply, *one does not observe the Wigner semicircle or Marchenko-Pastur eigenvalue distributions, implied by the Gaussian*



(a) Wigner semicircle (b) MLP (c) Logistic Regression

Figure 1.2: Comparison of different global spectral statistics (spectral densities). (a) We show actual GOE data to demonstrate the form of the Wigner semicircle. (b) Hessian of cross entropy loss for MLP on MNIST. (c) Hessian of cross entropy loss for logistic regression on MNIST. Note the log-scale on the y-axis. A few outliers have been clipped from logistic regression to aid visualisation.

Orthogonal or Wishart Ensembles. As shown in [Gra20a; Gra+19b; Pap18; Pap19; GKX19; SBL16; Sag+17] the spectral density of neural network Hessians contain outliers and a large number of near zero eigenvalues, features not seen in canonical random matrix ensembles. Furthermore, even allowing for this, as shown in [Gra+20] by specifically embedding outliers as a low rank perturbation to a random matrix, the remaining bulk spectral density still does not match the Wigner semicircle or Marchenko-Pastur distributions [Gra20a], bringing into question the validity of the underlying modelling. The fact that the experimental results differ markedly from the theoretical predictions has called into question the validity of neural network analyses based on canonical random matrix ensembles. Moreover, the compelling results of works such as [Cho+15; PB17] are obtained using very particular properties of the canonical ensembles, such as large deviation principles, as pointed out in [Gra20a]. The extent to which such results can be generalised is an open question. Hence, further work is required to better understand to what extent random matrix theory can be used to analyse the loss surfaces of neural networks. In Chapter 7, we show that the *local spectral statistics* (i.e. those measuring correlations on the scale of the mean eigenvalue spacing) of neural network Hessians are well modelled by those of GOE random matrices, even when the mean spectral density is different from the semicircle law. We display these results experimentally on MNIST trained multi-layer perceptrons and on the final layer of a ResNet-34 on CIFAR-10. The objective of Chapter 7 is to motivate a new use for Random Matrix Theory in the study of the theory of deep neural networks. In the context of more established applications of random matrix theory, this conclusion may not be so surprising – it has often been observed that the local spectral statistics are universal while the mean density is not – however, in the context of machine learning this important point has not previously been made, nor its consequences explored. In Chapter 7 we illustrate it in that setting, through numerical experiments, and start to examine some of its implications.

Having established experimentally the presence of universal local random matrix statistics in real-world neural networks (though admittedly very small ones by modern standards), we proceed in Chapter 8 to demonstrate how local random matrix statistics can be used as modeling assumptions for models of deep neural network Hessians to obtain surprisingly strong generalisations of prior spectral results. Works such as [AAC13; Cho+15; Fyo05] and our own contributions in Chapters 3 and 4 show how detailed calculations can be completed beyond in and beyond the standard spin glass case, however these results all depend on important properties of the GOE, to which the Hessians in those cases are closely related. In a recent work, [GZR20] showed how valuable practical insights about DNN optimisation can be obtained by considering the outliers in the spectrum of the loss surface Hessian. Once again, this work relies on special properties from random matrix theory, indeed an expression for the outliers follows from a known phase transition result whereby the largest eigenvalue "pops out" of the bulk. This result has been proven only for rotationally invariant matrix ensembles in [BN11], itself a generalisation of the celebrated BBP phase transition [BAP05], though it was conjectured in [BN11] to be more general (a point which we clarify in Chapter 8, section 8.1.3). In addition, the explicit form of a Wigner semi-circle density was used to obtain the concrete outlier expression used in practice.

Microscopic random matrix universality is known to be far more robust than universality on the macroscopic scale. Indeed, such results are well established for invariant ensembles and can be proved using Riemann-Hilbert methods [Dei99]. For more general random matrices, microscopic universality has been proved by quite different methods in a series of works over the last decade or so, of which a good review is [EY17a]. Crucial in these results is the notion of a *local law* for random matrices. The technical statement of local laws is given later in section 2.7, but roughly they assert that the spectrum of a random matrix is, with very high probability, close to the deterministic spectrum defined by its limiting spectral density (e.g. the semicircle law for Wigner matrices). Techniques vary by ensemble, but generally a local law for a random matrix ensemble provides the control required to demonstrate that certain matrix statistics are essentially invariant under the evolution of the Dyson Brownian motion. In the case of real symmetric matrices, the Dyson Brownian motion converges in finite time to the GOE, hence the statistics preserved under the Dyson Brownian motion must match the GOE. The n-point correlation functions of eigenvalues are one such preserved quantity, from which follows, amongst other properties, that the Wigner surmise is a good approximation to the adjacent spacings distribution.

At the macroscopic scale, there are results relevant to neural networks, for example [PSG18; Pas20] consider random neural networks with Gaussian weights and establish results that are generalised to arbitrary distributions with optimal conditions, so demonstrating universality. On the microscopic scale, our work in Chapter 7 provided the first evidence of universal random matrix theory statistics in neural networks and was subsequently to the weight matrices of neural networks in [TSR22], but no prior work has considered the implications of these statistics, that being the central contribution of Chapter 8. Our main mathematical result is a significant generalisation of the Hessian spectral outlier result recently presented by [GZR20]. This generalisation removes any need for GOE or Wigner forms of the Hessian and instead leverages much more universal properties of the eigenvectors and eigenvalues of random matrices which we argue are quite likely to hold for real networks. Our results make concrete predictions about the outliers of DNN Hessians which we compare with experiments on several real-world DNNs. These experiments provide indirect evidence of the presence of universal random matrix statistics in the Hessians of large DNNs, which is noteworthy as certainly these DNNs are far too large to permit exact eigendecomposition of their Hessians as done in Chapter 7. Along a similar line, we show how local random matrix laws in DNNs can dramatically simplify the dynamics of certain gradient descent optimisation algorithms and may be in part responsible for their success. Finally, we highlight another aspect of random matrix universality relevant to DNN loss surfaces. Recent work [ABM21a] has shown that the so-called 'self averaging' property of random matrix determinants is very much more universal than previously thought. The self-averaging of random matrix determinants has been used in the spin glass literature both rigorously and non-rigorously (e.g. [Fyo04; Fyo05; AAC13; Bas+21; Bas+22a] inter alia) and is the key property that produces the exponentially large/small number of local optima repeatedly observed. We argue that insights into the geometry of DNN loss surfaces can be conjectured from quite general assumptions about the Hessian and gradient noise and from the general self-averaging effect of random matrix determinants.

1.4.3 Correlated noise models for neural network loss surfaces

Spin glass and statistical physics based models provide an important perspective on the geometric and statistical properties of neural network loss surfaces, as is extensively explored in Chapters 3 and 4, alongside prior work in the literature. Part of the appeal of these approaches is their ontological separation from classical approaches to analysis and methods of proof in statistical learning theory. Having defined a model and settled on stochastic gradient descent (or a variant) as the optimisation approach, a natural question is: does stochastic gradient descent converge under some assumptions and what, if any, guarantees are there on the parameters to which it converges? Questions like this are well-studied in the statistical learning and optimisation literature [PJ92; VPF21], but in the context of neural network this work is of limited applicability as the known results all rely on properties that are not possessed by neural networks, such as convexity of the loss surface (as a function of the network weights). Some recent work has established convergence results using weaker assumptions like the PL inequality [Bel21] and [RYH22] has developed a theory of neural network training dynamics based on perturbation analysis. In aggregate, there are many separate results giving guarantees on SGD under a variety of assumptions, some of which are plausible for neural networks but what exists is far short of a complete theory. The results based on spin glass models, as seen in Chapter 3 and 4, are quite different in nature from these SGD convergence results, providing insight into the overall structure and complexity of the loss surfaces on which SGD operates. These approaches are able to capture much more of the genuine complexity of the loss surfaces of real neural networks than the classical SGD convergence analyses, however the results they provide are somewhat like descriptive sketches of the the loss surfaces, unlike the precise convergence guarantees of the classical analyses. In Chapter 5 we present work that bridges that gap between these two parallel streams of thought. Concretely, we obtain several variations on SGD convergence results, particularly in the case of *iterate averaging*. Iterate averaging is a well-known technique in stochastic optimisation, where the parameter iterates $\boldsymbol{w}_k$ are simply averaged to produce the new sequence

$$\hat{\boldsymbol{w}}_t = \frac{1}{t}\sum_{k=1}^{t} \boldsymbol{w}_k. \tag{1.11}$$

Intuitively, this simple averaging should have the effect of reducing the variance in the parameter estimates, and indeed this very fact is critical in some convergence proofs, such as that for Adam [KB14] given in [RKK19]. That being said, to the best of our knowledge there has been no explicit theoretical work analysing the generalisation benefit of iterate averaging. Whilst [Izm+18] propose that iterate averaging leads to "flatter minima which generalise better", flatness metrics are known to have limitations as a proxy for generalisation [Din+17b]. [Mar14] show that the iterate average convergence rate for both SGD and second-order methods are identical, but argue that second-order methods have an optimal pre-asymptotic convergence rate on a quadratic loss surface. Here, pre-asymptotic means before taking the number of iterations $t \to \infty$ and quadratic means that the Hessian is constant at all points in weight-space. The analysis does not extend to *generalisation* and no connection is made to adaptive gradient methods, nor to the importance of the high parameter-space dimensionality of the problem, both of which are addressed by our approach in Chapter 5. Amendments to improve the generalisation of adaptive methods include switching between Adam and SGD [KS17] and decoupled weight decay [LH18], limiting the extent of adaptivity [CG18; Zhu+20]. We incorporate these insights into our algorithms but significantly outperform them experimentally. The closest algorithmic contribution to our work is *Lookahead* [Zha+19], which combines adaptive methods with an exponentially moving average scheme.

The key contribution of Chapter 5 is to introduce spin glass like statistical models for neural network loss into the realm of SGD convergence results. In particular, we make use of a general stationary Gaussian process model for the noise of the loss surface which is a generalisation of the spin glass models used prior work and our own and bring two important benefits. Firstly, these models are intrinsically amenable to asymptotic analysis in the regime of very large parameter dimensionality, indeed this kind of asymptotic analysis is our focus in Chapters 3 and 4. As there, this is an important feature of any analysis of neural networks, as virtually all successful modern applications use large networks with very many parameters. Secondly, these loss surface models are inherently models of statistical dependence between the noise on loss surface gradient iterates, a feature which, again, is central to the calculations in Chapters 3 and 4. In the context of SGD convergence results and iterate averaging, statistical dependence between gradient iterates is essential for a realistic analysis, as the weights, and hence gradients, at each iteration of stochastic gradient descent are clearly not independent. Beginning with a simple model of independent, isotropic Gaussian gradient noise, we first establish a basic result for SGD with iterate averaging in the high-dimensional regime, exhibiting explicitly the variance reduction effect of iterate averaging compared to standard SGD. We then replace the inadequate and naïve assumption of independent gradient noise with a Gaussian process model for the loss noise, from which we derive a dependent model for the gradient noise. In this setting, we prove a generalised convergence result for SGD and SGD with iterate averaging, again demonstrating the variance reducing effect of iterate averaging but also providing insights into the effect of learning rate which derives directly from the dependence between gradient iterates. We additionally establish a sequence of results for variations on the basic Gaussian process noise model and also for certain adaptive gradient descent algorithms. Overall, our work provides an entirely novel approach to the modeling and analysis SGD algorithms which incorporates important properties of modern neural networks and creates connections between two previously separate approaches in the study of their training. Our novel perspective on the issue of SGD convergence and iterate averaging provides insight into the interaction between iterate averaging, adaptive gradient descent methods and learning rates, which helps to explain why most experimental results with iterate averaging may have historically been poor.

1.4.4 Practical application of random matrix loss surface models for hyperparameter tuning

A unifying feature of all work in this thesis is the study of neural networks via models of their loss surfaces. Our work shows how such models can be developed and analysed to shed light on the important features such as the configuration of local optima and the spectral outliers of loss surface Hessians, both of which are relevant to gradient-based optimisation of f neural networks' parameters. As important as these studies are for advancing the relatively primitive theoretical understanding of what has become a ubiquitous and indispensable approach to machine learning, the immediate practical applications are quite limited. The spin-glass models of Chapters 3 and 4 are largely without any direct practical application, being too crude a statistical modern for practical neural networks. We demonstrate in Chapters 7 and 8 that universal local random matrix theory statistics can be used to build much more realistic models of neural network loss surfaces and yield detailed predictions about spectral outliers of their Hessians. It is beyond doubt that such results about spectral outliers are of practical use, as clearly demonstrated in [GZR22], where the results are used to derive practical and effective scaling rules for learning rates. Our results considerably expand and substantiate those of [GZR22], but it has not been demonstrated that these much more precise results add anything practically over the cruder and less rigorous approach of their antecedents. Chapter 6 introduces an entirely new application of random matrix theory techniques to neural network loss surfaces, producing immediate practical benefit to the training of real-world networks.

The founding idea of Chapter 6 is a simple observation about a very common numerical 'hack' used in several standard variants of stochastic gradient descent. Let $L(\boldsymbol{w})$ be the loss surface of some neural network with parameters $\boldsymbol{w} \in \mathbb{R}^N$ and let $H = \nabla^2 L$ be its Hessian.

Stochastic gradient descent updates weights according to the rule

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k \nabla L \tag{1.12}$$

where $\boldsymbol{w}_k$ are the network parameters after k iterations of SGD and at each iteration a different batch is used. $\alpha_k > 0$ is the *learning rate* which, in the simplest setting for SGD, does not depend on k, but in general can be varied throughout training to achieve superior optimisation and generalisation. The general form of adaptive optimiser updates is

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k B^{-1} \nabla L \tag{1.13}$$

where B is a *pre-conditioning matrix*. The essential idea of adaptive methods is to use the pre-conditioning matrix to make the geometry of L more favourable to SGD.

One approach is to take B to be diagonal, which can be thought of as having per-parameter learning rates adapted to the local loss surface geometry. More generally, one might seek an approximation B to the local loss surface Hessian, effectively changing the basis of the update rule to a natural one, with per-direction learning rates. Alternatively, if $B \approx H$ then the local quadratic approximation to the loss surface, i.e. the second-order term in a Taylor expansion, is isotropic in weight space. What both of these approaches have in common is that they in principle allow for bigger steps (i.e. larger α_k, as the different scales of the ∇L in the different parameters are normalised. Indeed, a standard approach for diagonal B is to construct a diagonal approximation to H. Without this, α_k must essentially be tuned to be so small that the change of $\boldsymbol{w}$ in the direction of the largest component of ∇L is not too large. For Adam [KB14], the most commonplace adaptive optimiser in the deep learning community, B is given by the diagonal matrix with entries $\frac{\sqrt{\langle g_k^2 \rangle} + \varepsilon}{\langle g_k \rangle}$. Here $\boldsymbol{g}$ is the loss gradient and $\langle \cdot \rangle$ denotes an empirical exponential moving average or iterations.

For many practical problems of interest, the test set performance of adaptive gradient methods is significantly worse than SGD [Wil+17]—a phenomenon that we refer to as the *adaptive generalisation gap*. As a consequence of this effect, many state-of-the-art models, especially for image classification datasets such as CIFAR [Yun+19] and ImageNet [Xie+19; Cub+19], are still trained using SGD with momentum. Although less widely used, another class of adaptive methods which suffer from the same phenomenon [Tor20] are *stochastic second order methods*, which seek to alter the learning rate along the eigenvectors of the Hessian of the loss function. KFAC [MG15] uses a Kroenecker factored approximation of the Fisher information matrix (which can be seen as a positive definite approximation to the Hessian [Mar14]). Other methods use Hessian–vector products [Dau+14; Mar10] in conjunction with Lanczos methods and conjugate gradients [MS06]. All second order and adaptive gradient methods, are endowed with an extra hyper-parameter called the damping or numerical stability co-efficient respectively. This parameter limits the maximal learning rate along the eigenvectors or unit vectors in the parameter space respectively and is typically set to a very small value by practitioners.

In principle there is no reason why a certain parameter gradient should not be zero (or very small) and hence the inversion of B could cause numerical issues. This is the original reason given by [KB14] for the numerical stability coefficient ε. Similarly so for KFAC for which $B = \sum_i^P \lambda_i \phi_i \phi_i^T$ where $\{\lambda_i, \phi_i\}_{i=1}^P$ are the eigenvalue, eigenvector pairs of the kronecker factored approximation to the Hessian. Hence to each eigenvalue a small damping coefficient δ is added. Whilst for both adaptive and second order gradient methods, the numerical stability and damping coefficients are typically treated in the literature as extra nuisance parameters which are required to be non-zero but not of great theoretical or practical importance, we strongly challenge this view. In Chapter 6, we relate these coefficients to the well known linear shrinkage method from random matrix theory. It is clear from a random matrix theory perspective, that the sub-sampling of the Hessian will lead to the creation of a noise bulk in its spectrum around the origin, precisely the region where the damping coefficient is most relevant. We show, both experimentally and theoretically, that these coefficients should be considered as extremely important hyper-parameters whose tuning has a strong impact on generalisation. Furthermore, we derive from a random matrix theory additive noise model of the loss surface Hessian a novel algorithm for their online estimation, which we find effective in experiments on real networks and datasets.

1.4.5 Mathematical contributions

We end this section with a brief summary of the purely mathematical contributions of this thesis, much of which has been covered above but in the context of their applications.

Due to the presence of an additive term deforming the GOE matrix, in Chapter 3 we are forced to use different methods to obtain the complexity results analogous to [AAC13] and in so doing provide a novel approach to these calculations. [AAC13] starts by computing the index-specific complexity and then sums over index to obtain the non-specific complexity. By contrast, we use supersymmetric methods to first obtain the non-specific complexity and then use the large deviations principle to reintroduce the index dependence. To the best of our knowledge, this approach has not been used before, though there are of course many works that perform the first part of this calculation for various models.

In Chapter 4 we make use of the Coulomb gas method to calculate a random matrix determinant as part of the complexity calculation, which is entirely routine, however we also provide a proof of the validity of the Coulomb gas method for the relevant matrix ensemble. The proof structure is a standard matter of establishing complementary upper and lower bounds. The proof of the upper bound makes use of standard probabilistic inequalities and properties of Gaussians, however we use the supersymmetric method integral representations to derive error bounds on the mean spectral density which are the key ingredient in the proof of the lower bound.

Finally, in Chapter 8 we prove a novel result for the limiting spectral measures of additions of random matrices. It is well known [AGZ10; VDN92] that the sum of two free independent random matrices with well defined limiting spectral measures has a limiting spectral measure given by the free convolution of the two. We are able to establish the same free convolutional limiting spectral measure but requiring only that one of the matrices obeys quantum unique ergodicity. The proof of this result is also a novel application of quantum unique ergodicity, as we leverage a supersymmetric representation to compute the limiting spectral and use the defining quantum unique ergodicity property to compute the integral over the matrix eigenvectors.

1.5 Literature review of deep learning theory

We close this chapter with a broad review of the literature on deep learning theory. This is a field experiencing a tremendous amount of activity so our review shall be far from exhaustive. We will give particular attention to the literature related to random matrix theory, but shall also seek to highlight the other broad approaches that have attained some prevalence.

1.5.1 Random matrix theory

Random and complex landscapes The work most closely related to our own began with [Cho+15; CLA15; Sag+14] where the connections between neural network loss surfaces and spin glasses were first introduced and studied, with the underpinning mathematical results being drawn from the random matrix theory literature such as [Fyo04; Fyo05; AAC13]; we discuss these works in detail elsewhere in this chapter and the next. In the same lineage of work are more recent notable examples such as [Ros+19; Man+19b; Aro+19] which can be summarised as the study of high-dimensional signal-plus-noise models. These works avoid any direct connection to neural networks, instead focusing on much simpler random matrix and tensor models that act as playgrounds for stochastic gradient descent on high-dimensional loss surfaces. This approach is of course inspired by [Cho+15] and these works similarly consider issues of loss surface complexity, but with the explicit inclusion of extra structure, or 'signal'. This signal was notably lacking from [Cho+15], as the spin-glass is really just a model of pure noise. Intuitively, one expects that the loss surfaces of real neural networks contain some underlying structure induced by the structure of the data and the network itself, but that a considerable component of noise is also induced on the surface by the noise on the data and also possibly the weights and biases themselves. By creating simple, paired-down loss surface models containing the same kind of high-dimensional noise present in the spin glass, but with some signal (or structure) injected, these works are able to study questions about the presence and prevalence of *spurious minima* i.e. local minima of the noisy loss surface that are uncorrelated with the true minima of the noise-less surface. They uncover phase transitions between chaotic surfaces on which the structure-induced minima are swamped by spurious minima and surfaces which, though they contain many noise-induced minima, the structure of the minima is such that the signal is still recoverable.

Random neural networks In the line of work discussed above, random matrices arise somewhat indirectly in the study of neural networks via the Kac-Rice approach to landscape complexity analysis. Since neural networks are constructed using, and parametrised by, weight matrices in each of their layers, one can naturally seek a theory of *random* neural networks by considering these weight matrices to be random. [PB17] bridged the gap between studies of landscape complexity and random neural networks by considering networks with i.i.d. normal weights applied to i.i.d. normal data and computing the limiting spectral density of their Hessians in the large parameter number limit. They decompose the Hessian as the sum of a positive semi-definite matrix (often called to Gauss-Newton matrix elsewhere [Mar16a]) and a matrix that contains all the dependence on the residuals (i.e. the error terms between the network predictions and the truth values). With this decomposition, they make assumptions of free independence to enable the use of tools from free probability to compute the limiting spectral densities. By assuming also an i.i.d. Gaussian form of the residuals parameterised by some variance ε, they are able to describe the spectra of neural networks Hessian at different loss values a compare with experiment. Random networks were also considered in [LLC+18] in the context of random feature ridge regression i.e. a 1-layer neural network with MSE loss and an L2 ridge regularisation penalty for which only the final layer is trained. The first layer, being untrained, acts as a random transformation of the the input data and then the weights of the final layer have a unique solution known in closed form, since the final layer is simply a linear ridge regression on the random features. Since the final layer weights can be solved in closed form, a closed form is available for the training error which is found to be given in terms of the resolvent $Q = (N^{-1}\Sigma^T\Sigma + \gamma I)^{-1}$ where $\Sigma = \sigma(WX)$ are the random features produced by the random weights W and input data points X and N is the number of random features (i.e. the width of the hidden layer). The proofs rely largely on concentration properties of sub-Gaussian random variables to establish that various random matrix quantities concentrate on their expectations. In a related work [PW17] 1-layer neural networks with random weights were considered. The authors compute the limiting spectral density of the Gram matrix Y^TY of the network output Y. This work was the first in which the non-linearities introduced by neural network activation functions were handled directly and analytically in the setting of random matrix theory, since [LLC+18] was restricted to polynomial activation functions. The weight entries and the data entries are assumed to i.i.d. Gaussians and the proof of the limiting spectral density uses the moment method of random matrix theory. An interesting consequence of the results is that there exist certain non-linear activation functions for which the Gram matrix spectrum is the Marcenko-Pastur distribution, so that the spectrum is preserved through the non-linear activation function. The authors conjecture that these "isospectral" activation functions may have beneficial practical properties for training, as the spectral statistics remain constant through the layers, an idea somewhat reminiscent of batch norm [IS15]. [BP19] extends the results of [PW17] to more general (i.e. sub-Gaussian) entry distributions on the network weights and the data, using again a moment method proof. They also extend to the case of multiple layers, though the results in that case are very intricate and opaque. Continuing again in this line of work, [ALP22] extends the analysis to 1-layers random networks with random biases ans shows that the distribution of the biases induces something like a mixture over activation functions. [PSG18] considers the input-output Jacobian J of random multi-layer networks using the techniques of free probability theory to derive the spectrum of the Gram matrix JJ^T. Using these results, they are able to derive necessary and sufficient conditions on the spectra of the weight matrices to give a stable spectrum (i.e. not no explosion nor collapse) in the large network depth limit. These results were subsequently generalised and given a fully-rigorous proof in a series of papers by Pastur and collaborators [Pas20; PS22; Pas22]. The first paper in the sequence considers the Gaussian case, as in [PSG18], with the chief difficulty being that the free independence that is required to apply the streamlined free probability argument given in [PSG18] is not apparent. The second paper extends to general i.i.d. distributions with at least four finite moments and the third extends to weights matrices with orthogonal distributions (so not i.i.d. entries). Another perspective on random neural networks is given in the works [SPS17; Yan+19], where the techniques of mean field theory are applied to the standard multi-layer perceptron architectures, firstly with linear or ReLU activations and then with more general activations and batch normalisation. The training loss of the network plays the role of the Lagrangian and the partition function is computed by explicitly integrating out the random (i.i.d. Gaussian) weights and biases. In the case of batch normalisation, the authors are able to use the mean field techniques to make predictions about instabilities (e.e. due to gradient explosion) of very deep networks in the presence of batch normalisation. Beyond the question of why does SGD work at all for deep neural networks, there are various phenomena observed in their training and use that lack adequate theoretical explanations. One such is the *double/triple descent phenomenon*, which is commonly observed in large modern deep neural networks but is at odds with classical statistical learning theory. Standard results from statistical learning theory dictate that the best attainable test loss of a particular model decreases as the number of parameters N of the model increases, but only up to a point beyond which the loss increases again. This is a reflection of the classical *bias-variance* trade-off [Has+09] which states that the expected test error of a machine learning model can be decomposed into two additive terms, bias and variance, which account for different sources of error in the fitting process. High variance means that there is high variation in the estimated parameters between different sampled instances of the training set which indicates that the model tends to systematically fit to the noise in the training data, rather than the underlying structure (called *overfitting*). High bias means that the test error over different sampled instances of the training set is biased away from zero, indicating that the model tends to systematically fail to identify meaningful generalisable structure in the data (called *underfitting*). It is intuitive that a model with too few parameters will tend to underfit, as the model lacks the expressive capacity to capture the structure in the data. On the other hand, a model with too many parameters (i.e. more than are really needed to capture the structure in the data) will tend to overfit as it is has spare capacity that can be used to interpolate noise in the training data, which of course drives down the training loss, but at the expense of increasing the test loss. All of this holds for classical approaches to machine learning, i.e. broadly those before the deep learning revival of the 2010s, however repeated empirical observations with increasingly larger deep networks have revealed that this classical picture has its limits. Modern deep networks used in computer vision applications are routinely chosen to have 10s of millions of parameters, which by any reasonable measure is considerably more than would be required to express the true structure in the data and is indeed sufficient to allow for perfect interpolation of the training data [He+16]. Modern transformer networks used extensively in natural language processing are larger still [Bro+20] with 100s of *billions* of parameters. Repeatedly and in multiple domains, it has been observed that dramatically increasing the number of networks parameters and also the training time can lead to ever better test set performance *even when training data are near perfectly interpolated*. This phenomenon was dubbed the *double descent*, referring to the shape of the graph of test error against number of parameters. Classically, this graph has a single local minimum at the point of bias-variance balance, but very large deep neural networks have revealed a second, lower minimum in the greatly ("abundantly") over-parameterised region [Zha+16; Zha+21; BRT19; Bel+19]. Prior works attempted to analyse this phenomenon in the simplest cases of linear regression models [BHM18], but the key contribution of [AP20b] was to analyse the effect of parameter number on single hidden layer random networks. Neural networks with a single hidden layer are the simplest example of a model in which the number of trainable parameters N can be specified separately from the input data dimension d and target data dimension C, since in a model with no hidden layers (e.g. linear or logistic regression) N is necessarily equal to dC, whereas the width of even a single hidden layer can be specified arbitrarily. The authors were able to show that single hidden layer networks with random i.i.d. Gaussian weights trained on entirely random data with random labels display a double descent, even a *triple descent*, with a third test error minima in an extreme "hyperabdundant" parametrisation region. Much like the earlier work [Cou+19], the test error is expressed as a certain random matrix resolvent which is in turn computed by determining the limiting spectral density of a certain random matrix via tools from free probability theory and invoking notions of random matrix universality to replace the complicated, intractable matrix ensembles arising from the network with certain independent Gaussian matrices. This work produces an immediate insight: the double (triple) descent phenomenon is not unique to deep neural networks, nor even to the type of data on which they are typically trained or the training procedure, but rather it is a "background" property of over-parametrised non-linear models and generic data.

Spectra of neural networks The works discussed so far consider random neural networks and random matrices in neural networks *ex-ante*, i.e. modeling assumptions are made, or models constructed, that explicitly introduce randomness to neural networks or their loss surfaces. Their is another line of work which is better characterised as *ex-post* randomisation, wherein neural networks are directly studied and, for example, spectral properties of their loss surface Hessians or weights are analysed. For the fist time in [Pap18; Pap19], the spectra of loss surface Hessians of real-world neural networks were approximated and analysed. For practical modern neural networks, the loss surface Hessian is of course far too large to even store in memory, let alone compute via automatic differentiation or eigen-decompose, having N^2 entries, where the number of network parameters N is typically 10^7 or more. The key numerical advance in these works is the application of Lanczos iteration methods [Lan50; MS06] to compute high-quality approximations to the spectral density of very large matrices given only the matrix-vector multiplication function $\mathcal{M}_H : \mathbb{R}^N \to \mathbb{R}^N$ with

$\mathcal{M}_H(\boldsymbol{v}) = H\boldsymbol{v}$ and not the whole matrix H. This can be combined with the Pearlmutter trick [Pea94] which computes

$$\frac{\partial^2 l}{\partial w_i \partial w_j}\boldsymbol{v} = \frac{\partial}{\partial w_i}\left(\boldsymbol{v}^T \frac{\partial l}{\partial w_j}\right)$$

which is very much amenable to automatic differentiation in modern deep learning frameworks. Actually, this approach was pioneered contemporaneously by Granziol and collaborators in a sequence of pre-prints for which the best reference is [GZR22]. One of the key insights in those works was to highlight the very considerable discrepancy between the spectra of real neural network Hessians and those of standard canonical random matrix models such as the GOE that is assumed by spin glass models such as [Cho+15] and in [Gra20a], it was proposed that the spectra of products of canonical random matrix ensembles can be used to obtain agreement with certain aspects of the spectra of real neural networks, in particular their considerable rank degeneracy.

These empirical analyses uncover rich and interesting structure in the spectra of real deep neural networks, in particular the spectra clearly display a bulk and some large outliers. The outliers appear to be directly attributable to the classes in a typical classification problem (i.e. one outlier per class) and naturally one expects from random matrix theory that the bulk corresponds to noise [PB20]. There is further structure still, with the discovery in an later work [Pap20] of a group of eigenvalues outside of the bulk[6] but much smaller than the main outliers. There are typically $C(C-1)$ of these outliers, for a C class classification problem, so they appear to correspond somehow to inter-class correlations.

Rather than considering loss surface Hessians, another line of inquiry has directly analysed the spectra of neural network weight matrices before, during and after training. [MM18] consider several types of network trained on real datasets and look at the spectra of their weights matrices at initialisation and as training progresses. They identify several distinct phases of training from these spectra, beginning with full classical random matrix behaviour at initialisation and developing towards some heavy-tailed distribution leading to the conjecture that neural networks are implicitly regularised by some process inducing these heavy tailed spectra as training proceeds. Note that the idea of implicit regularisation of neural networks via stochastic gradient descent pre-dates this work by several years [NTS14; Ney+17a; Ney+17b; Ney17]. Finally, we mention [TSR22] in which the spectra of random and trained neural network weight matrices was analysed but on the *local* scale, rather than the global scale pursued by [MM18]. This work followed on from our own in Chapter 7 [BGK22] and similarly discovered the robust presence of universal GOE random matrix spacing statistics in the spectra.

[6] Though not stated by the author, this extra group of outlier eigenvalues must clearly be outside the Tracy-Widom region as well.

1.5.2 Other approaches

We mentioned above some mean-field approaches to the analysis of neural networks, but this would not be complete without also mentioning the recent work of Roberts and Yaida [RYH21] in which this subject is developed in considerable depth. The authors proceed incrementally from linear networks at initialisation (the simplest case), to non-linear networks and ultimately training dynamics via a perturbation theory approach. This analysis relies heavily on the *neural tangent kernel* which can be introduced quite simply by considering the loss derivatives via the chain rule:

$$\frac{\partial L}{\partial \theta_a} = \sum_i \frac{\partial L}{\partial z_i}\frac{\partial z_i}{\partial \theta_a}$$

where $\boldsymbol{z}$ is the network output which is fed into the loss L. A single step of stochastic gradient descent will update the weights $\boldsymbol{\theta}$ by taking a small step of scale η along the negative gradient direction, so that the leading order (in η) change in the loss is

$$\Delta L = -\eta \sum_{i,j}\sum_{a,b} \frac{\partial L}{\partial z_i}\frac{\partial L}{\partial z_i}\frac{\partial z_i}{\partial \theta_a}\,\frac{\partial z_i}{\partial \theta_b}$$

which leads to the identification of the neural tangent kernel

$$K_{i,j} = \sum_{a,b} \frac{\partial z_i}{\partial \theta_a}\,\frac{\partial z_i}{\partial \theta_b}.$$

The neural tangent kernel can be seen to largely govern the dynamics of stochastic gradient descent for very wide networks (i.e. those with some fixed number of layer but very many parameters in each layer), see e.g. [JGH18; AP20a].

Building on the above-mentioned decomposition of neural network Hessian spectra into components attributable to class centres and inter-class correlations [Pap20], the concept of *neural collapse* has been advanced. Empirical studies of network pre-activations in [PHD20] discovered that, in networks trained to good accuracy, the pre-activations coalesce around C clusters, one for each class in the classification problem. Indeed, as training progresses the pre-activations converge to very low variance around the class cluster centres and the cluster centres themselves converge to an equiangular tight frame.

Another recent line of work studies neural networks in their capacity as function approximators [E+20] and attempts to characterise using the tools of mathematical analysis the sets of functions that can be well approximate by neural networks. A 2-layer (i.e. 1 hidden layer) network can be expressed as a random feature model

$$f(\boldsymbol{x}, \boldsymbol{a}) = \frac{1}{m}\sum_{j=1}^{m} a_j \phi(\boldsymbol{x}; \boldsymbol{w}_j), \quad \phi(\boldsymbol{x}, \boldsymbol{w}) = \sigma(\boldsymbol{x}^T \boldsymbol{w}).$$

This expression can be rewritten as an integral by defining an atomic probability measure $\pi = m^{-1}\sum_{j=1}^{m}\delta_{\boldsymbol{w}_j}$ over the first layer weights $\{\boldsymbol{w}_j\}_j$

$$f(\boldsymbol{x}, \boldsymbol{a}) = \int a(\boldsymbol{w})\phi(\boldsymbol{x}; \boldsymbol{w})d\pi(\boldsymbol{w}),$$

which suggests the generalisation of this expression to any probability measure π, so producing a type of random neural network with marginalised first layer weights. In this construction, the 2-layer MLP network can be viewed as a Monte Carlo integration approximation to this more general object. An important insight about the role of the curse of dimensionality in deep learning is revealed by this formalism. Classical function approximation theory typically constructs approximations of a function f by defining some Sobolev space with a convenient basis, say of polynomials. If m is the number of free parameters in the approximation (e.g. the maximum degree of the polynomial basis) and d is the input dimension of f, then one obtains an approximation error that scales something like $m^{-\alpha/d}$ for some $\alpha > 0$ defined by the details of the chosen approximation space. As the input dimension d grows, this error term becomes less and less favourable, requiring exponentially more free parameters m to achieve the same approximation error. This contrasts sharply with the above Monte Carlo integration interpretation of a 2-layer MLP, which has an error term with the standard MC scaling of $m^{-1/2}$, which crucially is independent of the input dimension d. This analysis approach provides some insight into how neural networks appear to overcome the curse of dimensionality in their input space faced by other approaches to machine learning. The results in [E+20] go further and in fact identify precisely the function spaces for which 2 layer MLPs can provide good approximations.

[Bel21] considers the success of stochastic gradient descent at finding high quality minima for deep neural networks. As we have already discussed, classical optimisation theory holds that finding global minima of non-convex functions is generally intractable and [Bel21] argues that the considerable over-parametrisation of modern neural networks implies that their loss surfaces are filled with many local minima and they are generically not even locally convex around those minima. The *PL inequality* [Pol64; Loj63] for a loss function L with constant μ is $\frac{1}{2}\|\nabla L(\boldsymbol{w})\|^2 \geqslant \mu L(\boldsymbol{w})$ and, combined with a smoothness condition, is sufficient to guarantee exponential convergence of stochastic gradient descent [Bel21], but the PL condition is much weaker than even local convexity. The conclusion of this line of work is broadly that the classical picture that lack of convexity and numerous local minima mean that stochastic gradient descent on neural networks is doomed to fail is overly pessimistic and weaker, more plausible conditions may suffice to provide expectation of convergence.

Chapter 2

Mathematical tools

This chapter aims to provide a self-contained introduction to the main mathematical tools required in the subsequent chapters, intended to be accessible to a mathematical audience with no previous familiarity with random matrix theory.

2.1 Introduction to random matrix theory

Random matrix theory provides much of the mathematical context and insight for the results in this thesis, as well as providing most of the techniques used in the calculations. It is a large a diverse field touching many areas of pure and applied mathematics and physics and we shall not attempt to provide comprehensive introduction. The classic introduction is Mehta's book [Meh04]. Thorough and mathematically orientated modern treatments can be found in the books by Anderson, Guionnet and Zeitouni [AGZ10], Tau [Tao12] and Meckes [Mec19]. Accessible and application orientated introductions are given by [LNV18] and [PB20]. A detailed introduction to modern topics in a mathematically rigorous style can be found in [EY17a]. Given the breadth of random matrix theory, only a fraction of its concepts and tools are required in this thesis and so we restrict this introduction to those.

2.1.1 Random matrices

A random matrix is no more nor less than one would expect, namely a matrix-valued random variable. Such objects are entirely natural in almost any branch of applied mathematics or statistics. Consider for example a sample of N data points each being represented as a tuple of M real values, such as 2-tuples of latitude and longitude for locations of house or 500-long tuples of returns data for the S&P 500 index. It is natural, at least from the perspective of computational convenience, to stack these data points into an array X of shape $N \times M$ with each row corresponding to a single sample. From the perceptive of a pure mathematician thinking of matrices as representations of linear maps on vector spaces, X does not appear to be a matrix, but just a collection of number conveniently packed into a array. Suppose that the N samples are $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ drawn from a multivariate Gaussian distribution $\mathcal{N}(0, \Sigma)$. The information contained in the sample is entirely represented by this sequence in $\mathbb{R}^N$, so what is the purpose of stack them into a 'matrix' X? One answer is, of course, numerical convenience and efficiency. For example, suppose that Σ is known and we wish to construct the standardised variables $\boldsymbol{z}_i = \Sigma^{-1/2}\boldsymbol{x}_i$. One can view this as a sequence of N matrix-vector operations, but it is more mathematically compact and numerically efficient to instead view it as a single matrix-matrix operation $Z = \Sigma^{-1/2}X$. There are, however, deeper and richer reasons to consider X. Consider the matrix $S = \frac{1}{N}X^TX$ - an $M \times M$ positive semi-definite symmetric matrix. One can clearly write

$$S_{ij} = \frac{1}{N}\sum_{k=1}^{N}(\boldsymbol{x}_k)_i(\boldsymbol{x}_k)_j$$

and so S_{ij} is an empirical estimate from N samples of the covariance between the i-th and j-th coordinates in the data distribution. The eigenvalues and eigenvectors of S clearly have meaning, for example the eigenvector corresponding to the largest eigenvalue is the direction in $\mathbb{R}^M$ responsible for the most variance in the data. In the S&P 500 example above, this direction would correspond to 'the market', and in the coordinates example, it may correspond to a major river along whose banks most settlements are found. We need not restrict ourselves to matrices of the form of N samples of M dimensional variables. Consider data collected from a telecommunications network on N end-points (or nodes), examples of which include telephone numbers or registered users of instant messaging services. Let X_{ij} be the number of communication events between end-point i and end-point j observed over some time period. Properly normalised by the total number of events in the same period, X_{ij} could instead be an empirical estimate of the probability of communication between end-points i and j. Viewing X as a symmetric matrix, not merely and array, and computing its spectral decomposition, one will find that the eigenvectors corresponding to meaningful communities in the network, with the eigenvalues giving an estimate of the relative importance of each community in the network.

These examples illustrate a critical point: viewing arrays of random variables (or data) as matrices is not a mere numerical convenience, for one finds that bona fide linear algebraic objects such as eigenvalues and eigenvectors have meaning and structure. Let us return to the example of a matrix X containing financial data, e.g. share prices or returns, for M assets sampled over N days. If M is small compared to a large sample size N, then we can expect much of the noise in the samples to average out to produce a matrix S with M meaningful eigenvector representing genuine correlations between the M assets. In the opposite extreme where M is much larger that N, we expect that many of the genuine correlations in the data will be lost in the noise. But what of the intermediate case, where M and N are of comparable size? Intuitively, one expects that the strongest signals in the data (such as the the market) will be preserved and clearly visible through the noise in the data, while more subtle signals will be lost. Translating this into the language of random matrices, the largest eigenvalues (and their eigenvectors) correspond to genuine signal in the data, while the smallest correspond to sample noise. The obvious question is whether one can separate the signal from the noise, i.e. how many of the largest eigenvalues are signal? This question can be seen, conceptually, as motivating much of the work in random matrix theory. Consider any linear algebraic property of a matrix: eigenvalues, eigenvectors, determinant, trace, characteristic polynomial, condition number, etc. Given a distribution on a matrix, what is the distribution on any of these objects? If one can answer this question for pure noise random matrices, then one can easily identify matrices that contain signal. If one can answer the question in the case of signal-plus-noise random matrices, then one can separate the signal from the noise.

The above discussion has been rather statistically-focused, but historically random matrix theory was used by Wigner and Dyson [Wig67; Dys62a; Dys62b; Dys70] to provide elegant and powerful models for atomic nuclei. The governing quantum mechanical equation for an atomic nucleus is the Schrödinger equation

$$H\psi_i = E_i\psi_i \tag{2.1}$$

where H is an Hermitian operator (the *Hamiltonian*) on an Hilbert space, $\{\psi_i\}$ is a wave functions and E_i are corresponding energy levels. The physical observables here are the energy levels, but in all but the very simplest of cases (such as a Hydrogen nucleus) they cannot be computed analytically, or even numerically, owing to the complexity of the interaction between the nucleons. Dyson and Wigner's insight was that the general appearance of energy levels *on average* can be described by simple statistical models of (2.1) not requiring detailed knowledge of the equation or its solution. To quote Dyson [Dys62b]:

> *The statistical theory will not predict the detailed sequence of levels in any one nucleus, but it will describe the general appearance and the degree of irregularity of the level structure, that is expected to occur in any nucleus which is too complicated to be understood in detail.*

This aspect of random matrix theory will be of particular value in this thesis. We endeavour to understand properties of very large deep neural networks applied to complicated high-dimensional tasks on real-world data. Such models may contain millions of free parameters operating on datasets of millions of samples with many thousands of dimensions per sample and complicated statistical dependence between dimensions. The dynamics of the model parameters as they are trained are far too complicated to be studied directly. As with atomic nuclei many decades earlier, the central hypothesis of this thesis and other related contemporary work is that statistical theories of deep neural networks can describe their general properties and be used to understand their behaviour without reference to the intractable details of their training dynamics.

2.1.2 Random matrix ensembles

Probability distributions on matrices are commonly referred to as *ensembles* in random matrix theory. There are modest number of canonical random matrix ensembles that form the foundation of much of the work in random matrix theory and about which a great deal is known in considerable mathematical detail. The importance of each of the canonical ensembles tends to vary between application areas, so we shall restrict ourselves in this section to only those ensembles that feature in the coming chapters. We shall be exclusively interested in real matrices, as the matrices that arise when studying neural networks and machine learning are almost always real. Moreover, many of the matrices we shall be interested in will be symmetric. The most important random matrix ensemble for this thesis is the *Gaussian orthogonal ensemble* (GOE). There is some variation between authors on unimportant normalisation, but we shall say that a $N \times N$ matrix $X \in \mathbb{R}^{N\times N}$ is a GOE matrix, $X \sim \text{GOE}^N$ if

$$X_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \frac{1+\delta_{ij}}{2N}\right) \text{ for } i \leqslant j \text{ and } X_{ij} = X_{ji} \text{ for } i > j, \tag{2.2}$$

i.e. X has Gaussian entries, independent up-to symmetry and with twice the variance on the diagonal as off-diagonal. This specific variance structure allows for a powerful closed-form expression of the law of X:

$$d\mu(X) = \frac{1}{Z_N} \exp\left(-N\frac{\text{Tr}X^TX}{2}\right) dX \tag{2.3}$$

where Z_N is a normalisation constant and dX is simply the standard Lebesgue product measure on the upper-diagonal and diagonal entries of X. Note that the GOE is called an orthogonal ensemble because it possesses symmetry with respect to the real orthogonal group $O(N)$ on orthogonal matrices. Sampling a matrix X from GOE^N can be done with a very simple algorithm:

$$Y_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1), \quad X = \frac{Y+Y^T}{2N}.$$

The GOE is a specific case of the more general class of *Wigner* matrices, which have independent (up-to symmetry) Gaussian entries with variance σ_d^2/N on the diagonal and σ_u^2/N off the diagonal. Generalising even further, generalised Wigner matrices take the form

$$X_{ij} \overset{\text{i.i.d.}}{\sim} \mu \text{ for } i<j, \;\; X_{ii} \overset{\text{i.i.d.}}{\sim} \mu_d \text{ and } X_{ij} = X_{ji} \text{ for i > j}$$

for any sufficiently well-behaved measures μ and μ_d on $\mathbb{R}$. There are complex Hermitian and quaternionic version of GOE and Wigner matrices for details of which we refer the reader to any standard reference on random matrix theory.

An alternative generalisation of the GOE is born of (2.3), which we rewrite as

$$d\mu(X) = \frac{1}{Z_N} \exp\left(-N\text{Tr}V(X)\right) dX$$

where $V : \mathbb{R}^{N\times N} \to \mathbb{R}^{N\times N}$ is defined to be $V(X) = \frac{1}{2}X^TX$. In deference to its origins in statistical physics, V is often referred to as a potential. With this rewritten form of the GOE density, one can simply change the definition of V and so obtain different distributions on real symmetric matrices.

We shall also encounter matrices distributed with *Haar measure* on the orthogonal group $O(N)$. The Haar measure on any compact group G [Mec19] is the unique measure μ_{Haar} finite on all subsets of G such that

$$\mu(gS) = \mu(S)\ \forall g \in G \text{ and } S \subset G.$$

The Haar measure can be viewed as the 'flat random' measure on G and, in the case $G = O(N)$, a matrix distributed with Haar measure is a uniform random matrix on the real orthogonal group. Geometrically, a matrix with Haar measure on $O(N)$ is a uniform random basis rotation. Haar random orthogonal matrices O can be sampled quite simply by sampling N i.i.d. vectors $\boldsymbol{x}_i$ with i.i.d. $\mathcal{N}(0,1)$ entries and then applying the Gram-Schmidt algorithm the obtain an orthonormal set of vectors $\boldsymbol{o}_1,\ldots,\boldsymbol{o}_N$ which are the rows of the Haar-distributed matrix [Mez06].

Finally, we mention the real *Ginibre* [AGZ10] ensemble on $N \times M$ matrices which are simply matrices with i.i.d. entries and no symmetry constraint. The Ginibre analogue of the GOE is and ensemble of matrices with i.i.d. $\mathcal{N}(0,1/N)$ entries.

2.1.3 Eigenvalues and spectral measures

Let X_N be any real symmetric random matrix of shape $N \times N$. The following discussion could equally be presented for any Hermitian random matrix and could be generalised much further at the expense of having to account for non-real eigenvalues. For the purposes of this thesis, we may restrict our discussion to matrices with real eigenvalues and, to be concrete, let us stick to real symmetric matrices. Let $\lambda_1 < \lambda_2 < \ldots < \lambda_N$ be the eigenvalues of X_N. The *empirical spectral measure* of X_N is defined as

$$\hat{\mu}_N = \frac{1}{N}\sum_{i=1}^{N}\delta_{\lambda_i} \tag{2.4}$$

where δ_λ is a Dirac δ-function mass at location λ, i.e. defined by

$$\int_A \delta_\lambda = \mathbf{1}\{\lambda \in A\}$$

for any set $A \subset \mathbb{R}$. Since X_N is random, its eigenvalues $\{\lambda_1,\ldots,\lambda_N\}$ are random variable with some joint probably density $p(\lambda_1,\ldots,\lambda_N)$. $\hat{\mu}_N$ is a probability measure on $\mathbb{R}$ and moreover, it is a *random* probability measure, its distribution being induced by $p(\lambda_1,\ldots,\lambda_N)$. Imagine constructing many independent samples of X_N, hence from $p(\lambda_1,\ldots,\lambda_N)$ and hence of $\hat{\mu}_N$. Once could imagine averaging the samples of $\hat{\mu}_N$

$$\frac{1}{m}\sum_{j=1}^{m}\hat{\mu}_N$$

and so obtaining some indication of the average location of the eigenvalues of X_N. Intuitively, one would imagine this average measure becoming a better and better approximation to some absolutely continuous measure (though there is, of course, no general guarantee of such convergence). Extending this to a concrete mathematical question: does $\mathbb{E}\hat{\mu}_N$ exist, and what is it? In the same way that one can imagine growing the number of sampled eigenvalues by increasing the number of independent samples of X_N, one can also consider a family of distributions on X_N, parametrised by dimension $N \in \mathbb{N}$, and let the dimension N for a single sample grow. In this context, there is another natural question: does $\lim_{N\to\infty} \hat{\mu}_N$ exist, how strong is the convergence, and what it the limit measure? When it exists, we shall define

$$\mu_\infty = \lim_{N\to\infty} \hat{\mu}_N \tag{2.5}$$

to be the *limiting spectral measure* of X_N (being intentionally vague about the strength of convergence, for now). Likewise, when the expectation exists, we define

$$\mu_N = \mathbb{E}\hat{\mu}_N \tag{2.6}$$

to be the *mean spectral measure* of X_N. When either of these measure are absolutely continuous with respect to Lebesgue measure, we define

$$\rho_\infty(\lambda) = \frac{d\mu_\infty}{d\lambda}$$

to be the *limiting spectral density* (LSD) and similarly

$$\rho_N(\lambda) = \frac{d\mu_N}{d\lambda}$$

is the *mean spectral density*. Let us now be concrete and consider some specific examples, beginning with the most famous.

Example 2.1 (GOE). Recalling the form (2.3) of the GOE measure on $N \times N$ matrices, we can now explore why it was described as “powerful”. Let X be an $N \times N$ GOE random matrix. Since X is real symmetric, it is an elementary result of linear algebra that X can be written in the form $X = U^T \Lambda U$, where $U \in O(N)$ is a real orthogonal matrix and Λ is a real diagonal matrix. Of course, the diagonal entries of Λ are the eigenvalues of X and the rows of U are the corresponding eigenvectors. But now

$$\exp\left(-\frac{N\mathrm{Tr}X^TX}{2}\right) = \exp\left(-\frac{N\mathrm{Tr}U^T\Lambda UU^T\Lambda U}{2}\right) = \exp\left(-\frac{N\mathrm{Tr}U^T\Lambda^2 U}{2}\right) = \exp\left(-\frac{N\mathrm{Tr}\Lambda^2}{2}\right)$$

which depends only on the eigenvalues of X and not on the eigenvectors. We must deal with the Jacobian of the change of variables from X to (Λ, U). A standard calculation found in any introductory text on random matrix theory shows that

$$\prod_{1\leqslant i\leqslant j\leqslant N} dX_{ij} = \Delta(\{\lambda_i\}_{i=1}^N) d\mu_{Haar}(U) \prod_{i=1}^N d\lambda_i$$

where the *Vandermonde* determinant is defined by

$$\Delta(\{\lambda_1,\ldots,\lambda_N\}) = \begin{vmatrix} 1 & 1 & \cdots & 1 \\ \lambda_1 & \lambda_2 & \cdots & \lambda_N \\ \lambda_1^2 & \lambda_2^2 & \cdots & \lambda_N^2 \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_1^{N-1} & \lambda_2^{N-1} & \cdots & \lambda_N^{N-1} \end{vmatrix} = \prod_{1\leqslant i<j\leqslant N} |\lambda_i - \lambda_j|. \tag{2.7}$$

Overall, we see that

$$d\mu(X) = d\mu_{Haar}(U)\Delta(\{\lambda_i\}_{i=1}^N)\prod_{i=1}^N d\lambda_i \frac{e^{-\frac{N\lambda_i^2}{2}}}{\sqrt{2\pi N}} \tag{2.8}$$

where there was of course no need to compute the normalisation constant, as it can simply be written down from the simple Gaussian product measure form in the $\lambda_1,\ldots,\lambda_N$. The form (2.8) already reveals much about the statistics of the eigenvalues and eigenvectors of X. It is immediately obvious that the eigenvalues are independent of the eigenvectors. The eigenvectors are Haar-distributed, so they are simply a flat random orthonormal basis of $\mathbb{R}^N$. The eigenvalues have richer structure, but we can immediately make some heuristic comments on their statistics. Absent the Vandermonde term, the eigenvalues would be i.i.d. centred Gaussians with variance $1/N$, so the larger N is, the less dispersed the eigenvalues will be around 0. The Vandermonde term introduces dependence between all of the eigenvalue, and specifically it introduces *repulsion*, as $\Delta(\lambda_i)$ is a decreasing function of the distance between the eigenvalues. We can predict therefore that the distribution of the $\lambda_1,\ldots,\lambda_N$ is some equilibrium balancing the repulsion between all eigenvalues and the independent confining Gaussian potentials on each eigenvalue.

We shall now turn our attention to the mean and limiting spectral measures of the GOE. There are several quite different routes by which one can obtain these results. For a general and entirely rigorous approach, which in fact applies to *any* generalised Wigner matrix, we direct the reader to [AGZ10]. We present an approach using supersymmetric methods later in this chapter. For now, we shall present a derivation using the *Coulomb gas method* [CFV16] which, in addition to the supersymmetric method, is of great relevance to the central calculations of this thesis.

Let us introduce the following reformulation of the eigenvalue joint density function of the GOE:

$$p(\lambda_1,\ldots,\lambda_N) = \Delta(\{\lambda_i\}_{i=1}^N)\prod_{i=1}^N \frac{e^{-\frac{N\lambda_i^2}{2}}}{\sqrt{2\pi N}} = \frac{1}{(2\pi N)^{N/2}}\exp\left(-\frac{N}{2}\sum_{i=1}^N\left\{\lambda_i^2 - \frac{1}{N}\sum_{j\neq i}\log|\lambda_i-\lambda_j|\right\}\right).$$

Further, using the definition of the empirical spectral density, we can write

$$p(\lambda_1,\ldots,\lambda_N) = \frac{1}{(2\pi N)^{N/2}}\exp\left(-\frac{N^2}{2}\int d\hat{\mu}_N(\lambda)\left\{\lambda^2 - \int_{\lambda'\neq\lambda} d\hat{\mu}_N(\lambda')\log|\lambda'-\lambda|\right\}\right) \tag{2.9}$$

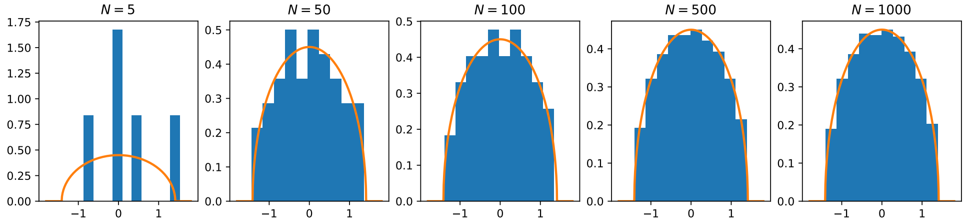


Figure 2.1: The Wigner semi-circle density (orange line plot) compared to histograms of eigenvalues computed from single samples of $N \times N$ GOE matrices (blue) for various values of N

from which the repulsion vs confinement statistics of the eigenvalues is made most clear. The logarithmic (Coulomb) potential has a singularity and 0 which penalises eigenvalue configurations with insufficient space between eigenvalues, whereas the quadratic potential penalises configurations with any eigenvalues too far from the origin. Continuing in the parlance of statistical physics, define the Lagrangian

$$\mathcal{E}(\lambda;\mu) = \lambda^2 - \int_{\lambda'\neq\lambda} d\mu(\lambda')\log|\lambda-\lambda'|$$

and thence the action

$$\mathcal{I}[\mu] = \int d\mu(\lambda)\mathcal{E}(\lambda;\mu)$$

with which we have

$$p(\lambda_1,\ldots,\lambda_N) = \frac{1}{Z_N}\exp\left(-\frac{N^2}{2}\mathcal{I}[\hat{\mu}_N]\right). \tag{2.10}$$

Let us consider $N\to\infty$ and for now assume that $\hat{\mu}_N$ converges, in some sense, to μ_∞. It is clear from the action principle (or Laplace's method for asymptotic evaluation of integrals) that μ_∞ must be a global minimiser of $\mathcal{I}$. As such, μ_∞ must be a deterministic probability measure on $\mathbb{R}$, so we shall assume weak almost sure convergence of $\hat{\mu}_N$ to μ_∞. It remains just to solve the variational problem

$$\text{argmin}_{\mu\in\mathcal{P}(\mathbb{R})}\mathcal{I}[\mu]$$

where $\mathcal{P}(\mathbb{R})$ is the set of all probability measures on $\mathbb{R}$. The solution for ρ_∞ can be found e.g. in [AG97] and is

$$\rho_\infty(\lambda) = \rho_{SC}(\lambda) = \frac{1}{\pi}\sqrt{2-\lambda^2} \tag{2.11}$$

which is the celebrated Wigner *semi-circle* density.

The semi-circle density is striking by its simplicity and elegance which, in fact, hint at a much deeper role in random matrix theory than just the limiting spectral density of a particular random matrix ensemble. Firstly, the semi-circle is not unique to the GOE but is shared by all generalised Wigner matrices (though, of course, the derivation above is possible only for the three canonical

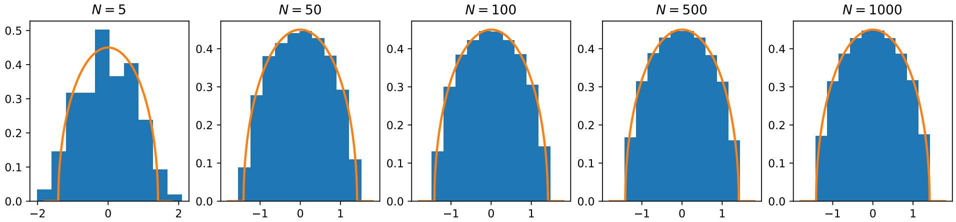


Figure 2.2: The Wigner semi-circle density (orange line plot) compared to histograms of eigenvalues computed from 100 i.i.d. samples of $N \times N$ GOE matrices (blue) for various N values.

Gaussian Wigner ensembles: GOE, GUE, and GSE). More importantly, the semi-circle takes the place of the Gaussian in an analogue of the the central limit theorem for random matrices, about which we provide more discussion in section 2.6.

So far, we have spoken only of the LSD, but what of the mean spectral density? We shall defer an explicit calculation for the GOE to section 2.3, but we shall see that the density ρ_N of the mean spectral measure μ_N can be written as

$$\rho_N(\lambda) = \rho_{SC}(\lambda) + o(1)$$

where the $o(1)$ term is uniformly small in N for $\lambda \in \mathbb{R}$. Once again, this property of the very special GOE ensemble points to a much deeper phenomenon in random matrix theory: *self-averaging*. To leading order in large N, the spectral density of a single random GOE matrix of size $N \times N$ is deterministic and identical to the mean spectral density, which is an average of the whole GOE ensemble of random matrices.

Example 2.2 (An invariant ensemble). Recall the Lagrangian defined above

$$\mathcal{E}(\lambda;\mu) = \lambda^2 - \int_{\lambda\neq\lambda'} d\mu(\lambda')\log|\lambda-\lambda'|$$

with which we were able to express the GOE joint eigenvalue density as

$$p(\lambda_1,\ldots,\lambda_N) = \frac{1}{Z_N}\exp\left(-\frac{N^2}{2}\int d\hat{\mu}_N(\lambda)\mathcal{E}(\lambda;\hat{\mu}_N)\right). \tag{2.12}$$

The origin of the two terms in $\mathcal{E}$ is quite plain: λ^2 simply comes from the Gaussian distribution of the GOE entries, while the logarithmic term comes from the Vandermonde determinant. The Vandermonde term is therefore universal to any real symmetric matrix ensemble, as it follows simply from the matrix change of variables. Similarly, if we were discussion complex Hermitian matrices there would be a universal Vandermonde term simply twice that for real symmetric matrices. So for any real symmetric matrix ensemble, we could in principle repeat the above procedure and arrive at a Lagrangian with exactly the same logarithmic Vandermonde term, along with some ensemble specific term. Of course, in general this term would not factorise nicely over the eigenvalues, so the above reduction to simply $\int \hat{\mu}_N(\lambda)\mathcal{E}(\lambda,\hat{\mu}_N)$ would not be possible. Let us then just consider ensembles for which this factorisation *does* occur, so that one would obtain the same Lagrangian form of the eigenvalues density but with Lagrangian

$$\mathcal{E}_V(\lambda;\mu) = V(\lambda) - \int_{\lambda\neq\lambda'} d\mu(\lambda')\log|\lambda-\lambda'|$$

where $V\colon \mathbb{R}\to\mathbb{R}$ is some function with sufficient smoothness and sufficiently fast growth at infinity to define a normalisable probability density. Such a random matrix ensemble is known as an *invariant ensemble* because it retains the same orthogonal invariance possessed by the GOE. The matrix density for an invariant ensemble can be simply written as

$$p(X)dX \propto e^{-\frac{N}{2}\mathrm{Tr}V(X)}dX.$$

For a real symmetric matrix argument $X = O^T\Lambda O$ one has the power series definition

$$\mathrm{Tr}V(X) = \sum_{r\geqslant 0} a_r \mathrm{Tr}X^r = \sum_{r\geqslant 0} a_r \mathrm{Tr}O^T\Lambda O\ldots O^T\Lambda O = \sum_{r\geqslant 0} a_r \mathrm{Tr}\Lambda^r = V(\Lambda) = \mathrm{Tr}V(\Lambda),$$

so

$$p(X)dX \propto e^{-\frac{N}{2}\sum_{j=1}^N V(\lambda_j)}\Delta(\{\lambda_i\}_{i=1}^N)d\lambda_1\ldots d\lambda_N d\mu_{Haar}(O)$$

which confirms the Lagrangian expression given above.

2.1.4 The Wigner surmise

As we have seen in the preceding section, though the semi-circle plays a deep role in random matrix theory, it is by no means a universal spectral density for random matrix ensembles. Simply change the potential to deviate from the simple quadratic case was sufficient to produce entirely different spectral densities with invariant ensembles. So, at the level of the mean (or limiting) spectral measure, the semi-circle is more general that the GOE and the Gaussian Wigner ensembles, but is specific to Wigner matrices. One of the most astonishing results in random matrix theory is that there are properties of GOE matrices that are, in fact, *universal* in the sense that they are properties shared by a very wide class of matrices beyond the GOE and Wigner ensembles. A full discussion of this kind of random matrix universality is deferred to the later Section 2.7.

Random matrix theory was first developed in physics to explain the statistical properties of nuclear energy levels, and later used to describe the spectral statistics in atomic spectra, condensed matter systems, quantum chaotic systems etc; see, for example [WM08; Bee97; Ber+87; Boh91]. *None of these physical systems exhibits a semicircular empirical spectral density*. However they all generically show agreement with random matrix theory at the level of the mean eigenvalue spacing when local spectral statistics are compared. The key insight here is that while almost any realistic physical system, model or even the machine learning systems which are the central objects of study for this thesis, will certainly not posses semi-circular densities at the macroscopic scale of the mean spectral density, but nevertheless random matrix theory can still describe spectral fluctuations on the microscopic scale of the mean eigenvalue spacing.

It is worth noting in passing that possibilities other than random-matrix statistics exist and occur. For example, in systems that are classically integrable, one finds instead Poisson statistics [BT77; Ber+87]; similarly, Poisson statistics also occur in disordered systems in the regime of strong Anderson localisation [Efe99]; and for systems close to integrable one finds a superposition of random-matrix and Poisson statistics [BR84]. So showing that random matrix theory applies is far from being a trivial observation. Indeed it remains one of the outstanding challenges of mathematical physics to prove that the spectral statistics of any individual Hamiltonian system are described by it in the semi-classical limit.

Random matrix calculations in physics re-scale the eigenvalues to have a mean level spacing of 1 and then typically look at the *nearest neighbour spacings distribution* (NNSD), i.e. the distribution of the distances between adjacent pairs of eigenvalues. One theoretical motivation for considering the NNSD is that it is independent of the Gaussianity assumption and reflects the symmetry of the underlying system. It is the NNSD that is universal (for systems of the same symmetry class) and not the average spectral density, which is best viewed as a parameter of the system. The aforementioned transformation to give mean spacing 1 is done precisely to remove the effect of the average spectral density on the pair correlations leaving behind only the universal correlations.

In contrast to the LSD, other k-point correlation functions are also normalised such that the mean spacing between adjacent eigenvalues is unity. At this *microscopic* scale, the LSD is locally constant and equal to 1 meaning that its effect on the eigenvalues' distribution has been removed and only microscopic correlations remain. In the case of Wigner random matrices, for which the LSD varies slowly across the support of the eigenvalue distribution, this corresponds to scaling by $\sqrt{P}$. On this scale the limiting eigenvalue correlations when $P \to \infty$ are *universal*; that is, they are the same for wide classes of random matrices, depending only on symmetry [GMW98]. For example, this universality is exhibited by the NNSD. Consider a 2×2 GOE matrix, in which case the j.p.d.f has a simple form:

$$p(\lambda_1, \lambda_2) \propto |\lambda_1 - \lambda_2| e^{-\frac{1}{2}(\lambda_1^2 + \lambda_2^2)}. \tag{2.13}$$

Making the change of variables $v_1 = \lambda_1 - \lambda_2, v_2 = \lambda_1 + \lambda_2$, integrating out v_2 and setting $s = |v_1|$ results in a density $\rho_{Wigner}(s) = \frac{\pi s}{2} e^{-\frac{\pi}{4}s^2}$, known as the *Wigner surmise* (see Figure 2.3). For larger matrices, the j.p.d.f must include an indicator function $\mathbb{1}\{\lambda_1 \leqslant \lambda_2 \leqslant \ldots \lambda_P\}$ before marginalisation so that one is studying pairs of *adjacent* eigenvalues. While the Wigner surmise can only be proved exactly, as above, for the 2 × 2 GOE, it holds to high accuracy for the NNSD of GOE matrices of any size provided that the eigenvalues have been scaled to give mean spacing 1.[1] The Wigner surmise density vanishes at 0, capturing 'repulsion' between eigenvalues that is characteristic of RMT statistics, in contrast to the distribution of entirely independent eigenvalues given by the *Poisson law*

[1] An exact formula for the NNSD of GOE matrices of any size, and one that holds in the large P limit, can be found in [Meh04].

$\rho_{Poisson}(s) = e^{-s}$. The Wigner surmise is universal in that the same density formula applies to all real-symmetric random matrices, not just the GOE or Wigner random matrices.

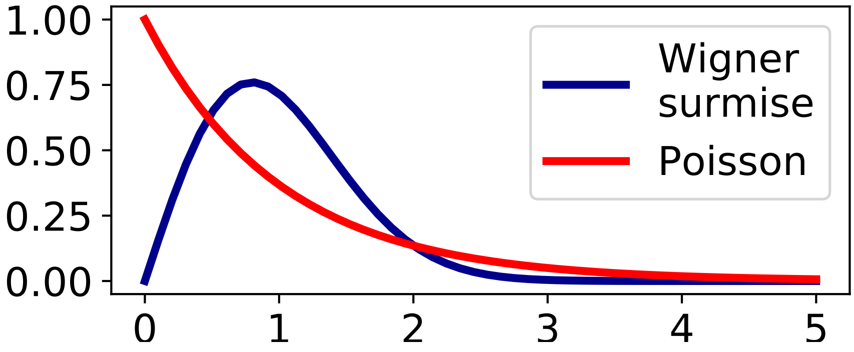


Figure 2.3: The density of the Wigner surmise.

2.1.5 Eigenvectors

What of the eigen*vectors* of random matrices? We have already seen that GOE matrices, and invariant ensembles in general, have Haar-distributed eigenvectors entirely independent of the eigenvalues. Just as the semi-circle is unique to Wigner matrices but the GOE Wigner surmise is seen in all matrices with orthogonal group symmetry, so Haar-distributed eigenvectors independent of the eigenvalues are seen only in invariant ensembles (not even in non-Gaussian Wigner matrices) but certain properties of Haar matrices are universal across a similarly wide class of random matrices. Once again, the discussion of these deep universality results will be given in Section 2.7, but we shall set the scene by first describing the *delocalisation* property of Haar-distributed eigenvectors.

Let U be an $N \times N$ Haar-distributed orthogonal matrix and let $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N$ be its rows. Recall from the discussion above wherein the Haar distribution was introduced the following construction:

$$\text{Let } \boldsymbol{g}_1, \ldots, \boldsymbol{g}_N \text{ be i.i.d. vectors from } \mathcal{N}(0, I_N); \tag{2.14}$$

$$\text{let } \boldsymbol{v}_1, \ldots, \boldsymbol{v}_N \text{ be the results of a Gram-Schmidt algorithm;} \tag{2.15}$$

$$\text{then, in distribution, } \{\boldsymbol{u}_i\}_{i=1}^N = \{\boldsymbol{v}_i / \|\boldsymbol{v}_i\|_2\}_{i=1}^N. \tag{2.16}$$

Fix some $r < N$ and introduce the event

$$B_N(v) := \left\{ |N^{-1}\langle \boldsymbol{g}_i, \boldsymbol{g}_j\rangle - \delta_{ij}| \leqslant N^{-v}, \quad 1 \leqslant i, j \leqslant r \right\}. \tag{2.17}$$

Then it is an exercise in Gaussian calculations and asymptotics, as given in [GM+05], to conclude that under the i.i.d Gaussian law of the $(\boldsymbol{g}_j)_{j=1}^N$ the complementary event has low probability for large N:

$$\mathbb{P}(B_N(v)^c) = \mathcal{O}(C(v)e^{-\alpha N^{1-2v}}), \tag{2.18}$$

where $\alpha, C(v) > 0$ and we take $0 < v < \frac{1}{2}$ to make this statement meaningful. What's more, one can directly obtain that, given B_N,

$$\|\boldsymbol{g}_i\|_2^2 = N^{1-v} \tag{2.19}$$

for any $v > 0$. So, restricting to only a fixed subset of the eigenvectors as $N \to \infty$, the simply i.i.d. Gaussian vectors $\boldsymbol{g}_i$ from which they are constructed are, with high probability, close to being orthogonal even before applying Gram-Schmidt algorithm and they all have the same L_2 norm to leading order in N. This line of reasoning leads to the fact that, with high probability,

$$||N^{-1/2}\tilde{\boldsymbol{g}}_j - \boldsymbol{u}_j|| \leqslant N^{-\frac{v}{2}}, \tag{2.20}$$

so, indeed, in the above precise probabilistic sense, any subset of Haar-distributed eigenvectors are extremely close to an corresponding set of i.i.d. standard Gaussian vectors, re-scaled by $N^{-1/2}$.

2.2 Kac-Rice formulae

The majority of chapters 3 and 4 is concerned with computing the expected complexity of certain loss surfaces in the limit as the number of parameters $N \to \infty$. Let us recall a basic definition of complexity as introduced above. Let $\mathcal{M}$ be a compact, oriented, N-dimensional C^1 manifold with a C^1 Riemannian metric g. Let $\psi : \mathcal{M} \to \mathbb{R}$ be a random field on $\mathcal{M}$. For an open set $A \subset \mathbb{R}$, let

$$C(A) \equiv \left|\{x \in \mathcal{M} \mid \nabla\psi(x) = 0,\ \psi(x) \in A\}\right|. \tag{2.21}$$

$C(A)$ simply counts the number of local optima of ψ for which ψ lies in the set A. Note that the condition of a compact manifold $\mathcal{M}$ is important here; without other constraints (for which see e.g. [ABM21b]) there is no guarantee of a finite value for $C(A)$ given a non-compact $\mathcal{M}$. Computing anything about $C(A)$ appears extremely challenging, but one can make some informal progress rather directly with an integral expression

$$C(A) = \int_{\nabla\psi(\mathcal{M})} d\boldsymbol{u}\ \delta(\boldsymbol{u})\mathbf{1}\{\psi(\nabla\psi^{-1}(\boldsymbol{u})) \in A\} \tag{2.22}$$

which one can write down simply from the sampling property of the δ-function. The the composition property of the δ-function gives

$$C(A) = \int_{\mathcal{M}} d\boldsymbol{x}\ |\det\nabla^2\psi(\boldsymbol{x})|\delta(\nabla\psi(\boldsymbol{x}))\mathbf{1}\{\psi(\boldsymbol{x}) \in A\}. \tag{2.23}$$

From this simple argument, we see that the *Hessian* of ψ, and in particular the absolute value of its determinant, will be central to calculation of $C(A)$. Recall that ψ is a random field, so its Hessian $\nabla^2\psi$ is a random matrix of size $N \times N$, so one can see already that the complexity of random functions is connected with random matrix theory. What these simple arguments lack is any reference to the probability density of ψ. Since ψ is random, so also is $C(A)$, so we must be more precise about what 'calculating $C(A)$' means. One could attempt to compute the entire density of $C(A)$, but this is clearly the most difficult objective. Let us restrict our consideration to simple statistics of $C(A)$ and, in particular, its expected value. Proceeding informally, we have

$$\mathbb{E}C(A) = \mathbb{E}\left\{\int_{\mathcal{M}} d\boldsymbol{x}\ |\det\nabla^2\psi(\boldsymbol{x})|\delta(\nabla\psi(\boldsymbol{x}))\mathbf{1}\{\psi(\boldsymbol{x})\in A\}\middle|\nabla\psi(\boldsymbol{x})=0\right\}p_{\boldsymbol{x}}(0) \tag{2.24}$$

where $p_{\boldsymbol{x}}$ is the density of $\nabla\psi$ at the point $\boldsymbol{x}\in\mathcal{M}$. Within the conditional expectation, the delta function can be dropped, giving simply

$$\mathbb{E}C(A) = \mathbb{E}\left[\int_{\mathcal{M}} d\boldsymbol{x}\ |\det\nabla^2\psi(\boldsymbol{x})|\mathbf{1}\{\psi(\boldsymbol{x})\in A\}\middle|\nabla\psi(\boldsymbol{x})=0\right]p_{\boldsymbol{x}}(0) \tag{2.25}$$

and finally swapping the order of integration informally gives

$$\mathbb{E}C(A) = \int_{\mathcal{M}} d\boldsymbol{x}\ p_{\boldsymbol{x}}(0)\mathbb{E}\left[|\det\nabla^2\psi(\boldsymbol{x})|\mathbf{1}\{\psi(\boldsymbol{x})\in A\}\middle|\nabla\psi(\boldsymbol{x})=0\right]. \tag{2.26}$$

We see now that $\mathbb{E}C(A)$ will be tractable if we can compute the joint distribution of $\psi,\nabla^2\psi$ conditional on $\nabla\psi$, and subsequently evaluate the random determinant's expected value. The expression (2.26) is an example of a *Kac-Rice formula* [Kac43; Ric44]. These kind of informal arguments have been extensively used in the mathematical physics literature to compute quantities such as expected landscape complexities and cardinalities of other level-sets of random functions [Ber02; BLO98; FS02; Fyo04; Fyo05]. These arguments can be made fully rigorous and cast in a more general setting as is shown in the important book by Adler and Taylor [AT09]. We repeat here the foundational Kac-Rice result from that work which is central to our complexity calculations in the coming chapters.

Theorem 2.1 ([AT09] Theorem 12.1.1)**.** *Let $\mathcal{M}$ be a compact, oriented, N-dimensional C^1 manifold with a C^1 Riemannian metric g. Let $\phi:\mathcal{M}\to\mathbb{R}^N$ and $\psi:\mathcal{M}\to\mathbb{R}^K$ be random fields on $\mathcal{M}$. For an open set $A\subset\mathbb{R}^K$ for which ∂A has dimension $K-1$ and a point $\boldsymbol{u}\in\mathbb{R}^N$ let*

$$N_{\boldsymbol{u}} \equiv \left|\{x\in\mathcal{M}\mid\phi(x)=\boldsymbol{u},\ \psi(x)\in A\}\right|. \tag{2.27}$$

Assume that the following conditions are satisfied for some orthonormal frame field E:

(a) All components of ϕ, $\nabla_E\phi$, and ψ are a.s. continuous and have finite variances (over $\mathcal{M}$).

(b) For all $x\in\mathcal{M}$, the marginal densities p_x of $\phi(x)$ (implicitly assumed to exist) are continuous at $\boldsymbol{u}$.

(c) The conditional densities $p_x(\cdot|\nabla_E\phi(x),\psi(x))$ of $\phi(x)$ given $\psi(x)$ and $\nabla_E\phi(x)$ (implicitly assumed to exist) are bounded above and continuous at $\boldsymbol{u}$, uniformly in $\mathcal{M}$.

(d) The conditional densities $p_x(\cdot|\phi(x)=\boldsymbol{z})$ of $\det(\nabla_{E_j}\phi^i(x))$ given are continuous in a neighbourhood of 0 for $\boldsymbol{z}$ in a neighbourhood of $\boldsymbol{u}$ uniformly in $\mathcal{M}$.

(e) The conditional densities $p_x(\cdot|\phi(x)=\boldsymbol{z})$ are continuous for $\boldsymbol{z}$ in a neighbourhood of $\boldsymbol{u}$ uniformly in $\mathcal{M}$.

(f) The following moment condition holds

$$\sup_{x\in\mathcal{M}}\max_{1\leqslant i,j\leqslant N}\mathbb{E}\left\{\left|\nabla_{E_j}\phi^i(x)\right|^N\right\}<\infty \tag{2.28}$$

(g) The moduli of continuity with respect to the (canonical) metric induced by g *of each component of* ψ, *each component of* ϕ *and each* $\nabla_{E_j}\phi^i$ *all satisfy, for any* $\varepsilon>0$

$$\mathbb{P}(\omega(\eta)>\varepsilon)=o(\eta^N),\quad \textit{as } \eta\downarrow 0 \tag{2.29}$$

where the modulus of continuity *of a real-valued function* G *on a metric space* (T,τ) *is defined as (c.f. [AT09] around (1.3.6))*

$$\omega(\eta):=\sup_{s,t:\tau(s,t)\leqslant\eta}|G(s)-G(t)| \tag{2.30}$$

Then

$$\mathbb{E}N_{\boldsymbol{u}}=\int_{\mathcal{M}}\mathbb{E}\left\{|\det\nabla_E\phi(x)|\mathbb{1}\{\psi(x)\in A\}\mid\phi(x)=\boldsymbol{u}\right\}p_x(\boldsymbol{u})Vol_g(x) \tag{2.31}$$

where p_x *is the density of* ϕ *and* Vol_g *is the volume element induced by* g *on* $\mathcal{M}$.

Note the greater generality of this theorem compared to the heuristic derivation above. The required result for complexity can be obtained as a special case by taking $\phi=\nabla\psi$ and $\boldsymbol{u}=0$.

2.3 Supermathematics

Grassmann variables are entirely algebraic objects defined by an *anti-commutation rule*. Let $\{\chi_i\}_i$ be a set of Grassmann variables, then by definition

$$\chi_i\chi_j=-\chi_j\chi_i,\quad \forall i,j. \tag{2.32}$$

The complex conjugates χ_i^* are separate objects, with the complex conjugation unary operator * defined so that $\left(\chi_i^*\right)^*=-\chi_i^*$, and Hermitian conjugation is then defined as usual by $\chi^\dagger=(\chi^T)^*$. The set of variables $\{\chi_i,\chi_i^*\}_{i=1}^N$ generate a *graded algebra* over $\mathbb{C}$. Mixed vectors of commuting and anti-commuting variables are called *supervectors*, and they belong to a vector space called *superspace*. The integration symbol $\int d\chi_i d\chi^*$ is defined as a formal algebraic linear operator by the properties

$$\int d\chi_i=0,\qquad \int d\chi_i\,\chi_j=\delta_{ij}, \tag{2.33}$$

and these are called *Berezin* integrals. Functions of the the Grassmann variables are defined by their formal power series, e.g.

$$e^{\chi_i}=1+\chi_i+\frac{1}{2}\chi_i^2+\ldots=1+\chi_i \tag{2.34}$$

where the termination of the series follows from $\chi_i^2=0\ \ \forall i$, which is an immediate consequence of (2.32). From this it is apparent that (2.33), along with (2.32), is sufficient to define Berezin integration over arbitrary functions of arbitrary combinations of Grassmann variables. Finally we establish our notation for supersymmetric (or *graded*) traces of supermatrices. We will encounter supermatrices of the form

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

where A, D are square block matrices of commuting variables and B, C are rectangular block matrices of Grassmann variables. In this case, the graded trace is given by $\mathrm{trg} M = \mathrm{Tr} A - \mathrm{Tr} D$ and such matrices are referred to as $(B+F)\times(B+F)$, where A is shape $B \times B$ and D is shape $F \times F$. We refer the reader to [Efe96] for a full introduction to supersymmetric variables and methods.

Grassmann variables play an important role in quantum field theory and related fields [Pes18; GSW12], being the algebraic representation of fermions, with bosons being represented by commuting variables. As such, even in applications unrelated to quantum physics, the particle nomenclature may be used; for example the diagonal blocks of the matrix M above may be referred to as *bosonic blocks* and the off-diagonals referred to as *fermionic blocks*. There are important connections between random matrix theory and quantum field theory in which the role of supersymmetry in both is made quite plain [Ver04], but for the purposes of this thesis, Grassmann variables and supersymmetric methods are simply mathematical tools that we use to compute certain random matrix quantities. Supersymmetric methods provide a powerful way of computing random matrix determinants, which in turn can have many applications to compute various quantities of interest [Ver04; Noc16]. We will focus on two such applications that are used in chapters 3, 4 and 8.

Consider a random $N \times N$ matrix X and suppose if has a limiting spectral measure μ with density ρ and Stieljtes transform g. Given a density on X, an important question is to determine the spectral density ρ, by the Stieljtes inversion formula, it is sufficient to compute g:

$$\rho(x) = \frac{1}{\pi} \lim_{\epsilon \to 0} \Im g(x + i\epsilon). \tag{2.35}$$

Let $G(z)$ be the Stieljtes transform of the empirical spectral measure of X, i.e.

$$G(z) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{z - \lambda_i} \tag{2.36}$$

where λ_i are the eigenvalues of X. G is a random function and for many matrix ensembles will have the convergence property $G \to g$ weakly almost surely as $N \to \infty$. Similarly, G will typically have the self-averaging property $\mathbb{E}G \to g$ in the sense of deterministic functions. It follows that computing ρ can be achieved by computing the leading order term in an asymptotic expansion for $\mathbb{E}G$ in the limit $N \to \infty$. The key to this approach is that, if the average $\mathbb{E}G$ can be computed over the matrix ensemble X, then the asymptotic analysis for large N can be performed on deterministic objects to obtain ρ, rather than having to deal with asymptotics of random functions. To see the connection with random determinants and thence supersymmetry, we can rewrite

$$G(z)=\frac{1}{N}\frac{1}{\det(zI-X)}\sum_{i=1}^{N}\prod_{k\neq i}^{N}(z-\lambda_k)=\frac{1}{N}\frac{\partial}{\partial \mathrm{j}}\bigg|_{\mathrm{j}=0}\frac{\det(zI-X+\mathrm{j}I)}{\det(zI-X)} \tag{2.37}$$

where the first equality is a simple algebraic identity and the second follows from the product rule of differentiation. It follows that computing $\mathbb{E}G$ is equivalent to computing the random matrix average

$$\mathbb{E}\frac{\det(zI-X+\mathrm{j}I)}{\det(zI-X)} \tag{2.38}$$

followed by some differentiation. Note that this 'trick' involving the introduction of the dummy variable j is widely used as well in the perturbation theory approach to quantum field theory [Pes18].

We have seen in the previous section that random matrix determinants are to be expected for example in complexity calculations, i.e. one needs to compute $\mathbb{E}|\det X|$ for a random $N\times N$ matrix X, and doing so in the large N limit may be sufficient. The presence of the absolute value here is a particular nuisance, but just like the Stieljtes transform above, ratios of determinants can be used to provide an alternate formulation:

$$|\det X|=\frac{(\det X)^2}{(\sqrt{\det X})^2}=\frac{\det X\det X}{\sqrt{\det X}\sqrt{\det X}} \tag{2.39}$$

where the principal branch of the square root is taken.

The general challenge here is to compute expectations of ratios of integer and half integer powers of random matrix determinants. This topic has been much explored in the literature, see e.g. [Fyo05; Ver04; Noc16; Noc17]. The role of supersymmetric methods in this approach stems from a familiar change of variables result. For a non-singular Hermitian $N\times N$ matrix X

$$\frac{1}{(-i)^N\pi^N}\int_{\mathbb{R}^N}d\boldsymbol{z}e^{-i\boldsymbol{z}^TX\boldsymbol{z}}=\frac{1}{\sqrt{\det X}} \tag{2.40}$$

where the determinant is the simply the Jacobian of the transformation from variables $\boldsymbol{z}$ to $X^{1/2}\boldsymbol{z}$. Similarly, using complex integration variables one can obtain

$$\frac{1}{(2\pi)^N}\int_{\mathbb{C}^N}d\boldsymbol{z}d\boldsymbol{z}^*e^{-i\boldsymbol{z}^\dagger X\boldsymbol{z}}=\frac{1}{\det X}. \tag{2.41}$$

The final ingredient is to use Grassmann integration variables to obtain an analogous expression for $\det X$, as opposed to powers of its reciprocal. Indeed, by introducing Grassmann variables χ_i,χ_i* and a Berezin integral, we obtain

$$\frac{1}{(-i)^N}\int\prod_{i=1}^{N}d\chi_i d\chi_i^* e^{-i\chi^\dagger X\chi}=\det X. \tag{2.42}$$

Rather than a change of variable result from multivariate calculus, this result is proved by expanding the exponential. Recall that $\int d\chi_i\ 1=0$, so the only therm in the exponential expansion that can be non-zero after Berezin integration are those that contain each χ_i and χ_i^* at least once. But also, since $\chi_i^2 = 0 = (\chi_i^*)^2$, the only non-zero terms are those that contain each χ_i, χ_i^* *exactly* once, hence

$$\begin{aligned}\int \prod_{i=1}^N d\chi_i d\chi_i^* e^{-i\chi^\dagger X \chi} &= \frac{1}{N!}\int \prod_{i=1}^N d\chi_i d\chi_i^* (-i\chi^\dagger X\chi)^N \\ &= (-i)^N \frac{1}{N!}\int \prod_{i=1}^N d\chi_i d\chi_i^* \sum_{j_1,k_1,\ldots,j_N,k_N=1}^N \chi_{j_1}^* X_{j_1 k_1}\chi_{k_1}\cdots\chi_{j_N}^* X_{j_N k_N}\chi_{k_N} \qquad (2.43)\end{aligned}$$

The only non-zero terms from the sum must have $j_1, \ldots, j_N$ equal to a permutation of $1, \ldots, N$ and similarly $k_1, \ldots, k_N$, so we can write

$$\int \prod_{i=1}^N d\chi_i d\chi_i^* e^{-i\chi^\dagger X\chi} = (-i)^N \frac{1}{N!}\int \prod_{i=1}^N d\chi_i d\chi_i^* \sum_{\sigma,\tau\in S_N} \chi_{\sigma(1)}^* X_{\sigma(1)\tau(1)}\chi_{\tau(1)}\cdots\chi_{\sigma(N)}^* X_{\sigma(N)\tau(N)}\chi_{\tau(N)}. \qquad (2.44)$$

Re-indexing the sum over the symmetric group by defining $\sigma = \sigma' \circ \tau$, we see that the sum over τ can be rendered trivial, giving just a constant factor of $N!$, so

$$\int \prod_{i=1}^N d\chi_i d\chi_i^* e^{-i\chi^\dagger X\chi} = (-i)^N \int \prod_{i=1}^N d\chi_i d\chi_i^* \sum_{\sigma'\in S_N} \chi_{\sigma'(1)}^* X_{\sigma'(1)1}\chi_1\cdots\chi_{\sigma'(N)}^* X_{\sigma'(N)N}\chi_N. \qquad (2.45)$$

Finally, the Grassmann terms must be commuted to render them in the correct order to agree with the differentials, i.e.

$$\int \prod_{i=1}^N d\chi_i d\chi_i^* e^{-i\chi^\dagger X\chi} = (-i)^N \int \prod_{i=1}^N d\chi_i d\chi_i^* \prod_{j=N}^1 \chi_j^*\chi_j \sum_{\sigma\in S_N} \operatorname{sgn}(\sigma)\prod_{k=1}^N X_{\sigma(k)k} = (-i)^N \det X. \qquad (2.46)$$

To conclude, ratios of certain powers of random matrix determinants can be written as Gaussian integrals over supersymmetric (i.e. mixed commuting and Grassmann) vectors. While this may seem at first like an increase in complexity, the supersymmetric representations have certain advantages, such as linearity since $e^{-i\phi^\dagger(X+Y)\phi} = e^{-i\phi^\dagger X\phi}e^{-i\phi^\dagger Y\phi}$. For example, if X and Y are independent, then a supersymmetric representation allows $\mathbb{E}|\det(X+Y)|$ to be computed as two separate and independent expectations of X and Y. It is this linearisation effect of supersymmetric representations that is at the heart of its application to many calculations, including those in chapters 3 and 4. In all applications of the supersymmetric method in random matrix theory, the random matrix calculation is reduced to 'simply' $\mathbb{E}e^{-i\mathrm{Tr}XK}$ where the matrix $K = \phi\phi^\dagger + \chi\chi^\dagger$ for some commuting vector ϕ and Grassmann χ of dimension N. The distribution of X is then encoded in this Fourier transform like object, and the remainder of the calculation is then an exercise in evaluating supersymmetric integrals. In the case of the GOE, this average is particular easy to compute:

$$\begin{aligned}\mathbb{E}e^{-i\mathrm{Tr}XK} &= \frac{N^N}{(2\pi)^{N/2}}\int dX \exp\left\{-\frac{N}{2}\mathrm{Tr}X^2 - i\mathrm{Tr}XK\right\} \\ &= \frac{N^N}{(2\pi)^{N/2}}\int dX \exp\left\{-\frac{N}{2}\mathrm{Tr}\left(X + i\frac{1}{N}K\right)^2 - \frac{1}{2N}\mathrm{Tr}K^2\right\} \\ &== e^{-\frac{1}{2N}\mathrm{Tr}K^2}. \qquad (2.47)\end{aligned}$$

The final technique we need to introduce for supersymmetric methods is the *Hubbard-Stratonovich* transformation. Consider complex commuting integration variables $\phi \in \mathbb{C}^N$ and Grassmann variables χ. Then $\mathrm{Tr}(\phi\phi^\dagger + \chi\chi^\dagger)^2 = \mathrm{trg} Q^2$ where

$$Q = \begin{pmatrix} \phi^\dagger\phi & \phi^\dagger\chi \\ \chi^\dagger\phi & \chi^\dagger\chi \end{pmatrix}. \tag{2.48}$$

The Hubbard-Stratonovich transformation introduces a 2×2 matrix integration variable σ which is of the same supersymmetric $1+1$ type as Q then

$$e^{-\frac{1}{2N}\mathrm{trg} Q^2} = \int d\sigma\; e^{-\frac{N}{2}\mathrm{trg}\sigma^2 - i\psi^\dagger\sigma\psi} \tag{2.49}$$

where

$$\psi = \phi \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \chi \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{2.50}$$

The power of the Hubbard-Stratonovich transformation is that it linearises the dependence of the supersymmetric integrand on the supersymmetric N-dimensional vectors at the cost only of introducing an integral over an extra 2×2 (or in general $k \times k$) supersymetric matrix. In many calculations, this transformation makes the N-dimensional supersymmetric integral easy to compute, leaving only an integration of a fixed number of supersymmetric variables, which is precisely the conditions required for applying standard techniques from asymptotic analysis for $N \to \infty$.

2.4 Large deviations principles

Consider as an example a $N \times N$ GOE matrix X normalised so that the semi-circular radius is 2. By the very existence of such a compact limiting spectral density, eigenvalues greater than 2 or less than -2 are in some sense unlikely for large N. Large deviations principles (LDPs) answer the question of precisely *how unlikely* these eigenvalues are. Let $\lambda_1 \leqslant \ldots \leqslant \lambda_N$ be the eigenvalues of X. The large deviation event for λ_1 is $\{\lambda_1 < x\}$ for $x < -2$, and similarly for λ_N is $\{\lambda_N > x\}$ for $x > 2$. Fixing an integer $k \geqslant 1$ as $N \to \infty$ there are also the large deviations events $\{\lambda_k < x\}$ for $x < -2$ (and similarly for λ_{N-k}). Formally, a large deviations principle for λ_k with speed $\alpha(N)$ and rate function $I_k(x)$ requires

$$\limsup_{N\to\infty} \frac{1}{\alpha(N)} \log \mathbb{P}(\lambda_k < x) = -I_k(x) \quad \text{for } x \leqslant -2, \tag{2.51}$$

$$\limsup_{N\to\infty} \frac{1}{\alpha(N)} \log \mathbb{P}(\lambda_k \geqslant x) = -\infty \quad \text{for } x \in (-2, 2). \tag{2.52}$$

For $x \in (-2,2)$, if $\lambda_k \geqslant x$, then there is an non-empty interval $(-2, x)$ in which there are at most k eigenvalues, so this represents a configuration of eigenvalues for which the difference between $\hat{\mu}_N$ and μ is not negligible, which must be extremely unlikely since $\hat{\mu}_N$ converges to μ. The large

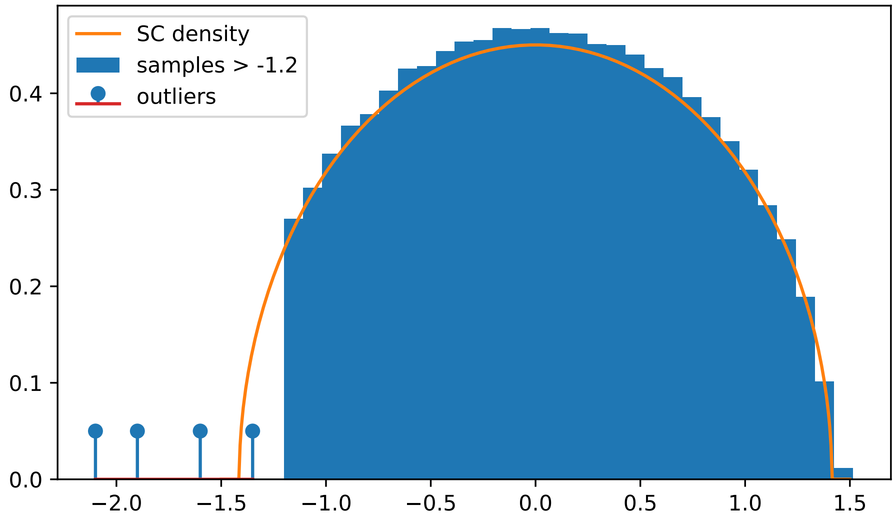


Figure 2.4: The samples show an example spectrum of a very large random matrix with only a small number of eigenvalues less than −1.2. One can see that the empirical spectral density $\hat{\mu}_N$ deviates in a non-negligible manner from μ_{SC}.

deviations principle encodes this as the infinity in (2.52), which says that $\{\lambda_k \geqslant x\}$ is very much more unlikely than even $\{\lambda_k < -2\}$. In the case of the GOE [ADG01; AAC13], $\alpha(N) = N$ and

$$I_k(x) = kI_1(x) = \begin{cases} \frac{k}{2}\int_x^{-2} dz\sqrt{z^2-4}, & \text{for } x \leqslant -2,\\ \infty & \text{otherwise.}\end{cases}$$

Note that the infinity in (2.52) should be expected from the expression (2.10) of the eigenvalue j.p.d.f. as $e^{-N^2\mathcal{I}[\hat{\mu}_N]}$, since $\lambda_k \geqslant x$ for $x \in (-2,2)$ implies $\mathcal{I}[\hat{\mu}_N] > 0$; figure 2.4 shows this argument pictorially.

Indeed, this intuition is a good representation of the full rigorous argument to prove this LDP. Note that that $\hat{\mu}_N$ is a random probability measure, so it appears as though $\hat{\mu}_N$ obeys a LDP with speed N^2 and rate function something like $\mathcal{I}$, where recall

$$\mathcal{I}[\mu] = \int d\mu(\lambda)\mathcal{E}(\lambda;\mu) = \int d\mu(\lambda)\lambda^2 - \int d\mu(\lambda)d\mu(\lambda')\log|\lambda-\lambda'|.$$

This is in fact the case and was established in [AG97], where the rate function was found for for $\beta = 1,2,4$ to be

$$J_\beta[\mu] = \frac{1}{2}\left(\int d\mu(\lambda)\lambda^2 - \beta\int d\mu(\lambda)d\mu(\lambda')\log|\lambda-\lambda'| + \frac{\beta}{2}\log\frac{\beta}{2} - \frac{3}{4}\beta\right), \qquad (2.53)$$

which has a unique minimiser, with with value 0, among all probability measures on $\mathbb{R}$ at the semi-circle measure with radius $\sqrt{2\beta}$. This fact alone is sufficient to establish (2.52) for the GOE. Indeed, consider the bounded Lipschitz distance on probability measure on $\mathbb{R}$

$$d_{Lip}(\mu,\nu) = \sup_{\|f\|_{Lip}\leqslant 1}\left|\int f(x)d(\mu-\nu)(x)\right|$$

where the supremum is taken over all Lipschitz function with Lipschitz constant at most 1. Using this metric, one can define a ball $B_\varepsilon(\mu_{SC})$ of radius ε centred on the minimiser μ_{SC} of J_1. For $x\in(-2,2)$, if $\lambda_k \geqslant x$ then

$$d_{Lip}(\mu_{SC},\hat{\mu}_N) \geqslant \left|\int_{-2}^{x} d\mu_{SC}(\lambda) - \frac{k}{N}\right| > \varepsilon$$

for all large enough N and for some fixed $\varepsilon > 0$ (independent of N). So if $\lambda_k \geqslant x$, then $\hat{\mu}_N$ lies outside the ball of radius ε centred on μ_{SC}, but since J_1 has a unique minimiser, it follows that $J_1[\hat{\mu}_N] > \delta$ for all large enough N and for some $\delta > 0$ (independent of N), so the LDP on $\hat{\mu}_N$ yields the infinite limit (2.52). The proof to establish the complementary limit (2.51) also makes use of the LDP on $\hat{\mu}_N$. The joint density $p(\lambda_1,\ldots,\lambda_N)$ can be split as

$$\begin{aligned}p(\lambda_1,\ldots,\lambda_N) &= p(\lambda_{k+1},\ldots,\lambda_N)f(\lambda_1,\ldots,\lambda_k;\lambda_{k+1},\ldots,\lambda_N)\\ &\propto \Delta(\{\lambda_i\}_{i=1}^k)p(\lambda_{k+1},\ldots,\lambda_N)\exp\left(-N\sum_{j=1}^{k}\left\{\frac{\lambda_j^2}{2} - \int d\hat{\mu}_{N-k}(\lambda)\log|\lambda-\lambda_j|\right\}\right).\end{aligned}$$

By placing all eigenvalues inside large ball of radius M, the left-over Vandermonde term $\Delta(\{\lambda_i\}_{i=1}^k)$ can be bounded by $2M^{k^2}$, say. Since N is very large and k fixed, the LDP for the empirical spectral density $\hat{\mu}_{N-K}$ of $\lambda_{k+1},\ldots,\lambda_N$ applies and $\hat{\mu}_{N-k}$ can be effectively replaced by μ_{SC}, incurring only an error term suppressed by an LDP bound of size e^{-cN^2} for some constant $c>0$. It then remain to bound the contribution from eigenvalues outside the ball of radius M and to evaluate the supremum over $\lambda < x$ of $\int d\mu_{SC}(\lambda')\log|\lambda'-\lambda| - \frac{1}{2}\lambda^2$, which gives precisely the result $I_k(x)$ stated above.

2.5 Random determinants

We have discussed how the supersymmetric method can be used in random determinant calculations such as $\mathbb{E}_X \log|\det X|$ and this in fact provides the basis for much of our work with random determinants arising in Kac-Rice formulae in chapters 3 and 4. In this section, we provide some broader background on random determinant calculations using different techniques and for other statistics.

A foundational work in random determinant calculations is [AAC13]. The focus of that work is the calculation of the complexity of the basic spherical p-spin glass model. Let $f\colon \mathbb{R}^N \to \mathbb{R}$ be the spin glass and consider the following sets of points on the N-sphere:

$$\begin{aligned}&\{\boldsymbol{x}\in S^N \mid \nabla f(\boldsymbol{x}) = 0,\ f(\boldsymbol{x}) < \sqrt{N}u\},\\ &\{\boldsymbol{x}\in S^N \mid \nabla f(\boldsymbol{x}) = 0,\ f(\boldsymbol{x}) < \sqrt{N}u,\ i(\nabla^2 f(\boldsymbol{x})) = k\},\end{aligned}$$

where $i(\cdot)$ is the *index*, which simply counts the number of negative eigenvalues of a real symmetric matrix. For fixed N, note that these sets are almost surely finite and so one can unambiguously define the following notions of complexity simply as the cardinality of these sets:

$$\mathbb{E}C_N(u) = |\{\boldsymbol{x} \in \mathbb{R}^N \mid \nabla f(\boldsymbol{x}) = 0,\ f(\boldsymbol{x}) < \sqrt{N}u\}|,$$
$$\mathbb{E}C_{N,k}(u) = |\{\boldsymbol{x} \in \mathbb{R}^N \mid \nabla f(\boldsymbol{x}) = 0,\ f(\boldsymbol{x}) < \sqrt{N}u,\ i(\nabla^2 f(\boldsymbol{x})) = k\}|,$$

The appropriateness of the scale $\sqrt{N}$ of the upper bound on f will become apparent below. The argument proceeds in the following steps:

1. Apply a Kac-Rice formula to express the complexity as integral involving the absolute value of the determinant of a random Hessian:
$$\mathbb{E}C_{N,k}(u) = \int_{S^N} d\boldsymbol{x}\ \mathbb{E}\left[|\det \nabla^2 f| \mathbf{1}\{f(\boldsymbol{x}) \leqslant \sqrt{N}u\}\mathbf{1}\{i(\nabla^2 f(\boldsymbol{x})) = k\} \mid \nabla f(\boldsymbol{x}) = 0\right] p_{\boldsymbol{x}}(0)$$
where $p_{\boldsymbol{x}}$ is the density of ∇f at $\boldsymbol{x}$.

2. Exploit spherical symmetry of the integrand to dispense with the integral over the N-sphere.

3. Use the covariance function of f to derive the joint distribution of f, its derivatives and its Hessian. The derivatives must be taken parallel to the N-sphere, so the Hessian is an $N-1 \times N-1$ matrix. One discovers that ∇f is independent of f and ∇f, which greatly simplifies the above expectation. Moreover, $\nabla^2 f$ just has Gaussian entries and Gaussian conditioning laws can be used to derive the distribution of $\nabla^2 f \mid f(\boldsymbol{x}) = y$. One finds that it is a shifted GOE $X - yI$, where X is a standard GOE. In addition, ∇f is an isotropic Gaussian vector with variance p.

4. The complexity is then given by
$$\mathbb{E}C_{N,k} \propto \int_{-\infty}^{u} dy\ e^{-\frac{Ny^2}{2}}\mathbb{E}\left[|\det(X - yI)| \mid \mathbf{1}\{i(X - yI) = k\}\right]. \tag{2.54}$$

5. The determinant simplifies greatly and can be written as a product over eigenvalues. Then the expectation can be rewritten as
$$\mathbb{E}\left[|\det(X - yI)| \mid \mathbf{1}\{i(X - yI) = k\}\right] \propto \int d\lambda_1 \ldots \lambda_{N-1} e^{-\frac{Ny^2}{2}} \prod_{j=1}^{N-1} e^{-\frac{(N-1)\lambda_j^2}{2}} \Delta(\{\lambda_i\}_{i=1}^{N-1}) \prod_{j=1}^{N-1} |\lambda_j - y|$$
$$\mathbf{1}\{\lambda_1 \leqslant \ldots \leqslant y \leqslant \lambda_{N-1}\}.$$

   Note that from the above expression it is clear that $\sqrt{N}$ is the correct scaling to make the density of f agree with that of the eigenvalues of $\nabla^2 f$. With some re-scaling of variables, the determinant and the Vandermonde terms combine to give an $N \times N$ Vandermonde, so overall
$$\mathbb{E}C_{N,k}(u) \propto \mathbb{P}(\lambda_k \leqslant A_N u)$$
   with the probability taken over an $N \times N$ GOE and for some constant A_N. $\mathbb{E}C_{N,k}$ can then be computed using a large deviations principle for the k-th eigenvalue of an $N \times N$ GOE.

6. C_N can be derived from $C_{N,k}$ by summing over all k.

In reality, the main results of [AAC13] and related work (such as our own) focus on computing the leading order term in a large N asymptotic expansion of $\log\mathbb{E}|\det(X-yI)|$, though in some cases it is possible to compute the sharp leading order term in $\mathbb{E}|\det(X-yI)|$, as done in [AAC13] and also in chapter 3. To state the precise results from [AAC13], we require the following definitions:

$$\Theta_p(u) = \begin{cases} \frac{1}{2}\log(p-1) - \frac{p2-2}{4(p-1)}u^2 - I_1(u;E_\infty) & \text{if } u \leqslant -E_\infty, \\ \frac{1}{2}\log(p-1) - \frac{p-2}{4(p-1)}u^2 & \text{if } -E_\infty \leqslant u \leqslant 0, \\ \frac{1}{2}\log(p-1) & \text{if } 0 \geqslant u, \end{cases} \tag{2.55}$$

where $E_\infty = 2\sqrt{\frac{p-1}{p}}$, and $I_1(\cdot;E)$ is defined on $(-\infty,-E]$ by

$$I_1(u;E) = \frac{2}{E^2}\int_u^{-E}(z^2-E^2)^{1/2}dz = -\frac{u}{E^2}\sqrt{u^2-E^2} - \log\left(-u+\sqrt{u^2-E^2}\right) + \log E, \tag{2.56}$$

and

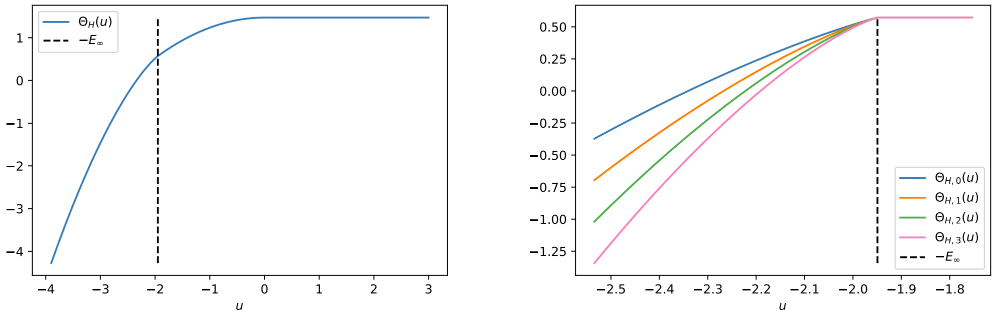
$$\Theta_{p,k}(u) = \begin{cases} \frac{1}{2}\log(p-1) - \frac{p-2}{4(p-1)}u^2 - (k+1)I_1(u;E_\infty) & \text{if } u \leqslant -E_\infty, \\ \frac{1}{2}\log(p-1) - \frac{p-2}{p} & \text{if } u > -E_\infty. \end{cases} \tag{2.57}$$

Then we have the following limit results

$$\lim_{N\to\infty}\frac{1}{N}\log\mathbb{E}C_N(u) = \Theta_p(u), \quad \lim_{N\to\infty}\frac{1}{N}\log\mathbb{E}C_{N,k}(u) = \Theta_{p,k}(u). \tag{2.58}$$

There are some important features to highlight about these results. Note that E_∞ plays the role of the left edge of the support of a semi-circle density which, of course, has its origin in the GOE distribution of f's Hessian. In particular, note that $\Theta_{p,k}$ includes large deviations terms for u below $-E_\infty$, the effective left edge of a semi-circle, but not above it. We also note the structure of stationary points of f that is encoded in Θ_p and $\Theta_{p,k}$ for which we show plots in Figure 2.5. Negative values of $\Theta_{p,k}(u)$ correspond to upper bounds on f below which it has 'exponentially few' stationary points of index k i.e. effectively none. Positive values, by contrast, correspond to exponentially many stationary points of index k. This therefore is the mathematical description of the 'layered structure' of spin glass stationary points on which [Cho+15] and our results in chapters 3 and 4 depend. There exist critical values $E_i{}_{i=1}^{\infty}$ such that $\Theta_{p,i}(-E_i) = 0$. For f below the critical value $-E_0$, there are effectively no stationary points of f. Between $-E_0$ and $-E_1$, there are exponentially many local minima, but effectively no stationary points of any other index. Between $-E_1$ and $-E_2$ there are exponentially many local minima and stationary points of index 1, but effectively none of any higher indices. The final critical value is $-E_\infty$, above which stationary points of all indices are found

The quantity $\log\mathbb{E}C_N$, where the logarithm is taken *after* the expectation is known as the *annealed average*, and so the corresponding complexity is known as the annealed complexity. The alternative



(a) Plot of Θ_H.

(b) Plot of $\Theta_{H,k}$ $k = 0,1,2,3$.

Figure 2.5: Plots of the functions Θ_H and $\Theta_{H,k}$ for $H = 20$.

is known as the *quenched complexity*, in which the expectation is taken after the logarithm. We shall discuss the differences between the two below. The first few steps outlined above are quite general and we shall see them repeated, mutatis mutandis, in chapters 3 and 4. The later steps, however, are clearly highly specific to the precise conditional Hessian distribution of the spin-glass. In particular, if the Hessian were a GOE shifted by a some matrix other than a multiple of the identity, then one would be unable to so easily dispense with the eigenvector component of the matrix expectation. Further, step 5 is an miraculous simplification wherein the conditional value of f is effectively inserted as an extra eigenvalue of the GOE, so reducing the whole calculation to a tail probability of the k-th eigenvalue of a GOE. We shall in chapters 3, 4 and 8 how supersymmetric techniques, among others, can be be employed to generalise these steps in more complicated settings. In a sequence of recent works [ABM21a; ABM21b; McK21] the question of random determinants was considered for considered for very general random matrices. Indeed, there is every reason to believe that the general framework developed particularly in [ABM21a] provides close to optimal conditions under which the annealed average over absolute values of random matrix determinants can be computed. The method developed in that work is, in essence, a rigorous mathematically justified version of a general mathematical physics approach known as the *Coulomb gas method* [CFV16; For10]. Consider a random $N \times N$ matrix X with (random) eigenvalues $\lambda_1, \ldots, \lambda_N$, empirical spectral density $\hat{\mu}_N$ and assume a limiting spectral density μ. Let us consider real symmetric X, but of course what we describe can be equally well presented for Hermitian X. One can simply express the determinant of X in terms of its eigenvalues alone and then use the definition of $\hat{\mu}_N$ to write

$$\mathbb{E}|\det X| = \mathbb{E}\prod_{j=1}^{N}|\lambda_j| = \mathbb{E}\exp\left\{N\int d\hat{\mu}_N(\lambda)\log|\lambda|\right\}.$$

Recall from (2.10) that the eigenvalue density can be written in the form

$$p(\lambda_1, \ldots, \lambda_N) = \frac{1}{Z_N}\exp\left(-\frac{N^2}{2}\mathcal{I}[\hat{\mu}_N]\right),$$

so that

$$\mathbb{E}|\det X| = \frac{1}{Z_N}\int d\lambda_1 \dots d\lambda_N \exp\left\{N\int d\hat{\mu}_N(\lambda)\log|\lambda|\right\}\exp\left(-\frac{N^2}{2}\mathcal{I}[\hat{\mu}_N]\right).$$

Heuristically, the Laplace method can be applied to conclude that the dominant leading order contribution to this integral as $N \to \infty$ comes from $\hat{\mu}_N$ in a small ball around μ, so

$$\mathbb{E}|\det X| \sim \exp\left\{N\int d\mu(\lambda)\log|\lambda|\right\}$$

and then

$$\frac{1}{N}\log\mathbb{E}|\det X| \sim \int d\mu(\lambda)\log|\lambda|.$$

This approach gives solid intuition for the asymptotic behaviour of $\mathbb{E}|\det X|$ in general, but is of course only heuristic. In chapter 4, the Coulomb gas method plays an important part in the Kac-Rice calculation of complexity, however we have to expend some effort to provide the rigorous justification for its use in that particular case and these arguments are quite specific to the matrix ensemble in question. The main theorems of [ABM21a] provide a general justification for the Coulomb gas method, or really the result above that can be derived using it. The theorems are quite general but rely on a number of technical conditions on the matrix ensemble and much of the effort in that paper and its companions [ABM21b; McK21] is devoted to proving satisfaction of these conditions for some particular matrix ensembles of interest. Interestingly, parts of the argument in [ABM21a] are not dissimilar to the Laplace method heuristic above, as one of the key ingredients is a condition on X giving good enough bounds on the convergence rate of $\hat{\mu}_N$ to $\mathbb{E}\hat{\mu}_N$ and $\mathbb{E}\hat{\mu}_N$ to μ. At the time of writing, these results are the most general and powerful tools for calculating $\mathbb{E}|\det X|$, however establishing satisfaction of their conditions is by no means straightforward so for some matrix ensembles, less general techniques may be easier to apply.

We close this section by mentioning the differences between the annealed averages that we have discussed in some detail and the alternative quenched averages. The Jensen-Shannon theorem gives the inequality

$$\mathbb{E}\log|\det X| \leqslant \log\mathbb{E}|\det X|$$

so the annealed average is an upper bound for the quenched average and likewise the annealed complexity of a random function is an upper bound for the quenched complexity. The annealed complexity has received much more attention in the literature, in part because it is more analytically tractable. At least heuristically, one can see why this should be by just trying to repeat the simple Coulomb gas argument above. Recall that the key to the argument's success (and, in some real sense, the success of [ABM21a]) is expressing $|\det X|$ as $\exp\left(N\int d\hat{mu}_N(\lambda)\log|\lambda|\right)$. This expression, written as a functional of $\hat{\mu}_N$ and in the form $e^{N\cdots}$ is exactly what is required for Laplace style asymptotic analysis when combined with the eigenvalue density inside the expectation. By contrast,

$\log|\det X| = N\int d\hat{\mu}_N(\lambda)\log|\lambda|$, which cannot be expressed in the above Laplace-amenable form. [Ros+19] is an important recent work that begins the extension of the Kac-Rice approach to quenched complexity via the non-rigorous replica method. The authors highlight that the quenched and annealed complexities do not in general agree even to leading order and argue that the quenched complexity is, in some sense, the better representation of a surface's complexity. In chapters 3 and 4 in which annealed complexity calculations feature significantly, we use highly simplified statistical physics models of much more complicated objects (deep neural networks), attempting to retain just enough of the original structure to provide some insight while still having an analytically tractable complexity. What's more, the complexity itself, annealed or quenched, is just a static snapshot of the already much simplified loss landscape, whereas real-world neural networks are trained over some complex stochastic trajectory in parameter space. As with any model of a complex system, these complexity calculations can only ever be expected to provide some limited insight into aspects of the underlying system. Given that the models themselves are very simplified and a focus on just their complexity is a considerable simplification of real training dynamics, we argue that the distinction between annealed on quenched complexity in this context, while important, is not the most significant factor affecting ecological validity.

Finally, we note that quantities other than the expectation of complexity (equivalently: absolute values of determinants) have been considered. In the context of spherical p-spin glass considered in [AAC13], the *variance* of the complexity is obtained in [Sub17] which is necessary to determine whether the expected value is typical. The proofs in this case are much more technical than those for the expectation and extensions to more complicated models such as those considered in Chapters 3 and 4 appears out of reach.

2.6 Free probability

Free probability theory is a rich and deep field describing probability distributions on non-commuting algebras. The notation of *freeness* itself provides the generalisation of the concept of independence from standard probability theory to non-commuting algebras. The theory extends beyond the boundaries of random matrix theory to probability distributions on more general algebras [VDN92], but its connection to random matrix theory is immediately clear: random matrices are non-commuting objects endowed with probability distributions. For the purposes of this thesis, we will need only a basic introduction to free probability in the context of random matrices.

Consider two $N \times N$ real matrices A and B, where A is random and B may be random or deterministic. Suppose that A is rotationally invariant, i.e. its eigenvectors follow Haar measure on the orthogonal group. A is then said to be *in general position* compared to B, which means roughly that there is entirely no correlation or dependence between their eigenspaces. In this case, A and B can be shown to be free independent of each other. Suppose that both A and B have limiting spectral measures μ and ν respectively and let $C = A + B$. Since A and B are free independent, it is known [VDN92; AGZ10] that C has the limiting spectral measure $\mu \boxplus \nu$, which is known as the free additive convolution between the measures μ and ν. To define the free additive convolution, we must introduce some integral transforms. Let g_μ, g_ν be the Stieljtes transforms of μ and ν and let $B_\mu = g_\mu^{-1}$ and $B_\nu = g_\nu^{-1}$ be their inverses. The R-transforms are then defined as $R_\mu(z) = B_\mu(z) - z^{-1}$ and $R_\nu(z) = B_\nu(z) - z^{-1}$. The R-transforms play the role of Fourier transforms for probability measures, as one has the result

$$R_{\mu \boxplus \nu} = R_\mu + R_\nu. \tag{2.59}$$

In fact, one must take care with the definitions of these transforms. The above expressions are just a consequence of their true definitions as formal power series in the complex plane.

The Stieljtes transform of a measure is given by the power series

$$g_\mu(z) = \sum_{n \geqslant 0} m_n^{(\mu)} z^{-(n+1)} \tag{2.60}$$

where $m_n^{(\mu)} = \int d\mu(x)\ x^n$ is the n-th moment of μ (likewise for ν). The R-transform of a measure is defined as a formal power series [AGZ10]

$$R_\mu(z) = \sum_{n=0}^{\infty} k_{n+1}^{(\mu)} z^n \tag{2.61}$$

where $k_n^{(\mu)}$ is the n-th cumulant of the measure μ. It is known [AGZ10] that $k_n^{(\mu)} = C_n^{(\mu)}$ where the functional inverse of the Stieljtes transform of the measure is given by the formal power series

$$B_\mu(z) = \frac{1}{z} + \sum_{n=1} C_n^{(\mu)} z^{n-1}. \tag{2.62}$$

So the key result (2.59) is really a statement about the cumulants of μ, ν and $\mu \boxplus \nu$, namely $k_n^{(\mu \boxplus \nu)} = k_n^{(\mu)} + k_n^{(\nu)}$.

There is a useful relation between cumulants and moments which can be found, for example, in the proof of Lemma 5.3.24 in [AGZ10]:

$$m_n = \sum_{r=1}^{n} \sum_{\substack{0 \leqslant i_1, \ldots, i_r \leqslant n-r \\ i_1 + \ldots + i_r = n-r}} k_r m_{i_1} \ldots m_{i_r}. \tag{2.63}$$

The final concept we need from free probability theory is *subordination functions*. Given measures μ, ν there exists a subordination function $\omega : \mathbb{C} \to \mathbb{C}$ such that $g_{\mu \boxplus \nu}(z) = g_\nu(\omega(z))$ [Bia97]. Depending on the context, the subordination function formulation relating $\mu \boxplus \nu$ to μ and ν can prove more convenient than the formulation via sums of R-transforms, see e.g. [Bia97; CD16] and chapter 8 below.

We conclude this briefest of introduction to free probability by providing a few concrete results for integral transforms of the a specific measure, namely the semi-circle μ_{SC} with density $\rho_{SC}(x) = \pi^{-1}\sqrt{2 - x^2}$. We shall include the calculations as we have been repeatedly frustrated to find them absent from the literature. Henceforth $\mu = \mu_{SC}$ and we will drop all μ and SC labels.

Stieltjes transform For odd n clearly $m_n = 0$ by the symmetry of the semi-circle measure. Now consider the even moments:

$$\begin{aligned} m_{2n} = \pi^{-1}\int dx\, x^{2n}\sqrt{2-x^2} &= 2^{1+n}\pi^{-1}\int_{-\pi/2}^{\pi/2} d\theta \cos^2\theta \sin^{2n}\theta \\ &= 2^{1+n}\pi^{-1}\int_{-\pi/2}^{\pi/2} d\theta (\sin^{2n}\theta - \sin^{2(n+1)}\theta) \end{aligned} \tag{2.64}$$

The trigonometric integrals are standard exercises in basic calculus[2]:

$$\int_{-\pi/2}^{\pi/2} d\theta \sin^{2n}\theta = \pi \frac{2n-1}{2n}\frac{2n-3}{2n-2}\cdots\frac{1}{2} \tag{2.65}$$

so

$$m_{2n} = 2^{1+n}\frac{2n-1}{2n}\frac{2n-3}{2n-2}\cdots\frac{1}{2}\left(1-\frac{2n+1}{2n+2}\right) = 2^{1+n}\frac{2n-1}{2n}\frac{2n-3}{2n-2}\cdots\frac{1}{2}\frac{1}{2n+2}. \tag{2.66}$$

Thus we have the Stieltjes transform

$$\begin{aligned} g(z) &= \sum_{n=0}^{\infty} z^{-(2n+1)}2^{1+n}\frac{1}{2n+2}\frac{2n-1}{2n}\frac{2n-3}{2n-2}\cdots\frac{1}{2} \\ &= z\sum_{n=0}^{\infty}\left(\frac{z^2}{2}\right)^{-(n+1)}\frac{1}{2n+2}\frac{2n-1}{2n}\frac{2n-3}{2n-2}\cdots\frac{1}{2} \\ &= z\sum_{n=0}^{\infty}\left(\frac{z^2}{2}\right)^{-(n+1)}\frac{1}{(n+1)!}\frac{(2n-1)(2n-3)\dots 1}{2^{n+1}} \\ &= z\sum_{n=0}^{\infty}\left(\frac{-z^2}{2}\right)^{-(n+1)}\frac{1}{(n+1)!}\frac{(2n-1)(2n-3)\dots 1}{2^{n+1}}(-1)^{n+1} \end{aligned} \tag{2.67}$$

and we can now identity the Taylor expansion of a familiar function, so

$$g(z) = z\left(1-\sqrt{1-\frac{2}{z^2}}\right) = z - \sqrt{z^2-2}. \tag{2.68}$$

For a general semi-circle with radius r, we can thence immediately write down its Stieltjes transform

$$\frac{2}{r}\left(z-\sqrt{z^2-r^2}\right) \tag{2.69}$$

where the pre-factor comes simply from the appropriate normalisation of the density $\sqrt{r^2-x^2}$ relative to $\sqrt{2-x^2}$. Inverting this Siteltjes transform is simple. Let $y(z) = g_r^{-1}(z)$, then

$$\begin{aligned} rz &= 2y - 2\sqrt{y^2-r^2} \\ \iff 4y^2 - 4r^2 &= 4y^2 - 4zry + z^2r^2 \\ \iff g_r^{-1}(z) = y(z) &= \frac{1}{z} + \frac{rz}{4} \end{aligned}$$

from which it follows that $R_r(z) = \frac{rz}{4}$.

[2] The usual approach is to write $\sin^{2n}\theta = \sin^{2n-2}\theta - \cos^2\theta\sin^{2n-2}\theta$, apply integration by parts to the second term and then iterate.

2.7 Local laws and universality

Earlier in this chapter, we introduced the Wigner surmise and the rough notion of local universality in random matrices. This section provides further details about universality, with particular emphasis on the rather stunning sequence of papers beginning around [EYY12] that are well on the way to answering quite definitively the question of local universality.

Broadly speaking, universality refers to the phenomenon that certain properties of special random matrix ensembles (such as the GOE) remain true for more general random matrices that share some key feature with the special ensembles. For example, the Wigner semicircle is the limiting spectral density of the Gaussian Wigner ensembles, i.e. matrices with Gaussian entries, independent up to symmetry (symmetric real matrices, Hermitian complex matrices) [Meh04]. The Gaussian case is the simplest to prove, and there are various powerful tools not available in the non-Gaussian case, however the Wigner semicircle has been established as the limiting spectral density for Wigner matrices with quite general distributions on their entries [AGZ10; Tao12]. While surprisingly general is some sense, the Wigner semicircle relies on independence (up to symmetry) of matrix entries, a condition which is not typically satisfied in real systems. The limiting form of the spectral density of a random matrix ensemble is a *macroscopic* property, i.e. the matrix is normalised such that the average distance between adjacent eigenvalues is on the order of $1/\sqrt{N}$, where N is the matrix size. At the opposite end of the scale is the *microscopic*, where the normalisation is such that eigenvalues are spaced on a scale of order 1; at this scale, random matrices display a remarkable universality. For example, any real symmetric matrix has a set of orthonormal eigenvectors and so the set of all real symmetric matrices is closed under conjugation by orthogonal matrices. Wigner conjectured that certain properties of GOE matrices hold for very general random matrices that share the same (orthogonal) symmetry class, namely symmetric random matrices (the same is true of Hermitian random matrices and the unitary symmetry class). The spacings between adjacent eigenvalues should follow a certain explicit distribution, the Wigner surmise, and the eigenvectors should be *delocalised*, i.e. the entries should all be of the same order as the matrix size grows. Both of these properties are true for the GOE and can be proved straightforwardly with quite elementary techniques. Indeed, in the case of 2×2 GOE, it is a standard first exercise in random matrix theory to prove that the eigenvalue spacing distribution is precisely the Wigner surmise (for $N \times N$ GOEs it is only a good approximation and improves as $N \to \infty$). Microscopic random matrix universality is known to be far more robust than universality on the macroscopic scale. Indeed, such results are well established for invariant ensembles and can be proved using Riemann-Hilbert methods [Dei99]. For more general random matrices, microscopic universality has been proved by quite different methods in a series of works over the last decade or so, of which a good review is [EY17a]. Crucial in these results is the notion of a *local law* for random matrices. The technical statements of some local laws are given below, but roughly they assert that the spectrum of a random matrix is, with very high probability, close to the deterministic spectrum defined by its limiting spectral density (e.g. the semicircle law for Wigner matrices). Techniques vary by ensemble, but generally a local law for a random matrix ensemble provides the control required to demonstrate that certain matrix statistics are essentially invariant under the evolution of the Dyson Brownian motion. In the case of real symmetric matrices, the Dyson Brownian motion converges in finite time to the GOE, hence the statistics preserved under the Dyson Brownian motion must match the GOE. The n-point correlation functions of eigenvalues are one such preserved quantity, from which follows, amongst other properties, that the Wigner surmise is a good approximation to the adjacent spacings distribution. The process we have just outlined is known as the 'three step strategy', which we now state in its entirety for real Wigner matrices, though the essence of the strategy is much more general.

1. Establish a local semi-circle law for the general Wigner ensemble X.

2. Universality for Gaussian divisible ensembles. Consider a random matrix $X_t = e^{-t/2}X + \sqrt{1-e^{-t}}G$, where G is a standard GOE matrix. One must show that X_t has universality for $t = N^{-\tau}$ for any $0 < tau < 1$. The clearest interpretation of this result is that, as X evolves under a matrix Ornstein-Uhlenbeck process, its local eigenvalue statistics have 'relaxed' to those of the GOE after any timescales greater than N^{-1}. Concretely this process is

$$dX_t = \frac{1}{\sqrt{N}}dB_t - \frac{1}{2}X_t dt$$

   where B_t is a standard symmetric Brownian motion and the initial data is $X_0 = X$. The local law on X is a key ingredient in establishing this result.

3. Approximation by a Gaussian divisible ensemble. This final step, sometimes called the 'comparison step', has to show that the local statistics of the matrix X can be well approximated by those of the Gaussian divisible ensemble X_t for short times scale $N^{-\tau}$ where $\tau < 1$. Combining with step 2, one then obtain universality for X.

We now make the preceding statements about correlation functions precise, following the treatment in [EY17b]. For an $N \times N$ matrix X, let $p_N^{(k)}$ be its k-point correlation function, i.e.

$$p_N^{(k)}(x_1, \ldots, x_k) = \int d\lambda_{k+1} \ldots d\lambda_N p_N(x_1, \ldots, x_k, \lambda_{k+1}, \ldots, \lambda_N)$$

where p_N is simply the symmetrised joint probability density of the eigenvalues of X (i.e. the joint density of the unordered eigenvalues). Assume that X has a limiting spectral density ρ with compact support and is normalised so that it the support is $[-\sqrt{2}, \sqrt{2}]$. Assume also that the symmetry group of X is $O(N)$, i.e. X is real-symmetric. One statement of spectral universality for X is that for any $\kappa > 0$ and for any $E \in [-\sqrt{2}+\kappa, \sqrt{2}-\kappa]$ we have

$$\lim_{N\to\infty} \frac{1}{\rho(E)^k} \int_{\mathbb{R}^k} d\boldsymbol{\alpha} F(\boldsymbol{\alpha}) p_N^{(n)}\left(E + \frac{\boldsymbol{\alpha}}{N\rho(E)}\right) = \int_{\mathbb{R}^k} d\boldsymbol{\alpha} F(\boldsymbol{\alpha}) q_{GOE}^{(k)}(\boldsymbol{\alpha})$$

for any smooth and compactly supported function $F\colon \mathbb{R}^k \to \mathbb{R}$. Here $q_{GOE}^{(k)}$ is simply the k-point correlation function for a GOE scaled so that its semi-circular radius is $\sqrt{2}$. This is so-called *spectral universality in the bulk*. From this statement, the local nature of spectral universality is quite plain. One fixes some location inside the bulk of the limiting spectral density of X, referred to as an *energy* E[3], then ones takes an fixed number k of eigenvalues and looks at their marginal joint probability density in a region of the spectrum centred tightly on E. As the matrix size N diverges, so the small region around E shrinks and the joint distribution of the k eigenvalues in the small region converges to simply the joint distribution of k eigenvalues of a standard GOE matrix. Note that the 'small region' around the location E in the spectral bulk has a precisely prescribed scaling of $1/N$, which is the scaling so that, with overwhelming probability, the number of eigenvalues in the small region is of order 1. Spectral universality as presented above is clearly good deal stronger than the Wigner surmise and is describing at least a similar phenomenon. We can go further however, an consider a different formulation of spectral universality that is a direct generalisation of the Wigner surmise, namely *spectral gap universality in the bulk*. Of course, we note that all of the above has been stated for real symmetric matrices and the GOE, but could equally well have been stated for Hermitian matrices and the GUE.

For an $0 < \alpha < 1$ and any integers $r, s \in [\alpha N, (1-\alpha)N]$

$$\begin{aligned}\lim_{N\to\infty}\Big|\mathbb{E}_X F\big(N\rho(\lambda_r)(\lambda_r - \lambda_{r+1}), \dots, N\rho(\lambda_r)(\lambda_r - \lambda_{r+k})\big)&\\ - \mathbb{E}_{GOE} F\big(N\rho_{SC}(\lambda_s)(\lambda_s - \lambda_{s+1}), \dots, N\rho_{SC}(\lambda_s)(\lambda_s - \lambda_{s+K})\big)\Big| &= 0\end{aligned}$$

where F is an arbitrary function as before. These two formulations of spectral universality are known to be equivalent [EY17b]. To recover the Wigner surmise, take $n = 1$ and then one obtains

$$\lim_{N\to\infty}\Big|\mathbb{E}_X F\big(N\rho(\lambda_r)(\lambda_r - \lambda_{r+1})\big) - \mathbb{E}_{GOE} F\big(N\rho_{SC}(\lambda_s)(\lambda_s - \lambda_{s+1})\big)\Big| = 0. \tag{2.70}$$

Note that $\rho_{SC}(\lambda_s)N$ is precisely the scaling required around λ_s to bring the GOE eigenvalues onto the scale on which the mean spacing is unity, thus for large N

$$\mathbb{E}_{GOE} F\big(N\rho_{SC}(\lambda_s)(\lambda_s - \lambda_{s+1})\big) = \int dr \rho_{\text{Wigner}}(r) F(r) + o(N),$$

and so (2.70) is indeed the precise statement of the universality of the Wigner surmise for X.

There are several forms of local law, but all provide high probability control on the error between the (random) matrix Green's function $G(z) = (z - X)^{-1}$ and certain deterministic equivalents. In all cases we use the set

$$\boldsymbol{S} = \{E + i\eta \in \mathbb{C} \mid |E| \leqslant \omega^{-1},\ N^{-1+\omega} \leqslant \eta \leqslant \omega^{-1}\} \tag{2.71}$$

[3] The physics terminology is due to the historical origins of spectral universality in the Wigner surmise within the context of random matrix models for quantum mechanical Hamiltonians.

for $\omega \in (0,1)$ and the local law statements holds for all (large) $D > 0$ and (small) $\xi > 0$ and for all large enough N. The *averaged local law* states:

$$\sup_{z\in \boldsymbol{S}} \mathbb{P}\left(\left|\frac{1}{N}\mathrm{Tr}G(z) - g_\mu(z)\right| > N^\xi\left(\frac{1}{N\eta} + \sqrt{\frac{\Im g_\mu(z)}{N\eta}}\right)\right) \leqslant N^{-D}. \tag{2.72}$$

The *isotropic local law* states:

$$\sup_{\|\boldsymbol{u}\|,\|\boldsymbol{v}\|=1, z\in \boldsymbol{S}} \mathbb{P}\left(|\boldsymbol{u}^T G(z)\boldsymbol{v} - g_\mu(z)| > N^\xi\left(\frac{1}{N\eta} + \sqrt{\frac{\Im g_\mu(z)}{N\eta}}\right)\right) \leqslant N^{-D}. \tag{2.73}$$

The *anisotropic local law* states:

$$\sup_{\|\boldsymbol{u}\|,\|\boldsymbol{v}\|=1, z\in \boldsymbol{S}} \mathbb{P}\left(|\boldsymbol{u}^T G(z)\boldsymbol{v} - \boldsymbol{u}^T\Pi(z)\boldsymbol{v}| > N^\xi\left(\frac{1}{N\eta} + \sqrt{\frac{\Im g_\mu(z)}{N\eta}}\right)\right) \leqslant N^{-D} \tag{2.74}$$

where $\Pi(\cdot)$ is an $N \times N$ deterministic matrix function on $\mathbb{C}$. The *entrywise local law* states:

$$\sup_{z\in \boldsymbol{S}, 1\leqslant i,j\leqslant N} \mathbb{P}\left(|G_{ij}(z) - \Pi_{ij}(z)| > N^\xi\left(\frac{1}{N\eta} + \sqrt{\frac{\Im g_\mu(z)}{N\eta}}\right)\right) \leqslant N^{-D}. \tag{2.75}$$

The anisotropic local law is a stronger version of the entrywise local law. The anisotropic local law is a more general version of the isotropic local law, which can be recovered in the isotropic case by taking $\Pi = g_\mu I$. The entrywise local law can also be applied in the isotropic case by taking $\Pi = g_\mu I$. The averaged local law is weaker than all of the other laws. General Wigner matrices are known to obey isotropic local semi-circle laws [ES17]. Anisotropic local laws are known for general deformations of Wigner matrices and general covariance matrices [KY17] as well as quite general classes of correlated random matrices [EKS19].

Local universality is not limited to the eigenvalues of random matrices. Recall that the eigenvectors of the canonical Gaussian orthogonal, unitary and symplectic ensembles are distributed with Haar measure on their respective symmetry groups. We have seen the precise and deep sense in which the eigenvalues of very general random matrices are similar to those of the very special canonical Gaussian orthogonal ensemble of the same symmetry class, but what of the eigenvectors? Is there some precise sense in which the eigenvectors of quite general random matrices are similar to Haar-distributed sets of vectors on their corresponding symmetry group? The first steps in this direction can be found in [BY17] where *quantum unique ergodicity* (QUE) is proved for generalised Wigner matrices. It is well known that the eigenvectors of quite general random matrices display a universal property of *delocalisation*, namely

$$|u_k|^2 \sim \frac{1}{N} \tag{2.76}$$

for any component u_k of an eigenvector $\boldsymbol{u}$. Universal delocalisation was conjectured by Wigner along with the Wigner surmise for adjacent eigenvalue spacing. QUE states that the eigenvectors of a random matrix are approximately Gaussian in the following sense ([BY17] Theorem 1.2):

$$\sup_{||\boldsymbol{q}||=1} \sup_{\substack{I\subset[N],\\ |I|=n}} \left|\mathbb{E}P\left(\left(N|\boldsymbol{q}^T\boldsymbol{u}_k|^2\right)_{k\in I}\right) - \mathbb{E}P\left(\left(|\mathcal{N}_j|^2\right)_{j=1}^n\right)\right| \leqslant N^{-\varepsilon},$$

for large enough N, where $\mathcal{N}_j$ are i.i.d. standard normal random variables, $(\boldsymbol{u}_k)_{k=1}^N$ are the normalised eigenvectors, P is any polynomial in n variables and $\varepsilon > 0$. Note that the set I in this statement is a subset of $[N] \equiv \{1,2,\ldots,N\}$ of *fixed size* n; n is not permitted to depend on N. Recall from earlier in this chapter, around (2.20), that fixed size subsets of Haar distributed eigenvectors of large random matrices can be well approximated by vectors of independent Gaussian entries. Note that the statement of QUE given above is of precisely the same character

Chapter 3

Neural networks with general activation functions

The content of this chapter was published first as a pre-print in April 2020 (`https://arxiv.org/abs/2004.03959`) and later as a journal article: "The loss surfaces of neural networks with general activation functions". **Nicholas P Baskerville**, Jonathan P Keating, Francesco Mezzadri and Joseph Najnudel. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(6):064001, 2021.

NPB suggested general activation functions as a focus, performed all of the calculations and experiments and wrote the paper. The other authors contributed ideas for possible approaches, provided feedback on results throughout and made small revisions to the drafts. Anonymous reviewers spotted some minor errors, advised on changes of presentation and provided useful references.

3.1 Introduction

3.1.1 Multi-layer perceptron neural networks

Let $f : \mathbb{R} \to \mathbb{R}$ be a suitably well-behaved (e.g. differentiable almost everywhere and with bounded gradient) non-linear *activation function* which is taken to applied entry-wise to vectors and matrices. We study multi-layer perceptron neural networks of the form

$$\boldsymbol{y}(\boldsymbol{x}) = f(W^{(H)} f(W^{(H-1)} f(\ldots f(W^{(1)}\boldsymbol{x})\ldots))) \tag{3.1}$$

where the input data vectors $\boldsymbol{x}$ lie in $\mathbb{R}^d$ and the *weight matrices* $\{W^{(\ell)}\}_{\ell=1}^{H}$ have any shapes compatible with $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{y}(\boldsymbol{x}) \in \mathbb{R}^c$. As discussed in Chapter 1, the matrices $W^{(\ell)}$ are parameters of the neural network f and in practice they will be randomly initialised with some standard distribution and then "learned" using some gradient descent algorithm on a data set. Their shapes are essentially arbitrary up-to compatibility constraints and the choice of *hidden layer* widths (i.e. the number of rows in each

$W^{\ell)}$) is an engineering decision unique to each concrete application. Note that, as in [Cho+15], we do not consider biases in the network.

3.1.2 Outline of results and methods

Following [Cho+15], we view $\boldsymbol{y}$ as a random function over a high-dimensional weight-space and explore its critical points, i.e. vanishing points of its gradient. The randomness will come from taking the input data to be random. We define the following key quantities[1]:

$$C_{k,H}(u) = \text{expected number of critical points of } \boldsymbol{y} \text{ of index } k \text{ taking values at most } u, \tag{3.2}$$

$$C_H(u) = \text{expected number of critical points of } \boldsymbol{y} \text{ taking values at most } u. \tag{3.3}$$

In Section 3.2 we make precise our heuristic definitions in (3.2)-(3.3). Following [AAC13] we obtain precise expressions for $C_{k,H}$ and C_H as expectations under the Gaussian Orthogonal Ensemble (GOE) and use them to study the asymptotics in the large-network limit. Our results reveal almost the same 'banded structure' of critical points as first found in [Cho+15]. In particular we establish the existence of the same critical values $E_0 > E_1 > ... > E_\infty$ such that, with overwhelming probability, critical points taking (scaled) values in $(-E_k, -E_{k+1})$ have index at-most $k+2$, and that there are exponentially many such critical points. We further obtain the exact leading order terms in the expansion of $C_H(u)$, this being the only point at which the generalised form of the activation function f affects the results. In passing, we also show that the network can be generalised to having any number of output neurons without much affecting the calculations of [Cho+15] who only consider single-output networks.

In Section 3.2 we extend the derivation of [Cho+15] to general activation functions by leveraging piece-wise linear approximations, and we extend to multiple outputs and new loss functions with a simple extension of the corresponding arguments in [Cho+15]. In Section 3.4 we obtain expressions for the complexities $C_{k,H}, C_H$ using a Kac-Rice formula as in [AAC13; FW07; Fyo04] but are forced to deal with a perturbed GOE matrix, preventing the replication of the remaining calculations in that work. Instead, in Section 3.5 we use the supersymmetric method following closely the work of [Noc16; FN15] and thereby reach the asymptotic results of [AAC13] by entirely different means.

3.2 Neural networks as random functions

In this section we show that, under certain assumptions, optimising the loss function of a neural network is approximately equivalent to minimising the value of a random function on a high dimensional hypersphere, closely related to the spin glass. Our approach is much the same as [Cho+15] but is extended to a general class of activation functions and also to networks with multiple output neurons.

[1] Recall that the *index* of a critical points is the number of negative eigenvalues of the Hessian at that point.

3.2.1 Modelling assumptions

We make the following assumptions, all of which are required for the specific analytic framework of the results in this chapter and are taken either exactly from, or by close analogy with [Cho+15]. We defer a discussion of their plausibility and necessity to Section 3.2.4.

1. Components of data vectors are i.i.d. standard Gaussians.
2. The neural network can be well approximated as a much sparser[2] network that achieves very similar accuracy.
3. The unique weights of the sparse network are approximately uniformly distributed over the graph of weight connections.
4. The activation function is twice-differentiable almost everywhere in $\mathbb{R}$ and can be well approximated as a piece-wise linear function with finitely many linear pieces.
5. The action of the piece-wise linear approximation to the activation function on the network graph can be modelled as i.i.d. discrete random variables, independent of the data at each node, indicating which linear piece is active.
6. The unique weights of a the sparse neural network lie on a hyper-sphere of some radius.

Remark 3.1. An alternative to assumption 5 would be to take the activation function to be *random* (and so too its piece-wise linear approximation). In this paradigm, we consider the ensuing analysis of this chapter to be a study of the *mean properties* of the induced ensemble of neural networks. Resorting to studying mean properties of complicated stochastic systems is a standard means of simplifying the analysis. We do not develop this remark further, but claim that the following calculations are not much affected by switching to this interpretation.

3.2.2 Linearising loss functions

In [Cho+15] the authors consider networks with a single output neuron with either L_1 or hinge loss and show that both losses are, in effect, just linear in the network output and with positive coefficient, so that minimising the loss can be replaced with minimising the network output. Our ensuing analysis can just as well be applied to precisely these situations, but here we present arguments to extend the applicability to multiple output neurons for L_1 regression loss and the widely-used cross-entropy loss [JC17] for classification.

L$_1$ loss. The L_1 loss is given by

$$\mathcal{L}_{L_1}(\boldsymbol{y}(\boldsymbol{X}),\boldsymbol{Y}) := \sum_{i=1}^{c} |y_i(\boldsymbol{X}) - Y_i| \tag{3.4}$$

[2] As in [Cho+15], a network with N weights is sparse if it has s unique weight values and $s \ll N$.

where $\boldsymbol{X}$ is a single random data vector and $\boldsymbol{Y}$ a single target output. Following [Cho+15], we assume that the absolute values in (3.4) can be modelled by using Bernoulli random variables, M_i say, taking values in $\{-1, 1\}$. Precisely, we replace $|y_i(\boldsymbol{X}) - Y_i|$ with $M_i(y_i(\boldsymbol{X}) - Y_i)$, so that the Bernoulli variables M_i model which section of the absolute value function $y_i(\boldsymbol{X}) - Y_i$ lies in. We do not expect $\boldsymbol{X}, \boldsymbol{Y}$ and the M_i to be independent, however it may be reasonable to assume that $\boldsymbol{X}$ and the M_i are conditionally independent conditioned on $\boldsymbol{Y}$. We then have

$$\begin{aligned}\mathbb{E}_{M|\boldsymbol{Y}} \mathcal{L}_{L_1}(\boldsymbol{y}(\boldsymbol{X}), \boldsymbol{Y}) = \mathbb{E}_{M|\boldsymbol{Y}} \sum_{i=1}^{c} M_i(y_i(\boldsymbol{X}) - Y_i) &= \sum_{i=1}^{c} (2\pi_i - 1) y_i(\boldsymbol{X}) - \sum_{i=1}^{c} \mathbb{E}_{M|\boldsymbol{Y}} M_i Y_i \\ &= \sum_{i=1}^{c} (2\pi_i - 1) y_i(\boldsymbol{X}) - \sum_{i=1}^{c} (2\pi_i - 1) Y_i \end{aligned} \tag{3.5}$$

where the M_i are Bernoulli random variables with $\mathbb{P}(M_i = 1) = \pi_i$. Observe that the second term in (3.5) is independent of the parameters of the network.

Cross-entropy loss. The cross-entropy loss is given by

$$\mathcal{L}_{\text{entr}}(\boldsymbol{y}(\boldsymbol{X}), \boldsymbol{Y}) := -\sum_{i=1}^{c} Y_i \log\big(\mathrm{SM}[\boldsymbol{y}(\boldsymbol{X})]_i\big) \tag{3.6}$$

where SM is the *soft-max* function:

$$\begin{aligned}\mathrm{SM} : &\mathbb{R}^c \to \mathbb{R}^c, \\ &\boldsymbol{z} \mapsto \frac{\exp(\boldsymbol{z})}{\sum_{i=1}^{m} \exp(z_i)}\end{aligned} \tag{3.7}$$

and $\exp(\cdot)$ is understood to be applied entry-wise. Note that we are applying the standard procedure of mapping network outputs onto the simplex Δ^{c-1} to allow us to calculate a mutual entropy. Restricting to c-class classification problems and using one-hot label vectors [Inc20], we obtain

$$\mathcal{L}_{\text{entr}}(\boldsymbol{y}(\boldsymbol{X}), \boldsymbol{Y}) = -\sum_{i=1}^{c} Y_i \left\{ y_i(\boldsymbol{X}) - \log\left(\sum_{j=1}^{c} \exp(y_j(\boldsymbol{X})) \right) \right\} \tag{3.8}$$

We note that classification networks typically produce very 'spiked' soft-max outputs [Guo+17], therefore we make the approximation

$$\sum_{i=1}^{c} \exp(y_i(\boldsymbol{X})) \approx \max_{i=1,\dots,c} \{\exp(y_i(\boldsymbol{X}))\} \tag{3.9}$$

and so we obtain from (3.8) and (3.9)

$$\mathcal{L}_{\text{entr}}(\boldsymbol{y}(\boldsymbol{X}), \boldsymbol{Y}) \approx -\sum_{i=1}^{c} \left\{ Y_i y_i(\boldsymbol{X}) - Y_i \max_{j=1,\dots,c} \{y_j(\boldsymbol{X})\} \right\} \tag{3.10}$$

We now model the max operation in (3.10) with a categorical variable, M'' say, over the indices $i = 1, \dots, c$ and take expectations (again assuming conditional independence of $\boldsymbol{X}$ and M'') to obtain

$$\mathbb{E}_{M''|\boldsymbol{Y}} \mathcal{L}_{\text{entr}}(\boldsymbol{y}(\boldsymbol{X}), \boldsymbol{Y}) = -\sum_{i=1}^{c} Y_i \left(y_i(\boldsymbol{x}) - \sum_{j=1}^{c} \pi''_j y_j(\boldsymbol{X}) \right) \tag{3.11}$$

Now $\boldsymbol{Y}$ is a one-hot vector and so (3.11) in fact reduces to

$$\mathbb{E}_{M''|\boldsymbol{Y}}\mathcal{L}_{\text{entr}}(\boldsymbol{y}(\boldsymbol{X}),\boldsymbol{Y}) = \sum_{j=1}^{c} \pi''_j y_j(\boldsymbol{x}) - y_i(\boldsymbol{x}) \tag{3.12}$$

for some i.

Remark 3.2. The arguments in this section are not intended to be anything more than heuristic, so as to justify our study of $\boldsymbol{a}^T\boldsymbol{y}$ for some constant $\boldsymbol{a}$ instead of the actual loss function of a neural network. The modelling assumptions required are no stronger than those used in [Cho+15].

3.2.3 Network outputs as spin glass-like objects

We assume that the activation function, f, can be well approximated by a piece-wise linear function with finitely many linear pieces. To be precise, given any $\varepsilon > 0$ there exists some positive integer L and real numbers $\{\alpha_i,\beta_i\}_{i=1}^L$ and real $a_1 < a_2 < \ldots < a_{L-1}$ such that

$$\begin{aligned} |f(x) - (\alpha_{i+1}x + \beta_{i+1})| < \varepsilon \quad &\forall x \in (a_i, a_{i+1}],\ 1 \leqslant i \leqslant L-2,\\ |f(x) - (\alpha_1 x + \beta_1)| < \varepsilon \quad &\forall x \in (-\infty, a_1],\\ |f(x) - (\alpha_L x + \beta_L)| < \varepsilon \quad &\forall x \in (a_{L-1}, \infty). \end{aligned} \tag{3.13}$$

Note that the $\{\alpha_i,\beta_i\}_{i=1}^L$ and $\{a_i\}_{i=1}^{L-1}$ are constrained by $L-1$ equations to enforce continuity, viz.

$$\alpha_{i+1}a_i + \beta_{i+1} = \alpha_i a_i + \beta_i, \quad 1 \leqslant i \leqslant L-1 \tag{3.14}$$

Definition 3.1. A continuous piece-wise linear function with L pieces $\hat{f}\left(x; \{\alpha_i,\beta_i\}_{i=1}^L, \{a_i\}_{i=1}^{L-1}\right)$ is an (L,ε)-*approximation* to to a function f if $\left|f(x) - \hat{f}\left(x; \{\alpha_i,\beta_i\}_{i=1}^L, \{a_i\}_{i=1}^{L-1}\right)\right| < \varepsilon$ for all $x \in \mathbb{R}$.

Given the above definition, we can establish the following.

Lemma 3.1. *Let $\hat{f}\left(\cdot; \{\alpha_i,\beta_i\}_{i=1}^L, \{a_i\}_{i=1}^{L-1}\right)$ be a (L,ε)-approximation to f. Assume that all the $W^{(i)}$ are bounded in Frobenius norm[3]. Then there exists some constant $K > 0$, independent of all $W^{(i)}$, such that*

$$\left\| f(W^{(H)}f(W^{(H-1)}f(\ldots f(W^{(1)}\boldsymbol{x})\ldots))) - \hat{f}(W^{(H)}\hat{f}(W^{(H-1)}\hat{f}(\ldots \hat{f}(W^{(1)}\boldsymbol{x})\ldots)))\right\|_2 < K\varepsilon \tag{3.15}$$

for all $\boldsymbol{x} \in \mathbb{R}^d$.

Proof. Suppose that (3.15) holds with $H-1$ in place of H. Because $\hat{f}$ is piece-wise linear and continuous then we clearly have

$$|\hat{f}(x) - \hat{f}(y)| \leqslant \max_{i=1,\ldots,L}\{|\alpha_i|\}|x-y| \equiv K'|x-y| \tag{3.16}$$

[3] Recall assumption 6, which is translated here to imply bounded Frobenius norm.

which can be seen by writing

$$\hat{f}(x)-\hat{f}(y)=(\hat{f}(x)-\hat{f}(a_i))+(\hat{f}(a_i)-\hat{f}(a_{i-1}))+\ldots+(\hat{f}(a_{j+1})-\hat{f}(a_j))+(\hat{f}(a_j)-\hat{f}(y)) \tag{3.17}$$

for all intermediate points $a_j,\ldots,a_i\in(y,x)$. Using (3.16) and our induction assumption we obtain

$$\begin{aligned}
&\left\|\hat{f}(W^{(H)}f(W^{(H-1)}f(W^{(H-2)}f(\ldots f(W^{(1)}\boldsymbol{x})\ldots)))-\hat{f}(W^{(H)}\hat{f}(W^{(H-1)}\hat{f}(W^{(H-2)}\hat{f}(\ldots\hat{f}(W^{(1)}\boldsymbol{x})\ldots)))\right\|_2\\
\leqslant& cK'\left\|W^{(H)}\left[f(W^{(H-1)}f(W^{(H-2)}f(\ldots f(W^{(1)}\boldsymbol{x})\ldots)))-\hat{f}(W^{(H-1)}\hat{f}(W^{(H-2)}\hat{f}(\ldots\hat{f}(W^{(1)}\boldsymbol{x})\ldots)))\right]\right\|_2\\
\leqslant& cKK'\left\|W^{(H)}\right\|_F\varepsilon\\
\leqslant& K''\varepsilon,
\end{aligned}$$

for some K'', where on the last line we have used the assumption that the network weights are bounded to bound $\|W^{(H)}\|_F$. The result for $H=1$ follows immediately from (3.16). ■

Remark 3.3. One could be more explicit in the construction of the piece-wise linear approximation $\hat{f}$ from f given the error tolerance ε by following e.g. [Ber+15]. We do not develop this further here as we do not believe it to be important to the practical implications of our results.

In much the same vein as [Cho+15] (c.f. Lemma 8.1 therein), we now use the following general result for classifiers to further justify our study of approximations to a neural network in the rest of the chapter.

Theorem 3.1. *Let Z_1 and Z_2 be the outputs of two arbitrary c-class classifiers on a dataset $\mathcal{X}$. That is, $Z_1(x),Z_2(x)$ take values in $\{1,2,\ldots,c\}$ for $x\in\mathcal{X}$. If Z_1 and Z_2 differ on no more than $\varepsilon|\mathcal{X}|$ points in $\mathcal{X}$, then*

$$corr(Z_1,Z_2)=1-\mathcal{O}(\varepsilon) \tag{3.18}$$

where, recall, the correlation of two random variables is given by

$$\frac{\mathbb{E}(Z_1Z_2)-\mathbb{E}Z_1\mathbb{E}Z_2}{std(Z_1)std(Z_2)}. \tag{3.19}$$

Proof. Let $\mathcal{X}_i\subset\mathcal{X}$ be the set of data points for which $Z_1=i$ for $i=1,2,\ldots,c$. Let $\mathcal{X}_{i,j}\subset\mathcal{X}_i$ be those points for which $Z_1=i$ but $Z_2=j$ where $j\neq i$. Define the following:

$$p_i=\frac{|\mathcal{X}_i|}{|\mathcal{X}|},\quad \varepsilon_i^+=\sum_{j\neq i}\frac{|\mathcal{X}_{i,j}|}{|\mathcal{X}|},\quad \varepsilon_i^-=\sum_{j\neq i}\frac{|\mathcal{X}_{j,i}|}{|\mathcal{X}|}. \tag{3.20}$$

We then have

$$\mathbb{E}Z_1 = \sum_{i=1}^{c} i p_i, \tag{3.21}$$

$$\mathbb{E}Z_2 = \sum_{i=1}^{c} i(p_i - \varepsilon_i^+ + \varepsilon_i^-) \tag{3.22}$$

$$\mathbb{E}Z_1 Z_2 = \sum_{i=1}^{c} i^2(p_i - \varepsilon_i^+) + \sum_{1\leqslant i<j\leqslant c} ij\frac{|\mathcal{X}_{i,j}|+|\mathcal{X}_{j,i}|}{|\mathcal{X}|} \tag{3.23}$$

$$std(Z_1) = \left[\sum_{i=1}^{c} i^2 p_i - \sum_{i,j} ijp_ip_j\right]^{1/2} \tag{3.24}$$

$$std(Z_2) = \left[\sum_{i=1}^{c} i^2(p_i - \varepsilon_i^+ + \varepsilon_i^-) - \sum_{i,j} ij(p_i - \varepsilon_i^+ + \varepsilon_i^-)(p_j - \varepsilon_j^+ + \varepsilon_j^-)\right]^{1/2}. \tag{3.25}$$

Now, by assumption $\sum_i \varepsilon_i^{\pm} \leqslant \mathcal{O}(\varepsilon)$ and so $\varepsilon_i^{\pm} \leqslant \mathcal{O}(\varepsilon)$ for all i. Similarly, $|\mathcal{X}_{i,j}|/|\mathcal{X}| \leqslant \mathcal{O}(\varepsilon)$ and so we quickly obtain from (3.21)-(3.23)

$$cov(Z_1,Z_2) = \sum_{i=1}^{c} i^2 p_i - \sum_{i,j} ijp_ip_j + \mathcal{O}(\varepsilon). \tag{3.26}$$

Finally, combining (3.24) - (3.26) we obtain

$$corr(Z_1,Z_2) = \frac{1+\mathcal{O}(\varepsilon)}{(1+\mathcal{O}(\varepsilon))^{1/2}} = 1+\mathcal{O}(\varepsilon). \tag{3.27}$$

■

The final intermediate result we require gives an explicit expression for the output of a neural network with a piece-wise linear activation function.

Lemma 3.2. *Consider the following neural network*

$$\hat{\boldsymbol{y}}(\boldsymbol{x}) = \hat{f}(W^{(H)}\hat{f}(\ldots\hat{f}(W^{(1)}\boldsymbol{x})\ldots)) \tag{3.28}$$

where $\hat{f}\left(\cdot;\{\alpha_i,\beta_i\}_{i=1}^{L},\{a_i\}_{i=1}^{L-1}\right)$ *is a piece-wise linear function with L pieces. Then there exist* $A_{i,j}$ *taking values in*

$$\mathcal{A} := \left\{\prod_{i=1}^{H}\alpha_{j_i} \;:\; j_1,\ldots,j_H \in \{1,\ldots,L\}\right\} \tag{3.29}$$

and $A_{i,j}^{(\ell)}$ *taking values in*

$$\mathcal{A}^{(\ell)} := \left\{\beta_k\prod_{r=1}^{H-\ell}\alpha_{j_r} \;:\; j_1,\ldots,j_{H-\ell},k \in \{1,\ldots,L\}\right\} \tag{3.30}$$

such that

$$\hat{y}_i(\boldsymbol{x}) = \sum_{j=1}^{d}\sum_{k\in\Gamma_i} x_{j,k}A_{j,k}\prod_{l=1}^{H} w_{j,k}^{(l)} + \sum_{\ell=1}^{H}\sum_{j=1}^{n_\ell}\sum_{k\in\Gamma_i^{(\ell)}} A_{j,k}^{(\ell)}\prod_{r=\ell+1}^{H} w_{j,k}^{(r)} \tag{3.31}$$

where Γ_i *is an indexing of all paths through the network to the* i*-th output neuron,* $\Gamma_i^{(\ell)}$ *is an indexing of all the paths through the network from the* ℓ*-th layer to the* i*-th output neuron,* $w_{j,k}^{(l)}$ *is the weight applied to the* j*-th input on the* k*-th path in the* l*-th layer,* $x_{j,k} = x_j$*, and* n_ℓ *is the number of neurons in layer* ℓ*.*

Proof. Firstly, for some $j = 1,\dots,L$

$$\hat{f}(W^{(1)}\boldsymbol{x})_i = \alpha_j (W^{(1)}\boldsymbol{x})_i + \beta_j \tag{3.32}$$

and so there exist $j_1, j_2, \dots \in \{1,\dots,L\}$ such that

$$[W^{(2)}\hat{f}(W^{(1)}\boldsymbol{x})]_i = \sum_k W^{(2)}_{ik}(\alpha_{j_k}(W^{(1)}\boldsymbol{x})_k + \beta_{j_k}) = \sum_k \alpha_{j_k} W^{(2)}_{ik} \sum_l W^{(1)}_{kl} x_l + \sum_k W^{(2)}_{ik}\beta_{j_k}. \tag{3.33}$$

Continuing in the vein of (3.33), there exist $k_1, k_2, \dots \in \{1,\dots,L\}$ such that

$$\hat{f}(W^{(2)}\hat{f}(W^{(1)}\boldsymbol{x}))_i = \alpha_{k_i} \sum_r \alpha_{j_r} W^{(2)}_{ir} \sum_l W^{(1)}_{kl} x_l + \alpha_{k_i} \sum_r W^{(2)}_{ir}\beta_{j_r} + \beta_{k_i} \tag{3.34}$$

from which we can see that the result follows by re-indexing and induction. ■

We now return to the neural network $\boldsymbol{y}(\cdot)$. Fix some small $\varepsilon > 0$, let $\hat{f}\left(\cdot; \{\alpha_i, \beta_i\}_{i=1}^{L}, \{x_i\}_{i=}^{L-1}\right)$ be a (L,ε)-approximation to f and let $\hat{\boldsymbol{y}}$ be the same network as $\boldsymbol{y}$ but with f replaced by $\hat{f}$. By Lemma 3.1, we have[4]

$$\|\boldsymbol{y}(\boldsymbol{x}) - \hat{\boldsymbol{y}}(\boldsymbol{x})\|_2 \lesssim \varepsilon \tag{3.35}$$

for all $\boldsymbol{x} \in \mathbb{R}^d$, and so we can adjust the weights of $\hat{\boldsymbol{y}}$ to obtain a network with accuracy within $\mathcal{O}(\varepsilon)$ of $\boldsymbol{y}$. We then apply Lemma 3.2 to $\hat{\boldsymbol{y}}$ and assume[5] that the $A_{i,j}$ and $A^{(\ell)}_{i,j}$ can be modelled as i.i.d. discrete random variables with

$$\mathbb{E}A_{i,j} = \rho, \quad \mathbb{E}A^{(\ell)}_{i,j} = \rho_\ell \tag{3.36}$$

and then

$$\mathbb{E}\hat{y}_i(\boldsymbol{X}) = \rho \mathbb{E}_{\boldsymbol{x}} \sum_{j=1}^{d} \sum_{k\in\Gamma_i} X_{j,k} \prod_{l=1}^{H} w^{(l)}_{j,k} + \sum_{\ell=1}^{H} \rho_\ell \sum_{j=1}^{n_\ell} \sum_{k\in\Gamma^{(\ell)}_i} \prod_{r=\ell+1}^{H} w^{(r)}_{j,k}. \tag{3.37}$$

Our reasoning is now identical to that in Section 3.3 of [Cho+15]. We use the assumptions of sparsity and uniformity (Section 3.2.1, assumptions 2, 3) and some further re-indexing to replace (3.37) by

$$\mathbb{E}\tilde{y}_i(\boldsymbol{X}) = \rho \mathbb{E}_{\boldsymbol{X}} \sum_{i_1,\dots,i_H=1}^{\Lambda} X_{i_1,\dots,i_H} \prod_{k=1}^{H} w_{i_k} + \sum_{\ell=1}^{H} \rho_\ell \sum_{i_{\ell+1},\dots,i_H=1}^{\Lambda} \prod_{k=\ell+1}^{H} w_{i_k} \tag{3.38}$$

where Λ is the number of unique weights of the network and, in particular, the sparsity and uniformity assumptions are chosen to give

$$\mathbb{E}_{\boldsymbol{X}} \|\tilde{\boldsymbol{y}}(\boldsymbol{X}) - \hat{\boldsymbol{y}}(\boldsymbol{X})\|_2 \lesssim \varepsilon. \tag{3.39}$$

(3.35) and (3.39) now give

$$\mathbb{E}_{\boldsymbol{X}} \|\tilde{\boldsymbol{y}}(\boldsymbol{X}) - \boldsymbol{y}(\boldsymbol{X})\|_2 \lesssim \varepsilon \tag{3.40}$$

[4] Here we use the standard notation that, for a function p on $\mathcal{B}$, $p \lesssim \varepsilon$ if there exists a constant K such that $p(x) \leqslant K\varepsilon$ for all $x \in \mathcal{B}$.

[5] This assumption is the natural analogue of the assumption used in [Cho+15].

and in the case of classifiers, (3.40) ensures that the conditions for Theorem 3.1 are met, so establishing that

$$corr(\tilde{\boldsymbol{y}}(\boldsymbol{X}), \boldsymbol{y}(\boldsymbol{X})) = 1 - \mathcal{O}(\varepsilon). \tag{3.41}$$

As in [Cho+15], we use these heuristics to justify studying $\tilde{\boldsymbol{y}}$ hereafter in place of $\boldsymbol{y}$.

Recalling the results of Section 3.2.2, in particular (3.5) and (3.12) we conclude that to study the loss surface of $\tilde{\boldsymbol{y}}$ under some loss function it is sufficient to study quantities of the form $\sum_{i=1}^{c} \eta_i \tilde{y}_i$ and, in particular, we study the critical points. The X are centred Gaussian random variables and so any finite weighted sum of some X is a centred Gaussian variable with some variance. We can re-scale variances and absorb constants into the ρ_ℓ and thereby replace $\sum_i \eta_i \tilde{y}_i(\boldsymbol{X})$ with $\tilde{y}_i(\boldsymbol{X})$.

Note that we assumed an L_2 constraint on the network weights (Section 3.2.1, point 6) and that now carries forward as

$$\frac{1}{\Lambda} \sum_{i=1}^{\Lambda} w_i^2 = \mathcal{C} \tag{3.42}$$

for some constant $\mathcal{C}$. For ease of notation in the rest of the chapter, we define

$$g(\boldsymbol{w}) = \sum_{i_1,\dots,i_H=1}^{\Lambda} X_{i_1,\dots,i_H} \prod_{k=1}^{H} w_{i_k} + \sum_{\ell=1}^{H} \rho'_\ell \sum_{i_{\ell+1},\dots,i_H=1}^{\Lambda} \prod_{k=\ell+1}^{H} w_{i_k} \tag{3.43}$$

where $\rho'_\ell := \rho_\ell/\rho$. Finally, recall that we assumed the data entries X_i are i.i.d standard Gaussians. To allow further analytic progress to be made, we follow [Cho+15] and now extend this assumption to $X_{i_1,\dots,i_H} \overset{\text{i.i.d}}{\sim} \mathcal{N}(0,1)$. The random function g is now our central object of study and, without loss of generality, we take $\mathcal{C} = 1$ in (3.42) so that g is a random function on the (Λ-1)-sphere of radius $\sqrt{\Lambda}$.

Observe that the first term in (3.43) is precisely the form of an H-spin glass as found in [Cho+15] and the second term is deterministic and contains (rather obliquely) all the dependence on the activation function. Having demonstrated the link between our results and those in [Cho+15], we now set $\Lambda = N$ for convenience and to make plain the similarities between what follows and [AAC13]. We also drop the primes on ρ'_ℓ.

3.2.4 Validity of the modelling assumptions.

The authors of [Cho+15] discuss the modelling assumptions in [CLA15]. We add to their comments that the hyper-sphere assumption 6 seems easily justifiable as merely L_2 weight regularisation.

Assumption 5 from Section 3.2.1 is perhaps the least palatable, as the section of a piece-wise linear activation function in which a pre-activation value lies is a deterministic function of that pre-activation value and so certainly not i.i.d. across the network and the data items. It is not clear how to directly test the assumption experimentally, but we can certainly perform some experiments to probe its plausibility.

For the sake of clarity, consider initially a `ReLU` activation function. Let $\mathscr{N}$ be the set of all nodes (neurons) in a neural network, and let $\mathscr{D}$ be a dataset of inputs for this network. Assumption 5 says that we can model the action of the activation function at any neuron $\mathfrak{n} \in \mathcal{N}$ and any data point $\boldsymbol{x} \in \mathcal{D}$ as i.i.d. Bernoulli random variables. In particular, this is why the the expectations over the activation function indicators and the data distribution can be taken independently in (3.37). If one fixes some neuron $\mathfrak{n} \in \mathcal{N}$, and observes its pre-activations over all data points in $\mathcal{D}$, one will observe some proportion $\rho^{\mathfrak{n}}$ of positive values. Assumption 5 implies that this proportion should be approximately the same for each $\mathfrak{n} \in \mathcal{N}$, namely p, where p is the success probability of the Bernoulli. Taking all of the $\rho^{\mathfrak{n}}$ together, their empirical distribution should have low variance and be centred on p. More precisely, for large $|\mathcal{D}|$ each $\rho^{\mathfrak{n}}$ should be close in distribution to i.i.d. Gaussian with mean p and variance of order $|\mathcal{D}|^{-1}$, a fact that can be derived simply from the central limit theorem applied to i.i.d. Bernoulli random variables. Similarly, assumption 5 implies that one can exchange data points and neurons in the previous discussion and so observe proportions $\bar{\rho}^{\boldsymbol{x}}$ for each $\boldsymbol{x} \in \mathcal{D}$, which again should have an empirical distribution centred on p and with low variance. The value of p is not prescribed by any of our assumptions and nor is it important, all that matters is that the distributions of $\{\rho^{\mathfrak{n}}\}_{\mathfrak{n} \in \mathcal{N}}$ and $\{\bar{\rho}^{\boldsymbol{x}}\}_{\boldsymbol{x} \in \mathcal{D}}$ are strongly peaked around some common mean.

We will now generalise the previous discussion to the case of any number of linear pieces of the activation function. Suppose that the activation function is piece-wise linear in L pieces and denote by $I_1, \ldots, I_L$ the disjoint intervals on which the activation function is linear; $\{I_i\}_{i=1}^{L}$ partition $\mathbb{R}$. Let $\iota(\boldsymbol{x}, \mathfrak{n})$ be defined so that the pre-activation to neuron $\mathfrak{n} \in \mathcal{N}$ when evaluating at $\boldsymbol{x} \in \mathcal{D}$ lies in $I_{\iota(\boldsymbol{x},\mathfrak{n})}$. We consider two scenarios, *data averaging* and *neuron averaging*. Under data averaging, we fix a neuron and observe the pre-activations observed over all $\mathcal{D}$, i.e. define for $j = 1, \ldots, L$ the counts

$$\chi_j^{\mathfrak{n}} = |\{\boldsymbol{x} \in \mathcal{D} \ : \ \iota(\boldsymbol{x}, \mathfrak{n}) = j\}| \tag{3.44}$$

and thence the $L - 1$ independent ratios

$$\rho_j^{\mathfrak{n}} = \frac{\chi_j^{\mathfrak{n}}}{\sum_{i=1}^{L} \chi_1^{\mathfrak{n}}} \tag{3.45}$$

for $j = 2, \ldots, L$. Similarly, in neuron averaging we define

$$\bar{\chi}_j^{\boldsymbol{x}} = |\{\mathfrak{n} \in \mathcal{N} \ : \ \iota(\boldsymbol{x}, \mathfrak{n}) = j\}|, \tag{3.46}$$

$$\bar{\rho}_j^{\boldsymbol{x}} = \frac{\bar{\chi}_j^{\boldsymbol{x}}}{\sum_{i=1}^{L} \bar{\chi}_1^{\boldsymbol{x}}}. \tag{3.47}$$

We thus have the sets of observed real quantities

$$R_j = \{\rho_j^{\mathfrak{n}} \ : \ \mathfrak{n} \in \mathcal{N}\}, \tag{3.48}$$

$$\bar{R}_j = \{\bar{\rho}_j^{\boldsymbol{x}} \ : \ \boldsymbol{x} \in \mathcal{D}\}. \tag{3.49}$$

$$\tag{3.50}$$

Under assumption 5, the empirical variance of the values in R_j and $\bar{R}_j$ should be small. We run experiments to interrogate this hypothesis under a variety of conditions. In particular:

1. Standard Gaussian i.i.d. data vs. 'real' data (MNIST digits [LC10]).
2. Multi-layer perceptron (MLP) vs. convolutional (CNN) architecture.
3. Trained vs. randomly initialised weights.
4. Various piece-wise linear activation functions.

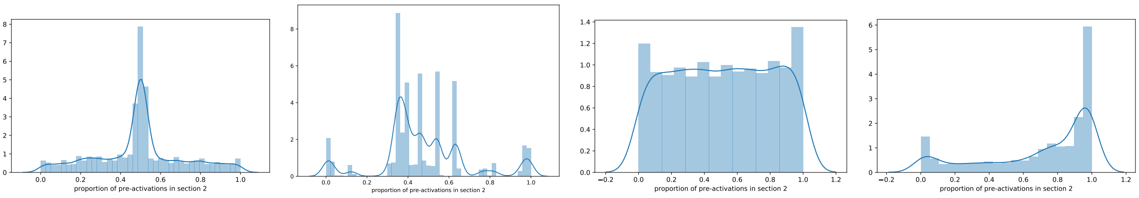
In particular:

1. We generate 10000 i.i.d. Gaussian data vectors of length 784 (to match the size of MNIST digits).
2. We fix a MLP architecture of 5 layers and a CNN architecture with 3 convolutional layers and 2 fully-connected. The exact architecture details are given in the Appendix.
3. We train all networks to test accuracy of at least 97% and use dropout with rate 0.1 during training.
4. We test `ReLU` (2 pieces), `HardTanh` (3 pieces) and a custom 5 piece function. Full details are given in Appendix A.2.

To examine the R_j and $\bar{R}_j$, we produce histograms of R_2 for $L=2$ (i.e. `ReLU`), joint density plots of (R_2, R_3) for $L=3$ (i.e. `HardTanh`) and pair-plots of (R_2, R_3, R_4, R_5) for $L=5$. We are presently only interested in the size of the variance shown, but these full distribution plots are included in-case any further interesting observations can be made in the future. Figures 3.1-3.4 show the results for `ReLU` activations and Figures 3.5-3.8 show the results for `HardTanh`. The qualitative trends are much the same for all three activation functions, but the plots for the 5-piece function are very large and so are relegated to the supplementary material[6]. We make the following observations:

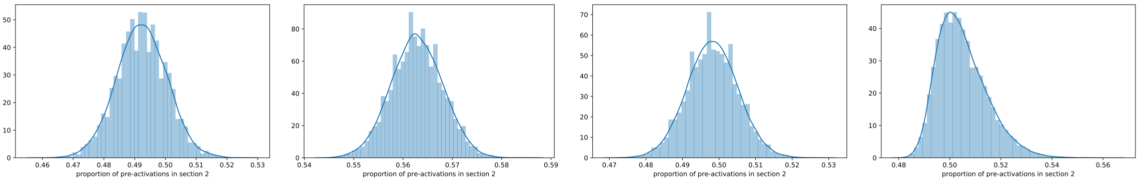
1. The variance of $\bar{R}_2$ is 'small' in all cases for `ReLU` networks except when evaluating MNIST-trained MLP networks on i.i.d. random normal data. This is the least relevant case practically.
2. For R_2, the results are much less convincing, though we do note that, with random weights and i.i.d. data, the MLP network does have quite a strongly peaked distribution. In other cases the variance is undeniably large.
3. The variance of $\bar{R}_{2,3}$ is 'small' in all cases for `HardTanh` except when evaluating LeNet architectures on MNIST data.
4. For R_3 in `HardTanh` networks, the variance seems to be low when the weights are random, but not when trained.

[6] `https://github.com/npbaskerville/loss-surfaces-general-activation-functions/blob/master/Loss_surfaces_of_neural_networks_with_general_activation_functions___supplimentary.pdf`



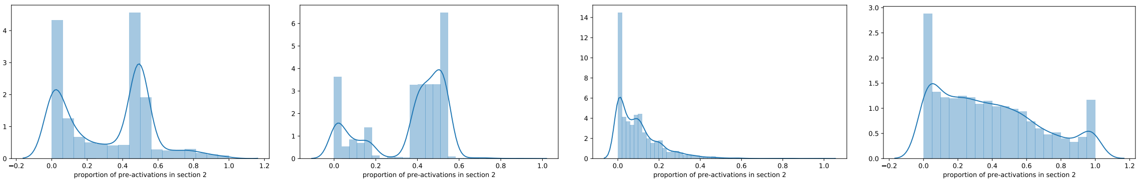
(a) MLP, i.i.d. data. (b) LeNet, i.i.d. data. (c) MLP, MNIST data. (d) LeNet, MNIST data.

Figure 3.1: Experimental distribution of R_2 (data averaging; each sample is a single neuron) for random MLP and LeNet `ReLU` networks, and i.i.d. normal and MNIST data. The blue line is a kernel density estimation fit.



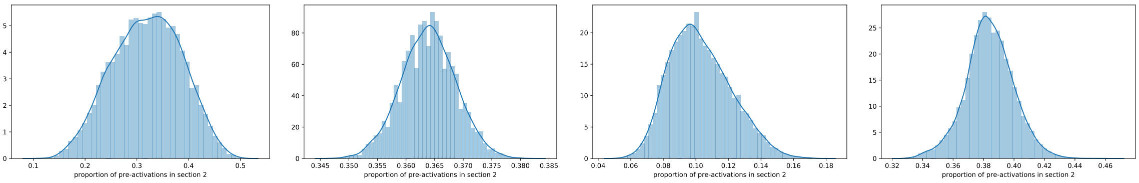
(a) MLP, i.i.d. data. (b) LeNet, i.i.d. data. (c) MLP, MNIST data. (d) LeNet, MNIST data.

Figure 3.2: Experimental distribution of $\bar{R}_2$ (neuron averaging; each sample is a single datum) for random MLP and LeNet `ReLU` networks, and i.i.d. normal and MNIST data. The blue line is a kernel density estimation fit.



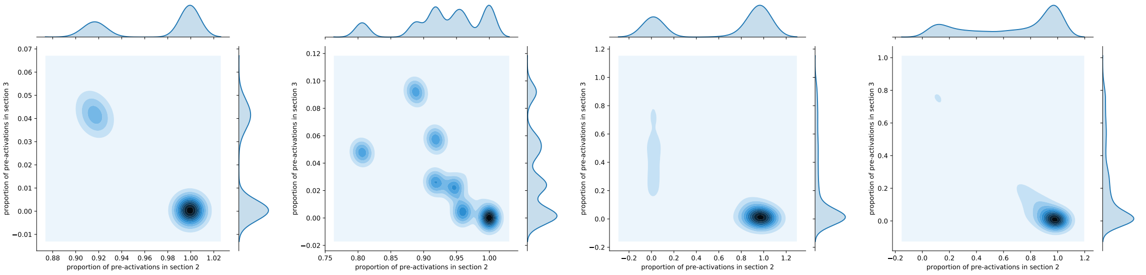
(a) MLP, i.i.d. data. (b) LeNet, i.i.d. data. (c) MLP, MNIST data. (d) LeNet, MNIST data.

Figure 3.3: Experimental distribution of R_2 (data averaging; each sample is a single neuron) for MLP and LeNet `ReLU` networks trained to high validation accuracy on MNIST, and evaluated on i.i.d. normal and MNIST data. The blue line is a kernel density estimation fit.



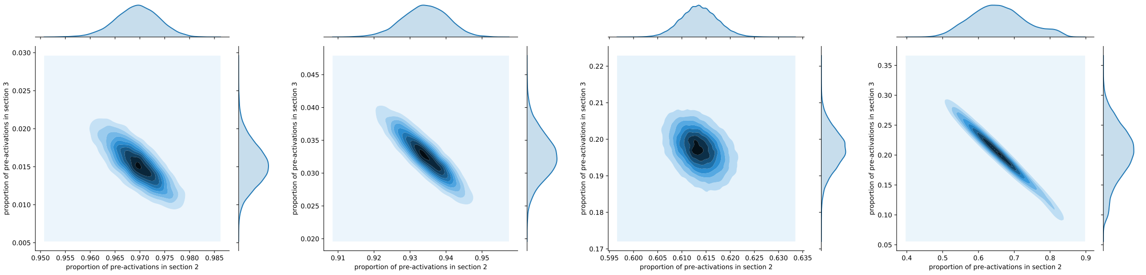
(a) MLP, i.i.d. data. (b) LeNet, i.i.d. data. (c) MLP, MNIST data. (d) LeNet, MNIST data.

Figure 3.4: Experimental distribution of $\bar{R}_2$ (neuron averaging; each sample is a single datum) for MLP and LeNet `ReLU` networks trained to high validation accuracy on MNIST, and evaluated on i.i.d. normal and MNIST data. The blue line is a kernel density estimation fit.



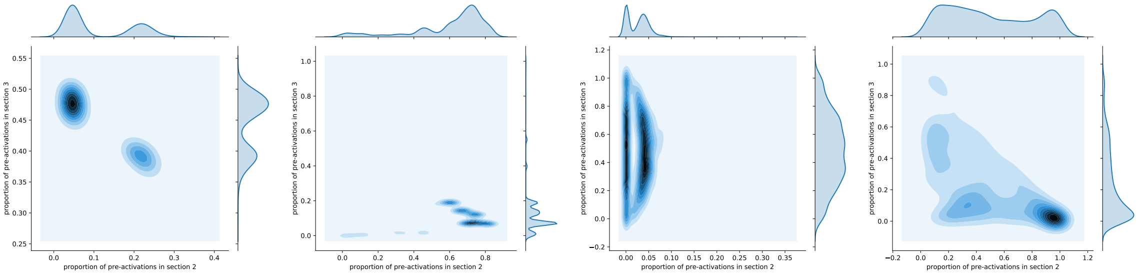
(a) MLP, i.i.d. data. (b) LeNet, i.i.d. data. (c) MLP, MNIST data. (d) LeNet, MNIST data.

Figure 3.5: Experimental distribution of (R_2, R_3) (data averaging; each sample is a single neuron) for random MLP and LeNet HardTanh networks, and i.i.d. normal and MNIST data. The plots show 2d kernel density estimation fits of the joint and 1d fits of the marginals.



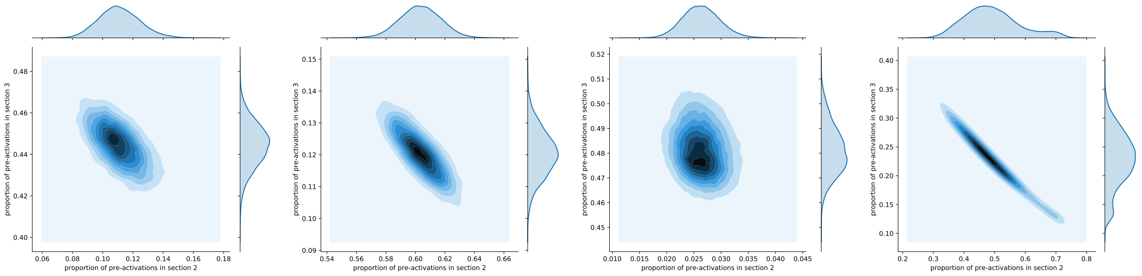
(a) MLP, i.i.d. data. (b) LeNet, i.i.d. data. (c) MLP, MNIST data. (d) LeNet, MNIST data.

Figure 3.6: Experimental distribution of $(\bar{R}_2, \bar{R}_3)$ (neuron averaging; each sample is a single datum) for random HardTanh MLP and LeNet networks, and i.i.d. normal and MNIST data. The plots show 2d kernel density estimation fits of the joint and 1d fits of the marginals.



(a) MLP, i.i.d. data. (b) LeNet, i.i.d. data. (c) MLP, MNIST data. (d) LeNet, MNIST data.

Figure 3.7: Experimental distribution of (R_2, R_3) (data averaging; each sample is a single neuron) for MLP and LeNet HardTanh networks trained to high validation accuracy on MNIST, and evaluated on i.i.d. normal and MNIST data. The plots show 2d kernel density estimation fits of the joint and 1d fits of the marginals.



(a) MLP, i.i.d. data. (b) LeNet, i.i.d. data. (c) MLP, MNIST data. (d) LeNet, MNIST data.

Figure 3.8: Experimental distribution of $(\bar{R}_2, \bar{R}_3)$ (neuron averaging; each sample is a single datum) for MLP and LeNet `HardTanh` networks trained to high validation accuracy on MNIST, and evaluated on i.i.d. normal and MNIST data. The plots show 2d kernel density estimation fits of the joint and 1d fits of the marginals.

Overall, we see that in some circumstances, particularly with un-trained weights, the assumption 5 is not as unreasonable as it first sounds. More importantly for the present work, comparing the three examined activation functions supports the hypothesis that, insofar as modeling the action of the `ReLU` activation function by independent Bernoulli random variables was valid in [Cho+15], our analogous modelling of the action of general piece-wise linear functions by independent discrete random variables is also valid. Put another way, it does not appear that the assumptions we make here are any stronger than those made in [Cho+15]. We finally note an interesting comparison between, for example, Figures 3.2a and 3.2c, or equally Figures 3.6a and 3.6c. In both cases, the variance is low for both distributions, and the only difference between the two experiments is the evaluation data, being i.i.d. Gaussian in the one case, and MNIST in the other. These results seem to demonstrate that the assumption of i.i.d. Gaussian data distribution is not trivialising the problem as one might expect a priori.

Taking all of the results of this section together, we see that the case for our extension of [Cho+15] is quite strong, but there are clearly realistic cases where the modelling assumptions applied to activation functions in [Cho+15] are convincingly violated.

3.3 Statement of results

We shall use *complexity* to refer to any of the following defined quantities which we define precisely as they appear in [AAC13].

Definition 3.2. For a Borel set $B \subset \mathbb{R}$ and non-negative integer k, let

$$C^{g}_{N,k}(B) = \left|\left\{\boldsymbol{w} \in \sqrt{N}S^{N-1} \,:\, \nabla g(\boldsymbol{w}) = 0, g(\boldsymbol{w}) \in B,\ i(\nabla^2 g) = k\right\}\right| \tag{3.51}$$

where $i(M)$ for a square matrix M is the *index* of M, i.e. the number of negative eigenvalues of M. We also define the useful generalisation $i_{\leqslant x}(M)$ to be the number of eigenvalues of M less than x, so $i_{\leqslant 0}(M) = i(M)$.

Definition 3.3. For a Borel set $B \subset \mathbb{R}$, let

$$C_N^g(B) = \left|\left\{\boldsymbol{w} \in \sqrt{N}S^{N-1} \;:\; \nabla g(\boldsymbol{w}) = 0, g(\boldsymbol{w}) \in B\right\}\right|. \tag{3.52}$$

We now state our main identities, which we find simpler to prove by scaling $\boldsymbol{w}$ to lie on the hyper-sphere of unit radius: $h(\boldsymbol{w}) := N^{-H/2}g(\sqrt{N}\boldsymbol{w})$. For convenience, we define

$$\rho_\ell^{(N)} = \rho_\ell N^{-\ell/2} \tag{3.53}$$

so that, recalling the form of g in (3.43), we obtain

$$h(\boldsymbol{w}) = \sum_{i_1,\dots,i_H=1}^{\Lambda} X_{i_1,\dots,i_H} \prod_{k=1}^{H} w_{i_k} + \sum_{\ell=1}^{H} \rho_\ell^{(N)} \sum_{i_{\ell+1},\dots,i_H=1}^{\Lambda} \prod_{k=\ell+1}^{H} w_{i_k}. \tag{3.54}$$

Though the complexities have been defined using general Borel sets, as in [AAC13], we focus on half-infinite intervals $(-\infty, u)$, acknowledging that everything that follows could be repeated instead with general Borel sets *mutatis mutandis*. We will henceforth be studying the following central quantities (note the minor abuse of notation):

$$C_{N,k}^h(\sqrt{N}u) = \left|\left\{\boldsymbol{w} \in S^{N-1} \;:\; \nabla h(\boldsymbol{w}) = 0, h(\boldsymbol{w}) \in \sqrt{N}u,\; i(\nabla^2 h) = k\right\}\right|, \tag{3.55}$$

$$C_N^h(\sqrt{N}u) = \left|\left\{\boldsymbol{w} \in S^{N-1} \;:\; \nabla h(\boldsymbol{w}) = 0, h(\boldsymbol{w}) \in \sqrt{N}u\right\}\right| \tag{3.56}$$

and it will be useful to define a relaxed version of (3.55) for $\mathcal{K} \subset \{0, 1, \dots, N\}$:

$$C_{N,\mathcal{K}}^h(\sqrt{N}u) = \left|\left\{\boldsymbol{w} \in S^{N-1} \;:\; \nabla h(\boldsymbol{w}) = 0, h(\boldsymbol{w}) \in \sqrt{N}u,\; i(\nabla^2 h) \in \mathcal{K}\right\}\right|. \tag{3.57}$$

Our main results take the form of two theorems that extend Theorems 2.5 and 2.8 from [AAC13] to our more general spin glass like object g, and a third theorem with partially extends Theorem 2.17 of [AAC13]. In the case of Theorem 2.8, we are able to obtain exactly the same result in this generalised setting. For Theorem 2.5, we have been unable to avoid slackening the result slightly, hence the introduction of the quantity $C_{N,\mathcal{K}}^h$ above. In the case of Theorem 2.17, we are only able to perform the calculations of the exact leading order term in one case and obtain a term very similar to that in [AAC13] but with an extra factor dependent on the piece-wise linear approximation to the generalised activation function. This exact term correctly falls-back to the term found in [AAC13] when we take $f = \texttt{ReLU}$.

Theorem 3.2. *Recall the definition of C_N^h in (3.56) and let Θ_H be defined as in [AAC13]:*

$$\Theta_H(u) = \begin{cases} \frac{1}{2}\log(H-1) - \frac{H-2}{4(H-1)}u^2 - I_1(u; E_\infty) & \text{if } u \leqslant -E_\infty, \\ \frac{1}{2}\log(H-1) - \frac{H-2}{4(H-1)}u^2 & \text{if } -E_\infty \leqslant u \leqslant 0, \\ \frac{1}{2}\log(H-1) & \text{if} 0 \geqslant u, \end{cases} \tag{3.58}$$

where $E_\infty = 2\sqrt{\frac{H-1}{H}}$, *and* $I_1(\cdot;E)$ *is defined on* $(-\infty,-E]$ *as in [AAC13] by*

$$I_1(u;E) = \frac{2}{E^2}\int_u^{-E}(z^2-E^2)^{1/2}dz = -\frac{u}{E^2}\sqrt{u^2-E^2} - \log\left(-u+\sqrt{u^2-E^2}\right)+\log E, \tag{3.59}$$

then

$$\lim_{N\to\infty}\frac{1}{N}\log\mathbb{E}C_N^h(\sqrt{N}u) = \Theta_H(u). \tag{3.60}$$

Theorem 3.3. *Recall the definition of* $C_{N,\mathcal{K}}^h$ *in (3.57) and let* $\Theta_{H,k}$ *be defined as in [AAC13]:*

$$\Theta_{H,k}(u) = \begin{cases} \frac{1}{2}\log(H-1) - \frac{H-2}{4(H-1)}u^2 - (k+1)I_1(u;E_\infty) & \text{if } u \leqslant -E_\infty, \\ \frac{1}{2}\log(H-1) - \frac{H-2}{H} & \text{if } u > -E_\infty, \end{cases} \tag{3.61}$$

then, with $\mathcal{K} = \{k-1,k,k+1\}$ *for* $k>0$,

$$\Theta_{H,k+1}(u) \leqslant \lim_{N\to\infty}\frac{1}{N}\log\mathbb{E}C_{N,\mathcal{K}}^h(\sqrt{N}u) \leqslant \Theta_{H,k-1}(u) \tag{3.62}$$

and similarly with $\mathcal{K} = \{0,1\}$

$$\Theta_{H,1}(u) \leqslant \lim_{N\to\infty}\frac{1}{N}\log\mathbb{E}C_{N,\mathcal{K}}^h(\sqrt{N}u) \leqslant \Theta_{H,0}(u). \tag{3.63}$$

Remark 3.4. Note that Theorem 3.3 holds for `ReLU` networks (equivalently, pure multi-spin glass models), as indeed it must. It can be seen as an immediate (weaker) consequence of the Theorem 2.5 in [AAC13] of which it is an analogue in our more general setting.

Theorem 3.4. *Let* $u < -E_\infty$ *and define* $v = -\frac{\sqrt{2}u}{E_\infty}$. *Define the function h by (c.f. (7.10) in [AAC13])*

$$h(v) = \left(\frac{|v-\sqrt{2}|}{|v+\sqrt{2}|}\right)^{1/4} + \left(\frac{|v+\sqrt{2}|}{|v-\sqrt{2}|}\right)^{1/4}, \tag{3.64}$$

and the functions

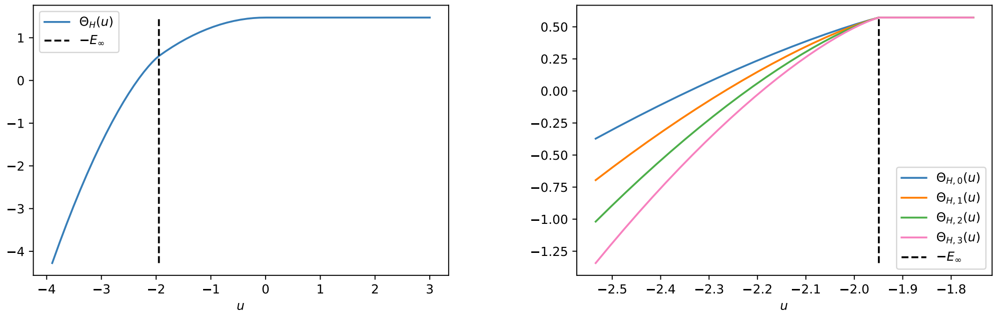
$$q(\theta') = \frac{1}{2}\sin^2 2\theta' + \frac{1}{4}\left(3+4\cos 4\theta'\right), \tag{3.65}$$

$$j(x,s_1,\theta') = 1 + \frac{1}{2}s_1\sqrt{x^2-2}h(x)^2 - s_1^2 q(\theta')|x^2-2|h(x)^2, \tag{3.66}$$

$$T(v,s_1) = \frac{2}{\pi}\int_0^{\pi/2} j(-v,s_1,\theta')d\theta'. \tag{3.67}$$

The $N-1\times N-1$ *deterministic matrix S is defined subsequently around (3.88). S has fixed rank* $r=2$ *and non-zero eigenvalues* $\{s_1, N^{-1/2}s_2\}$ *where* $s_j = \mathcal{O}(1)$. *The specific form of S is rather cumbersome and uninformative and so is relegated to Appendix A.1, and the vector* $\boldsymbol{v}$ *is defined in Lemma 3.4. Then we have*

$$\mathbb{E}C_N^h(\sqrt{N}u) \sim \frac{N^{-\frac{1}{2}}}{\sqrt{2\pi H}}e^{-\frac{v^2}{2H}}T(v,s_1)h(v)e^{N\Theta_H(u)}\frac{e^{I_1(u;E_\infty)-\frac{1}{2}uI_1'(u;E_\infty)}}{\frac{H-2}{2(H-1)}u + I_1'(u;E_\infty)}. \tag{3.68}$$



(a) Plot of Θ_H.

(b) Plot of $\Theta_{H,k}$ $k = 0, 1, 2, 3$.

Figure 3.9: Plots of the functions Θ_H and $\Theta_{H,k}$ for $H = 20$.

We include in Figures 3.9a and 3.9b plots of the functions Θ_H and $\Theta_{H,k}$ for completeness, though these figures are precisely the same as those appearing in [Cho+15; AAC13]. The critical observation from these plots is that each of the $\Theta_{H,k}$ and Θ_H are monotonically increasing and that there exist unique $E_0 > E_1 > \ldots > E_\infty$ such that $\Theta_{H,k}(-E_k) = 0$ and so the critical values $-E_k$ are the boundaries between regions of exponentially many and ‘exponentially few’ critical points of each respective index.

Remark 3.5. It is interesting to compare the expression (3.54) to the analogous expression for the model of [Ros+19]. In that work, when scaled to the unit hypersphere and scaled so that the spin glass term is composed of $\mathcal{O}(1)$ terms, the scale of the deterministic term is $\mathcal{O}(N^{1/2})$, while the corresponding scale in (3.54) is $\mathcal{O}(N^{-1/2})$. Based on this, one might well *conjecture* Theorem 3.2 and Theorem 3.3, however one would have no means by which to conjecture Theorem 3.4, and as far we can see no means to *prove* Theorem 3.2 and Theorem 3.3. As mentioned in the introduction, the single fixed distinguished direction in [Ros+19] is quite a special feature and is not present in (3.54).

3.4 GOE expressions for the complexity from Kac-Rice formulae

In this section we conduct analysis similar to that in [AAC13; FW07; Fyo04] to obtain expressions for the the expected number of critical points of the function h as defined in (3.54). We start with an elementary lemma deriving the 2-point covariance function for h.

Lemma 3.3. *For $\boldsymbol{w} \in S^{N-1}$, h is defined as in (3.54):*

$$h(\boldsymbol{w}) = \sum_{i_1,\ldots,i_H=1}^{\Lambda} X_{i_1,\ldots,i_H} \prod_{k=1}^{H} w_{i_k} + \sum_{\ell=1}^{H} \rho_\ell^{(N)} \sum_{i_{\ell+1},\ldots,i_H=1}^{\Lambda} \prod_{k=\ell+1}^{H} w_{i_k}, \quad X_{i_1,\ldots,i_H} \overset{i.i.d.}{\sim} \mathcal{N}(0,1).$$

For any $\boldsymbol{w}, \boldsymbol{w}' \in S^{N-1}$ the following holds

$$Cov(h(\boldsymbol{w}), h(\boldsymbol{w}')) = (\boldsymbol{w} \cdot \boldsymbol{w}')^H. \tag{3.69}$$

Proof. Let us begin by writing

$$h(\boldsymbol{w}) = \sum_{i_1,\dots,i_H=1}^{N} X_{i_1,\dots,i_H} \prod_{k=1}^{H} w_{i_k} + h^{(2)}(\boldsymbol{w}) \equiv h^{(1)}(\boldsymbol{w}) + h^{(2)}(\boldsymbol{w}) \tag{3.70}$$

where $h^{(2)}$ is deterministic. Then we have

$$\begin{aligned} Cov(h(\boldsymbol{w}), h(\boldsymbol{w}')) &\equiv \mathbb{E}\left[h(\boldsymbol{w})h(\boldsymbol{w}')\right] - \mathbb{E}h(\boldsymbol{w})\mathbb{E}h(\boldsymbol{w}') \\ &= \mathbb{E}\left[h^{(1)}(\boldsymbol{w})h^{(1)}(\boldsymbol{w}') - h^{(1)}(\boldsymbol{w})h^{(2)}(\boldsymbol{w}') - h^{(2)}(\boldsymbol{w})h^{(1)}(\boldsymbol{w}') + h^{(2)}(\boldsymbol{w})h^{(2)}(\boldsymbol{w}')\right] \\ &\quad - h^{(2)}(\boldsymbol{w})h^{(2)}(\boldsymbol{w}') \\ &= \mathbb{E}\left[h^{(1)}(\boldsymbol{w})h^{(1)}(\boldsymbol{w}')\right] \\ &= \sum_{i_1,\dots i_H=1}^{N} \prod_{k=1}^{H} w_{i_k} w'_{i_k} \\ &= \prod_{k=1}^{H} \sum_{i_k=1}^{N} w_{i_k} w'_{i_k} \\ &= (\boldsymbol{w}\cdot\boldsymbol{w}')^H \end{aligned} \tag{3.71}$$

where we have used $\mathbb{E}h^{(1)} = 0$ in going from the first to the second and the second to the third lines. ■

The following lemma calculates the full joint and thence conditional distribution of h and its first and second derivatives. The calculations follow closely those of [AAC13] and the results are required for later use in a Kac-Rice formula.

Lemma 3.4. *Pick some Cartesian coordinates on S^{N-1} and let $\boldsymbol{w}$ be the north-pole of the sphere $\boldsymbol{w} = (1,0,0,\dots)$. Let $h_i = \partial_i h(\boldsymbol{w})$ and $h_{ij} = \partial_i\partial_j h(\boldsymbol{w})$ where $\{\partial_i\}_{i=1}^{N-1}$ are the coordinate basis around $\boldsymbol{w}$ on the sphere. Then the following results hold.*

(a) For all $1 \leqslant i,j,k < N$, $h(\boldsymbol{w}), h_i(\boldsymbol{w}), h_{jk}(\boldsymbol{w})$ are Gaussian random variables whose distributions are

given by

$$\mathbb{E}[h(\boldsymbol{w})] = \sum_{\ell=1}^{H} \rho_\ell^{(N)} \tag{3.72}$$

$$Var[h(\boldsymbol{w})] = 1 \tag{3.73}$$

$$\mathbb{E}h_i(\boldsymbol{w}) = \sum_{\ell=1}^{H-1} \rho_\ell^{(N)} \left[(H-\ell) + (H-\ell-1)\delta_{i1}\right] \equiv v_i \tag{3.74}$$

$$\mathbb{E}[h_{ij}(\boldsymbol{w})] = \sum_{\ell=1}^{H-2} \rho_\ell^{(N)} \left\{ [(H-\ell)(H-\ell-1)+1]\delta_{i1}\delta_{j1} + (H-\ell-2)(\delta_{i1}+\delta_{j1}) + 1 \right\} \tag{3.75}$$

$$Cov(h(\boldsymbol{w}), h_i(\boldsymbol{w})) = 0 \tag{3.76}$$

$$Cov(h_i(\boldsymbol{w}), h_{jk}(\boldsymbol{w})) = 0 \tag{3.77}$$

$$Cov(h_i(\boldsymbol{w}), h_j(\boldsymbol{w})) = H\delta_{ij} \tag{3.78}$$

$$Cov(h(\boldsymbol{w}), h_{ij}(\boldsymbol{w})) = -H\delta_{ij} \tag{3.79}$$

$$Cov(h_{ij}(\boldsymbol{w}), h_{kl}(\boldsymbol{w})) = H(H-1)(\delta_{ik}\delta jl + \delta_{il}\delta_{kl}) + H^2\delta_{ij}\delta_{kl}. \tag{3.80}$$

To reiterate, note that we define the vector $\boldsymbol{v}$ *in (3.74) as*

$$v_i = \sum_{\ell=1}^{H-1} \rho_\ell^{(N)} \left[(H-\ell) + (H-\ell-1)\delta_{i1}\right].$$

(b) Make the following definitions:

$$\xi_0 = \sum_{\ell=1}^{H} \rho_\ell^{(N)} \tag{3.81}$$

$$\xi_1 = \sum_{\ell=1}^{H-2} \rho_\ell^{(N)} \left[(H-\ell)(H-\ell-1)+1\right] \tag{3.82}$$

$$\xi_2 = \sum_{\ell=1}^{H-2} \rho_\ell^{(N)} (H-\ell-2) \tag{3.83}$$

$$\xi_3 = \sum_{\ell=1}^{H-2} \rho_\ell^{(N)} \tag{3.84}$$

Then, conditional on $h(\boldsymbol{w}) = x$*, for* $x \in \mathbb{R}$*, the random variables* $h_{ij}(\boldsymbol{w})$ *are independent Gaussians satisfying*

$$\mathbb{E}[h_{ij}(\boldsymbol{w}) \mid h(\boldsymbol{w}) = x] = \xi_3 + \xi_2(\delta_{i1}+\delta_{j1}) + \xi_1\delta_{i1}\delta_{j1} - (x-\xi_0)\delta_{ij} \tag{3.85}$$

$$Var[h_{ij}(\boldsymbol{w}) \mid h(\boldsymbol{w}) = x] = H(H-1)(1+\delta_{ij}). \tag{3.86}$$

Or, equivalently,

$$\left(h_{ij}(\boldsymbol{w}) \mid h(\boldsymbol{w}) = x\right) \sim \sqrt{2(N-1)H(H-1)}\left(M^{N-1} - \frac{1}{\sqrt{2(N-1)H(H-1)}} H(x-\xi_0)\, I + S\right) \tag{3.87}$$

where $M^{N-1} \sim GOE^{N-1}$ and the matrix S is given by

$$S_{ij} = \frac{1}{\sqrt{2(N-1)H(H-1)}}\left(\xi_3 + \xi_2(\delta_{i1} + \delta_{j1}) + \xi_1\delta_{i1}\delta_{j1}\right). \tag{3.88}$$

Clearly all entries of S are of order N^{-1}, recalling the scale of $\rho_\ell^{(N)}$ given in (3.53). Moreover, S is of rank 2 and has eigenvalues $\{s_1, N^{-1/2}s_2\}$ for real $s_i = \mathcal{O}(1)$.

Proof. (a) Becuase the $X_{i_1,\dots,i_H}$ are centred Gaussians and $\boldsymbol{w} = (1,0,0,\dots,0)$, we immediately obtain (3.72). (3.74)-(3.75) can be seen to be true similarly, e.g. (3.75) by observing that the stochastic term is again zeroed-out by taking the expectation and the only terms that survive in the non-stochastic part are of the form

$$\frac{\partial^2}{\partial w_i \partial w_j} w_i w_j w_1^{H-\ell-2} \ (i,j \neq 1), \quad \frac{\partial^2}{\partial w_i \partial w_1} w_i w_1^{H-\ell-1} \ (i \neq 1), \quad \frac{\partial^2}{\partial w_1^2} w_1^{H-\ell}. \tag{3.89}$$

The remaining results (3.73), (3.76)-(3.80) all match those in Lemma 3.2 of [AAC13] and follow similarly from Lemma 3.3 and the following ([AT09]):

$$Cov\left(\frac{\partial^k \bar{h}(x)}{\partial x_{i_1}\dots\partial x_{i_k}}, \frac{\partial^l \bar{h}(y)}{\partial y_{j_1}\dots\partial y_{j_l}}\right) = \frac{\partial^{k+l} Cov(\bar{h}(x), \bar{h}(y))}{\partial x_{i_1}\dots\partial x_{i_k}\partial y_{j_1}\dots\partial y_{j_l}} \tag{3.90}$$

where $\bar{h} := h \circ \Phi^{-1}$ and Φ is a coordinate chart around $\boldsymbol{w}$.

(b) (3.85), (3.86) and the conditional independence result follow from (3.72), (3.73), (3.75), (3.80) and the standard result for the conditional distribution of one Gaussian under another (see e.g. [And62] Section 2.5), just as in the proof of Lemma 3.2 in [AAC13].

To show (3.87), recall that a GOE^N matrix is a real symmetric random matrix M and whose entries are independent centred Gaussians with with

$$\mathbb{E}M_{ij}^2 = \frac{1+\delta_{ij}}{2N}. \tag{3.91}$$

Finally we have to determine the eigenvalues of S. With $a = \xi_1 + 2\xi_2 + \xi_3, b = \xi_2 + \xi_3$ and $c = \xi_3$, S has entries

$$S = \frac{1}{\sqrt{2(N-1)H(H-1)}}\begin{pmatrix} a & b & b & \dots & b \\ b & c & c & \dots & c \\ b & c & c & \dots & c \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ b & c & c & \dots & c \end{pmatrix}, \tag{3.92}$$

and so has non-null eigenvectors $(1,u,u,\dots,u)^T$ with eigenvalues $(2(N-1)H(H-1))^{-1/2}\lambda$, where (after some simple manipulation)

$$\lambda^2 - (a - c(N-1))\lambda + ca(N-1) - b^2(N-1) = 0, \qquad u = \frac{\lambda - a}{(N-1)b}. \tag{3.93}$$

Recalling the scale of $\rho_\ell^{(N)} = \mathcal{O}(N^{-\ell/2})$ in (3.53) and the definitions ξ_1, ξ_2, ξ_3, we see that $a, b, c = \mathcal{O}(N^{-1/2})$ and so one easily obtains two solutions for λ, one of order $N^{1/2}$ and another of order $N^{-1/2}$, hence S has two non-zero eigenvalues of order 1 and $N^{-1/2}$.

■

Our next lemma establishes for use in this context a Kac-Rice fomula that will provide the first step in the computation of C_N^h and $C_{N,\mathcal{K}}^h$.

Lemma 3.5. *Let $\hat{F}$ be a real-valued centred Gaussian field on S^{N-1} that is almost surely (a.s.) C^2, $\tilde{F}$ be some non-random, real-valued C^2 function on S^{N-1} and let $F := \hat{F} + \tilde{F}$. Let $\mathcal{A} = \{U_\alpha, \Phi_\alpha\}_{\alpha\in I}$ be a finite atlas on S^{N-1}. Let $h^\alpha = h \circ \Phi_\alpha^{-1}$, and let $h_i^\alpha, h_{ij}^\alpha$ denote derivatives of h in the coordinate basis of the chart (U_α, Φ_α). Assume that the joint distribution $(F_i^\alpha(\boldsymbol{x}), F_{ij}^\alpha(\boldsymbol{x}))$ is non-degenerate for all α and for all $\boldsymbol{x} \in S^{N-1}$ and that there exist constants $K_\alpha, \beta > 0$ such that*

$$\max_{i,j} \left| Var(\hat{F}_{ij}^\alpha(\boldsymbol{x})) + Var(\hat{F}_{ij}^\alpha(\boldsymbol{y})) - 2Cov(\hat{F}_{ij}^\alpha(\boldsymbol{x}), \hat{F}_{ij}^\alpha(\boldsymbol{y})) \right| \leqslant K_\alpha \left|\log|x-y|\right|^{-1-\beta} \tag{3.94}$$

Then the following holds

$$C_{N,k}^F(B) = \int_{S^{N-1}} p_{\boldsymbol{x}}(0) \mathcal{S}_{N-1}(d\boldsymbol{x}) \mathbb{E}\left[|\det \nabla^2 F(\boldsymbol{x})| \mathbb{1}\left\{ F(\boldsymbol{x}) \in B,\ i(\nabla^2 F(\boldsymbol{x})) = k \right\} \mid \nabla F(\boldsymbol{x}) = 0 \right] \tag{3.95}$$

where $p_{\boldsymbol{x}}$ is the density of ∇F at $\boldsymbol{x}$ and $\mathcal{S}_{N-1}$ is the usual surface measure on S^{N-1}. Similarly,

$$C_N^F(B) = \int_{S^{N-1}} p_{\boldsymbol{x}}(0) \mathcal{S}_{N-1}(d\boldsymbol{x}) \mathbb{E}\left[|\det \nabla^2 F(\boldsymbol{x})| \mathbb{1}\left\{ F(\boldsymbol{x}) \in B \right\} \mid \nabla F(\boldsymbol{x}) = 0 \right] \tag{3.96}$$

The proof of Lemma 3.5 shall rely heavily on the Kac-Rice result Theorem 2.1.

Proof of Lemma 3.5 Following the proofs of Theorem 12.4.1 in [AT09] and Lemma 3.1 in [AAC13], we will apply Theorem 2.1 to the choices

$$\begin{aligned}
&\phi := \nabla F \\
&\psi := (F, \nabla_i \nabla_j F) \\
&A := B \times A_k \equiv B \times \{H \in \mathrm{Sym}_{N-1\times N-1} \mid i(H) = k\} \subset \mathbb{R} \times \mathrm{Sym}_{N-1\times N-1}, \\
&\boldsymbol{u} = 0
\end{aligned} \tag{3.97}$$

Then, if the conditions of Theorem 2.1 hold for these choices, we immediately obtain the result. It remains therefore to check the conditions of Theorem 2.1. Firstly, A is indeed an open subset of of $\mathbb{R} \times \mathrm{Sym}_{N-1\times N-1}$ (in turn, isomorphic to some $\mathbb{R}^K$) as can be easily deduced from the continuity of a matrix's eigenvalues in its entries. Condition (a) follows from the assumption of $\hat{F}$ being a.s. C^2 and $\tilde{F}$ being C^2. Conditions (b)-(f) all follow immediately from the Gaussianity of $\hat{F}$. To establish condition (g), we define $\hat{\omega}(\eta)$ and $\tilde{\omega}(\eta)$ in the obvious way and note that $\tilde{\omega}$ is non-random. Then, because $\tilde{F}$ is continuous, given $\varepsilon > 0$ there exists some $\eta_0 > 0$ such that for all $\eta < \eta_0$, $\tilde{\omega}(\eta) \leqslant \varepsilon$. Let

$\tilde{\omega}_0 := \tilde{\omega}(\eta_0)$ and choose some η_1 such that for all $\eta < \eta_1$, $\tilde{\omega}(\eta) < \tilde{\omega}_0$. We have $\omega(\eta) \leqslant \hat{\omega}(\eta) + \tilde{\omega}(\eta)$ and so for $\eta < \eta_1$

$$\begin{aligned}\mathbb{P}(\omega(\eta) > \varepsilon) &\leqslant \mathbb{P}(\hat{\omega}(\eta) + \tilde{\omega}(\eta) > \varepsilon)\\ &= \mathbb{P}(\hat{\omega}(\eta) > \varepsilon - \tilde{\omega}(\eta))\\ &\leqslant \mathbb{P}(\hat{\omega}(\eta) > \varepsilon - \tilde{\omega}_0) \qquad (3.98)\end{aligned}$$

and we note that $\varepsilon - \tilde{\omega}_0 \geqslant 0$ by construction. $\hat{\omega}$ is the modulus of continuity for a centred Gaussian field and so the condition (g) follows from (3.98) and the assumption (3.94) by the Borell-TIS inequality [AT09], just as in the proof of Corollary 11.2.2 in [AT09]. (3.96) is obtained in precisely the same way but simply dropping the $i(H) = k$ condition. ■

3.5 Asymptotic evaluation of complexity

In this section we conduct an asymptotic analysis of the GOE expressions for the complexity found in the preceding section. We first consider the case of counting critical points without any condition of the signature of the Hessian, which turns out to be easier. We then introduce the exact signature condition on the Hessian and proceed by presenting the necessary modifications to certain parts of our arguments.

3.5.1 Complexity results with no Hessian signature prescription

We need to establish a central lemma, which is a key step towards a generalisation of the results presented in [AAC13] but established by entirely different means, following the supersymmetric calculations of [Noc16]. Before this main lemma, we require a generalisation of a result from [FS02], whose proof is given at the end of the chapter (Section 3.6).

Lemma 3.6. *Given m vectors in $\mathbb{R}^N$ $\boldsymbol{x}_1, \dots, \boldsymbol{x}_m$, denote by $Q(\boldsymbol{x}_1, \dots, \boldsymbol{x}_m)$ the $m \times m$ matrix whose entries are given by $Q_{ij} = \boldsymbol{x}_i^T \boldsymbol{x}_j$. Let F be any function of an $m \times m$ matrix such that the integral*

$$\int_{\mathbb{R}^N} \dots \int_{\mathbb{R}^N} d\boldsymbol{x}_1 \dots d\boldsymbol{x}_m |F(Q)| \qquad (3.99)$$

exists, and let S be a real symmetric $N \times N$ matrix of fixed rank r and with non-zero eigenvalues $\{N^\alpha s_i\}_{i=1}^r$ for some $\alpha < 1/2$. Define the integral

$$\mathcal{J}_{N,m}(F;S) := \int_{\mathbb{R}^N} \dots \int_{\mathbb{R}^N} d\boldsymbol{x}_1 \dots d\boldsymbol{x}_m F(Q) e^{-iN\sum_{i=1}^N \boldsymbol{x}_i^T S \boldsymbol{x}_i}. \qquad (3.100)$$

Then as $N \to \infty$ we have

$$\mathcal{J}_{N,m}(F;S) = (1 + o(1))) \frac{\pi^{\frac{m}{2}\left(N - \frac{m-1}{2}\right)}}{\prod_{k=0}^{m-1} \Gamma\left(\frac{N-k}{2}\right)} \int_{Sym_{\geqslant 0}(m)} d\hat{Q} \left(\det \hat{Q}\right)^{\frac{N-m-1}{2}} F(\hat{Q}) \prod_{i=1}^N \prod_{j=1}^r \left(1 + 2iN^\alpha \hat{Q}_{ii} s_j\right)^{-1/2}. \qquad (3.101)$$

Now we state and prove the main lemma.

Lemma 3.7. *Let S be a rank r $N \times N$ symmetric matrix with non-zero eigenvalues $\{s_j\}_{j=1}^r$, where $r = \mathcal{O}(1)$ and $s_j = \mathcal{O}(1)$, and suppose S has all entries of order $\mathcal{O}(N^{-1})$ in a fixed basis. Let $x < 0$ and let M denote an $N \times N$ GOE matrix with respect to whose law expectations are understood to be taken. Then*

$$\begin{aligned}\mathbb{E}^N_{GOE}|\det(M - xI + S)| = K_N \lim_{\varepsilon \searrow 0} e^{2N(x^2-\varepsilon^2)}\,(1+o(1)) \iiint_0^{\pi/2} d\theta d\theta' d\hat{\theta} \iint_0^{\infty} dp_1 dp_2 \iint_{\Gamma} dr_1 dr_2 \\ J_1(p_1, p_2, \theta'; S, N) J_2(r_1, r_2, p_1, p_2) \cos^2 2\theta \sin 2\theta \sin 2\hat{\theta} \\ \exp\Bigg\{ -N\bigg(2\psi_L^{(+)}(r_1; x; \varepsilon\cos 2\theta \cos 2\hat{\theta}) \\ + 2\psi_U^{(+)}(r_2; x; \varepsilon\cos 2\theta\cos 2\hat{\theta}) \\ + \psi_L^{(-)}(p_1; x; \varepsilon\cos 2\theta') + \psi_U^{(-)}(p_2; x; \varepsilon \cos 2\theta')\bigg)\Bigg\}\end{aligned} \tag{3.102}$$

where

$$J_1(p_1, p_2, \theta'; \{s_j\}_{j=1}^r, N) = \prod_{j=1}^r \left(1 + 2iN^{1/2}s_j(p_1+p_2) - Ns_j^2\left[\sin^2 2\theta'(p_1^2+p_2^2) + \left(3 + 4\cos 4\theta'\right)p_1p_2\right]\right)^{-1/2}, \tag{3.103}$$

$$J_2(r_1, r_2, p_1, p_2; \varepsilon) = (r_1+p_1)(r_2+p_1)(r_1+p_2)(r_2+p_2)|r_1 - r_4|^4|p_1-p_2|(r_1r_2)^{-2}(p_1p_2)^{-3/2} \tag{3.104}$$

and

$$K_N = \frac{N^{N+3}(-i)^N}{\Gamma\left(\frac{N}{2}\right)\Gamma\left(\frac{N-1}{2}\right)\pi^{3/2}} \tag{3.105}$$

and the functions $\psi_L^{\pm}, \psi_U^{(\pm)}$ are given by

$$\psi_L^{(\pm)}(z; x, \varepsilon) = \frac{1}{2}z^2 \pm i(x+i\varepsilon)z - \frac{1}{2}\log z, \tag{3.106}$$

$$\psi_U^{(\pm)}(z; x, \varepsilon) = \frac{1}{2}z^2 \pm i(x-i\varepsilon)z - \frac{1}{2}\log z, \tag{3.107}$$

and Γ is a contour bounded away from zero in $\mathbb{C}$, e.g. that shown in Figure 3.10.

Proof. We begin with the useful expression for real symmetric matrices A [Fyo05; Fyo04]

$$|\det A| = \lim_{\varepsilon\to 0}\frac{\det A \det A}{\sqrt{\det(A - i\varepsilon)}\sqrt{\det(A+i\varepsilon)}} \tag{3.108}$$

where the limit is taken over real ε, and WLOG $\varepsilon > 0$. We're free to deform the matrices in the numerator for the sake of symmetry in the ensuing calculations, so

$$|\det A| = \lim_{\varepsilon\searrow 0}\frac{\det(A-i\varepsilon)\det(A+i\varepsilon)}{\sqrt{\det(A - i\varepsilon)}\sqrt{\det(A+i\varepsilon)}}. \tag{3.109}$$

For convenience of notation we put

$$\Delta_\varepsilon(M;x,S) = \frac{\det(M-xI+S-i\varepsilon)\det(M-xI+S+i\varepsilon)}{\sqrt{\det(M-xI+S-i\varepsilon)}\sqrt{\det(M-xI+S+i\varepsilon)}}. \tag{3.110}$$

Then we express the determinants and half-integer powers of determinants as Gaussian integrals over anti-commuting and commuting variables respectively as in [Noc16] and [FN15]:

$$\begin{aligned}&\Delta_\varepsilon(M;x,S)\\ =&K_N^{(1)}\int d\boldsymbol{x}_1 d\boldsymbol{x}_2 d\zeta_1 d\zeta_1^\dagger d\zeta_2 d\zeta_2^\dagger \exp\left\{-i\boldsymbol{x}_1^T(M-(x+i\varepsilon)I+S)\boldsymbol{x}_1 - i\boldsymbol{x}_2^T(M-(x-i\varepsilon)I+S)\boldsymbol{x}_2\right\}\\ &+\exp\left\{i\zeta_1^\dagger(M-(x+i\varepsilon)I+S)\zeta_1 + i\zeta_2^\dagger(M-(x-i\varepsilon)I+S)\zeta_2\right\}\end{aligned} \tag{3.111}$$

where $K_N^{(1)} = (-i)^N\pi^{-N}$, which follows from standard facts about commuting Gaussian integrals and Berezin integration. The remainder of the calculation is very similar to that presented in [Noc16; FN15] but we present it in full to keep track of the slight differences. Let

$$A = \boldsymbol{x}_1\boldsymbol{x}_1^T + \boldsymbol{x}_2\boldsymbol{x}_2^T + \zeta_1\zeta_1^\dagger + \zeta_2\zeta_2^\dagger \tag{3.112}$$

and note that, by the cyclicity of the trace,

$$\boldsymbol{x}_j^T(M-(x\pm i\varepsilon)I+S)\boldsymbol{x}_j = \mathrm{Tr}\left((M-(x\pm i\varepsilon)I+S)\boldsymbol{x}_j\boldsymbol{x}_j^T\right) \tag{3.113}$$

$$\zeta_j^\dagger(M-(x\pm i\varepsilon)I+S)\zeta_j = -\mathrm{Tr}\left((M-(x\pm i\varepsilon)I+S)\zeta_j\zeta_j^\dagger\right) \tag{3.114}$$

and so we can rewrite (3.111) as

$$\begin{aligned}\Delta_\varepsilon(M;x,S) = K_N^{(1)}\int d\boldsymbol{x}_1 d\boldsymbol{x}_2 d\zeta_1 d\zeta_1^\dagger d\zeta_2 d\zeta_2^\dagger \exp\left\{-i\mathrm{Tr}MA - i\mathrm{Tr}SA + i(x+i\varepsilon)\boldsymbol{x}_1^T\boldsymbol{x}_1 + i(x-i\varepsilon)\boldsymbol{x}_2^T\boldsymbol{x}_2\right\}\\ \exp\left\{-i(x+i\varepsilon)\zeta_1^\dagger\zeta_1 - i(x-i\varepsilon)\zeta_2^\dagger\zeta_2\right\}.\end{aligned} \tag{3.115}$$

We then define the Bosonic and Fermionic matrices

$$Q_B = \begin{pmatrix} \boldsymbol{x}_1^T\boldsymbol{x}_1 & \boldsymbol{x}_1^T\boldsymbol{x}_2 \\ \boldsymbol{x}_2^T\boldsymbol{x}_1 & \boldsymbol{x}_2^T\boldsymbol{x}_2 \end{pmatrix}, \quad Q_F = \begin{pmatrix} \zeta_1^\dagger\zeta_1 & \zeta_1^\dagger\zeta_2 \\ \zeta_2^\dagger\zeta_1 & \zeta_2^\dagger\zeta_2 \end{pmatrix} \tag{3.116}$$

and also $B = \boldsymbol{x}_1\boldsymbol{x}_1^T + \boldsymbol{x}_2\boldsymbol{x}_2^T$. Note that (3.109) is true for all real symmetric matrices A and so for *all* real symmetric M, S and real values x we have

$$\lim_{\varepsilon\searrow 0}\Delta_\varepsilon(M;x,S) = |\det(M-xI+S)| \tag{3.117}$$

and so with respect to the GOE law for M we certainly have

$$\Delta_\varepsilon(M;x,S) \overset{\text{a.s.}}{\to} |\det(M-xI+S)| \quad \text{as } \varepsilon\searrow 0 \tag{3.118}$$

thus meaning that the $\varepsilon\searrow 0$ limit can be exchanged with a GOE expectation over M. We therefore proceed with fixed $\varepsilon > 0$ to compute the GOE expectation of Δ_ε.

We have the standard Gaussian Fourier transform result for matrices:

$$\mathbb{E}_{GOE}^{N} e^{-i\mathrm{Tr}MA} = \exp\left\{-\frac{1}{8N}\mathrm{Tr}(A+A^{T})^{2}\right\} \tag{3.119}$$

and from [Noc16][7]

$$\mathrm{Tr}(A+A^{T})^{2} = 4\mathrm{Tr}Q_{B}^{2} - 2\mathrm{Tr}Q_{F}^{2} + 4\zeta_{1}^{T}\zeta_{2}\zeta_{2}^{\dagger}\zeta_{1}^{*} - 8\zeta_{1}^{\dagger}B\zeta_{1} - 8\zeta_{2}^{\dagger}B\zeta_{2} \tag{3.120}$$

so we can take the GOE average in (3.115) and obtain

$$\begin{aligned}\mathbb{E}_{GOE}^{N}\Delta_{\varepsilon}(M;x,S) = K_{N}^{(1)}\int d\boldsymbol{x}_{1}d\boldsymbol{x}_{2}d\zeta_{1}d\zeta_{1}^{\dagger}d\zeta_{2}d\zeta_{2}^{\dagger}\exp\left\{-\frac{1}{2N}\mathrm{Tr}Q_{B}^{2} - i\mathrm{Tr}SB + ix\mathrm{Tr}Q_{B} + \varepsilon\mathrm{Tr}Q_{B}\sigma\right\}\\ \exp\left\{\frac{1}{4N}\mathrm{Tr}Q_{F}^{2} - \frac{1}{2N}\zeta_{1}^{T}\zeta_{2}\zeta_{2}^{\dagger}\zeta_{1}^{*} + \sum_{j=1}^{2}\zeta_{j}^{\dagger}\left(\frac{B}{N} + iS - i(x + i(-1)^{j-1}\varepsilon)\right)\zeta_{j}\right\}.\end{aligned} \tag{3.121}$$

where we have defined

$$\sigma = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}.$$

We can then use the transformation

$$\exp\left\{\frac{1}{4N}\mathrm{Tr}Q_{F}^{2}\right\} = \frac{N^{2}}{\pi Vol(U(2))}\int d\hat{Q}_{F}\exp\left\{-N\mathrm{Tr}\hat{Q}_{F}^{2} + \mathrm{Tr}Q_{F}\hat{Q}_{F}\right\} \tag{3.122}$$

to obtain

$$\begin{aligned}\mathbb{E}_{GOE}^{N}\Delta_{\varepsilon}(M;x,S) = K_{N}^{(2)}\int d\boldsymbol{x}_{1}d\boldsymbol{x}_{2}d\zeta_{1}d\zeta_{1}^{\dagger}d\zeta_{2}d\zeta_{2}^{\dagger}d\hat{Q}_{F}\exp\left\{-\frac{1}{2N}\mathrm{Tr}Q_{B}^{2} - i\mathrm{Tr}SB + ix\mathrm{Tr}Q_{B} + \varepsilon\mathrm{Tr}Q_{B}\sigma\right\}\\ \exp\left\{-N\mathrm{Tr}\hat{Q}_{F}^{2} + \mathrm{Tr}\hat{Q}_{F}Q_{F} - \frac{1}{2N}\zeta_{1}^{T}\zeta_{2}\zeta_{2}^{\dagger}\zeta_{1}^{*} + \sum_{j=1}^{2}\zeta_{j}^{\dagger}\left(\frac{B}{N} + iS - i(x + i(-1)^{j-1}\varepsilon\right)\zeta_{j}\right\}\end{aligned} \tag{3.123}$$

where $K_{N}^{(2)} = K_{N}^{(1)}\frac{N^{2}}{\pi Vol(U(2))}$. The Fermionic cross-term in (3.123) can be dealt with using (see [Noc16] (4.104))

$$\exp\left(-\frac{1}{2N}\zeta_{1}^{T}\zeta_{2}\zeta_{2}^{\dagger}\zeta_{1}^{*}\right) = \frac{2N}{\pi}\int d^{2}u\exp\left(-2N\bar{u}u - i\left(u\zeta_{1}^{\dagger}\zeta_{2}^{*} + \bar{u}\zeta_{2}^{\dagger}\zeta_{1}\right)\right) \tag{3.124}$$

where $d^{2}u = d\Re u\ d\Im u$, and so we obtain

$$\begin{aligned}\mathbb{E}_{GOE}^{N}\Delta_{\varepsilon}(M;x,S) = K_{N}^{(3)}\int d\boldsymbol{x}_{1}d\boldsymbol{x}_{2}d\zeta_{1}d\zeta_{1}^{\dagger}d\zeta_{2}d\zeta_{2}^{\dagger}d\hat{Q}_{F}d^{2}u\exp\left\{-\frac{1}{2N}\mathrm{Tr}Q_{B}^{2} - i\mathrm{Tr}SB + ix\mathrm{Tr}Q_{B} + \varepsilon\mathrm{Tr}Q_{B}\sigma\right\}\\ \exp\left\{-N\mathrm{Tr}\hat{Q}_{F}^{2} - 2Nu\bar{u}\right\}\\ \exp\left\{\mathrm{Tr}\hat{Q}_{F}Q_{F} - i(u\zeta_{1}^{\dagger}\zeta_{2}^{*} + \bar{u}\zeta_{2}^{T}\zeta_{1}) + \sum_{j=1}^{2}\zeta_{j}^{\dagger}\left(\frac{B}{N} + iS - i(x + i(-1)^{j-1}\varepsilon\right)\zeta_{j}\right\}\end{aligned} \tag{3.125}$$

[7] Note that (4.100) in [Noc16] contains a trivial factor of 4 error that has non-trivial consequences in our calculations.

where $K_N^{(3)} = K_N^{(2)}\frac{2N}{\pi}$. To simplify the Fermionic component of (3.125) and make apparent its form, we introduce $\zeta^T = (\zeta_1^\dagger, \zeta_1^T, \zeta_2^\dagger, \zeta_2^T)$ and then (3.125) reads

$$\begin{aligned}
\mathbb{E}_{GOE}^N \Delta_\varepsilon(M; x, S) = K_N^{(3)} \int & d\boldsymbol{x}_1 d\boldsymbol{x}_2 d\zeta d\hat{Q}_F d^2u \exp\left\{-\frac{1}{2N}\mathrm{Tr}Q_B^2 - i\mathrm{Tr}SB + ix\mathrm{Tr}Q_B + \varepsilon\mathrm{Tr}Q_B\sigma\right\} \\
& \exp\left\{-N\mathrm{Tr}\hat{Q}_F^2 - 2Nu\bar{u}\right\} \\
& \exp\left\{\frac{1}{2}\zeta^T\mathcal{M}\zeta\right\} \\
= K_N^{(3)} \int & d\boldsymbol{x}_1 d\boldsymbol{x}_2 d\hat{Q}_F d^2u \exp\left\{-\frac{1}{2N}\mathrm{Tr}Q_B^2 - i\mathrm{Tr}SB + ix\mathrm{Tr}Q_B + \varepsilon\mathrm{Tr}Q_B\sigma\right\} \\
& \exp\left\{-N\mathrm{Tr}\hat{Q}_F^2 - 2Nu\bar{u}\right\} \\
& \sqrt{\det\mathcal{M}}
\end{aligned} \tag{3.126}$$

where the matrix $\mathcal{M}$ is given by

$$\mathcal{M} = \begin{pmatrix} 0 & A_1 & -iu & q_{12}^* \\ -A_1 & 0 & -q_{12} & i\bar{u} \\ iu & q_{12} & 0 & A_2 \\ -q_{12}^* & -i\bar{u} & -A_2 & 0 \end{pmatrix} \tag{3.127}$$

and, by analogy with (4.107) in [Noc16],

$$A_j = q_{jj} - i(x + i(-1)^{j-1}\varepsilon) + \frac{1}{N}B + iS, \tag{3.128}$$

where q_{ij} are the entries of $\hat{Q}_F$. To evaluate $\det\mathcal{M}$, we make repeated applications of the well-known result for block 2×2 matrices consisting of $N \times N$ blocks:

$$\det\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \det(A - BD^{-1}C)\det(D).$$

This process quickly results in

$$\begin{aligned}
\sqrt{\det\mathcal{M}} &= \det(A_1A_2 - (u\bar{u} + q_{12}\bar{q}_{12})) \\
&= \det\left(\left[\det(\hat{Q}_F - ix - \varepsilon\sigma) - \bar{u}u\right]I + \mathrm{Tr}(\hat{Q}_F - ix - \varepsilon\sigma)\left(\frac{1}{N}B + iS\right) + \left(\frac{1}{N}B + iS\right)^2\right) \\
&= \det\left(G_1 + N^{-1}B + iS\right)\det\left(G_2 + N^{-1}B + iS\right)
\end{aligned} \tag{3.129}$$

where we have chosen G_1, G_2 to be solutions to

$$G_1G_2 = \det(\hat{Q}_F - ix - \varepsilon\sigma) - \bar{u}u \tag{3.130}$$

$$G_1 + G_2 = \mathrm{Tr}(\hat{Q}_F - ix - \varepsilon\sigma). \tag{3.131}$$

Recalling the B has rank 2 we let O_B be the $N\times 2$ matrix of the non-null eigenvectors of B and $\lambda_{1,2}^{(B)}$ be its non-null eigenvalues and use the determinantal identity found in equation (3) of [BGM12] to write[8]

$$\det\left(G_j I_N + N^{-1}B + iS\right) = \det\left(G_j I_N + iS\right)\det\left(I_2 + N^{-1}O_B^T\left(G_j I_N + iS\right)^{-1} O_B \mathrm{diag}\left(\lambda_1^{(B)}, \lambda_2^{(B)}\right)\right). \tag{3.132}$$

We would now like to apply the integral formula found in Appendix D of [FS02] to re-write the integrals over the N-dimensional vectors $\boldsymbol{x}_1, \boldsymbol{x}_2$ as a single integral over a 2×2 symmetric matrix Q_B. However, the integrand does not only depend on $\boldsymbol{x}_1, \boldsymbol{x}_2$ through $Q_B \equiv \begin{pmatrix} \boldsymbol{x}_1^T\boldsymbol{x}_1 & \boldsymbol{x}_1^T\boldsymbol{x}_2 \\ \boldsymbol{x}_2^T\boldsymbol{x}_1 & \boldsymbol{x}_2^T\boldsymbol{x}_2 \end{pmatrix}$ thanks to the dependence on the eigenvectors of B in (3.132) and also in the term $\mathrm{Tr}SB$ in (3.126). Before addressing this problem, we will continue to manipulate the $\hat{Q}_F$ and u integrals along the lines of [Noc16].

First make the change of variables $\hat{Q}_F \leftarrow \hat{Q}_F + ix + \epsilon\sigma$ and $\boldsymbol{x}_j \leftarrow \sqrt{N}\boldsymbol{x}_j$ in (3.126) using (3.129) to obtain

$$\begin{aligned}\mathbb{E}_{GOE}^N \Delta_\epsilon(M; x, S) = K_N^{(4)} \int d\boldsymbol{x}_1 d\boldsymbol{x}_2 d\hat{Q}_F d^2u \exp&\left\{-\frac{N}{2}\mathrm{Tr}Q_B^2 - iN\mathrm{Tr}SB + ixN\mathrm{Tr}Q_B + \epsilon N\mathrm{Tr}Q_B\sigma\right\} \\ &\exp\left\{-N\mathrm{Tr}\hat{Q}_F^2 - 2N\mathrm{Tr}(ix+\epsilon\sigma)\hat{Q}_F - N\mathrm{Tr}(ix+\epsilon\sigma)^2 - 2Nu\bar{u}\right\} \\ &\prod_{j=1}^2 \det\left(G_j + B + iS\right)\end{aligned} \tag{3.133}$$

where $K_N^{(4)} = N^N K_N^{(3)}$ and now the terms G_1, G_2 are given by the modified versions of (3.130)-(3.131):

$$G_1 G_2 = \det\hat{Q}_F - \bar{u}u \tag{3.134}$$

$$G_1 + G_2 = \mathrm{Tr}\hat{Q}_F. \tag{3.135}$$

We now diagonalise the Hermitian matrix $\hat{Q}_F = \hat{U}\mathrm{diag}(q_1, q_2)\hat{U}^\dagger$ in (3.133), but the term $\mathrm{Tr}\sigma\hat{Q}_F$ is not unitarily invariant, so we follow [Noc16] and introduce an explicit parametrization[9] of the unitary matrix $\hat{U}$

$$\hat{U} = e^{i\hat{\phi}/2}\begin{pmatrix} e^{i\hat{\alpha}/2} & 0 \\ 0 & e^{-i\hat{\alpha}/2}\end{pmatrix}\begin{pmatrix} \cos\hat{\theta} & \sin\hat{\theta} \\ -\sin\hat{\theta} & \cos\hat{\theta}\end{pmatrix}\begin{pmatrix} e^{i\hat{\beta}/2} & 0 \\ 0 & e^{-i\hat{\beta}/2}\end{pmatrix}$$

where $\hat{\phi}, \hat{\alpha}, \hat{\beta} \in [0, 2\pi)$, $\hat{\theta} \in [0, \pi/2)$ and elementary calculations give the Jacobian factor $|q_1 - q_2|^2 \sin(2\hat{\theta})$. Further brief elementary calculations give

$$\mathrm{Tr}\hat{Q}_F\sigma = (q_2 - q_1)\cos(2\hat{\theta}). \tag{3.136}$$

[8] Note that we here include explicitly the identity matrix symbols to make plain the dimension of the determinants.

[9] [Noc16] uses an incorrect parametrization with only two angles. The calculations are are invariant in the extra angles α, β and so this detail only matters if one is tracking the multiplicative constants, as we do here.

and so, integrating out $\hat{\phi}, \hat{\alpha}, \hat{\beta}$,

$$\begin{aligned}\mathbb{E}^N_{GOE}\Delta_\varepsilon(M;x,S) = K^{(5)}_N e^{2N(x^2-\varepsilon^2)} \int d\boldsymbol{x}_1 d\boldsymbol{x}_2 \iint_{-\infty}^{\infty} dq_1 dq_2 \int d^2u \int_0^{\pi/2} d\theta \sin 2\hat{\theta} \\ \exp\left\{-\frac{N}{2}\mathrm{Tr}Q_B^2 - iN\mathrm{Tr}SB + ixN\mathrm{Tr}Q_B + \varepsilon N\mathrm{Tr}Q_B\sigma\right\} \\ \exp\left\{-N(q_1^2+q_2^2) - 2Nix(q_1+q_2) - 2N\varepsilon(q_2-q_1)\cos 2\hat{\theta} - 2Nu\bar{u}\right\} \\ \prod_{j=1}^2 \det\left(G_j + B + iS\right)|q_1-q_2|^2 \end{aligned} \tag{3.137}$$

with $K^{(5)} = (2\pi)^3 K^{(4)}_N$ and now

$$G_1G_2 = q_1q_2 - \bar{u}u \tag{3.138}$$

$$G_1 + G_2 = q_1 + q_2. \tag{3.139}$$

We form an Hermitian matrix

$$R = \left(\begin{array}{cc} q_1 & \bar{u} \\ u & q_2 \end{array}\right) \tag{3.140}$$

and so (3.137) is rewritten as

$$\begin{aligned}\mathbb{E}^N_{GOE}\Delta_\varepsilon(M;x,S) = K^{(6)}_N e^{2N(x^2-\varepsilon^2)} \int d\boldsymbol{x}_1 d\boldsymbol{x}_2 \int dR|R_{11}-R_{22}|^2 \int_0^{\pi/2} d\theta \sin 2\hat{\theta} \\ \exp\left\{-\frac{N}{2}\mathrm{Tr}Q_B^2 - iN\mathrm{Tr}SB + ixN\mathrm{Tr}Q_B + \varepsilon N\mathrm{Tr}Q_B\sigma\right\} \\ \exp\left\{-N\mathrm{Tr}R^2 - 2Nix\mathrm{Tr}R - 2\varepsilon N(R_{22}-R_{11})\cos 2\hat{\theta}\right\} \\ \prod_{j=1}^2 \det\left(G_j + B + iS\right) \end{aligned} \tag{3.141}$$

with $K^{(6)}_N = \frac{1}{16\pi^2}K^{(5)}_N$ and

$$G_1G_2 = \det R \tag{3.142}$$

$$G_1 + G_2 = \mathrm{Tr}R. \tag{3.143}$$

The factor of $(16\pi^2)^{-1}$ comes from the change of variables $(q_1, q_2, u, \bar{u}) \mapsto R$. Indeed, clearly $dq_1dq_2dud\bar{u} = Z^{-1}dR$ for some constant Jacobian factor Z. We can most easily determine Z by integrating against a test function:

$$\begin{aligned}\frac{4\pi Vol(U(2))}{Z} = \frac{1}{Z}\int_{\mathrm{Herm}(2)} dR e^{-\frac{1}{2}\mathrm{Tr}R^2} = \iint_{-\infty}^{\infty} dq_1dq_2 \iint_{-\infty}^{\infty} d\Re u\, d\Im u e^{-\frac{1}{2}(q_1^2+q_2^2+2u\bar{u})} = 2\pi^2 \\ \implies Z = \frac{2Vol(U(2))}{\pi} = 16\pi^2.\end{aligned}$$

We diagonalise $R = U\mathrm{diag}(r_1, r_2)U^\dagger$, but again the integrand in (3.141) is not unitarily invariant in R so we repeat the previous procedure using

$$U = e^{i\phi/2}\left(\begin{array}{cc} e^{i\alpha/2} & 0 \\ 0 & e^{-i\alpha/2}\end{array}\right)\left(\begin{array}{cc} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta\end{array}\right)\left(\begin{array}{cc} e^{i\beta/2} & 0 \\ 0 & e^{-i\beta/2}\end{array}\right).$$

Overall, integrating out ϕ, α, β, (3.141) becomes

$$\begin{aligned}\mathbb{E}^N_{GOE}\Delta_\epsilon(M;x,S) = K^{(7)}_N e^{2N(x^2-\epsilon^2)} \iint_0^{\pi/2} d\theta d\hat{\theta} \int d\boldsymbol{x}_1 d\boldsymbol{x}_2 \iint_{-\infty}^{\infty} dr_1 dr_2 |r_1 - r_2|^4 \sin 2\theta \cos^2 2\theta \sin 2\hat{\theta}\\ \exp\left\{-\frac{N}{2}\text{Tr}Q_B^2 - iN\text{Tr}SB + ixN\text{Tr}Q_B + \epsilon N\text{Tr}Q_B\sigma\right\}\\ \exp\{-N(r_1^2 + r_2^2) - 2Ni(x - i\epsilon\cos 2\theta\cos 2\hat{\theta})r_1\\ - 2Nix(x + i\epsilon\cos 2\theta\cos 2\hat{\theta})\}\\ \prod_{j=1}^2 \det\left(G_j + B + iS\right)\end{aligned} \tag{3.144}$$

where $K^{(7)} = (2\pi)^3 K^{(6)}$ and now

$$G_1 G_2 = r_1 r_2, \tag{3.145}$$

$$G_1 + G_2 = r_1 + r_2 \tag{3.146}$$

$$\iff \{G_1, G_2\} = \{r_1, r_2\}. \tag{3.147}$$

We can now clearly take $r_j = G_j$ without loss of generality. The terms $\det(r_j + B + iS)$ and $e^{-iN\text{Tr}SB}$ depend on the eigenvectors of B and prevent an application of the integral formula of [FS02] as used by [Noc16]. In fact, it is possible the adapt this integral formula for use in the presence of the term $e^{-iN\text{Tr}SB}$, as seen in Lemma 3.6.

Since S has all entries of order N^{-1}, we can expand the nuisance determinants:

$$\det(r_j + B + iS) = \prod_{i=1}^2 (r_j + \lambda_i^{(B)})(1 + o(1)). \tag{3.148}$$

For this step to be legitimate in the sense of asymptotic expansions, we must have that the error term is uniformly small in the integration variables $\boldsymbol{x}_1, \boldsymbol{x}_2, r_1, r_2, \theta, \hat{\theta}$. Note that the integrand in (3.144) is analytic in r_1, r_2 and so we can deform the contours of integration from $(-\infty, \infty)$ to Γ, a contour that, say, runs from $-\infty$ along the real line to -1 and then follows the unit semi-circle in the upper half plane to 1 before continuing to ∞ along the real line. We show an example contour in Figure 3.10. It is now clear that r_1, r_2 are bounded away from 0 and so the error terms in (3.148) are uniform, so giving

$$\begin{aligned}\mathbb{E}^N_{GOE}\Delta_\epsilon(M;x,S) = K^{(7)}_N e^{2N(x^2-\epsilon^2)} \iint_0^{\pi/2} d\theta d\hat{\theta} \int d\boldsymbol{x}_1 d\boldsymbol{x}_2 \iint_{-\infty}^{\infty} dr_1 dr_2 |r_1 - r_2|^4 \sin 2\theta \cos^2 2\theta \sin 2\hat{\theta}\\ \exp\left\{-\frac{N}{2}\text{Tr}Q_B^2 - iN\text{Tr}SB + ixN\text{Tr}Q_B + \epsilon N\text{Tr}Q_B\sigma\right\}\\ \exp\{-N(r_1^2 + r_2^2) - 2Ni(x - i\epsilon\cos 2\theta\cos 2\hat{\theta})r_1\\ - 2Nix(x + i\epsilon\cos 2\theta\cos 2\hat{\theta})\}\\ \prod_{i,j=1}^2 \det\left(r_j + \lambda_i^{(B)}\right)(1 + o(1))\end{aligned} \tag{3.149}$$

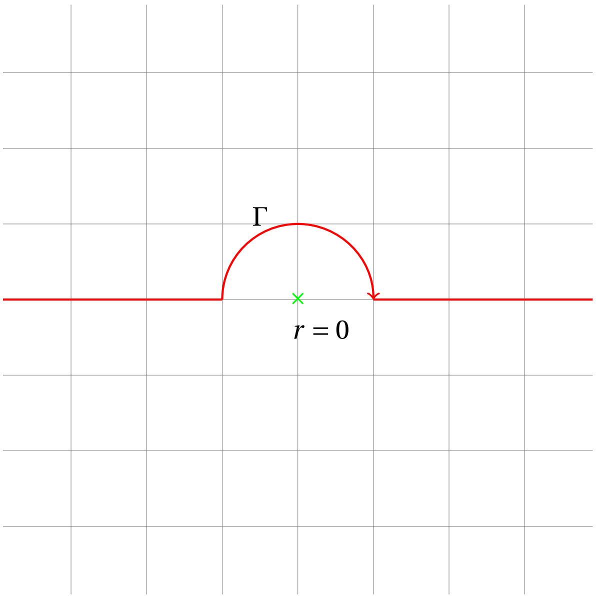


Figure 3.10: Example contour Γ used for the r_1, r_2 integrals to keep away from the origin (denoted by the green cross).

Lemma 3.6 can now be applied:

$$\begin{aligned}\mathbb{E}^N_{GOE}\Delta_\varepsilon(M;x,S) = K^{(8)}_N e^{2N(x^2-\varepsilon^2)}\,(1+o(1)) \\ \iint_0^{\pi/2} d\theta d\hat{\theta} \int_{\mathrm{Sym}_{\geqslant 0}(2)} dQ_B \iint_\Gamma dr_1 dr_2 \cos^2 2\theta \sin 2\theta \sin 2\hat{\theta} \\ \exp\left\{-\frac{N}{2}\mathrm{Tr}Q_B^2 + ixN\mathrm{Tr}Q_B + \varepsilon N \mathrm{Tr}Q_B\sigma\right\} \\ \exp\{-N(r_1^2+r_2^2) - 2Ni(x - i\varepsilon\cos 2\theta\cos 2\hat{\theta})r_1 \\ -2Nix(x+i\varepsilon\cos 2\theta \cos 2\hat{\theta})\} \\ \prod_{j=1}^r \left(1 + 2is_j\mathrm{Tr}Q_B - 4p_{11}p_{22}s_j^2\right)^{-1/2} \\ \prod_{i,j=1}^2 \left(r_j + \lambda_i^{(B)}\right)|r_1-r_2|^4 (r_1r_2)^{N-2}(\det Q_B)^{\frac{N-3}{2}},\end{aligned} \tag{3.150}$$

where p_{ij} are the entries of the matrix Q_B and $K^{(8)} = \frac{\pi^N \pi^{-1/2}}{\Gamma\left(\frac{N}{2}\right)\Gamma\left(\frac{N-1}{2}\right)} K_N^{(7)}$.

We now wish to diagonalise Q_B and integrate out its eigenvectors, but as before (around (3.141)) the integrand is not invariant under the action of the orthogonal group on Q_B and so we instead diagonalise $Q_B = O\mathrm{diag}(p_1,p_2)O^T$ and parametrize O as

$$O = \begin{pmatrix} \cos\theta' & \sin\theta' \\ -\sin\theta' & \cos\theta' \end{pmatrix} \tag{3.151}$$

but we must be careful to choose domain of integration for θ and (p_1, p_2) such that the transformation is a bijection. Consider a general positive semi-definite symmetric matrix

$$Q_B = \begin{pmatrix} a & c \\ c & b \end{pmatrix}.$$

Solving for the eigenvalues gives two choices for (p_1, p_2) because of the arbitrary ordering of the eigenvalues. We want a simple product domain for the (p_1, p_2) integrals and both eigenvalues are non-negative, so we choose $(p_1, p_2) \in (\mathbb{R}_{\geqslant 0})^2$. One can easily find that

$$c = \frac{p_2 - p_1}{2} \sin 2\theta \tag{3.152}$$

$$a = \frac{p_1 + p_2 + (p_1 - p_2)\cos 2\theta}{2} \tag{3.153}$$

$$b = \frac{p_1 + p_2 + (p_2 - p_1)\cos 2\theta}{2} \tag{3.154}$$

and so we see immediately that the domain of integration of θ must be restricted to an interval of length π to obtain a bijection. But further, because of the chosen domain for (p_1, p_2) the quantity $(p_1 - p_2)$ takes all values in $\mathbb{R}$ and thus we must in fact restrict θ to, say, $[0, \pi/2)$ to obtain a bijection. The Jacobian of this transformation is $|p_1 - p_2|$ and further

$$\begin{aligned} p_{11}p_{22} &= (p_1 \cos^2\theta' + p_2 \sin^2\theta')(p_2 \cos^2\theta' + p_1 \sin^2\theta') \\ &= (p_1^2 + p_2^2)(\cos\theta' \sin\theta')^2 + p_1 p_2 (\cos^4\theta' + \sin^4\theta') \\ &= \frac{1}{4}\sin^2 2\theta' (p_1^2 + p_2^2) + \frac{1}{4}\left(3 + 4\cos 4\theta'\right) p_1 p_2 \end{aligned} \tag{3.155}$$

and so we get

$$\begin{aligned} \mathbb{E}_{GOE}^N \Delta_\varepsilon(M; x, S) = K_N^{(8)} e^{2N(x^2-\varepsilon^2)} (1 + o(1)) & \iiint_0^{\pi/2} d\theta d\theta' d\hat{\theta} \iint_0^\infty dp_1 dp_2 \iint_\Gamma dr_1 dr_2 \\ & |r_1 - r_2|^4 (r_1 r_2)^{N-2} (p_1 p_2)^{\frac{N-3}{2}} \cos^2 2\theta \sin 2\theta \sin 2\hat{\theta} \\ & \exp\left\{ -\frac{N}{2}(p_1^2 + p_2^2) + iN(x - i\varepsilon \cos 2\theta')p_1 + iN(x + i\varepsilon\cos 2\theta')p_2 \right\} \\ & \exp\{ -N(r_1^2 + r_2^2) - 2Ni(x - i\varepsilon \cos 2\theta \cos 2\hat{\theta}) r_1 \\ & \quad - 2Nix(x + i\varepsilon \cos 2\theta \cos 2\hat{\theta}) \} \\ & \prod_{i,j=1}^2 \left(r_j + p_i \right) J_1(p_1, p_2, \theta'; \{s_j\}_{j=1}^r, N) \end{aligned} \tag{3.156}$$

where

$$J_1(p_1, p_2, \theta'; \{s_j\}_{j=1}^r, N) = \prod_{j=1}^r \left(1 + 2is_j(p_1 + p_2) - s_j^2 \left[\sin^2 2\theta' (p_1^2 + p_2^2) + \left(3 + 4\cos 4\theta'\right) p_1 p_2 \right] \right)^{-1/2}. \tag{3.157}$$

Now let us define the functions

$$\psi_U^{(\pm)}(z;x;\epsilon) = \frac{1}{2}z^2 \pm i(x-i\epsilon)z - \frac{1}{2}\log z \tag{3.158}$$

$$\psi_L^{(\pm)}(z;x;\epsilon) = \frac{1}{2}z^2 \pm i(x+i\epsilon)z - \frac{1}{2}\log z \tag{3.159}$$

$$\tag{3.160}$$

and also

$$J_2(r_1,r_2,p_1,p_2) = |r_1-r_2|^4|p_1-p_2|(r_1r_2)^{-2}(p_1p_2)^{-\frac{3}{2}}(r_1+p_1)(r_1+p_2)(r_2+p_1)(r_2+p_2) \tag{3.161}$$

and then we finally rewrite (3.156) as

$$\begin{aligned}
&\mathbb{E}_{GOE}^N \Delta_\epsilon(M;x,S)\\
&= K_N^{(8)} e^{2N(x^2-\epsilon^2)}(1+o(1)) \iiint_0^{\pi/2} d\theta d\theta' d\hat{\theta} \iint_0^\infty dp_1 dp_2 \iint_\Gamma dr_1 dr_2\\
&\quad J_1(p_1,p_2,\theta';S,N)J_2(r_1,r_2,p_1,p_2)\cos^2 2\theta \sin 2\theta \sin 2\hat{\theta}\\
&\quad \exp\Bigg\{-N\Big(2\psi_L^{(+)}(r_1;x;\epsilon\cos 2\theta\cos 2\hat{\theta}) + 2\psi_U^{(+)}(r_2;x;\epsilon\cos 2\theta\cos 2\hat{\theta})\\
&\quad + \psi_L^{(-)}(p_1;x;\epsilon\cos 2\theta') + \psi_U^{(-)}(p_2;x;\epsilon\cos 2\theta')\Big)\Bigg\}.
\end{aligned} \tag{3.162}$$

■

We will need the asymptotic behaviour of the constant K_N defined in Lemma 3.7.

Lemma 3.8. *As $N \to \infty$*

$$K_N \sim \frac{(-i)^N N^{\frac{9}{2}}}{4\sqrt{2}\pi^{\frac{5}{2}}}(2e)^N. \tag{3.163}$$

Proof. Using Stirling's formula for the Gamma function gives

$$\begin{aligned}
K_N &\sim \frac{N^{N+3}(-i)^N}{\pi^{3/2}} N^{-\frac{N}{2}+\frac{1}{2}}(N-1)^{-\frac{N}{2}+1}2^{\frac{N}{2}-\frac{1}{2}}2^{\frac{N}{2}-1}e^{\frac{N}{2}}e^{\frac{N}{2}-\frac{1}{2}}(2\pi)^{-1}\\
&= \frac{N^{N+3}(-i)^N}{\pi^{3/2}} N^{-N}N^{\frac{3}{2}}2^N 2^{-\frac{5}{2}}e^N e^{-\frac{1}{2}}\pi^{-1}\left(\frac{N-1}{N}\right)^{-\frac{N}{2}+1}\\
&\sim \frac{(-i)^N N^{\frac{9}{2}}}{4\sqrt{2}\pi^{\frac{5}{2}}}(2e)^N.
\end{aligned} \tag{3.164}$$

■

Building on Lemma 3.7, we can prove a generalisation of Theorem 2.8 from [AAC13], namely Theorem 3.2.

Theorem 3.2. *Recall the definition of C_N^h in (3.56) and let Θ_H be defined as in [AAC13]:*

$$\Theta_H(u) = \begin{cases} \frac{1}{2}\log(H-1) - \frac{H-2}{4(H-1)}u^2 - I_1(u; E_\infty) & \text{if } u \leqslant -E_\infty, \\ \frac{1}{2}\log(H-1) - \frac{H-2}{4(H-1)}u^2 & \text{if } -E_\infty \leqslant u \leqslant 0, \\ \frac{1}{2}\log(H-1) & \text{if } 0 \geqslant u, \end{cases} \tag{3.58}$$

where $E_\infty = 2\sqrt{\frac{H-1}{H}}$, and $I_1(\cdot; E)$ is defined on $(-\infty, -E]$ as in [AAC13] by

$$I_1(u; E) = \frac{2}{E^2}\int_u^{-E} (z^2 - E^2)^{1/2} dz = -\frac{u}{E^2}\sqrt{u^2 - E^2} - \log\left(-u + \sqrt{u^2 - E^2}\right) + \log E, \tag{3.59}$$

then

$$\lim_{N\to\infty} \frac{1}{N} \log \mathbb{E} C_N^h(\sqrt{N}u) = \Theta_H(u). \tag{3.60}$$

Proof. Combining Lemmata 3.4 and 3.5 and observing that the integrand in the Kac-Rice formula of Lemma 3.5 is spherically symmetric, we obtain

$$\mathbb{E}C_N^h(\sqrt{N}u) = \underbrace{(2(N-1)(H-1)H)^{\frac{N-1}{2}}\omega_N \frac{e^{-\frac{v^2}{2H}}}{(2\pi H)^{(N-1)/2}}}_{:=\Omega_N} \int_{-\infty}^{u_N} dx \, \frac{1}{\sqrt{2\pi}t} e^{-\frac{x^2}{2t^2}} \mathbb{E}_{GOE}^{N-1} |\det(M - xI + S)| \tag{3.165}$$

where

$$u_N = u\sqrt{\frac{HN}{2(N-1)(H-1)}},$$

the variance $t^2 = \frac{H}{2(N-1)(H-1)}$, $\omega_N = 2\pi^{N/2}/\Gamma(N/2)$ is the surface area of the $N-1$ sphere and S and $\boldsymbol{v}$ are defined in Lemma 3.4. Note that the first term in Ω_N comes from the expression (3.87) and the third term from (3.74) and (3.78), i.e. this is the density of ∇h evaluated at 0 as appears in Lemma

3.5. The conditions for Lemma 3.7 are shown to be met in Lemma 3.4, so we obtain

$$\begin{aligned}
\mathbb{E}C_N^h(\sqrt{N}u) =&\Omega_N K_{N-1}\sqrt{\frac{2(N-1)(H-1)}{H}}(1+o(1))\\
&\int_{-\infty}^{u_N} dx\,\frac{1}{\sqrt{2\pi}}\lim_{\epsilon\searrow 0}\iiint_0^{\pi/2} d\theta d\hat{\theta} d\theta' \iint_0^\infty dp_1dp_2\iint_\Gamma dr_1dr_2\\
&J_1(p_1,p_2,\theta';\{s_j\}_{j=1}^r,N-1)J_2(r_1,r_2,p_1,p_2)\cos^2 2\theta\sin 2\theta\sin 2\hat{\theta}\\
&\exp\Bigg\{-(N-1)\Bigg(2\psi_L^{(+)}(r_1;x;\epsilon\cos 2\theta\cos 2\hat{\theta})+2\psi_U^{(+)}(r_2;x;\epsilon\cos 2\theta\cos 2\hat{\theta})\\
&+\psi_L^{(-)}(p_1;x;\epsilon\cos 2\theta')+\psi_U^{(-)}(p_2;x;\epsilon\cos 2\theta')-\frac{H+1}{H}x^2\Bigg)\Bigg\}\\
=&c_{N,H}\int_{-\infty}^{u_N} dx\,\lim_{\epsilon\searrow 0}\iiint_0^{\pi/2} d\theta d\hat{\theta} d\theta' \iint_0^\infty dp_1dp_2\iint_\Gamma dr_1dr_2\\
&J_1(p_1,p_2,\theta';\{s_j\}_{j=1}^r,N-1)J_2(r_1,r_2,p_1,p_2)\cos^2 2\theta\sin 2\theta\sin 2\hat{\theta}\\
&\exp\Bigg\{-(N-1)\Bigg(2\psi_L^{(+)}(r_1;x;\epsilon\cos 2\theta\cos 2\hat{\theta})+2\psi_U^{(+)}(r_2;x;\epsilon\cos 2\theta\cos 2\hat{\theta})\\
&+\psi_L^{(-)}(p_1;x;\epsilon\cos 2\theta')+\psi_U^{(-)}(p_2;x;\epsilon\cos 2\theta')-\frac{H+1}{H}x^2\Bigg)\Bigg\} \qquad (3.166)
\end{aligned}$$

where we have defined the constant

$$c_{N,H}=\frac{\Omega_N K_{N-1}\sqrt{(H-1)(N-1)}}{\sqrt{H\pi}}(1+o(1)). \qquad (3.167)$$

We pause now to derive the asymptotic form of $c_{N,H}$. The vector $\boldsymbol{v}$ was defined in Lemma 3.4 and has entries of order $N^{-1/2}$, so $\boldsymbol{v}^2=\mathcal{O}(1)$. Using Stirling's formula for the Gamma function

$$\begin{aligned}
\Omega_N &\sim 2(N-1)^{\frac{N-1}{2}}(H-1)^{\frac{N-1}{2}}\pi^{1/2}N^{-\frac{N}{2}+\frac{1}{2}}2^{\frac{N}{2}-\frac{1}{2}}e^{\frac{N}{2}}(2\pi)^{-1/2}e^{-\frac{\boldsymbol{v}^2}{2H}}\\
&=(H-1)^{\frac{N-1}{2}}(2e)^{\frac{N}{2}}\left(\frac{N-1}{N}\right)^{\frac{N-1}{2}}e^{-\frac{\boldsymbol{v}^2}{2H}}\\
&\sim (H-1)^{\frac{N-1}{2}}(2e)^{\frac{N}{2}}e^{-1/2}e^{-\frac{\boldsymbol{v}^2}{2H}}\\
\implies \frac{\Omega_N\sqrt{(H-1)(N-1)}}{\sqrt{H\pi}} &\sim (H-1)^{\frac{N}{2}}(2e)^{\frac{N}{2}}e^{-1/2}H^{-1/2}\pi^{-1/2}(N-1)^{1/2}e^{-\frac{\boldsymbol{v}^2}{2H}} \qquad (3.168)
\end{aligned}$$

and so Lemma 3.8 gives

$$\begin{aligned}
c_{N,H} &\sim \frac{(-i)^{N-1}(N-1)^{\frac{9}{2}}}{4\sqrt{2}\pi^{\frac{5}{2}}}(2e)^{N-1}(H-1)^{\frac{N}{2}}(2e)^{\frac{N}{2}}e^{-1/2}H^{-1/2}\pi^{-1/2}(N-1)^{1/2}e^{-\frac{\boldsymbol{v}^2}{2H}}\\
&\sim \frac{(-i)^{N-1}N^5}{4\pi^3H^{1/2}}(2e)^{\frac{3}{2}(N-1)}(H-1)^{\frac{N}{2}}e^{-\frac{\boldsymbol{v}^2}{2H}}. \qquad (3.169)
\end{aligned}$$

In the style of [DH02], the multiple integral in (3.166) can be written as an expansion over saddle points and saddle points of the integrand restricted to sections of the boundary. Recalling the form of $\psi_U^{(\pm)}$ and $\psi_L^{(\pm)}$, we see that the integrand vanishes on the boundary and so we focus on the interior saddle points. Let us define the exponent function

$$\Phi(r_1,r_2,p_1,p_2,x;S,\epsilon)=2\psi_L^{(+)}(r_1;x,\epsilon)+2\psi_U^{(+)}(r_2;x,\epsilon)+\psi_L^{(-)}(p_1;x,\epsilon)+\psi_U^{(-)}(p_2;x,\epsilon)-\frac{(H+1)}{H}x^2 \tag{3.170}$$

It is clear that the $\cos\theta, \cos\hat{\theta}$ and $\cos\theta'$ terms in the exponent of (3.166) do not affect the saddle point asymptotic analysis, since we take the limit $\epsilon \to 0$, and $\theta,\hat{\theta},\theta' \in [0,\pi/2)$ and it is only the signs of the $\mathcal{O}(\epsilon)$ terms that are significant. Therefore, to simplify the exposition, we will suppress these terms. The (r_1,r_2,p_1,p_2) components of $\nabla\Phi$ are of the form

$$z \mapsto z \pm i(x \pm i\epsilon) - \frac{1}{2z} \tag{3.171}$$

and so the only saddle in Φ restricted to those components is at

$$r_1 = \frac{-i(x+i\epsilon)+(2-(x+i\epsilon)^2)^{1/2}}{2} := z_L^{(+)} \tag{3.172}$$

$$r_2 = \frac{-i(x-i\epsilon)+(2-(x-i\epsilon)^2)^{1/2}}{2} := z_U^{(+)} \tag{3.173}$$

$$p_1 = \frac{i(x+i\epsilon)+(2-(x+i\epsilon)^2)^{1/2}}{2} := z_L^{(-)} \tag{3.174}$$

$$p_2 = \frac{i(x-i\epsilon)+(2-(x-i\epsilon)^2)^{1/2}}{2} := z_U^{(-)}. \tag{3.175}$$

To deform the (r_1,r_2,p_1,p_2) contours through this saddle, we are required to choose a branch of the functions in (3.172 - 3.175). Each has branch points at $\pm\sqrt{2}+i\epsilon$ or $\pm\sqrt{2}-i\epsilon$. Since the initial contour of x integration lies along the real line, we take the following branch cuts in the complex x plane and respective angle ranges (see Figure 3.11)

$$[\sqrt{2}+i\epsilon,\sqrt{2}+i\infty],\quad [\pi/2,5\pi/2] \tag{3.176}$$

$$[\sqrt{2}-i\epsilon,\sqrt{2}-i\infty],\quad [-\pi/2,3\pi/2] \tag{3.177}$$

$$[-\sqrt{2}+i\epsilon,-\sqrt{2}+i\infty],\quad [\pi/2,5\pi/2] \tag{3.178}$$

$$[-\sqrt{2}-i\epsilon,-\sqrt{2}-i\infty],\quad [-\pi/2,3\pi/2]. \tag{3.179}$$

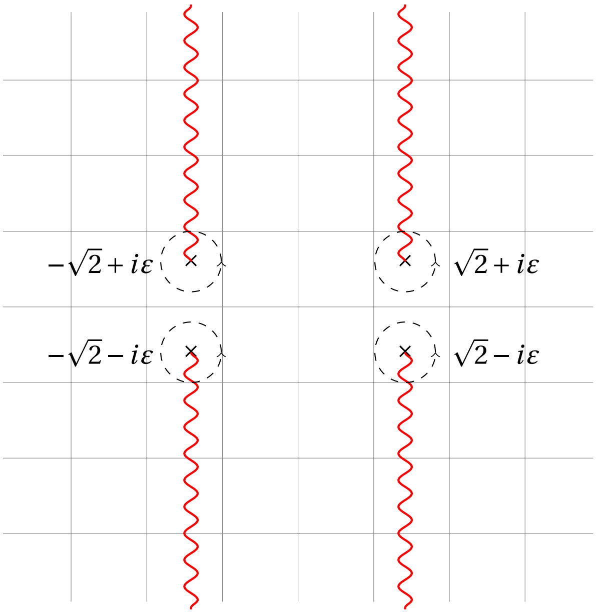


Figure 3.11: The choice of branch for the x integral in the proof of Theorem 3.2.

It is simple to compute $\psi_U^{(\pm)}(z_U^{(\pm)})$ and $\psi_L^{(\pm)}(z_L^{(\pm)})$:

$$\begin{aligned}
\psi_L^{(+)}(z_L^{(+)}) =& \frac{1}{4}\left(1+(x+i\varepsilon)^2+\log 2\right)+\frac{1}{4}\log 2+\frac{1}{4}i(x+i\varepsilon)\left(2-(x+i\varepsilon)^2\right)^{1/2} \\
&-\frac{1}{2}\log\left[-i(x+i\varepsilon)+\left(2-(x+i\varepsilon)^2\right)^{1/2}\right] && (3.180)\\
\psi_U^{(+)}(z_U^{(+)}) =& \frac{1}{4}\left(1+(x-i\varepsilon)^2+\log 2\right)+\frac{1}{4}\log 2+\frac{1}{4}i(x-i\varepsilon)\left(2-(x-i\varepsilon)^2\right)^{1/2} \\
&-\frac{1}{2}\log\left[-i(x-i\varepsilon)+\left(2-(x-i\varepsilon)^2\right)^{1/2}\right] && (3.181)\\
\psi_L^{(-)}(z_L^{(-)}) =& \frac{1}{4}\left(1+(x+i\varepsilon)^2+\log 2\right)+\frac{1}{4}\log 2-\frac{1}{4}i(x+i\varepsilon)\left(2-(x+i\varepsilon)^2\right)^{1/2} \\
&-\frac{1}{2}\log\left[i(x+i\varepsilon)+\left(2-(x+i\varepsilon)^2\right)^{1/2}\right] && (3.182)\\
\psi_U^{(-)}(z_U^{(-)}) =& \frac{1}{4}\left(1+(x-i\varepsilon)^2+\log 2\right)+\frac{1}{4}\log 2-\frac{1}{4}i(x-i\varepsilon)\left(2-(x-i\varepsilon)^2\right)^{1/2} \\
&-\frac{1}{2}\log\left[i(x-i\varepsilon)+\left(2-(x-i\varepsilon)^2\right)^{1/2}\right]. && (3.183)
\end{aligned}$$

Let us consider x still restricted to the real line. We are free to restrict to $\varepsilon > 0$ and then $x \pm i\varepsilon$ lies just above (below) the real line. For $x < -\sqrt{2}$ the angle from all four branch points is π and so we obtain

$$\begin{aligned}\Phi_{(4)}(x) := \lim_{\varepsilon\to 0}\Phi\left(z_L^{(+)}, z_U^{(+)}, z_L^{(-)}, z_U^{(-)}, x;\varepsilon\right) &= \frac{3}{2}\left(1+x^2+\log 2\right)+\frac{3}{2}\log 2-\frac{1}{2}x\sqrt{x^2-2}-2\log\left[-ix+i\sqrt{x^2-2}\right]\\ &-\log\left[ix+i\sqrt{x^2-2}\right]-\frac{H+1}{H}x^2\\ &=\frac{3}{2}\left(1+\log 2\right)+\frac{H-2}{2H}x^2+\frac{3}{2}\log 2-\frac{1}{2}x\sqrt{x^2-2}-\log\left[-ix+i\sqrt{x^2-2}\right]\\ &-\log 2\\ &=\frac{3}{2}\left(1+\log 2\right)+\frac{H-2}{2H}x^2+\frac{1}{2}\log 2-\frac{1}{2}x\sqrt{x^2-2}-\log\left[-x+\sqrt{x^2-2}\right]\\ &-\log i\\ &=\frac{3}{2}\left(1+\log 2\right)+\frac{H-2}{2H}x^2+I_1(x;\sqrt{2})-\log i \qquad (3.184)\end{aligned}$$

However for $-\sqrt{x}<x<\sqrt{2}$ the angles about the branch points are $\pi,\pi,2\pi,0$ in the order of (3.176-3.179). It follows that the square root terms in both of $\psi_L^{(\pm)}(z_L^{(\pm)})$ and both of $\psi_U^{(\pm)}(z_U^{(\pm)})$ have opposite signs and so

$$\begin{aligned}\Phi_{(4)}(x) &= \frac{3}{2}\left(1+\log 2\right)+\frac{H-2}{2H}x^2-\frac{3}{2}\log(-2)+\frac{3}{2}\log 2\\ &=\frac{3}{2}\left(1+\log 2\right)+\frac{H-2}{2H}x^2-\frac{3}{2}\log(-1). \qquad (3.185)\end{aligned}$$

Finally, the above reasoning can be trivially extended to $x>\sqrt{2}$ to obtain

$$\Phi_{(4)}(x)=\frac{3}{2}\left(1+\log 2\right)+\frac{H-2}{2H}x^2+I_1(-x;\sqrt{2})-\log i. \qquad (3.186)$$

It is apparent from (3.184)[10], (3.185) and (3.186) that the branch choice (3.176-3.179) and deforming through each of the saddles of in (r_1,r_2,p_1,p_2) gives a contour of steepest descent in x with the critical point being at $x=0$.

We are thus able to write down the leading order asymptotics for (3.166) for all real u coming either from the end-point $x=\sqrt{2}u/E_\infty$ or the critical point $x=0$. We begin with $u<-E_\infty$ by using (3.184):

$$\begin{aligned}\frac{1}{N}\log\mathbb{E}C_N^h(\sqrt{N}u) &\sim -\frac{3}{2}\log 2-\frac{3}{2}-\frac{H-2}{2H}\frac{Hu^2}{2(H-1)}-I_1(u;E_\infty)+\log i+\frac{1}{N}\log c_{N,H}\\ &\sim\frac{1}{2}\log(H-1)-\frac{H-2}{4(H-1)}u^2-I_1(u;E_\infty) \qquad (3.187)\end{aligned}$$

since by (3.169)

$$\log c_{N,H}\sim\frac{1}{2}N\log(H-1)+\frac{3}{2}(N-1)(1+\log 2)+(N-1)\log(-i). \qquad (3.188)$$

[10] Note that $I_1(x;\sqrt{2})$ is monotonically decreasing on $(-\infty,-\sqrt{2}]$.

For $-E_\infty \leqslant u < 0$ we use (3.185):

$$\begin{aligned}\frac{1}{N}\log\mathbb{E}C_N^h(\sqrt{N}u) &\sim -\frac{3}{2}\log 2 - \frac{3}{2} - \frac{H-2}{2H}\frac{Hu^2}{2(H-1)} + \frac{3}{2}\log(-1) + \frac{1}{N}\log c_{N,H}\\ &\sim \frac{1}{2}\log(H-1) - \frac{H-2}{4(H-1)}u^2 \end{aligned} \tag{3.189}$$

since $\frac{3}{2}\log(-1) = \log\left((-1)^{1/2}\right) = \log i$. Finally, for $u \geqslant 0$ the leading contribution comes from the critical point, so

$$\begin{aligned}\frac{1}{N}\log\mathbb{E}C_N^h(\sqrt{N}u) &\sim -\frac{3}{2}\log 2 - \frac{3}{2} + \frac{3}{2}\log(-1) + \frac{1}{N}\log c_{N,H}\\ &\sim \frac{1}{2}\log(H-1). \end{aligned} \tag{3.190}$$

■

We are in-fact able to obtain the exact leading order term in the expansion of $\mathbb{E}C_N^h(\sqrt{N}u)$ in the case $u < -E_\infty$, namely Theorem 3.4.

Theorem 3.4. *Let $u < -E_\infty$ and define $v = -\frac{\sqrt{2}u}{E_\infty}$. Define the function h by (c.f. (7.10) in [AAC13])*

$$h(v) = \left(\frac{|v-\sqrt{2}|}{|v+\sqrt{2}|}\right)^{1/4} + \left(\frac{|v+\sqrt{2}|}{|v-\sqrt{2}|}\right)^{1/4}, \tag{3.64}$$

and the functions

$$q(\theta') = \frac{1}{2}\sin^2 2\theta' + \frac{1}{4}\left(3 + 4\cos 4\theta'\right), \tag{3.65}$$

$$j(x, s_1, \theta') = 1 + \frac{1}{2}s_1\sqrt{x^2-2}h(x)^2 - s_1^2 q(\theta')|x^2-2|h(x)^2, \tag{3.66}$$

$$T(v, s_1) = \frac{2}{\pi}\int_0^{\pi/2} j(-v, s_1, \theta')d\theta'. \tag{3.67}$$

The $N-1\times N-1$ deterministic matrix S is defined subsequently around (3.88). S has fixed rank $r=2$ and non-zero eigenvalues $\{s_1, N^{-1/2}s_2\}$ where $s_j = \mathcal{O}(1)$. The specific form of S is rather cumbersome and uninformative and so is relegated to Appendix A.1, and the vector $\boldsymbol{v}$ is defined in Lemma 3.4. Then we have

$$\mathbb{E}C_N^h(\sqrt{N}u) \sim \frac{N^{-\frac{1}{2}}}{\sqrt{2\pi H}}e^{-\frac{v^2}{2H}}T(v,s_1)h(v)e^{N\Theta_H(u)}\frac{e^{I_1(u;E_\infty)-\frac{1}{2}uI_1'(u;E_\infty)}}{\frac{H-2}{2(H-1)}u + I_1'(u;E_\infty)}. \tag{3.68}$$

Proof. We begin by deriving an alternative form for h. For $v > \sqrt{2}$

$$\begin{aligned} h(v)^2 &= \frac{|v-\sqrt{2}| + |v+\sqrt{2}| + 2|v^2-2|^{\frac{1}{2}}}{|v^2-2|^{\frac{1}{2}}}\\ &= 2\left(v + |v^2-2|^{\frac{1}{2}}\right)|v^2-2|^{-\frac{1}{2}}\\ \implies h(v) &= \sqrt{2}\left(v + |v^2-2|^{\frac{1}{2}}\right)^{\frac{1}{2}}|v^2-2|^{-\frac{1}{4}}\\ &= 2|-v+|v^2-2|^{\frac{1}{2}}|^{-\frac{1}{2}}|v^2-2|^{-\frac{1}{4}}. \end{aligned} \tag{3.191}$$

This proof now proceeds like that of Theorem 3.2 except that we are required to keep track of the exact factors in (3.166) and evaluate the $\mathcal{O}(1)$ integrals arising from the saddle point approximation. First note that (using primes to denote z derivatives)

$$\psi_{U,L}^{(\pm)}{}''(z;x;\epsilon) = 1 + \frac{1}{2z^2} \tag{3.192}$$

and so we abbreviate $\psi_{U,L}^{(\pm)}{}'' = \psi''$. We get the following useful relation (now letting $\epsilon \to 0$ implicitly for simplicity of exposition)

$$\begin{aligned}\psi''(z_{U,L}^{(\pm)}) &= (z_{U,L}^{(\pm)})^{-2}\left(1 \mp ixz_{U,L}^{(\pm)}\right)\\ &= \frac{1}{2}(z_{U,L}^{(\pm)})^{-2}\left(2 - x^2 \pm x\sqrt{x^2-2}\right)\\ &= i\sqrt{x^2-2}(z_{U,L}^{(\pm)})^{-1}\end{aligned} \tag{3.193}$$

where, using our branch choice shown in Figure 3.11, for $x < -\sqrt{2}$ the saddle points are

$$z_{U,L}^{(\pm)} = \frac{\mp ix + i\sqrt{x^2-2}}{2}. \tag{3.194}$$

We recall the central expression (3.166) from the proof of Theorem 3.2:

$$\begin{aligned}\mathbb{E}C_N^h(\sqrt{N}u) = c_{N,H}\int_{-\infty}^{u_N} dx \lim_{\epsilon\searrow 0}\iiint_0^{\pi/2} d\theta d\hat{\theta} d\theta' \iint_0^{\infty} dp_1 dp_2 \iint_\Gamma dr_1 dr_2\\ J_1(p_1,p_2,\theta';\{s_j\}_{j=1}^r, N-1) J_2(r_1,r_2,p_1,p_2)\cos^2 2\theta \sin 2\theta \sin 2\hat{\theta}\\ \exp\Bigg\{-(N-1)\Big(2\psi_L^{(+)}(r_1;x;\epsilon\cos 2\theta\cos 2\hat{\theta}) + 2\psi_U^{(+)}(r_2;x;\epsilon\cos 2\theta\cos 2\hat{\theta})\\ + \psi_L^{(-)}(p_1;x;\epsilon\cos 2\theta') + \psi_U^{(-)}(p_2;x;\epsilon\cos 2\theta') - \frac{H+1}{H}x^2\Big)\Bigg\}\end{aligned}$$

and we recall the expressions for J_1, J_2 from Lemma 3.7:

$$\begin{aligned}J_1(p_1,p_2,\theta';\{s_j\}_{j=1}^r, N) = \left(1 + iN^{-1/2}s_2(p_1+p_2) - N^{-1}s_2^2\left[\frac{1}{4}\sin^2 2\theta'(p_1^2+p_2^2) + \frac{1}{4}\left(3+4\cos 4\theta'\right)p_1p_2\right]\right)^{-1/2}\\ \cdot\left(1 + is_1(p_1+p_2) - s_1^2\left[\frac{1}{4}\sin^2 2\theta'(p_1^2+p_2^2) + \frac{1}{4}\left(3+4\cos 4\theta'\right)p_1p_2\right]\right)^{-1/2},\\ J_2(r_1,r_2,p_1,p_2) = (r_1+p_1)(r_2+p_1)(r_1+p_2)(r_2+p_2)|r_1-r_2|^4|p_1-p_2|(r_1r_2)^{-2}(p_1p_2)^{-3/2}.\end{aligned}$$

We begin by evaluating J_1 to leading order at the saddle points:

$$\begin{aligned}&\frac{1}{2}\sin^2 2\theta'(z^{(-)})^2 + \frac{1}{4}\left(3+4\cos 4\theta'\right)(z^{(-)})^2 \equiv q(\theta')(z^{(-)})^2\\ &\implies J_1(z^{(-)},z^{(-)},\theta';\{s_j\}_{j=1}^r, N) \sim \left(1 + 4iz^{(-)}s_1 - 2q(\theta')\left(z^{(-)}\right)^2 s_1^2\right)^{-1/2}.\end{aligned} \tag{3.195}$$

Recalling

$$x + \sqrt{x^2-2} = \frac{-2}{-x+\sqrt{x^2-2}} = -\frac{h(x)^2}{2}\sqrt{x^2-2}, \qquad (z^{(-)})^2 = -\frac{1}{2}\sqrt{x^2-2}\left(x+\sqrt{x^2-2}\right) \tag{3.196}$$

we obtain

$$J_1 \sim 1 + \frac{1}{2} s_1 \sqrt{x^2-2}\, h(x)^2 - s_1^2 q(\theta')|x^2-2| h(x)^2 \equiv j(x, s_1, \theta'). \tag{3.197}$$

We see that $J_2(z^{(+)}, z^{(+)}, z^{(-)}, z^{(-)}) = 0$ and so we are required to expand J_2 in the region of

$$(r_1, r_2, p_1, p_2) = (z^{(+)}, z^{(+)}, z^{(-)}, z^{(-)}).$$

Following standard steepest descents practice, the integration variables r_1, r_2, p_1, p_2 are replaced by scaled variables in the region of the saddle point, i.e.

$$r_i = z^{(+)} + (N-1)^{-\frac{1}{2}} |\psi^{(+)''}(z^{(+)})|^{-\frac{1}{2}} \rho_i \tag{3.198}$$
$$p_i = z^{(-)} + (N-1)^{-\frac{1}{2}} |\psi^{(-)''}(z^{(-)})|^{-\frac{1}{2}} \pi_i \tag{3.199}$$

and so

$$\begin{aligned} &J_2(r_1, r_2, p_1, p_2) \\ =&(N-1)^{-\frac{5}{2}} |x^2-2|^2 (z^{(+)})^{-4} (z^{(-)})^{-3} |\psi^{(-)''}(z^{(-)})|^{-\frac{1}{2}} |\psi^{(+)''}(z^{(+)})|^{-2} |\rho_1-\rho_2|^4 |\pi_1-\pi_2| + o(N^{-\frac{5}{2}}). \end{aligned} \tag{3.200}$$

Piecing these components together gives

$$\begin{aligned} J_2 J_1 dr_1 dr_2 dp_1 dp_2 = &(N-1)^{-\frac{9}{2}} j(x, s_1, \theta')|x^2-2|^2 \\ &|\psi^{(-)''}(z^{(-)})|^{-\frac{3}{2}} |\psi^{(+)''}(z^{(+)})|^{-3} (z^{(+)})^{-4} (z^{(-)})^{-3} \\ &|\rho_1-\rho_2|^4 |\pi_1-\pi_2| d\rho_1 d\rho_2 d\pi_1 d\pi_2 \\ = &(N-1)^{-\frac{9}{2}} j(x, s_1, \theta')|x^2-2|^{-\frac{1}{4}} (z^{(+)})^{-1} (z^{(-)})^{-\frac{3}{2}} \\ &|\rho_1-\rho_2|^4 |\pi_1-\pi_2| d\rho_1 d\rho_2 d\pi_1 d\pi_2 \\ = &2(N-1)^{-\frac{9}{2}} j(x, s_1, \theta')|x^2-2|^{-\frac{1}{4}} (z^{(-)})^{-\frac{1}{2}} \\ &|\rho_1-\rho_2|^4 |\pi_1-\pi_2| d\rho_1 d\rho_2 d\pi_1 d\pi_2 \\ = &2^{\frac{3}{2}} (N-1)^{-\frac{9}{2}} j(x, s_1, \theta')|x^2-2|^{-\frac{1}{4}} \left(x + \sqrt{x^2-2}\right)^{-\frac{1}{2}} \\ &|\rho_1-\rho_2|^4 |\pi_1-\pi_2| d\rho_1 d\rho_2 d\pi_1 d\pi_2. \end{aligned} \tag{3.201}$$

Recalling the expression (3.191), we can then write

$$\begin{aligned} J_2 J_1 dr_1 dr_2 dp_1 dp_2 &= 2^{\frac{3}{2}} (N-1)^{-\frac{9}{2}} j(x, s_1, \theta') h(-x) 2^{-1} |\rho_1-\rho_2|^4 |\pi_1-\pi_2| d\rho_1 d\rho_2 d\pi_1 d\pi_2 \\ &= 2^{\frac{1}{2}} (N-1)^{-\frac{9}{2}} j(x, s_1, \theta') h(-x) |\rho_1-\rho_2|^4 |\pi_1-\pi_2| d\rho_1 d\rho_2 d\pi_1 d\pi_2 \end{aligned} \tag{3.202}$$

and so using (3.169), we obtain

$$\begin{aligned} \mathbb{E} C_N^h(\sqrt{N} u) \sim \frac{2^{-\frac{3}{2}} N^{\frac{1}{2}}}{\pi^3 \sqrt{H}} e^{-\frac{\boldsymbol{v}^2}{2H}} \frac{Y_2^{(4)}}{8} Y_2^{(1)} \iint_0^{\pi/2} d\theta d\hat{\theta}\ \cos^2 2\theta \sin 2\theta \sin 2\hat{\theta} \\ \sqrt{H-1} \int_0^{\pi/2} d\theta' \int_{-\infty}^{\frac{\sqrt{2}u}{E_\infty}\sqrt{\frac{N}{N-1}}} dx\ h(-x) j(x, s_1, \theta') e^{(N-1)\Theta_H(2^{-\frac{1}{2}} E_\infty x)} \end{aligned} \tag{3.203}$$

where we have defined the integrals

$$Y_n^{(\beta)} = \int_{\mathbb{R}^n} d\boldsymbol{y}\ e^{-\frac{1}{2}\boldsymbol{y}^2}|\Delta(\boldsymbol{y})|^\beta \tag{3.204}$$

and Δ is the Vandermonde determinant. Recall that, as in Theorem 3.2, the x integration contour in (3.203) is a steepest descent contour and so the leading order term comes from the end point. Now

$$\begin{aligned}&(N-1)\Theta_H\left(\sqrt{\frac{N}{N-1}}u\right)\\ =&(N-1)\frac{1}{2}\log(H-1) - N\frac{H-2}{4(H-1)}u^2 - (N-1)I_1\left(\sqrt{\frac{N}{N-1}}u;E_\infty\right)\\ =&(N-1)\frac{1}{2}\log(H-1) - N\frac{H-2}{4(H-1)}u^2 - (N-1)I_1(u;E_\infty) - \frac{N-1}{2N}uI_1'(u;E_\infty) + \mathcal{O}(N^{-1})\\ =&N\Theta_H(u) - \frac{1}{2}\log(H-1) + I_1(u;E_\infty) - \frac{1}{2}uI_1'(u;E_\infty) + \mathcal{O}(N^{-1})\end{aligned} \tag{3.205}$$

and so

$$\mathbb{E}C_N^h(\sqrt{N}u) \sim \frac{2^{-\frac{3}{2}}N^{\frac{-1}{2}}}{24\pi^3\sqrt{H}}e^{-\frac{v^2}{2H}}Y_2^{(4)}Y_2^{(1)}\left(\int_0^{\pi/2}d\theta' j(-v,s_1,\theta')\right)h(v)e^{N\Theta_H(u)}\frac{e^{I_1(u;E_\infty)-\frac{1}{2}uI_1'(u;E_\infty)}}{\frac{H-2}{2(H-1)}u + I_1'(u;E_\infty)} \tag{3.206}$$

where we have defined (c.f. [AAC13] Theorem 2.17) $v = -\sqrt{2}uE_\infty^{-1}$. It now remains only to evaluate the various constants in (3.206) where possible. Firstly observe

$$Y_2^{(1)} = 2\pi\mathbb{E}_{X_1,X_2\overset{i.i.d.}{\sim}\mathcal{N}(0,1)}|X_1-X_2| = 2\pi\mathbb{E}_{X\sim\mathcal{N}(0,2)}|X| = 2\sqrt{\pi}\int_0^\infty xe^{-\frac{x^2}{4}} = 4\sqrt{\pi} \tag{3.207}$$

and similarly

$$Y_2^{(4)} = 2\pi\mathbb{E}_{X_1,X_2\overset{i.i.d.}{\sim}\mathcal{N}(0,1)}(X_1-X_2)^4 = 2\pi\mathbb{E}_{X\sim\mathcal{N}(0,2)}X^4 = 24\pi. \tag{3.208}$$

For convenience we define

$$T(v,s_1) = \frac{2}{\pi}\int_0^{\pi/2}j(-v,s_1,\theta')d\theta', \tag{3.209}$$

and then collating our results:

$$\mathbb{E}C_N^h(\sqrt{N}u) \sim \frac{N^{-\frac{1}{2}}}{\sqrt{2\pi H}}e^{-\frac{v^2}{2H}}T(v,s_1)h(v)e^{N\Theta_H(u)}\frac{e^{I_1(u;E_\infty)-\frac{1}{2}uI_1'(u;E_\infty)}}{\frac{H-2}{2(H-1)}u + I_1'(u;E_\infty)}. \tag{3.210}$$

■

Remark 3.6. Having completed the proof of Theorem 3.4, we can now explain why this result generalises only part (a) of the analogous Theorem (2.17) from [AAC13], namely only the case $u < -E_\infty$. Recall that, following standard steepest descent practice, we introduced scaled integration variables in the region of the saddle point (3.198)-(3.199) and so arrived at (3.203) with the constant factors $Y_2^{(1)}, Y_2^{(4)}$ resulting from the Laplace approximation integrals over the scaled variables. If we take $-E_\infty < u < 0$, say, then $z_U^{(+)} + z_L^{(-)} = 0$ and $z_L^{(+)} + z_U^{(-)} = 0$ and so it is the terms $(r_1 + p_2), (r_2 + p_1)$ that vanish at the saddle point rather than $|r_1 - r_2|^4$ and $|p_1 - p_2|$. It follows that the terms $Y_2^{(1)}, Y_2^{(4)}$ are replaced by the integrals

$$\int_{\mathbb{R}} d\pi_1 d\pi_2 d\rho_1 d\rho_2 \; e^{-\frac{1}{2}(\pi_1^2+\pi_2^2+\rho_1^2+\rho_2^2)}(\rho_1 + \pi_2)(\rho_2 + \pi_1) = 0. \tag{3.211}$$

It is therefore necessary to keep terms to at least the first sub-leading order in the expansion of $J_1 J_2$ around the saddle point, however we cannot do this owing the presence of the $o(1)$ term in the constant $c_{N,H}$ as defined in (3.167) which we cannot evaluate.

Remark 3.7. Note that setting all the $\rho_\ell^{(N)} = 0$ gives $v = 0$, $S = 0$, hence $s_1 = 0$ and so $T = 1$. Consequently (3.210) recovers the exact spherical H-spin glass expression in part (a) of Theorem 2.17 in [AAC13].

Remark 3.8. The function $h(v)$ shows up in [AAC13] in the asymptotic evaluation of Hermite polynomials but arises here by an entirely different route.

3.5.2 Complexity results with prescribed Hessian signature

The next theorem again builds on Lemma 3.7 to prove a generalisation of Theorem 2.5 from [AAC13]. In fact, we will need a modified version of Lemma 3.7 which we now prove.

Lemma 3.9. *Let S be a rank 2 $N \times N$ symmetric matrix with non-zero eigenvalues $\{s_j\}_{j=1}^2$, where and $s_j = \mathcal{O}(1)$. Let $x < -\sqrt{2}$ and let M denote an $N \times N$ GOE matrix with respect to whose law expectations are understood to be taken. Then*

$$\begin{aligned}
&\mathbb{E}_{GOE}^N \left[|\det(M - xI + S)| \mathbb{1}[i_{\leqslant x}(M+S) \in \{k-1, k, k+1\}] \right] \\
\leqslant\; & v_U K_N e^{2Nx^2} (1 + o(1))\, e^{-N(k-1)I_1(x;\sqrt{2})} \lim_{\varepsilon \searrow 0} \iiint_0^{\pi/2} d\theta d\hat{\theta} d\theta' \iint_0^\infty dp_1 dp_2 \iint_\Gamma dr_1 dr_2 \\
& J_1(p_1, p_2, \theta'; \{s_j\}_{j=1}^r, N) J_2(r_1, r_2, p_1, p_2) \cos^2 2\theta \sin 2\theta \sin 2\hat{\theta} \\
& \exp\Big\{ -N\Big(2\psi_L^{(+)}(r_1; x; \varepsilon \cos 2\theta \cos 2\hat{\theta}) + 2\psi_U^{(+)}(r_2; x; \varepsilon \cos 2\theta \cos 2\hat{\theta}) \\
& \quad + \psi_L^{(-)}(p_1; x; \varepsilon \cos 2\theta') + \psi_U^{(-)}(p_2; x; \varepsilon \cos 2\theta') \Big) \Big\}
\end{aligned} \tag{3.212}$$

and

$$\begin{aligned}
&\mathbb{E}_{GOE}^{N}\left[|\det(M-xI+S)|\mathbb{1}[i_{\leqslant x}(M+S)\in\{k-1,k,k+1\}]\right]\\
\geqslant\ & v_L K_N e^{2Nx^2}(1+o(1))\,e^{-N(k+1)I_1(x;\sqrt{2})}\lim_{\varepsilon\searrow 0}\iiint_0^{\pi/2} d\theta d\hat{\theta}d\theta' \iint_0^{\infty} dp_1 dp_2 \iint_{\Gamma} dr_1 dr_2\\
&\quad J_1(p_1,p_2,\theta';\{s_j\}_{j=1}^r,N)J_2(r_1,r_2,p_1,p_2)\cos^2 2\theta\sin 2\theta\sin 2\hat{\theta}\\
&\quad \exp\left\{-N\Big(2\psi_L^{(+)}(r_1;x;\varepsilon\cos 2\theta\cos\hat{\theta})+2\psi_U^{(+)}(r_2;x;\varepsilon\cos 2\theta\cos\hat{\theta})\right.\\
&\qquad \left.+\psi_L^{(-)}(p_1;x;\varepsilon\cos 2\theta')+\psi_U^{(-)}(p_2;x;\varepsilon\cos 2\theta')\Big)\right\} \qquad (3.213)
\end{aligned}$$

where the functions J_1, J_2, the constant K_N and the functions $\psi_{U,L}^{(\pm)}$ are defined as in Lemma 3.7, and the v_L, v_U are some constants independent of N.

Remark 3.9. A more general version of this lemma holds with S having any fixed rank r. In that case, one considers

$$\mathbb{E}_{GOE}^{N}\left[|\det(M-xI+S)|\mathbb{1}[i_{\leqslant x}(M+S)\in\{k-(r-1),\ldots,k,\ldots,k+(r-1)\}]\right] \qquad (3.214)$$

and the statement and proof of the result are immediate extensions of what is given here. We omit this generality, since it is not required here.

Proof. This proof is largely the same as that of Lemma 3.7. The first difference arises at (3.119), where we are required to compute

$$\mathbb{E}_{GOE}^{N}\left[e^{-i\mathrm{Tr}MA}\mathbb{1}[i_{\leqslant x}(M+S)=k]\right]. \qquad (3.215)$$

As will become apparent towards the end of this proof, we do not know how to maintain the exact equality constraint[11] on index when $S\neq 0$, hence the slightly relaxed results that we are proving, however we will proceed by performing the calculation for $S=0$ and then show that S can be reintroduced one eigendirection at a time. As in the proof of Theorem A.1 in [AAC13], we split this expectation by fixing a bound, R, for the largest eigenvalue, i.e.

$$\begin{aligned}
&\mathbb{E}_{GOE}^{N}\left[e^{-i\mathrm{Tr}MA}\mathbb{1}[i_{\leqslant x}(M)=k]\right]\\
=&\mathbb{E}_{GOE}^{N}\left[e^{-i\mathrm{Tr}MA}\mathbb{1}[i_{\leqslant x}(M)=k,\max\{|\lambda_i(M)|\}_{i=1}^N\leqslant R]\right]\\
+&\mathbb{E}_{GOE}^{N}\left[e^{-i\mathrm{Tr}MA}\mathbb{1}[i_{\leqslant x}(M)=k,\max\{|\lambda_i(M)|\}_{i=1}^N> R]\right] \qquad (3.216)
\end{aligned}$$

We will focus initially on the first expectation on the RHS of (3.216) and deal with the second term later. Let us abbreviate the notation using

$$\mathcal{I}_R(M)=\{\max\{|\lambda_i(M)|\}_{i=1}^N\leqslant R\}.$$

[11] See Remark 3.10 below.

Recall that A has finite rank and note that A is symmetric without loss of generality, since

$$\mathrm{Tr}M\frac{A+A^T}{2} = \frac{1}{2}\left(\mathrm{Tr}MA + \mathrm{Tr}MA^T\right) = \frac{1}{2}\left(\mathrm{Tr}MA + \mathrm{Tr}AM^T\right) = \mathrm{Tr}MA \tag{3.217}$$

and hence $A = \mathrm{diag}(a_1, \ldots, a_{r_A}, 0\ldots, 0)$ without loss of generality. We begin by factorising the symmetric matrix M in the GOE integral:

$$\begin{aligned}&\mathbb{E}_M\left[e^{-i\mathrm{Tr}MA}\mathbb{1}[i_{\leqslant x}(M)=k, \mathcal{I}_R(M)]\right]\\ =&\int\frac{d\mu_E(\Lambda)}{Z_N}\mathbb{1}[-R\leqslant\lambda_1\ldots\leqslant\lambda_k\leqslant x\leqslant\lambda_{k+1}\leqslant\ldots\lambda_N\leqslant R]\int d\mu_{Haar}(O)e^{-i\sum_{j=1}^{r_A}a_j\boldsymbol{o}_j^T\Lambda\boldsymbol{o}_j}\end{aligned} \tag{3.218}$$

where μ_E is the un-normalised joint density of ordered GOE eigenvalues, μ_{Haar} is the Haar measure on the orthogonal group $O(N)$, $\boldsymbol{o}_j$ are the rows of the orthogonal matrix O and Z_N is normalisation for the ordered GOE eigenvalues given by the Selberg integral:

$$Z_N = \frac{1}{N!}(2\sqrt{2})^N N^{-N(N+1)/4}\prod_{i=1}^N\Gamma\left(1+\frac{i}{2}\right). \tag{3.219}$$

Much like the proof of Theorem A.1 in [AAC13], we proceed by splitting the eigenvalues in (3.218) to enforce the constraint given by the indicator function:

$$\begin{aligned}&\mathbb{E}_M\left[e^{-i\mathrm{Tr}MA}\mathbb{1}[i_{\leqslant x}(M)=k,\mathcal{I}_R(M)]\right]\\ =&\int d\mu_{Haar}(O)\frac{1}{Z_N}\int_{[-R,x]^k}\prod_{i=1}^k\left(d\lambda_i e^{-N\lambda_i^2/2}\right)\Delta\left(\{\lambda_i\}_{i=1}^k\right)\mathbb{1}[\lambda_1\leqslant\ldots\leqslant\lambda_k]\\ &\int_{(x,R]^{N-k}}\prod_{i=k+1}^N\left(d\lambda_i e^{-N\lambda_i^2/2}\right)\Delta\left(\{\lambda_i\}_{i=k+1}^N\right)\mathbb{1}[\lambda_{k+1}\leqslant\ldots\leqslant\lambda_N]\\ &e^{-i\sum_{j=1}^{r_A}a_j\boldsymbol{o}_j^T\Lambda\boldsymbol{o}_j}\exp\left(\sum_{j=1}^k\sum_{\ell=k+1}^N\log|\lambda_j-\lambda_\ell|\right)\\ =&\int d\mu_{Haar}(O)\int_{[-R,x]^k}\prod_{i=1}^k\left(d\lambda_i e^{-N\lambda_i^2/2}\right)\Delta\left(\{\lambda_i\}_{i=1}^k\right)\frac{Z_{N-k}}{k!Z_N}\\ &\frac{1}{Z_{N-k}(N-k)!}\int_{(x,R]^{N-k}}\prod_{i=k+1}^N\left(d\lambda_i e^{-N\lambda_i^2/2}\right)\Delta\left(\{\lambda_i\}_{i=k+1}^N\right)\\ &e^{-i\sum_{j=1}^{r_A}a_j\boldsymbol{o}_j^T\Lambda\boldsymbol{o}_j}\exp\left(\sum_{j=1}^k\sum_{\ell=k+1}^N\log|\lambda_j-\lambda_\ell|\right)\\ =&\int_{[-R_N,x_N]^k}\prod_{i=1}^k\left(d\lambda_i e^{-(N-k)\lambda_i^2/2}\right)\Delta\left(\{\lambda_i\}_{i=1}^k\right)\\ &\int_{(x_N,R_N]^{N-k}}d\bar{\mu}_E(\Lambda_{N-k})\int d\mu_{Haar}(O)e^{-i\sum_{j=1}^{r_A}\sqrt{\frac{N-k}{N}}a_j\boldsymbol{o}_j^T\Lambda\boldsymbol{o}_j}\\ &\exp\left(\sum_{j=1}^k\sum_{\ell=k+1}^N\log|\lambda_j-\lambda_\ell|\right)\frac{Z_{N-k}}{k!Z_N}\left(\sqrt{\frac{N-k}{N}}\right)^{N+N(N+1)/2}\end{aligned} \tag{3.220}$$

where $x_N := \sqrt{\frac{N}{N-k}}x$, $R_N := \sqrt{\frac{N}{N-k}}R$ and $\bar{\mu}_E$ is the normalised joint density of un-ordered GOE eigenvalues.

We will first need to deal with the Itzykson-Zuber integral in (3.220) before dealing with the eigenvalue integrals. We follow [GM+05], in particular the proof of Theorem 7 therein. We have the well-known result (Fact 8 in [GM+05]) that in the sense of distributions

$$(\boldsymbol{o}_1,\ldots,\boldsymbol{o}_{r_A}) \sim \left(\frac{\tilde{\boldsymbol{g}}_1}{||\tilde{\boldsymbol{g}}_1||},\ldots,\frac{\tilde{\boldsymbol{g}}_{r_A}}{||\tilde{\boldsymbol{g}}_{r_A}||}\right) \tag{3.221}$$

where the $(\tilde{\boldsymbol{g}}_j)_{j=1}^{r_A}$ are constructed via the Gram-Schmidt process from $(\boldsymbol{g}_j)_{j=1}^{r_A} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\boldsymbol{0},1)$. (3.221) exactly gives

$$\int d\mu_{Haar}(O)e^{-i\sum_{j=1}^{r_A}\sqrt{\frac{N-k}{N}}a_j\boldsymbol{o}_j^T\Lambda\boldsymbol{o}_j} = \int\prod_{j=1}^{r_A}\frac{d\boldsymbol{g}_j}{\sqrt{2\pi}^N}e^{-\frac{\boldsymbol{g}_j^2}{2}}\exp\left(-i\sqrt{\frac{N-k}{N}}\sum_{j=1}^{r_A}a_j\frac{\tilde{\boldsymbol{g}}_j^T\Lambda\tilde{\boldsymbol{g}}_j}{||\tilde{\boldsymbol{g}}_j||^2}\right) \tag{3.222}$$

and we will now seek to replace the $\tilde{\boldsymbol{g}}_j$ with $\boldsymbol{g}_j$ via appropriate approximations. Introduce the event

$$B_N(v) := \left\{|N^{-1}\langle \boldsymbol{g}_i,\boldsymbol{g}_j\rangle - \delta_{ij}| \leqslant N^{-v}, \quad 1\leqslant i,j\leqslant r_A\right\} \tag{3.223}$$

and then from [GM+05] we immediately conclude that under the i.i.d Gaussian law of the $(\boldsymbol{g}_j)_{j=1}^{r_A}$ the complementary event has low probability:

$$\mathbb{P}(B_N(v)^c) = \mathcal{O}(C(v)e^{-\alpha N^{1-2v}}) \tag{3.224}$$

where $\alpha, C(v) > 0$ and we take $0 < v < \frac{1}{2}$ to make this statement meaningful. This enables us to write

$$\begin{aligned}&\int d\mu_{Haar}(O)e^{-i\sum_{j=1}^{r_A}\sqrt{\frac{N-k}{N}}a_j\boldsymbol{o}_j^T\Lambda\boldsymbol{o}_j}\\ &=\left(1+\mathcal{O}(e^{-\alpha N^{1-2v}})\right)\int\prod_{j=1}^{r_A}\frac{d\boldsymbol{g}_j}{\sqrt{2\pi}^N}e^{-\frac{\boldsymbol{g}_j^2}{2}}\exp\left(-i\sqrt{\frac{N-k}{N}}\sum_{j=1}^{r_A}a_j\frac{\tilde{\boldsymbol{g}}_j^T\Lambda\tilde{\boldsymbol{g}}_j}{||\tilde{\boldsymbol{g}}_j||^2}\right)\mathbb{1}\{B_N(v)\}.\end{aligned} \tag{3.225}$$

Again, directly from [GM+05], given $B_N(v)$ we have

$$||\tilde{\boldsymbol{g}}_j - \boldsymbol{g}_j|| \leqslant N^{\frac{1}{2}-\frac{v}{2}} \tag{3.226}$$

and therefore

$$||\tilde{\boldsymbol{g}}_j||^2 = N\left[1+N^{-1}\left(||\tilde{\boldsymbol{g}}_j||^2 - ||\boldsymbol{g}_j||^2\right) + \left(N^{-1}||\boldsymbol{g}_j||^2 - 1\right)\right] = N(1+\mathcal{O}(N^{-v})) \tag{3.227}$$

and

$$\begin{aligned}\tilde{\boldsymbol{g}}_j^T\Lambda\tilde{\boldsymbol{g}}_j &= \boldsymbol{g}^T\Lambda\boldsymbol{g} + \sum_{i=1}^N(\tilde{g}_i - g_i)^2\lambda_i + 2\sum_{i=1}^N g_i(\tilde{g}_i - g_i)\lambda_i\\ \implies \left|\frac{\tilde{\boldsymbol{g}}_j^T\Lambda\tilde{\boldsymbol{g}}_j}{||\tilde{\boldsymbol{g}}_j||^2} - \frac{\boldsymbol{g}_j^T\Lambda\boldsymbol{g}_j}{||\boldsymbol{g}_j||^2}\right| &\lesssim N^{-\frac{v}{2}}||\Lambda||_\infty.\end{aligned} \tag{3.228}$$

We see therefore that, in approximating the $\{\tilde{\boldsymbol{g}}_j\}_j$ by $\{\boldsymbol{g}_j\}_j$ in (3.225) we introduce an error term in the exponential that is uniformly small in the integration variables $\{\boldsymbol{g}_j\}_j$. Combining (3.225), (3.227) and (3.228) and noting that $||\Lambda||_\infty = R_N \sim R$ under the eigenvalue integral in (3.220) gives

$$
\begin{aligned}
&\int d\mu_{Haar}(O) e^{-i\sum_{j=1}^{r_A}\sqrt{\frac{N-k}{N}}a_j \boldsymbol{o}_j^T \Lambda \boldsymbol{o}_j} \\
&= \left(1+\mathcal{O}(N^{-\frac{v}{2}})\right)\int \prod_{j=1}^{r_A} \frac{d\boldsymbol{g}_j}{\sqrt{2\pi}^N} e^{-\frac{g_j^2}{2}} \exp\left(-i\sqrt{\frac{N-k}{N}}\sum_{j=1}^{r_A} a_j \frac{\boldsymbol{g}_j^T\Lambda \boldsymbol{g}_j}{N(1+\mathcal{O}(N^{-v}))}\right) \\
&= \prod_{j=1}^{r_A}\prod_{i=1}^{N}\left(1+2iN^{-1}a_j\lambda_i\right)^{-\frac{1}{2}}\left(1+\mathcal{O}(N^{-\frac{v}{2}})\right) \\
&= \exp\left\{-\frac{N-k}{2}\sum_{j=1}^{r_A}\int d\hat{\mu}_{N-k}(z)\log(1+2iN^{-1}a_j z)\right\} \\
&\quad \exp\left\{-\frac{1}{2}\sum_{j=1}^{r_A}\sum_{i=1}^{k}\log(1+2iN^{-1}a_j\lambda_i)\right\}\left(1+\mathcal{O}(N^{-\frac{v}{2}})\right) \qquad (3.229)
\end{aligned}
$$

where we have defined

$$
\hat{\mu}_{N-k} = \frac{1}{N-k}\sum_{i=k+1}^{N}\delta_{\lambda_i}. \qquad (3.230)
$$

Following [AAC13], we now introduce the following function

$$
\Phi(z,\mu) = -\frac{z^2}{2} + \int d\mu(z')\log|z-z'| \qquad (3.231)
$$

and so and then (3.220) and (3.229) can be rewritten as

$$
\begin{aligned}
&\mathbb{E}_M\left[e^{-i\mathrm{Tr}MA}\mathbb{1}[i_{\leqslant x}(M)=k,\mathcal{I}_R(M)]\right] \\
&= \int_{[-R_N,x_N]^k}\prod_{i=1}^{k}d\lambda_i\ \Delta\left(\{\lambda_j\}_{j=1}^k\right)\exp\left\{-\frac{1}{2}\sum_{j=1}^{r_A}\sum_{i=1}^{k}\log(1+2iN^{-1}a_j\lambda_i)\right\}\left(1+\mathcal{O}(N^{-\frac{v}{2}})\right) \\
&\quad \int_{(x_N,R_N]^{N-k}} d\bar{\mu}_E(\Lambda_{N-k})\exp\left\{-\frac{N-k}{2}\sum_{j=1}^{r_A}\int d\hat{\mu}_{N-k}(z)\log(1+2iN^{-1}a_j z)\right\} \\
&\quad \exp\left((N-k)\sum_{j=1}^{k}\Phi(\lambda_j,\hat{\mu}_{N-k})\right)\frac{Z_{N-k}}{k!Z_N}\left(\sqrt{\frac{N-k}{N}}\right)^{N+N(N+1)/2}. \qquad (3.232)
\end{aligned}
$$

We now appeal to the Coulomb gas method [CFV16] and in particular the formulation found in [Maj+11]. We replace the joint integral of $N-k$ eigenvalues in (3.232) with a functional integral over the continuum eigenvalues density:

$$
\begin{aligned}
&\int_{(x_N,R_N]^{N-k}} d\bar{\mu}_E(\Lambda_{N-K})\exp\left((N-k)\sum_{j=1}^{k}\Phi(\lambda_j,\hat{\mu}_{N-k})\right)\exp\left\{-\frac{N-k}{2}\sum_{j=1}^{r_A}\int d\hat{\mu}_{N-k}(z)\log\left(1+\frac{2ia_jz}{N}\right)\right\} \\
&= \int \mathcal{D}[\mu]e^{-N^2\mathcal{S}_x[\mu]}\exp\left((N-k)\sum_{j=1}^{k}\Phi(\lambda_j,\mu)\right)\exp\left\{-\frac{N-k}{2}\sum_{j=1}^{r_A}\int d\mu(z)\log\left(1+\frac{2ia_jz}{N}\right)\right\} \qquad (3.233)
\end{aligned}
$$

where the action is defined as

$$\begin{aligned}\mathcal{S}_x[\mu] =&\frac{1}{2}\int dz\mu(z)z^2 - \iint_{z\neq z} dzdz'\mu(z)\mu(z')\log|z-z'|\\&+A_1\left(\int dz\theta(R_N-z)\mu(z)-1\right)+A_2\left(\int dz\mu(z)\theta(z-x)-1\right)-\Omega\end{aligned}\tag{3.234}$$

where θ is the Heaviside step function, Ω is the constant resulting from the normalisation of the eigenvalue joint density and A_1, A_2 are Lagrange multipliers.

Owing to the N^2 rate in (3.233), the integral concentrates around the minimiser of the action. Since $x < -\sqrt{2}$ and we have chosen $R > |x|$, it is clear following [Maj+11] that the semi-circle law $\mu_{SC}(z) = \pi^{-1}\sqrt{2-z^2}$ minimises this action and further that $\mathcal{S}_x[\mu_{SC}] = 0$, so we have

$$\begin{aligned}&\int \mathcal{D}[\mu]e^{-N^2\mathcal{S}_x[\mu]}\exp\left((N-k)\sum_{j=1}^k\Phi(\lambda_j,\mu)\right)\exp\left\{-\frac{N-k}{2}\sum_{j=1}^{r_A}\int d\mu(z)\log(1+2iN^{-1}a_jz)\right\}\\=&\int_{B_\delta(\mu_{SC})}\mathcal{D}[\mu]e^{-N^2\mathcal{S}_x[\mu]}\exp\left((N-k)\sum_{j=1}^k\Phi(\lambda_j,\mu)\right)\exp\left\{-\frac{N-k}{2}\sum_{j=1}^{r_A}\int d\mu(z)\log(1+2iN^{-1}a_jz)\right\}\\&+e^{-N^2c_\delta}\mathcal{O}(1)\end{aligned}\tag{3.235}$$

where $\delta = \mathcal{O}(N^{-1})$ and $c_\delta > 0$ is some constant. Performing the usual Laplace method expansion of the action in (3.235) and re-scaling the first non-vanishing derivative to be $\mathcal{O}(1)$, it is clear that the action only contributes a real factor of $\mathcal{O}(1)$ that is independent of the dummy integration variables $\boldsymbol{x}_1, \boldsymbol{x}_2, \zeta_1, \zeta_1^\dagger, \zeta_2, \zeta_2^\dagger$ and the other eigenvalues $\lambda_1,\ldots,\lambda_k$ and can therefore be safely summarised as $\mathcal{O}(1)$. Whence

$$\begin{aligned}&\int \mathcal{D}[\mu]e^{-N^2\mathcal{S}_x[\mu]}\exp\left((N-k)\sum_{j=1}^k\Phi(\lambda_j,\mu)\right)\exp\left\{-\frac{N-k}{2}\sum_{j=1}^{r_A}\int d\mu(z)\log(1+2iN^{-1}a_jz)\right\}\\=&\mathcal{O}(1)\exp\left((N-k)\sum_{j=1}^k\Phi(\lambda_j,\mu_{SC})\right)\exp\left\{-\frac{N-k}{2}\sum_{j=1}^{r_A}\int d\mu_{SC}(z)\log(1+2iN^{-1}a_jz)\right\}\\&+e^{-N^2c_\delta}\mathcal{O}(1).\end{aligned}\tag{3.236}$$

Now elementary calculations give, noting that the integrand is uniformly convergent in N owing to the compact support of μ_{SC},

$$\begin{aligned}\int d\mu_{SC}(z)\log(1+2iN^{-1}a_jz) &= -\frac{2ia_j}{N}\int d\mu_{SC}(z)z+\frac{2a_j^2}{N^2}\int d\mu_{SC}(z)z^2+\mathcal{O}(a_j^3N^{-3})\\&=\frac{a_j^2}{N^2}(1+\mathcal{O}(a_jN^{-1}))\\\implies \frac{N-k}{2}\sum_{j=1}^{r_A}\int d\mu_{SC}(z)\log(1+2iN^{-1}a_jz) &= \frac{\mathrm{Tr}A^2}{2N}(1+\|A\|_\infty\mathcal{O}(N^{-1}))\end{aligned}\tag{3.237}$$

where we have implicitly assumed that the spectral radius $\|A\|_\infty \ll N$. This constraint can be introduced by restricting the domains of integration for $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ in the anaologue of (3.115) from all of $\mathbb{R}^N$ to balls of radius $o(\sqrt{N})$. It is a standard result for Gaussian integrals that this can be achieved at the cost of an exponentially smaller term. Summarising (3.232), (3.233), (3.236) and (3.237):

$$\begin{aligned}
&\mathbb{E}_M\left[e^{-i\mathrm{Tr}MA}\mathbb{1}[i_{\leqslant x}(M)=k,\mathcal{I}_R(M)]\right]\\
=&\int_{[-R_N,x_N]^k}\prod_{i=1}^k d\lambda_i\ \Delta\left(\{\lambda_j\}_{j=1}^k\right)\exp\left\{-\frac{1}{2}\sum_{j=1}^{r_A}\sum_{i=1}^k\log(1+2iN^{-1}a_j\lambda_i)\right\}\exp\left((N-k)\sum_{j=1}^k\Phi(\lambda_j,\mu_{SC})\right)\\
&e^{-\frac{\mathrm{Tr}A^2}{2N}}\left(\mathcal{O}(1)+\mathcal{O}(N^{-\frac{v}{2}})+\mathcal{O}(N^{-1})||A||_\infty\right)\frac{Z_{N-k}}{k!Z_N}\left(\sqrt{\frac{N-k}{N}}\right)^{N+N(N+1)/2}\\
=&\int_{[-R_N,x_N]^k}\prod_{i=1}^k d\lambda_i\ \Delta\left(\{\lambda_j\}_{j=1}^k\right)\exp\left((N-k)\sum_{j=1}^k\Phi(\lambda_j,\mu_{SC})\right)\\
&e^{-\frac{\mathrm{Tr}A^2}{2N}}\left(\mathcal{O}(1)+\mathcal{O}(N^{-\frac{v}{2}})+\mathcal{O}(N^{-1})||A||_\infty\right)\frac{Z_{N-k}}{k!Z_N}\left(\sqrt{\frac{N-k}{N}}\right)^{N+N(N+1)/2}\\
=&\int_{[-R_N,x_N]^k}\prod_{i=1}^k d\lambda_i\ \Delta\left(\{\lambda_j\}_{j=1}^k\right)\exp\left((N-k)\sum_{j=1}^k\Phi(\lambda_j,\mu_{SC})\right)\\
&e^{-\frac{\mathrm{Tr}A^2}{2N}}\mathcal{O}(1)\frac{Z_{N-k}}{k!Z_N}\left(\sqrt{\frac{N-k}{N}}\right)^{N+N(N+1)/2}
\end{aligned}\tag{3.238}$$

where in the second equality we have Taylor expanded the remaining logarithm and summarised the result with another factor of $(1+\mathcal{O}(N^{-1})||A||_\infty)$.

We now wish to follow the proof of Theorem A.1 in [AAC13] and use $\Delta(\{\lambda_j\}_{j=1}^k)\leqslant(2R_N)^k\leqslant(3R)^k$ for $\lambda_j\in[-R_N,R_N]$ with bound (3.238), however the expectation on the left hand side of (3.238) is not necessarily real. We do however know that the $\mathcal{O}(1)$ term in (3.238) is real to leading order and so we can write

$$\mathbb{E}_M\left[e^{-i\mathrm{Tr}MA}\mathbb{1}[i_{\leqslant x}(M)=k,\mathcal{I}_R(M)]\right]=\Re\mathbb{E}_M\left[e^{-i\mathrm{Tr}MA}\mathbb{1}[i_{\leqslant x}(M)=k,\mathcal{I}_R(M)]\right](1+io(1))\tag{3.239}$$

and thence focus on bounding the real part of the expectation to obtain

$$\begin{aligned}
&\Re\mathbb{E}_M\left[e^{-i\mathrm{Tr}MA}\mathbb{1}[i_{\leqslant x}(M)=k,\mathcal{I}_R(M)]\right]\\
\leqslant&K(3R)^k\frac{Z_{N-k}}{k!Z_N}\left(\sqrt{\frac{N-k}{N}}\right)^{N+N(N+1)/2}\quad e^{-\frac{\mathrm{Tr}A^2}{2N}}\left(\int_{-R_N}^{x_N}dze^{(N-k)\Phi(z,\mu)}\right)^k
\end{aligned}\tag{3.240}$$

where we have exchanged $\mathcal{O}(1)$ terms for some appropriate constant K. Continuing to bound (3.240):

$$\begin{aligned}&\Re\mathbb{E}_M\left[e^{-i\mathrm{Tr}MA}\mathbb{1}[i_{\leqslant x}(M)=k,\mathcal{I}_R(M)]\right]\\ \leqslant&K(3R)^{2k}\frac{Z_{N-k}}{k!Z_N}\left(\sqrt{\frac{N-k}{N}}\right)^{N+N(N+1)/2}e^{-\frac{\mathrm{Tr}A^2}{2N}}\exp\left(k(N-k)\sup_{\substack{z\in[-2R,x]\\ \nu\in B_\delta(\mu_{SC})}}\Phi(z,\nu)\right)\\ \leqslant&K(3R)^{2k}\frac{Z_{N-k}}{k!Z_N}\left(\sqrt{\frac{N-k}{N}}\right)^{N+N(N+1)/2}e^{-\frac{\mathrm{Tr}A^2}{2N}}e^{-k(N-k)(1/2+I_1(x;\sqrt{2}))}\end{aligned} \tag{3.241}$$

where we have used the same result as used around (A.18) in [AAC13] to take the supremum.

Recalling (3.216), we can now use (3.241) and the GOE large deviations principle [ADG01] as in [AAC13] to obtain

$$\Re\mathbb{E}_M\left[e^{-i\mathrm{Tr}MA}\mathbb{1}[i_{\leqslant x}(M)=k]\right]\leqslant K''(3R)^k\frac{Z_{N-k}}{k!Z_N}\left(\sqrt{\frac{N-k}{N}}\right)^{N+N(N+1)/2}e^{-k(N-k)(1/2+I_1(x;\sqrt{2}))}e^{-\frac{1}{2N}\mathrm{Tr}A^2}+e^{-NR^2} \tag{3.242}$$

We now seek to obtain a complementary lower bound and again follow [AAC13] in choosing some y and R' such that $y<x<R'<-\sqrt{2}$. We then, following a similar procedure as above, find

$$\Re\mathbb{E}_M\left[e^{-i\mathrm{Tr}MA}\mathbb{1}[i_{\leqslant x}(M)=k]\right]\geqslant\tilde{K}\frac{Z_{N-k}}{k!Z_N}\left(\sqrt{\frac{N-k}{N}}\right)^{N+N(N+1)/2}e^{-\frac{1}{2N}\mathrm{Tr}A^2}\exp\left(k(N-k)\sup_{\substack{z\in[y,x]\\ \nu\in B_\delta(\mu_{SC})}}\Phi(z,\nu)\right) \tag{3.243}$$

and taking $y\nearrow x$ we obtain the complement to (3.242):

$$\Re\mathbb{E}_M\left[e^{-i\mathrm{Tr}MA}\mathbb{1}[i_{\leqslant x}(M)=k]\right]\geqslant\tilde{K}\frac{Z_{N-k}}{k!Z_N}\left(\sqrt{\frac{N-k}{N}}\right)^{N+N(N+1)/2}e^{-k(N-k)(1/2+I_1(x;\sqrt{2}))}e^{-\frac{1}{2N}\mathrm{Tr}A^2}. \tag{3.244}$$

Next we need the asymptotic beahviour of the Selberg term in (3.242) and (3.244)

$$T_{N,k}:=\frac{Z_{N-k}}{k!Z_N}\left(\sqrt{\frac{N-k}{N}}\right)^{N+N(N+1)/2}=\underbrace{\frac{Z_{N-k}(N-k)!}{Z_NN!}\left(\frac{N-k}{N}\right)^{\frac{(N-k)(N-k+1)}{4}}}_{T'_{N,k}}\frac{N!}{(N-k)!k!}\left(\frac{N-k}{N}\right)^{\frac{N}{2}+\frac{N(N+1)-(N-k)(N-k+1)}{4}}. \tag{3.245}$$

The term $T'_{N,k}$ appears in [AAC13] (defined in A.13) and it is shown there that

$$\lim_{N\to\infty}N^{-1}\log T'_{N,k}=\frac{k}{2}. \tag{3.246}$$

Clearly

$$\lim_{N\to\infty} N^{-1}\log\frac{N!}{(N-k)!k!} = 0 \tag{3.247}$$

and it is simple to show that

$$\lim_{N\to\infty}\left(\frac{N-k}{N}\right)^{\frac{N}{2}+\frac{N(N+1)-(N-k)(N-k+1)}{4}} = e^{-\frac{k(k+1)}{2}} \tag{3.248}$$

and so we have overall

$$\lim_{N\to\infty} N^{-1}\log T_{N,k} = \frac{k}{2}. \tag{3.249}$$

So absorbing any $\mathcal{O}(1)$ terms into constants K_L and K_U we have

$$K_L e^{-kN(1+o(1))I_1(x;\sqrt{2})}e^{-\frac{1}{2N}\mathrm{Tr}A^2} \leqslant \Re\mathbb{E}_M\left[e^{-i\mathrm{Tr}MA}\mathbb{1}[i_{\leqslant x}(M)=k]\right] \leqslant K_U e^{-kN(1+o(1))I_1(x;\sqrt{2})}e^{-\frac{1}{2N}\mathrm{Tr}A^2} \tag{3.250}$$

Set $S = s_1\boldsymbol{e}_1\boldsymbol{e}_1^T + s_2\boldsymbol{e}_2\boldsymbol{e}_2^T$ and $S_1 = s_1\boldsymbol{e}_1\boldsymbol{e}_1^T$. Suppose $s_1>0$ and $s_2>0$. By the interlacing property of eigenvalues, we have

$$\lambda_1^{(M)} \leqslant \lambda_1^{(M+S_1)} \leqslant \lambda_2^{(M)} \leqslant \ldots \leqslant \lambda_k^{(M)} \leqslant \lambda_k^{(M+S_1)} \leqslant \lambda_{k+1}^{(M)} \leqslant \lambda_{k+1}^{(M+S_1)} \leqslant \ldots \leqslant \lambda_N^{(M)} \leqslant \lambda_N^{(M+S_1)} \tag{3.251}$$

Therefore we have

$$\begin{cases} \{i_{\leqslant x}(M)=k\} \subset \{i_{\leqslant x}(M+S_1)\in\{k-1,k\}\} \subset \{i_{\leqslant x}(M)\in\{k-1,k,k+1\}\} = \bigsqcup_{j=-1}^{1}\{i_{\leqslant x}(M)=k+j\} & \text{for } k>0, \\ \{i_{\leqslant x}(M)=k\} \subset \{i_{\leqslant x}(M+S_1)=k\} \subset \{i_{\leqslant x}(M)\in\{k,k+1\}\} = \bigsqcup_{j=0}^{1}\{i_{\leqslant x}(M)=k+j\} & \text{for } k=0, \end{cases} \tag{3.252}$$

and so (3.250) gives

$$\begin{aligned} K_L e^{-kN(1+o(1))I_1(x;\sqrt{2})}e^{-\frac{1}{2N}\mathrm{Tr}A^2} \leqslant& \Re\mathbb{E}_M\left[e^{-i\mathrm{Tr}MA}\mathbb{1}[i_{\leqslant x}(M+S_1)\in\{k-1,k\}]\right] \\ \leqslant& 3K_U e^{-(k-1)N(1+o(1))I_1(x;\sqrt{2})}e^{-\frac{1}{2N}\mathrm{Tr}A^2}, \\ e^{-\frac{1}{2N}\mathrm{Tr}A^2} \leqslant& \Re\mathbb{E}_M\left[e^{-i\mathrm{Tr}MA}\mathbb{1}[i_{\leqslant x}(M+S_1)=0]\right] \leqslant 2K_U e^{-\frac{1}{2N}\mathrm{Tr}A^2}. \end{aligned}$$

We can then extend to S likewise by observing that interlacing gives

$$\{i_{\leqslant x}(M+S_1)\in\{k,k+1\}\} \subset \{i_{\leqslant x}(M+S)\in\{k-1,k,k+1\}\} \subset \{i_{\leqslant x}(M+S_1)\in\{k-1,k,k+1,k+2\}\} \tag{3.253}$$

and iterating using (3.252) yields

$$\begin{cases} \{i_{\leqslant x}(M)=k+1\} \subset \{i_{\leqslant x}(M+S)\in\{k-1,k,k+1\}\} \subset \bigsqcup_{j=-1}^{3}\{i_{\leqslant x}(M)=k+j\}, & \text{for } k>0 \\ \{i_{\leqslant x}(M)=k+1\} \subset \{i_{\leqslant x}(M+S)\in\{k,k+1\}\} \subset \bigsqcup_{j=0}^{3}\{i_{\leqslant x}(M)=k+j\}, & \text{for } k=0 \end{cases} \tag{3.254}$$

and (3.250) then gives

$$\begin{aligned}K_L e^{-(k+1)N(1+o(1))I_1(x;\sqrt{2})}e^{-\frac{1}{2N}\text{Tr}A^2} &\leqslant \Re\mathbb{E}_M\left[e^{-i\text{Tr}MA}\mathbb{1}[i_{\leqslant x}(M+S)\in\{k-1,k,k+1\}]\right]\\ &\leqslant 5K_U e^{-(k-1)N(1+o(1))I_1(x;\sqrt{2})}e^{-\frac{1}{2N}\text{Tr}A^2} \qquad (3.255)\\ K_L e^{-N(1+o(1))I_1(x;\sqrt{2})}e^{-\frac{1}{2N}\text{Tr}A^2} &\leqslant \Re\mathbb{E}_M\left[e^{-i\text{Tr}MA}\mathbb{1}[i_{\leqslant x}(M+S)\in\{0,1\}]\right]\leqslant 4K_U e^{-\frac{1}{2N}\text{Tr}A^2}.\end{aligned}$$

If instead the signs of s_1, s_2 are different, then the interlacing will be in the reverse orders, but the conclusion of (3.255) will be unchanged. Finally using (3.239) in the analogue of (3.111)

$$\begin{aligned}&\mathbb{E}_M[|\det(M-xI+S)|\mathbb{1}[i_{\leqslant x}(M+S)\in\{k-1,k,k+1\}]]\\ =&\Re\mathbb{E}_M[|\det(M-xI+S)|\mathbb{1}[i_{\leqslant x}(M+S)\in\{k-1,k,k+1\}]]\\ =&\Re\left\{K_N^{(1)}\lim_{\epsilon\searrow 0}\int d\boldsymbol{x}_1 d\boldsymbol{x}_2 d\zeta_1 d\zeta_1^\dagger d\zeta_2 d\zeta_2^\dagger \exp\left\{-i\boldsymbol{x}_1^T(M-(x+i\epsilon)I+S)\boldsymbol{x}_1 - i\boldsymbol{x}_2^T(M-(x-i\epsilon)I+S)\boldsymbol{x}_2\right\}\right.\\ &\quad\exp\left\{i\zeta_1^\dagger(M-(x+i\epsilon)I+S)\zeta_1 + i\zeta_2^\dagger(M-(x-i\epsilon)I+S)\zeta_2\right\} \qquad (3.256)\\ &\quad\left.\mathbb{E}_M\left[e^{-i\text{Tr}MA}\mathbb{1}[i_{\leqslant x}(M+S)\in\{k-1,k,k+1\}]\right]\right\}\\ =&\Re\left\{K_N^{(1)}\lim_{\epsilon\searrow 0}\int d\boldsymbol{x}_1 d\boldsymbol{x}_2 d\zeta_1 d\zeta_1^\dagger d\zeta_2 d\zeta_2^\dagger \exp\left\{-i\boldsymbol{x}_1^T(M-(x+i\epsilon)I+S)\boldsymbol{x}_1 - i\boldsymbol{x}_2^T(M-(x-i\epsilon)I+S)\boldsymbol{x}_2\right\}\right.\\ &\quad\exp\left\{i\zeta_1^\dagger(M-(x+i\epsilon)I+S)\zeta_1 + i\zeta_2^\dagger(M-(x-i\epsilon)I+S)\zeta_2\right\} \qquad (3.257)\\ &\quad\left.\Re\mathbb{E}_M\left[e^{-i\text{Tr}MA}\mathbb{1}[i_{\leqslant x}(M+S)\in\{k-1,k,k+1\}]\right](1+io(1))\right\}\\ =&\Re\left\{K_N^{(1)}\lim_{\epsilon\searrow 0}\int d\boldsymbol{x}_1 d\boldsymbol{x}_2 d\zeta_1 d\zeta_1^\dagger d\zeta_2 d\zeta_2^\dagger \exp\left\{-i\boldsymbol{x}_1^T(M-(x+i\epsilon)I+S)\boldsymbol{x}_1 - i\boldsymbol{x}_2^T(M-(x-i\epsilon)I+S)\boldsymbol{x}_2\right\}\right.\\ &\quad\exp\left\{i\zeta_1^\dagger(M-(x+i\epsilon)I+S)\zeta_1 + i\zeta_2^\dagger(M-(x-i\epsilon)I+S)\zeta_2\right\} \qquad (3.258)\\ &\quad\left.\Re\mathbb{E}_M\left[e^{-i\text{Tr}MA}\mathbb{1}[i_{\leqslant x}(M+S)\in\{k-1,k,k+1\}]\right]\right\}(1+io(1)) \qquad (3.259)\end{aligned}$$

From this point on, the proof proceeds, *mutatis mutandis*, as that for Lemma 3.7 but applied to the upper and lower bounds on (3.259) obtained from (3.255). The final range of integration for p_1 and p_2 will be some intervals $(0, o(1))$ owing to the change of variables used around (3.133), but this does not affect the ensuing asymptotics in which the p_1, p_2 integration contours are deformed through the saddle point at $z_{U,L}^{(-)}$. ■

Remark 3.10. We note that if an appropriate generating function for $\mathbb{1}\left[i_{\leqslant x}(M+S)=k\right]$ could be found, that would allow for a straightforward taking of the expectation in (3.215), then the calculations of Lemma 3.7 could be modified to include this extra term and then the desired expectation $\mathbb{E}_{GOE}^N\left[|\det(M-xI+S)|\mathbb{1}[i_{\leqslant x}(M+S)=k]\right]$ could be read-off in comparison with the result of Lemma 3.7.

We have established all we need to prove Theorem 3.3.

Theorem 3.3. *Recall the definition of $C^h_{N,\mathcal{K}}$ in (3.57) and let $\Theta_{H,k}$ be defined as in [AAC13]:*

$$\Theta_{H,k}(u) = \begin{cases} \frac{1}{2}\log(H-1) - \frac{H-2}{4(H-1)}u^2 - (k+1)I_1(u;E_\infty) & \text{if } u \leqslant -E_\infty, \\ \frac{1}{2}\log(H-1) - \frac{H-2}{H} & \text{if } u > -E_\infty, \end{cases} \tag{3.61}$$

then, with $\mathcal{K} = \{k-1, k, k+1\}$ for $k > 0$,

$$\Theta_{H,k+1}(u) \leqslant \lim_{N\to\infty} \frac{1}{N}\log\mathbb{E}C^h_{N,\mathcal{K}}(\sqrt{N}u) \leqslant \Theta_{H,k-1}(u) \tag{3.62}$$

and similarly with $\mathcal{K} = \{0, 1\}$

$$\Theta_{H,1}(u) \leqslant \lim_{N\to\infty} \frac{1}{N}\log\mathbb{E}C^h_{N,\mathcal{K}}(\sqrt{N}u) \leqslant \Theta_{H,0}(u). \tag{3.63}$$

Proof. First consider $u < -E_\infty$. The proof proceeds just as that of Theorem 3.2 but applying Lemma 3.9 instead of Lemma 3.7 and working identically on the upper and lower bounds from Lemma 3.9.

Now consider $u > -E_\infty$. By the interlacing property as used around (3.251), $i_{\leqslant x}(M)$ and $i_{\leqslant x}(M+S)$ differ by no more than 2. Hence

$$i_{\leqslant x}(M+S) \in \mathcal{K} \implies i_{\leqslant x}(M) = \mathcal{O}(1) \tag{3.260}$$

but for $0 > x > -\sqrt{2}$, and $M \sim GOE_N$, the large deviations principle for the GOE [AG97] gives

$$\mathbb{P}(i_{\leqslant x}(M) = \mathcal{O}(1)) \leqslant e^{-cN^2} \tag{3.261}$$

for some constant c, hence the x integral analogous to (3.166) is exponentially suppressed with quadratic speed in N for $x > -\sqrt{2}$. But we have already seen that the integral is only suppressed with linear speed in N for $x < -\sqrt{2}$, and further that $\Theta_{H,k}(u)$ is increasing on $(-\infty, -E_\infty)$ and so, by the Laplace principle, the leading order contribution is from around $x = -\sqrt{2}$ and so

$$\lim_{N\to\infty} \frac{1}{N}\log\mathbb{E}C^h_{N,\mathcal{K}}(\sqrt{N}u) = \lim_{N\to\infty} \frac{1}{N}\log\mathbb{E}C^h_{N,\mathcal{K}}(-\sqrt{N}E_\infty) \tag{3.262}$$

for $u > -E_\infty$, which completes the proof.

■

Remark 3.11. We are clearly unable to provide an exact leading term for $C^h_{N,\mathcal{K}}(\sqrt{N}u)$ for any value of u as we did for $C^h_N(\sqrt{N}u)$ for $u < -E_\infty$ in Theorem 3.4 because the presence of S in $i_{\leqslant x}(M+S)$ has forced us in Lemma 3.9 to resort to upper and lower bounds on the leading order term. We note that in [AAC13] the authors are also not able to obtain the exact leading term in this case by their rather different methods. Recalling Remark 3.10, we conjecture that this term could be obtained by variants of our methods if only a suitable (perhaps approximate) generating function for $\mathbb{1}[i_{\leqslant x}(M+S) = k]$ could be discovered.

3.6 Low rank perturbation of a matrix identity

In this section we establish a modified version of Theorem I from [FS02] required in the proof in Lemma 3.7. In that Lemma, we are faced with an integral of the form

$$\mathcal{I}_N(F;S) = \iint_{\mathbb{R}^N} d\boldsymbol{x}_1 d\boldsymbol{x}_2 F(Q_B) e^{-iN\mathrm{Tr}SB} \tag{3.263}$$

where the $N \times N$ matrix B is defined as $B = \boldsymbol{x}_1\boldsymbol{x}_1^T + \boldsymbol{x}_2\boldsymbol{x}_2^T$, the 2×2 matrix Q_B is given by

$$Q_B = \begin{pmatrix} \boldsymbol{x}_1^T\boldsymbol{x}_1 & \boldsymbol{x}_1^T\boldsymbol{x}_2 \\ \boldsymbol{x}_2^T\boldsymbol{x}_1 & \boldsymbol{x}_2^T\boldsymbol{x}_2 \end{pmatrix}, \tag{3.264}$$

F is some suitably nice function and S is some real symmetric matrix of rank $r = \mathcal{O}(1)$ as $N \to \infty$ and with non-zero eigenvalues $\{N^{-1/2}s_i\}_{i=1}^r$ for $s_i = \mathcal{O}(1)$. It is sufficient to be able to evaluate a leading order term of $\mathcal{I}_N$ in an expansion for large N. [FS02] proves the following related result:

Lemma 3.10 ([FS02] Theorem I). *Given m vectors in $\mathbb{R}^N$ $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m$, denote by $Q(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m)$ the $m \times m$ matrix whose entries are given by $Q_{ij} = \boldsymbol{x}_i^T\boldsymbol{x}_j$. Let F be any function of an $m \times m$ matrix such that the integral*

$$\int_{\mathbb{R}^N}\cdots\int_{\mathbb{R}^N} d\boldsymbol{x}_1\ldots d\boldsymbol{x}_m |F(Q)| \tag{3.265}$$

exists and define the integral

$$\mathcal{J}_{N,m}(F) := \int_{\mathbb{R}^N}\cdots\int_{\mathbb{R}^N} d\boldsymbol{x}_1\ldots d\boldsymbol{x}_m F(Q). \tag{3.266}$$

Then we have

$$\mathcal{J}_{N,m}(F) = \frac{\pi^{\frac{m}{2}\left(N-\frac{m-1}{2}\right)}}{\prod_{k=0}^{m-1}\Gamma\left(\frac{N-k}{2}\right)} \int_{Sym_{\geqslant 0}(m)} d\hat{Q}\left(\det\hat{Q}\right)^{\frac{N-m-1}{2}} F(\hat{Q}). \tag{3.267}$$

We will prove the following perturbed version of this result and in greater generality than is required in the present work.

Lemma 3.6. *Given m vectors in $\mathbb{R}^N$ $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m$, denote by $Q(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m)$ the $m \times m$ matrix whose entries are given by $Q_{ij} = \boldsymbol{x}_i^T\boldsymbol{x}_j$. Let F be any function of an $m \times m$ matrix such that the integral*

$$\int_{\mathbb{R}^N}\cdots\int_{\mathbb{R}^N} d\boldsymbol{x}_1\ldots d\boldsymbol{x}_m |F(Q)| \tag{3.99}$$

exists, and let S be a real symmetric $N \times N$ matrix of fixed rank r and with non-zero eigenvalues $\{N^\alpha s_i\}_{i=1}^r$ for some $\alpha < 1/2$. Define the integral

$$\mathcal{J}_{N,m}(F;S) := \int_{\mathbb{R}^N}\cdots\int_{\mathbb{R}^N} d\boldsymbol{x}_1\ldots d\boldsymbol{x}_m F(Q) e^{-iN\sum_{i=1}^N \boldsymbol{x}_i^T S\boldsymbol{x}_i}. \tag{3.100}$$

Then as $N \to \infty$ we have

$$\mathcal{J}_{N,m}(F;S) = (1+o(1)))\frac{\pi^{\frac{m}{2}\left(N-\frac{m-1}{2}\right)}}{\prod_{k=0}^{m-1}\Gamma\left(\frac{N-k}{2}\right)} \int_{Sym_{\geqslant 0}(m)} d\hat{Q}\left(\det\hat{Q}\right)^{\frac{N-m-1}{2}} F(\hat{Q}) \prod_{i=1}^N\prod_{j=1}^r \left(1+2iN^\alpha\hat{Q}_{ii}s_j\right)^{-1/2}. \tag{3.101}$$

Proof. The proof of Lemma 3.10 presented in Appendix D of [FS02] proceeds by induction on m and relies on writing the integration vector $\boldsymbol{x}_m$ as $\boldsymbol{x}_m = \rho_m O_m \boldsymbol{e}_N$ where $\boldsymbol{e}_N$ is the N-th basis vector in the chosen orthonormal basis, $\rho_m > 0$ is a scalar variable and O_m is an orthogonal matrix. The proof proceeds by making a change of variables for the first $m-1$ integration vectors and then finding that the integrand does not depend on O_m and so the integral over O_m with respect to the Haar measure just contributes a volume factor of

$$\frac{2\pi^{N/2}}{\Gamma(N/2)}. \tag{3.268}$$

It is at this point where the $e^{-iN\text{Tr}SB}$ term in (3.263) causes problems because a dependence on O_m remains. Indeed, we have

$$\boldsymbol{x}_m^T S \boldsymbol{x}_m = \rho_m \boldsymbol{e}_N^T O_m^T S O_m \boldsymbol{e}_N. \tag{3.269}$$

Since S is real symmetric we may take, WLOG, $S = N^{\alpha}\text{diag}(s_1, \ldots, s_r, 0, \ldots, 0)$. Then

$$e^{-iN\boldsymbol{x}_m^T S \boldsymbol{x}_m} = e^{-iN^{1+\alpha}\rho_m \sum_{j=1}^r s_j (o_{Nj})^2} \tag{3.270}$$

where o_{Nj} is the j-th component of the N-th column of O. Proceeding with an evaluation of an integral like (3.263) then requires the evaluation of the integral

$$\int_{O(N)} d\mu_{\text{Haar}}(O_m) e^{-iN^{1+\alpha}\rho_m \sum_{j=1}^r s_j (o_{Nj})^2}. \tag{3.271}$$

We can now follow [GM+05], in particular the proof of Theorem 7 therein. We have the well-known result (Fact 8 in [GM+05]) that in the sense of distributions

$$(\boldsymbol{o}_1, \ldots, \boldsymbol{o}_p) \sim \left(\frac{\tilde{\boldsymbol{g}}_1}{||\tilde{\boldsymbol{g}}_1||}, \ldots, \frac{\tilde{\boldsymbol{g}}_p}{||\tilde{\boldsymbol{g}}_p||}\right) \tag{3.272}$$

for any $p = \mathcal{O}(1)$ and where the $(\tilde{\boldsymbol{g}}_j)_{j=1}^p$ are constructed via the Gram-Schmidt process from $(\boldsymbol{g}_j)_{j=1}^{r_A} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\boldsymbol{0}, 1)$. So in particular

$$\boldsymbol{o}_N \sim \frac{\boldsymbol{g}}{||\boldsymbol{g}||}, \ \ \boldsymbol{g} \sim \mathcal{N}(0, 1). \tag{3.273}$$

(3.273) then exactly gives

$$\int_{O(N)} d\mu_{\text{Haar}}(O_m) e^{-iN^{1+\alpha}\rho_m \sum_{j=1}^r s_j (o_{Nj})^2} = \int_{\mathbb{R}^N} \frac{d\boldsymbol{g}}{(2\pi)^{N/2}} e^{-\frac{\boldsymbol{g}^2}{2}} \exp\left(-iN^{1+\alpha}\rho_m \sum_{j=1}^r s_j \frac{g_j^2}{||\boldsymbol{g}||^2}\right) \tag{3.274}$$

Introduce the event

$$B_N(v) := \left\{|N^{-1}\langle \boldsymbol{g}, \boldsymbol{g}\rangle - 1| \leqslant N^{-v}\right\} \tag{3.275}$$

and then from [GM+05] we immediately conclude that under the i.i.d Gaussian law of $\boldsymbol{g}$ the complementary event has low probability:

$$\mathbb{P}(B_N(v)^c) = \mathcal{O}(C(v)e^{-\beta N^{1-2v}}) \tag{3.276}$$

where $\beta, C(v) > 0$ and we take $0 < v < \frac{1}{2}$ to make this statement meaningful. This enables us to write

$$\begin{aligned}
&\int_{O(N)} d\mu_{\text{Haar}}(O_m) e^{-iN^{1+\alpha}\rho_m \sum_{j=1}^{r} s_j (o_{Nj})^2} \\
&= \left(1 + \mathcal{O}(e^{-\beta N^{1-2v}})\right) \int_{\mathbb{R}^N} \frac{dg}{(2\pi)^{N/2}} e^{-\frac{g^2}{2}} \exp\left(-iN^{1+\alpha}\rho_m \sum_{j=1}^{r} s_j \frac{g_j^2}{||g||^2}\right) \mathbb{1}\{B_N(v)\} \\
&= \left(1 + \mathcal{O}(e^{-\beta N^{1-2v}})\right) \int_{\mathbb{R}^N} \frac{dg}{(2\pi)^{N/2}} \mathbb{1}\{B_N(v)\} \\
&\qquad e^{-\frac{g^2}{2}} \exp\left(-iN^{\alpha}(1 + \mathcal{O}(N^{-v}))\rho_m \sum_{j=1}^{r} s_j g_j^2\right) \qquad (3.277)
\end{aligned}$$

but given $B_N(v)$ we have $g_j^2 \lesssim N$ for all $j = 1, \ldots, N$ and so we do not, as it stands, have uniformly small error terms. We can circumvent this by introducing the following event for $0 < \eta < \frac{1}{2}$:

$$E_N^{(r)}(\eta) = \{|g_j| \leqslant N^{\frac{1}{2}-\eta} \ \text{ for } j = 1, \ldots, r\}. \qquad (3.278)$$

Let us use $\hat{g}$ to denote the $N - r$ dimensional vector with components $(g_{r+1}, \ldots, g_N)$.

Then we have

$$\left| |N^{-1}||\hat{g}||^2 - 1| - N^{-1} \sum_{i=1}^{r} g_j^2 \right| \leqslant |N^{-1}||g||^2 - 1| \leqslant |N^{-1}||\hat{g}||^2 - 1| + N^{-1} \sum_{i=1}^{r} g_j^2 \qquad (3.279)$$

so if $\eta > \frac{v}{2}$ then it follows that

$$B_N(v) \mid E_N^{(r)}(\eta) = B_{N-r}(v). \qquad (3.280)$$

But we also have (e.g. [AAR99] Appendix C)

$$\mathbb{P}(E_N^{(r)}(\eta)) = \left[\text{erf}\left(N^{\frac{1}{2}-\eta}\right)\right]^r = \left[1 - \mathcal{O}(N^{\frac{1}{2}-\eta} e^{-N^{1-2\eta}})\right]^r = 1 - \mathcal{O}(N^{\frac{1}{2}-\eta} e^{-N^{1-2\eta}}) \qquad (3.281)$$

and so (taking $\eta > v$, say)

$$\mathbb{P}\left(B_N(v) \cap E_N^{(r)}(\eta)\right) = \mathbb{P}\left(B_N(v) \mid E_N^{(r)}(\eta)\right) \mathbb{P}\left(E_N^{(r)}(\eta)\right) = 1 - \mathcal{O}(e^{-\alpha N^{1-2v}}) \qquad (3.282)$$

and thus we can replace (3.277) with

$$\begin{aligned}
\int_{O(N)} d\mu_{\text{Haar}}(O_m) e^{-iN^{1+\alpha}\rho_m \sum_{j=1}^{r} s_j (o_{Nj})^2} &= \left(1 + \mathcal{O}(e^{-\beta N^{1-2v}})\right) \int_{\mathbb{R}^N} \frac{dg}{(2\pi)^{N/2}} \mathbb{1}\{B_N(v) \cap E_N^{(r)}(\eta)\} \\
&\qquad e^{-\frac{g^2}{2}} \exp\left(-iN^{\alpha}(1 + \mathcal{O}(N^{-v}))\rho_m \sum_{j=1}^{r} s_j g_j^2\right)
\end{aligned} \qquad (3.283)$$

but now $N^{\alpha-v} g_j^2 \leqslant N^{\alpha+1-v-2\eta} \leqslant N^{\alpha+1-3v} \to 0$ as $N \to \infty$ so long as we choose $v > \frac{\alpha+1}{3}$. Given that $\alpha < 1/2$, this choice is always possible for $0 < v < 1/2$. Thus the error term in the exponent of (3.283) is in fact uniformly small in $\boldsymbol{g}$ and so we obtain

$$\begin{aligned}
&\int_{O(N)} d\mu_{\text{Haar}}(O_m) e^{-iN^{1+\alpha}\rho_m \sum_{j=1}^{r} s_j (o_{Nj})^2} \\
&= (1+o(1)) \int_{\mathbb{R}^N} \frac{d\boldsymbol{g}}{(2\pi)^{N/2}} \mathbb{1}\{B_N(v) \cap E_N^{(r)}(\eta)\} \exp\left(-\frac{\boldsymbol{g}^2}{2} - iN^{\alpha}\rho_m \sum_{j=1}^{r} s_j g_j^2\right) \\
&= (1+o(1)) \int_{\mathbb{R}^r} \frac{dg_1 \dots dg_r}{(2\pi)^{r/2}} \exp\left(-\frac{1}{2}\sum_{j=1}^{r} \left\{1+2iN^{\alpha}\rho_m s_j\right\} g_j^2\right) \\
&= (1+o(1)) \prod_{j=1}^{r} \left(1+2iN^{\alpha}\rho_m s_j\right)^{-\frac{1}{2}}. \qquad (3.284)
\end{aligned}$$

In the induction step in the proof of [FS02], ρ_m becomes the new diagonal entry of the expanded $\hat{Q}$ matrix. Combining (3.284) with that proof gives the result

$$\mathcal{I}_N(F;S) = (1+o(1)) \frac{\pi^{N-\frac{1}{2}}(1+o(1))}{\Gamma\left(\frac{N}{2}\right)\Gamma\left(\frac{N-1}{2}\right)} \int_{\text{Sym}_{\geqslant 0}(m)} d\hat{Q} \left(\det \hat{Q}\right)^{\frac{N-3}{2}} F(\hat{Q}) \prod_{j=1}^{r}\prod_{i=1}^{N} \left(1+2iN^{\alpha}\hat{Q}_{ii}s_j\right)^{-1/2}. \qquad (3.285)$$

■

Remark 3.12. We note a comparison between Lemma 3.6 and the theorem in Appendix C of [Fyo19]. That result is exact and holds for general functions of projections $\boldsymbol{x}_i^T\boldsymbol{s}$ onto some arbitrary fixed vector $\boldsymbol{s}$, so it is a generalisation of our Lemma 3.6 for $r=1$, however it *only* applies to $r=1$. In [Fyo19], the function $F(Q_B)$ (in our notation) is replaced by the more general $\mathcal{F}(Q_B;\boldsymbol{s}_B)$ where the vector $\boldsymbol{s}_B$ has entries $(\boldsymbol{s}_B)_i = \boldsymbol{s}^T\boldsymbol{x}_i$ and $\boldsymbol{s}$ is an arbitrary vector. The result analogous Lemma 3.6 is

$$\mathcal{J}_{N,m}(\mathcal{F};\boldsymbol{s}) \propto \int_{\text{Sym}_{\geqslant 0}(m)} d\hat{Q} \int_{\mathbb{R}^m} d\boldsymbol{t} \left(\det \hat{Q}\right)^{\frac{N-m-2}{2}} \mathcal{F}(\hat{Q}+\boldsymbol{t}\boldsymbol{t}^T; \|\boldsymbol{s}\|\boldsymbol{t}), \qquad (3.286)$$

where we omit the constant multiplicative factor since we are content to verify that the functional form agrees with Lemma 3.6. To use this theorem in the case of Lemma 3.6, the vector $\boldsymbol{s}$ is chosen to have norm $\|\boldsymbol{s}\|_2 = N^{\alpha/2}s_1^{1/2}$, where s_1 is the single non-zero eigenvalue of the rank 1 matrix S and $\mathcal{F}(Q_B;\boldsymbol{s}_B) = F(Q_B)e^{-iN\sum_{j=1}^{m}(\boldsymbol{x}_j^T\boldsymbol{s})^2}$. With these choices

$$\begin{aligned}
\mathcal{J}_{N,m}(\mathcal{F};\boldsymbol{s}) &\propto \int_{\text{Sym}_{\geqslant 0}(m)} d\hat{Q} \int_{\mathbb{R}^m} d\boldsymbol{t} \left(\det \hat{Q}\right)^{\frac{N-m-2}{2}} F(\hat{Q}+\boldsymbol{t}\boldsymbol{t}^T) e^{-iN^{1+\alpha}s_1\boldsymbol{t}^2} \\
&= \int_{\mathbb{R}^m} d\boldsymbol{t} \int_{\text{Sym}_{\geqslant 0}(m)} d\hat{Q} \mathbf{1}\{\boldsymbol{t}^T\hat{Q}^{-1}\boldsymbol{t} < 1\} \left(\det \hat{Q} - \boldsymbol{t}\boldsymbol{t}^T\right)^{\frac{N-m-2}{2}} F(\hat{Q}) e^{-iN^{1+\alpha}s_1\boldsymbol{t}^2} \\
&= \int_{\mathbb{R}^m} d\boldsymbol{t} \int_{\text{Sym}_{\geqslant 0}(m)} d\hat{Q} \mathbf{1}\{\boldsymbol{t}^T\hat{Q}^{-1}\boldsymbol{t} < 1\} \det \hat{Q}^{\frac{N-m-2}{2}} (1-\boldsymbol{t}^T\hat{Q}^{-1}\boldsymbol{t})^{\frac{N-m-2}{2}} F(\hat{Q}) e^{-iN^{1+\alpha}s_1\boldsymbol{t}^2} \\
&= \int_{\|\boldsymbol{t}\|_2<1} d\boldsymbol{t} \int_{\text{Sym}_{\geqslant 0}(m)} d\hat{Q} \det \hat{Q}^{\frac{N-m-1}{2}} (1-\boldsymbol{t}^2)^{\frac{N-m-2}{2}} F(\hat{Q}) e^{-iN^{1+\alpha}s_1\boldsymbol{t}^T\hat{Q}\boldsymbol{t}}. \qquad (3.287)
\end{aligned}$$

Now

$$\int_{\|\boldsymbol{t}\|_2<1} d\boldsymbol{t}(1-\boldsymbol{t}^2)^{\frac{N-m-2}{2}} e^{-iN^{1+\alpha}s_1\boldsymbol{t}^T\hat{Q}\boldsymbol{t}} = \int_{\|\boldsymbol{t}\|_2<1} d\boldsymbol{t} \exp\left\{-N\left(iN^{\alpha}s_1\boldsymbol{t}^T\hat{Q}\boldsymbol{t} - \frac{N-m-2}{2N}\log(1-\boldsymbol{t}^2)\right)\right\},$$

and so we can evaluate the integral over $\boldsymbol{t}$ asymptotically. The saddle point is clearly at $\boldsymbol{t}=0$, so the leading order contribution as $N\to\infty$ is from around this point. We proceed by expanding the logarithm and evaluating the integral one coordinate at a time. Also note that $\frac{N-m-2}{2N}\sim\frac{1}{2}$ for large N. Thus, writing $\boldsymbol{t}=(\boldsymbol{t}', t_m)$,

$$\begin{aligned}\int_{\|\boldsymbol{t}\|_2<1} d\boldsymbol{t}(1-\boldsymbol{t}^2)^{\frac{N-m-2}{2}} e^{-iN^{1+\alpha}s_1\boldsymbol{t}^T\hat{Q}\boldsymbol{t}} &\sim \int_{\|\boldsymbol{t}\|_2<1} d\boldsymbol{t} \exp\left\{-N\left(iN^{\alpha}s\boldsymbol{t}^T\hat{Q}\boldsymbol{t} + \frac{1}{2}\boldsymbol{t}^2\right)\right\} \\ &\sim \int_{\|\boldsymbol{t}'\|_2<1-\varepsilon^2} d\boldsymbol{t}' \int_{-\varepsilon}^{\varepsilon} dt_m \exp\left\{-N\left(\frac{1}{2}t_m^2(2\hat{Q}_{mm}iN^{\alpha}s_1+1)\right.\right. \\ &\qquad \left.\left. + 2t_m\sum_{j\neq m}\hat{Q}_{mj}t_j' + \boldsymbol{t}'^T\hat{Q}'\boldsymbol{t}' + \frac{1}{2}\boldsymbol{t}'^2\right)\right\}\end{aligned}$$

where $\hat{Q}'$ is the $m-1\times m-1$ top left block of $\hat{Q}$ and $\varepsilon \ll 1$. Completing the square and applying Laplace's method to the t_m integral gives

$$\int_{\|\boldsymbol{t}\|_2<1} d\boldsymbol{t}(1-\boldsymbol{t}^2)^{\frac{N-m-2}{2}} e^{-iN^{1+\alpha}s_1\boldsymbol{t}^T\hat{Q}\boldsymbol{t}} \sim N^{-1/2}\int_{\|\boldsymbol{t}'\|_2<1} d\boldsymbol{t}'(2\hat{Q}_{mm}iN^{\alpha}s_1+1)^{-1/2}\exp\left\{-N\left(\boldsymbol{t}'^T\hat{Q}'\boldsymbol{t}' + \frac{1}{2}\boldsymbol{t}'^2\right)\right\}$$

and so one can clearly iterate to obtain

$$\int_{\|\boldsymbol{t}\|_2<1} d\boldsymbol{t}(1-\boldsymbol{t}^2)^{\frac{N-m-2}{2}} e^{-iN^{1+\alpha}s_1\boldsymbol{t}^T\hat{Q}\boldsymbol{t}} \sim N^{-m/2}\prod_{j=1}^{m}(2\hat{Q}_{jj}iN^{\alpha}s+1)^{-1/2}$$

and so, recalling (3.287), we obtain the same expression as Lemma 3.6 (up-to un-tracked constants).

3.7 Conclusion

The interpretation of the results we have presented in this chapter is largely the same as that first given in [Cho+15]. Under the chosen modeling assumptions, the local optima of the the neural network loss surface are arranged so that, above a critical value $-\sqrt{N}E_{\infty}$, it is overwhelmingly likely that gradient descent will encounter high-index optima and so 'escape' and descend to lower loss. Below $-\sqrt{N}E_{\infty}$, the low-index optima are arranged in a 'banded' structure, however, due to the imprecision of Theorem 3.3, the bands are slightly blurred when compared with [Cho+15]. We display the differences in Table 3.1.

Our results have plugged a gap in the analysis of [Cho+15] by demonstrating that the specific `ReLU` activation function required by the technicalities of their derivation is not, in fact, a requirement of the results themselves, which we have shown to hold for any reasonable choice of activation function. At the same time, experimental results imply that a sufficiently precise model for deep neural network loss surfaces should display some non-trivial dependence on the choice of activation function, but

band	possible indices [Cho+15]	**possible indices**
$(-\sqrt{N}E_0, -\sqrt{N}E_1)$	0	0,1,2
$(-\sqrt{N}E_1, -\sqrt{N}E_2)$	0,1	0,1,2,3
$(-\sqrt{N}E_2, -\sqrt{N}E_3)$	0,1,2	0,1,2,3,4
$(-\sqrt{N}E_3, -\sqrt{N}E_4)$	0,1,2,3	0,1,2,3,4,5

Table 3.1: Illustration of the banded low-index local optima structure obtained here for neural networks with general activation functions and compared to the analogous results in [Cho+15].

we have shown that no dependence at all is seen at the relevant level of logarithmic asymptotic complexity, but is visible in the sharp leading order complexity. In defense of [Cho+15], we have reduced the scope for their results to be some spurious apparition of an intersection of several unrealistic simplifications. However, with the same result, we have demonstrated an important aspect of neural network architectural design to which the multi-spin glass correspondence is entirely insensitive, so limiting the precision of any statements about real neural networks that can be made using this analysis.

In the pursuit of our aims, we have been forced to approximately reproduce the work of [AAC13] by means of the supersymmetric method of Random Matrix Theory, which we believe is quite novel and have also demonstrated how various steps in these supersymmetric calculations can be adapted to the setting of a GOE matrix deformed by some low-rank fixed matrix including utilising Gaussian approximations to orthogonal matrices in ways we have not previously seen in the literature. We believe some of our intermediate results and methods may be of use in other contexts in Random Matrix Theory.

As highlighted in the main text, there are a few areas for future work that stem immediately from our calculations. We list them here along with other possibilities.

1. Constructing an appropriate indicator function (or approximate indicator function) for the index of a matrix so that Theorem 3.3 can be precised and to obtain exact leading order terms for $C^h_{N,k}$ that could not be obtained in [AAC13] (see Remark 3.6).

2. The 'path-independence' assumption (Section 3.2.1, assumption 5) is the weakest link in this work (and that of [Cho+15]) and we have shed further light on its validity through experimentation (Section 3.2.4). The supersymmetric calculations used here have shown themselves to be powerful and quite adaptable. We therefore suggest that it may be possible to somehow encapsulate the failure of assumption 5 as a first-order correlation term and repeat the presented analysis in an expansion when this term is small.

3. Further, this work and others mentioned in the introduction have shown that studying spin glass like objects in this context is a fruitful area of research and so we would like to study more exotic glassy objects inspired by different neural network architectures and applications and hope to be able to adapt the calculations presented here to such new scenarios.

Chapter 4

A spin glass model for generative adversarial networks

The content of this chapter was published first as a pre-print in January 2021 (`https://arxiv.org/abs/2101.02524`) and later as a journal article: “A spin glass model for the loss surfaces of generative adversarial networks”. **Nicholas P Baskerville**, Jonathan P Keating, Francesco Mezzadri and Joseph Najnudel. *Journal of Statistical Physics*, 186(2):1-45 2022.

NPB suggested the topic, performed most of the calculations and experiments and wrote the paper. The other authors contributed ideas for possible approaches, provided feedback on results throughout and made small revisions to the drafts. NPB and JN collaborated on the proof of Lemma 4. Jonathan Hodgson helped considerably with the design of Figure 6. Anonymous reviewers spotted some minor errors, advised on changes of presentation and extra experiments and provided useful references.

4.1 An interacting spin glass model

We use multi-spin glasses in high dimensions as a toy model for neural network loss surfaces without any further justification, beyond that found in [Cho+15] and Chapter 3. GANs are composed of two networks: *generator* (G) and *discriminator* (D). G is a map $\mathbb{R}^m \to \mathbb{R}^d$ and D is a map $\mathbb{R}^d \to \mathbb{R}$. G's purpose is to generate synthetic data samples by transforming random input noise, while D's is to distinguish between real data samples and those generated by G. Given some probability distribution $\mathbb{P}_{data}$ on some $\mathbb{R}^d$, GANs have the following minimax training objective

$$\min_{\Theta_G}\max_{\Theta_D}\left\{\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_{data}}\log D(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(0,\sigma_z^2)}\log(1-D(G(\boldsymbol{z})))\right\}, \tag{4.1}$$

where Θ_D, Θ_G are the parameters of the discriminator and generator respectively. With $\boldsymbol{z}\sim\mathcal{N}(0,\sigma_z^2)$, $G(\boldsymbol{z})$ has some probability distribution $\mathbb{P}_{gen}$. When successfully trained, the initially unstructured

$\mathbb{P}_{gen}$ examples are easily distinguished by D, this in turn drives improvements in G, bring $\mathbb{P}_{gen}$ closer to $\mathbb{P}_{data}$. Ultimately, the process successfully terminates when $\mathbb{P}_{gen}$ is very close to $\mathbb{P}_{data}$ and D performs little better than random at the distinguishing task. To construct our model, we introduce two spin glasses:

$$\ell^{(D)}(\boldsymbol{w}^{(D)}) = \sum_{i_1,\dots,i_p=1}^{N_D} X_{i_1,\dots,i_p} \prod_{k=1}^{p} w^{(D)}_{i_k} \tag{4.2}$$

$$\ell^{(G)}(\boldsymbol{w}^{(D)}, \boldsymbol{w}^{(G)}) = \sum_{i_1,\dots,i_{p+q}=1}^{N_D+N_G} Z_{i_1,\dots,i_{p+q}} \prod_{k=1}^{p+q} w_k \tag{4.3}$$

where $\boldsymbol{w}^T = (\boldsymbol{w}^{(D)^T}, \boldsymbol{w}^{(G)^T})$, all the $X_{i_1,\dots,i_p}$ are i.i.d. $\mathcal{N}(0,1)$ and $Z_{j_1,\dots,j_{p+q}}$ are similarly i.i.d. $\mathcal{N}(0,1)$. We then define the models for the discriminator and generator losses:

$$L^{(D)}(\boldsymbol{w}^{(D)}, \boldsymbol{w}^{(G)}) = \ell^{(D)}(\boldsymbol{w}^{(D)}) - \sigma_z \ell^{(G)}(\boldsymbol{w}^{(D)}, \boldsymbol{w}^{(G)}), \tag{4.4}$$

$$L^{(G)}(\boldsymbol{w}^{(D)}, \boldsymbol{w}^{(G)}) = \sigma_z \ell^{(G)}(\boldsymbol{w}^{(D)}, \boldsymbol{w}^{(G)}). \tag{4.5}$$

$\ell^{(D)}$ plays the role of the loss of the discriminator network when trying to classify genuine examples as such. $\ell^{(G)}$ plays the role of loss of the discriminator when applied to samples produced by the generator, hence the sign difference between $L^{(D)}$ and $L^{(G)}$. $\boldsymbol{w}^{(D)}$ are the weights of the discriminator, and $\boldsymbol{w}^{(G)}$ the weights of the generator. The X_i are surrogates for the training data (i.e. samples from $\mathbb{P}_{data}$) and the Z_j are surrogates for the noise distribution of the generator. For convenience, we have chosen to pull the σ_z scale outside of the Z_j and include it as a constant multiplier in (4.4)-(4.5). In reality, we should like to keep Z_j as i.i.d. $\mathcal{N}(0,1)$ but take X_i to have some other more interesting distribution, e.g. normally or uniformly distributed on some manifold. Using $[x]$ to denote the integer part of x, we take $N_D = [\kappa N], N_G = [\kappa' N]$ for fixed $\kappa \in (0,1)$, $\kappa' = 1-\kappa$, and study the regime $N \to \infty$. Note that there is no need to distinguish between $[\kappa N]$ and κN in the $N \to \infty$ limit.

Remark 4.1. Our model is not supposed to have any direct relationship to GANs. Rather, we have used two spin glasses as models for high-dimensional random surfaces. The spin glasses are related by sharing some of their variables, namely the $\boldsymbol{w}^{(D)}$, just as the two training objectives in GANs share the discriminator weights. In prior work modeling neural network loss surfaces as spin glasses, the number of spins corresponds to the number of layers in the network, therefore we have chosen p spins for $\ell^{(D)}$ and $p+q$ for $\ell^{(G)}$, corresponding to p layers in the discriminator and q layers in the generator, but the generator is only ever seen in the losses composed with the discriminator. One could make other choices of $\ell^{(D)}$ and $\ell^{(G)}$ to couple the two glasses and we consider one such example in the appendix Section B.1.

4.2 Kac-Rice formulae for complexity

Training GANs involves jointly minimising the losses of the discriminator and the generator. Therefore, rather than being interested simply in upper-bounding a single spin-glass and counting its stationary points, the complexity of interest comes from jointly upper bounding both $L^{(D)}$ and $L^{(G)}$ and counting points where both are stationary. Using S^M to denote the M-sphere[1], we define the complexity

$$C_N = \left|\{\boldsymbol{w}^{(D)} \in S^{N_D}, \boldsymbol{w}^{(G)} \in S^{N_G} \ : \ \nabla_D L^{(D)} = 0, \nabla_G L^{(G)} = 0, L^{(D)} \in B_D, L^{(G)} \in B_G\}\right| \tag{4.6}$$

for some Borel sets $B_D, B_G \subset \mathbb{R}$ and where ∇_D, ∇_G denote the Riemannian covariate derivatives on the hyperspheres with respect to the discriminator and generator weights respectively. Note:

1. We have chosen to treat the parameters of each network as somewhat separate by placing them on their own hyper-spheres. This reflects the minimax nature of GAN training, where there really are 2 networks being optimised in an adversarial manner rather than one network with some peculiar structure.

2. We could have taken $\nabla = (\nabla_D, \nabla_G)$ and required $\nabla L^{(D)} = \nabla L^{(G)} = 0$ but, as in the previous comment, our choice is more in keeping with the adversarial set-up, with each network seeking to optimize separately its own parameters in spite of the other.

3. We will only be interested in the case $B_D = (-\infty, \sqrt{N}u_D)$ and $B_G = (-\infty, \sqrt{N}u_G)$, for $u_D, u_G \in \mathbb{R}$.

So that the finer structure of local minima and saddle points can be probed, we also define the corresponding complexity with Hessian index prescription

$$\begin{aligned} C_{N,k_D,k_G} = \Bigg|\Bigg\{\boldsymbol{w}^{(D)} \in S^{N_D}, \boldsymbol{w}^{(G)} \in S^{N_G} \ : \ \nabla_D L^{(D)} = 0, \nabla_G L^{(G)} = 0, L^{(D)} \in B_D, L^{(G)} \in B_G \\ i(\nabla_D^2 L^{(D)}) = k_D, \ i(\nabla_G^2 L^{(G)}) = k_G\Bigg\}\Bigg|, \end{aligned} \tag{4.7}$$

where $i(M)$ is the index of M (i.e. the number of negative eigenvalues of M). We have chosen to consider the indices of the Hessians $\nabla_D^2 L^{(D)}$ and $\nabla_G^2 L^{(G)}$ separately, just as we chose to consider separately vanishing derivatives $\nabla_D L^{(D)}$ and $\nabla_G L^{(G)}$. We believe this choice best reflects the standard training loop of GANs, where each iteration updates the discriminator and generator parameters in separate steps.

To calculate the complexities, we follow the well-trodden route of Kac-Rice formulae as pioneered by [Fyo04; FW07]. For a fully rigorous treatment, we proceed as in [AAC13] and Chapter 3.

[1] We use the convention of the M-sphere being the sphere embedded in $\mathbb{R}^M$.

Lemma 4.1.

$$C_N = \int_{S^{N_D}\times S^{N_G}} d\boldsymbol{w}^{(G)} d\boldsymbol{w}^{(D)} \;\; \varphi_{(\nabla_D L^{(D)}, \nabla_G L^{(G)})}(0)$$
$$\mathbb{E}\left[\left|\det\begin{pmatrix} \nabla^2_D L^{(D)} & \nabla_{GD} L^{(D)} \\ \nabla_{DG} L^{(G)} & \nabla^2_G L^{(G)} \end{pmatrix}\right| \mid \nabla_G L^{(G)} = 0, \nabla_D L^{(D)} = 0\right] \mathbb{1}\left\{L^{(D)} \in B_D, L^{(G)} \in B_G\right\} \tag{4.8}$$

and therefore

$$C_N = \int_{S^{N_D}\times S^{N_G}} d\boldsymbol{w}^{(G)} d\boldsymbol{w}^{(D)} \;\; \varphi_{(\nabla_D L^{(D)}, \nabla_G L^{(G)})}(0) \int_{B_D} dx_D \int_{B_G} dx_G \; \varphi_{L^{(D)}}(x_D)\varphi_{L^{(G)}}(x_G)$$
$$\mathbb{E}\left[\left|\det\begin{pmatrix} \nabla^2_D L^{(D)} & \nabla_{GD} L^{(D)} \\ \nabla_{DG} L^{(G)} & \nabla^2_G L^{(G)} \end{pmatrix}\right| \mid \nabla_G L^{(G)} = 0, \nabla_D L^{(D)} = 0, L^{(D)} = x_D, L^{(G)} = x_G\right]. \tag{4.9}$$

where $\varphi_{(\nabla_D L^{(D)}, \nabla_G L^{(G)})}$ is the joint density of $(\nabla_D L^{(D)}, \nabla_G L^{(G)})^T$, $\varphi_{L^{(D)}}$ the density of $L^{(D)}$, and $\varphi_{L^{(G)}}$ the density of $L^{(G)}$, all implicitly evaluated at $(\boldsymbol{w}^{(D)}, \boldsymbol{w}^{(G)})$.

Proof. In the notation of Theorem 2.1, we make the following choices:

$$\phi = \begin{pmatrix} \nabla_D L^{(D)} \\ \nabla_G L^{(G)} \end{pmatrix}, \quad \psi = \begin{pmatrix} L^{(D)} \\ L^{(G)} \end{pmatrix}$$

and so

$$A = B_D \times B_G, \quad \boldsymbol{u} = 0.$$

and the manifold $\mathcal{M}$ is taken to be $S^{N_D} \times S^{N_G}$ with the product topology. It is sufficient to check the conditions of Theorem 2.1 with the above choices.

Conditions (a)-(f) are satisfied due to Gaussianity and the manifestly smooth definition of $L^{(D)}, L^{(G)}$. The moduli of continuity conditions as in (g) are satisfied separately for $L^{(D)}$ and its derivatives on S^{N_D} and for $L^{(G)}$ and its derivatives on S^{N_G}, as seen in the proof of the analogous result for a single spin glass in [AAC13]. But since $\mathcal{M}$ is just a direct product with product topology, it immediately follows that (g) is satisfied, so Theorem 2.31 applies and we obtain (4.8). (4.9) follows simply, using the rules of conditional expectation. ■

With Lemma 4.1 in place, we can now establish the following Kac-Rice expression specialised to our model:

Lemma 4.2. *For $(N-2)\times(N-2)$ GOE matrix M and independent $(N_D-1)\times(N_D-1)$ GOE matrix M_1, define*

$$H(x, x_1) \overset{d}{=} bM + b_1 \begin{pmatrix} M_1 & 0 \\ 0 & 0 \end{pmatrix} - x - x_1 \begin{pmatrix} I_{N_D} & 0 \\ 0 & 0 \end{pmatrix}. \tag{4.10}$$

For $u_G, u_D \in \mathbb{R}$, define

$$B = \left\{ (x, x_1) \in \mathbb{R}^2 \;:\; x \leqslant \frac{1}{\sqrt{2}}(p+q)2^{p+q}u_G, \;\; x_1 \geqslant -(p+q)^{-1}2^{-(p+q)}px - \frac{p}{\sqrt{2}}u_D \right\}. \tag{4.11}$$

Define the constant

$$K_N = \omega_{\kappa N}\omega_{\kappa' N}(2(N-2))^{\frac{N-2}{2}}(2\pi)^{-\frac{N-2}{2}}\left(p+\sigma_z^2 2^{p+1}(p+q)\right)^{-\frac{\kappa N-1}{2}}\left(\sigma_z^2 2^{p+q}(p+q)\right)^{-\frac{\kappa' N-1}{2}} \tag{4.12}$$

where the variances are

$$s^2 = \frac{1}{2}\sigma_z^2(p+q)^2 2^{3(p+q)}, \quad s_1^2 = \frac{p^2}{2}. \tag{4.13}$$

and $\omega_N = \frac{2\pi^{N/2}}{\Gamma(N/2)}$ is the surface area of the N sphere. The expected complexity C_N is then

$$\mathbb{E}C_N = K_N \int_B \sqrt{\frac{N}{2\pi s^2}} e^{-\frac{N}{2s^2}x^2} dx \sqrt{\frac{N}{2\pi s_1^2}} e^{-\frac{N}{2s_1^2}x_1^2} dx_1 \; \mathbb{E}|\det H(x, x_1)|. \tag{4.14}$$

Proof. Define the matrix

$$\tilde{H} = \begin{pmatrix} \nabla_D^2 L^{(D)} & \nabla_{GD} L^{(D)} \\ \nabla_{DG} L^{(G)} & \nabla_G^2 L^{(G)} \end{pmatrix}$$

appearing in the expression for C_N in Lemma 4.1. Note that $\tilde{H}$ takes the place of a Hessian (though it is not symmetric). We begin with the distribution of

$$\tilde{H} \mid \{(\ell^{(D)}, \ell^{(G)}) = (x_D, x_G), \; (\nabla_D \ell^{(D)}, \nabla \ell^{(G)}) = (0,0)\}.$$

Note that the integrand in (4.14) is jointly spherically symmetric in both $\boldsymbol{w}^{(D)}$ and $\boldsymbol{w}^{(G)}$. It is therefore sufficient to consider $\tilde{H}$ in the region of a single point on each sphere. We choose the north poles and coordinate bases on both spheres in the region of their north poles. The remaining calculations are routine Gaussian manipulations, very similar in character to those in the previous chapter, so they are given at the end of this chapter (section 4.6). One finds

$$\begin{aligned}\tilde{H} \overset{d}{=} \sqrt{2p(p-1)}\begin{pmatrix} \sqrt{N_D-1}M_2^{(D)} & 0 \\ 0 & 0 \end{pmatrix} + \sigma_z\sqrt{2^{p+q+1}(p+q)(p+q-1)}\begin{pmatrix} \sqrt{N_D-1}M_1^{(D)} & -2^{-1/2}G \\ 2^{-1/2}G^T & \sqrt{N_G-1}M^{(G)} \end{pmatrix} \\ -\sigma_z(p+q)x_G 2^{p+q}\begin{pmatrix} -I_{N_D} & 0 \\ 0 & I_{N_G} \end{pmatrix} - px_D\begin{pmatrix} I_{N_D} & 0 \\ 0 & 0 \end{pmatrix}\end{aligned} \tag{4.15}$$

where $M_1^{(D)}, M_2^{(D)}$ are independent GOE^{N_D-1} matrices, $M^{(G)}$ is an independent GOE^{N_G-1} matrix and G is an independent $(N_D-1)\times(N_G-1)$ Ginibre matrix. Note that the dimensions are N_D-1 and N_G-1 rather N_D and N_G. This is simply because the hypersphere S^{N_D} is an N_D-1 dimensional manifold, and similarly S^{N_G}.

We can simplify by summing independent Gaussians to obtain

$$\tilde{H} = \begin{pmatrix} \sigma_D\sqrt{N_D-1}M^{(D)} & -2^{-1/2}\sigma_G G, \\ 2^{-1/2}\sigma_G G^T & \sigma_G\sqrt{N_G-1}M^{(G)} \end{pmatrix} - \sigma_z(p+q)x_G 2^{p+q}\begin{pmatrix} -I_{N_D} & 0 \\ 0 & I_{N_G} \end{pmatrix} - px_D\begin{pmatrix} I_{N_D} & 0 \\ 0 & 0 \end{pmatrix} \tag{4.16}$$

where

$$\sigma_G = \sigma_z\sqrt{2^{p+q+1}(p+q)(p+q-1)} \tag{4.17}$$

$$\sigma_D = \sqrt{\sigma_G^2 + 2p(p-1)} \tag{4.18}$$

and $M^{(D)} \sim GOE^{N_D-1}$ is a GOE matrix independent of $M^{(G)}$ and G.

There is an alternative reformulation of $\tilde{H}$ that will also be useful. Indeed, because $M^{(D)}_{1,2} \overset{d}{=} -M^{(D)}_{1,2}$, let us write $\tilde{H}$ as

$$\begin{aligned}\tilde{H} =& \sigma_z J\left(\sqrt{2^{p+q+1}(p+q)(p+q-1)(N_D+N_G-2)}M_1 - (p+q)x_G 2^{p+q} I\right) \\ &+ \left(\sqrt{2p(p-1)(N_D-1)}\begin{pmatrix} M_2 & 0 \\ 0 & 0 \end{pmatrix} - px_D\begin{pmatrix} I_{N_D} & 0 \\ 0 & 0 \end{pmatrix}\right) \\ \overset{d}{=}& J\Bigg[\sigma_z\sqrt{2^{p+q+1}(p+q)(p+q-1)(N_D+N_G-2)}M_1 - \sigma_z(p+q)x_G 2^{p+q} I \\ &+ \sqrt{2p(p-1)(N_D-1)}\begin{pmatrix} M_2 & 0 \\ 0 & 0 \end{pmatrix} + px_D\begin{pmatrix} I_{N_D} & 0 \\ 0 & 0 \end{pmatrix}\Bigg]\end{aligned} \tag{4.19}$$

where $M_1 \sim GOE^{N_D+N_G-2}$ is a GOE matrix of size N_D+N_G-2, $M_2 \sim GOE^{N_D-1}$ is an independent GOE matrix of size N_D-1 and

$$J = \begin{pmatrix} -I_{N_D} & 0 \\ 0 & I_{N_G} \end{pmatrix}. \tag{4.20}$$

If follows that

$$\begin{aligned}|\det\tilde{H}| \overset{d}{=}& \Bigg|\det\Bigg[\sigma_z\sqrt{2^{p+q+1}(p+q)(p+q-1)(N_D+N_G-2)}M_1 - \sigma_z(p+q)x_G 2^{p+q} I \\ &+ \sqrt{2p(p-1)(N_D-1)}\begin{pmatrix} M_2 & 0 \\ 0 & 0 \end{pmatrix} + px_D\begin{pmatrix} I_{N_D} & 0 \\ 0 & 0 \end{pmatrix}\Bigg]\Bigg|.\end{aligned} \tag{4.21}$$

Now define the constants

$$b = \sqrt{2^{p+q}(p+q)(p+q-1)}\sigma_z, \quad b_1 = \sqrt{p(p-1)\kappa} \tag{4.22}$$

$$x = \frac{\sigma_z(p+q)2^{p+q}}{\sqrt{N}}x_G, \quad x_1 = -\frac{p}{\sqrt{N}}x_D, \tag{4.23}$$

and then we arrive at

$$|\det\tilde{H}| \overset{d}{=} (2(N-2))^{\frac{N-2}{2}}|\det H(x,x_1)|. \tag{4.24}$$

The variances of $L^{(D)}$ and $L^{(G)}$ are derived from those of $\ell^{(G)}, \ell^{(D)}$ computed in Section 4.6 (see (4.131), (4.135)):

$$Var(\ell^{(D)}) = 1, \quad Var(\ell^{(G)}) = 2^{p+q}.$$

Similarly the density $\varphi_{(\nabla_D L^{(D)}, \nabla_G L^{(G)})}$ is found in (4.147):

$$\varphi_{(\nabla_D L^{(D)}, \nabla_G L^{(G)})}(0) = (2\pi)^{-\frac{N-2}{2}} \left(p + \sigma_z^2 2^{p+1}(p+q)\right)^{-\frac{N_D-1}{2}} \left(\sigma_z^2 2^{p+q}(p+q)\right)^{-\frac{N_G-1}{2}}.$$

We have now collected all the inputs required for Lemma 4.1. The domain of integration B arises from the constraints $L^{(D)} \in (-\infty, \sqrt{N}u_D)$ and $L^{(G)} \in (-\infty, \sqrt{N}u_G)$ and the re-scaled variables (4.23). This completes the proof. ■

We will need the asymptotic behaviour of the constant K_N, which we now record in a small lemma.

Lemma 4.3. *As $N \to \infty$,*

$$K_N \sim 2^{\frac{N}{2}} \pi^{N/2} \left(\kappa^\kappa \kappa'^{\kappa'}\right)^{-N/2} \sqrt{\kappa\kappa'} \left(p + \sigma_z^2 2^{p+1}(p+q)\right)^{-\frac{\kappa N-1}{2}} \left(\sigma_z^2 2^{p+q}(p+q)\right)^{-\frac{\kappa' N-1}{2}} \tag{4.25}$$

Proof. By Stirling's formula

$$\begin{aligned} K_N \sim\ & 4\pi^N \left(\frac{4\pi}{\kappa N}\right)^{-1/2} \left(\frac{4\pi}{\kappa' N}\right)^{-1/2} \left(\frac{\kappa N}{2e}\right)^{-\kappa N/2} \left(\frac{\kappa' N}{2e}\right)^{-\kappa' N/2} (2(N-2))^{\frac{N-2}{2}} (2\pi)^{-\frac{N-2}{2}} \\ & \left(p + \sigma_z^2 2^{p+1}(p+q)\right)^{-\frac{\kappa N-1}{2}} \left(\sigma_z^2 2^{p+q}(p+q)\right)^{-\frac{\kappa' N-1}{2}} \\ \sim\ & 2^{\frac{N}{2}} \pi^{N/2} \left(\kappa^\kappa \kappa'^{\kappa'}\right)^{-N/2} \sqrt{\kappa\kappa'} \left(p + \sigma_z^2 2^{p+1}(p+q)\right)^{-\frac{\kappa N-1}{2}} \left(\sigma_z^2 2^{p+q}(p+q)\right)^{-\frac{\kappa' N-1}{2}} \end{aligned} \tag{4.26}$$

where we have used $(N-2)^{\frac{N-2}{2}} = N^{\frac{N-2}{2}} \left(1 - \frac{2}{N}\right)^{\frac{N-2}{2}} \sim N^{\frac{N-2}{2}} e^{-N/2}$. ■

4.3 Limiting spectral density of the Hessian

Our intention now is to compute the the expected complexity $\mathbb{E}C_N$ via the Coulomb gas method. The first step in this calculation is to obtain the limiting spectral density of the random matrix

$$H' = bM + b_1 \begin{pmatrix} M_1 & 0 \\ 0 & 0 \end{pmatrix} - x_1 \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix}, \tag{4.27}$$

where, note, $H' = H + xI$ is just a shifted version of H as defined in Lemma 4.2. Here the upper-left block is of dimension κN, and the overall dimension is N. Let μ_{eq} be the limiting spectral measure of H' and ρ_{eq} its density. The supersymmetric method provides a way of calculating the expected Stieltjes transforms of ρ_{eq} [Ver04]:

$$\langle G(z) \rangle = \frac{1}{N} \frac{\partial}{\partial J}\bigg|_{J=0} Z(J) \tag{4.28}$$

$$Z(J) := \mathbb{E}_{H'} \frac{\det(z - H' + J)}{\det(z - H')}. \tag{4.29}$$

Recall that a density and its Stieltjes transform are related by the Stieltjes inversion formula

$$\rho_{eq}(z) = \frac{1}{\pi}\lim_{\varepsilon\to 0}\Im\langle G(z+i\varepsilon)\rangle. \tag{4.30}$$

The function $Z(J)$ can be computed using a supersymmetric representation of the ratio of determinants. Firstly, we recall an elementary result from multivariate calculus, where M is a real matrix:

$$\int\prod_{i=1}^{N}\frac{d\phi_i d\phi_i^*}{2\pi}e^{-i\phi^\dagger M\phi} = \frac{1}{\det M}. \tag{4.31}$$

By introducing Grassmann varibables χ_i, χ_i* and a Berezin integral, we obtain a complimentary expression:

$$\int\frac{1}{-i}\prod_{i=1}^{N}d\chi_i d\chi_i^* e^{-i\chi^\dagger M\chi} = \det M, \tag{4.32}$$

Using the integral results (4.31), (4.32) we can then write

$$\frac{\det(z-H'+J)}{\det(z-H')} = \int d\Psi\exp\left\{-i\phi^\dagger(z-H')\phi - i\chi^\dagger(z+J-H')\chi\right\} \tag{4.33}$$

where the measure is

$$d\Psi = \frac{1}{-i(2\pi)^N}\prod_{t=1}^{2}d\phi[t]d\phi^*[t]d\chi[t]d\chi^*[t], \tag{4.34}$$

ϕ is a vector of N complex commuting variables, χ and χ^* are vectors of N Grassmann variables, and we use the $[t]$ notation to denote the splitting of each of the vectors into the first κN and last $(1-\kappa)N$ components, as seen in [GW90]:

$$\phi = \begin{pmatrix}\phi[1]\\ \phi[2]\end{pmatrix}. \tag{4.35}$$

We then split the quadratic form expressions in (4.33)

$$\begin{aligned}&-\phi^\dagger(z-H')\phi - \chi^\dagger(z+J-H')\chi\\ =&-\phi[1]^\dagger(x_1-b_1M_1)\phi[1] - \phi^\dagger(z-bM)\phi - \chi[1]^\dagger(x_1-b_1M_1)\chi[1] - \chi^\dagger(z+J-bM)\chi.\end{aligned} \tag{4.36}$$

Taking the GOE averages is now simple [Ver04; Noc17]:

$$\mathbb{E}_M\exp\left\{-ib\phi^\dagger M\phi - ib\chi^\dagger M\chi\right\} = \exp\left\{-\frac{b^2}{4N}\text{trg}Q^2\right\}, \tag{4.37}$$

$$\mathbb{E}_M\exp\left\{-ib_1\phi[1]^\dagger M_1\phi[1] - ib_1\chi[1]^\dagger M_1\chi[1]\right\} = \exp\left\{-\frac{b_1^2}{4\kappa N}\text{trg}Q[1]^2\right\}, \tag{4.38}$$

where the supersymmetric matrices are given by

$$Q = \begin{pmatrix}\phi^\dagger\phi & \phi^\dagger\chi\\ \chi^\dagger\phi & \chi^\dagger\chi\end{pmatrix},\quad Q[1] = \begin{pmatrix}\phi[1]^\dagger\phi[1] & \phi[1]^\dagger\chi[1]\\ \chi[1]^\dagger\phi[1] & \chi[1]^\dagger\chi[1]\end{pmatrix}. \tag{4.39}$$

Introducing the tensor notation

$$\psi = \phi \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \chi \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \psi[1] = \phi[1] \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \chi[1] \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{4.40}$$

and

$$\zeta = \begin{pmatrix} z & 0 \\ 0 & z+J \end{pmatrix} \tag{4.41}$$

we can compactly write

$$Z(J) = \int d\Psi \exp\left\{ -\frac{b^2}{4N}\mathrm{trg}Q^2 - \frac{b_1^2}{4\kappa N}\mathrm{trg}Q[1]^2 - i\psi[1]^\dagger \psi[1] x_1 - i\psi^\dagger \zeta \psi \right\}. \tag{4.42}$$

We now perform two Hubbard-Stratonovich transformations [Ver04]

$$Z(J) = \int d\Psi d\sigma d\sigma[1] \exp\left\{ -\frac{N}{b^2}\mathrm{trg}\sigma^2 - \frac{\kappa N}{b_1^2}\mathrm{trg}\sigma[1]^2 - i\psi[1]^\dagger (x_1 + \sigma[1])\psi[1] - i\psi^\dagger (\sigma + \zeta)\psi \right\}, \tag{4.43}$$

where σ and $\sigma[1]$ inherit their form from $Q, Q[1]$

$$\sigma = \begin{pmatrix} \sigma_{BB} & \sigma_{BF} \\ \sigma_{FB} & i\sigma_{FF} \end{pmatrix}, \quad \sigma[1] = \begin{pmatrix} \sigma_{BB}[1] & \sigma_{BF}[1] \\ \sigma_{FB}[1] & i\sigma_{FF}[1] \end{pmatrix} \tag{4.44}$$

with $\sigma_{BB}, \sigma_{FF}, \sigma_{BB}[1], \sigma_{FF}[1]$ real commuting variables, and $\sigma_{BF}, \sigma_{FB}, \sigma_{BF}[1], \sigma_{FB}[1]$ Grassmanns; the factor i is introduced to ensure convergence. Integrating out over $d\Psi$ is now a straightforward Gaussian integral in superspace, giving

$$\begin{aligned} Z(J) &= \int d\Psi d\sigma d\sigma[1] \exp\left\{ -\frac{N}{b^2}\mathrm{trg}\sigma^2 - \frac{\kappa N}{b_1^2}\mathrm{trg}\sigma[1]^2 - i\psi[1]^\dagger (x_1 + \zeta + \sigma + \sigma[1])\psi[1] - i\psi[2]^\dagger (\sigma + \zeta)\psi[2] \right\} \\ &= \int d\sigma d\sigma[1] \exp\left\{ -\frac{N}{b^2}\mathrm{trg}\sigma^2 - \frac{\kappa N}{b_1^2}\mathrm{trg}\sigma[1]^2 - \kappa N \mathrm{trglog}(x_1 + \zeta + \sigma + \sigma[1]) - \kappa' N \mathrm{trglog}(\sigma + \zeta) \right\} \\ &= \int d\sigma d\sigma[1] \exp\left\{ -\frac{N}{b^2}\mathrm{trg}(\sigma - \zeta)^2 - \frac{\kappa N}{b_1^2}\mathrm{trg}\sigma[1]^2 - \kappa N \mathrm{trglog}(x_1 + \sigma + \sigma[1]) - \kappa' N \mathrm{trglog}\sigma \right\}. \end{aligned} \tag{4.45}$$

Recalling the definition of ζ, we have

$$\mathrm{trg}(\sigma - \zeta)^2 = (\sigma_{BB} - z)^2 - (i\sigma_{FF} - z - J)^2 \tag{4.46}$$

and so one immediately obtains

$$\begin{aligned}\frac{1}{N}\frac{\partial}{\partial J}\bigg|_{J=0} Z(J) = \frac{2}{b^2}\int d\sigma d\sigma[1](z - i\sigma_{FF})\exp\bigg\{&-\frac{N}{b^2}\mathrm{trg}(\sigma - z)^2 - \frac{\kappa N}{b_1^2}\mathrm{trg}\sigma[1]^2\\ &-\kappa N\mathrm{trg}\log(x_1+\sigma+\sigma[1]) - \kappa' N\mathrm{trg}\log\sigma\bigg\}\\ = \frac{2}{b^2}\int d\sigma d\sigma[1](z - i\sigma_{FF})\exp\bigg\{&-\frac{N}{b^2}\mathrm{trg}\sigma^2 - \frac{\kappa N}{b_1^2}\mathrm{trg}\sigma[1]^2\\ &-\kappa N\mathrm{trg}\log(x_1+z+\sigma+\sigma[1]) - \kappa' N\mathrm{trg}\log(z+\sigma)\bigg\}\end{aligned} \tag{4.47}$$

To obtain the limiting spectral density (LSD), or rather its Stieltjes transform, one must find the leading order term in the $N\to\infty$ expansion for (4.47). This can be done by using the saddle point method on the $\sigma,\sigma[1]$ manifolds. We know that the contents of the exponential must vanish at the saddle point, since the LSD is $\mathcal{O}(1)$, so we in fact need only compute σ_{FF} at the saddle point. We can diagonalise σ within the integrand of (4.47) and absorb the diagonalising graded $U(1/1)$ matrix into $\sigma[1]$. The resulting saddle point equations for the off-diagonal entries of the new (rotated) $\sigma[1]$ dummy variable are trivial and immediately give that $\sigma[1]$ is also diagonal at the saddle point. The saddle point equations are then

$$\frac{2}{b_1^2}\sigma_{BB}[1] + \frac{1}{\sigma_{BB}[1]+\sigma_{BB}+x_1+z} = 0 \tag{4.48}$$

$$\frac{2}{b^2}\sigma_{BB} + \frac{\kappa}{\sigma_{BB}[1]+\sigma_{BB}+x_1+z} + \frac{\kappa'}{\sigma_{BB}+x} = 0 \tag{4.49}$$

$$\frac{2}{b_1^2}\sigma_{FF}[1] - \frac{1}{\sigma_{FF}[1]+\sigma_{FF}-ix_1-iz} = 0 \tag{4.50}$$

$$\frac{2}{b^2}\sigma_{FF} - \frac{\kappa}{\sigma_{FF}[1]+\sigma_{FF}-ix_1-iz} - \frac{\kappa'}{\sigma_{FF}-iz} = 0. \tag{4.51}$$

(4.50) and (4.51) combine to give an explicit expression for $\sigma_{FF}[1]$:

$$\sigma_{FF}[1] = \frac{b_1^2}{2\kappa}\left(\frac{2}{b^2}\sigma_{FF} - \kappa'(\sigma_{FF}-iz)^{-1}\right). \tag{4.52}$$

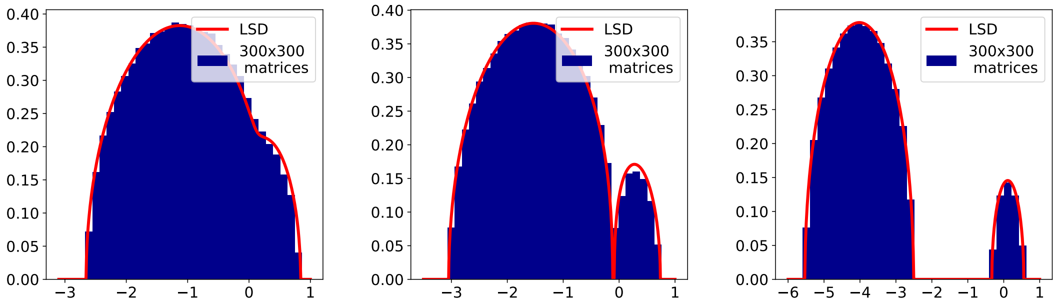
With a view to simplifying the numerical solution of the coming quartic, we define $t = i(\sigma_{FF} - iz)$ and then a line of manipulation with (4.51) and (4.52) gives

$$\left(t^2 - zt - \kappa' b^2\right)\left((1+\kappa^{-1}b^{-2}b_1^2)t^2 - (\kappa^{-1}b_1^2b^{-2}z - x_1)t - \kappa'\kappa^{-1}b_1^2\right) + b^2\kappa t^2 = 0. \tag{4.53}$$

By solving (4.53) numerically for fixed values of κ, b, b_1, x_1, we can obtain the four solutions $t_1(z), t_2(z), t_3(z), t_4(z)$. These four solution functions arise from choices of branch for $(z, x_1)\in\mathbb{C}^2$ and determining the correct branch directly is highly non-trivial. However, for any $z\in\mathbb{R}$, at most one of the t_i will lead to a positive LSD, which gives a simple way to compute ρ_{eq} numerically using (4.30) and (4.47):

$$\rho_{eq}(z) = \max_i\left\{-\frac{2}{b^2\pi}\Im t_i(z)\right\}. \tag{4.54}$$

Plots generated using (4.54) and eigendecompositions of matrices sampled from the distribution of H' are given in Figure 4.1 and show good agreement between the two. Note the three different forms: single component support, two component support and the transition point between the two, according to the various parameters. In these plots, the larger lobes on the left correspond to the upper left block, which is much larger than the lower-right block (since $\kappa = 0.9$ here). One can see this by considering large x_1, for which there must be a body of eigenvalues in the region of $-x_1$ owing to the upper left block. Since x_1 only features in the upper-left block, not all of the eigenvalues can be located around $-x_1$, and the remainder are found in the other lobe of the density which is around 0 in Figure 4.1.



(a) Merged

(b) Touching

(c) Separate

Figure 4.1: Example spectra of H' showing empirical spectra from 100 300×300 matrices and the corresponding LSDs computed from (4.53). Here $b = b_1 = 1$, $\kappa = 0.9$, σ_z=1 and x_1 is varied to give the three different behaviours.

4.4 The asymptotic complexity

In the previous section, we have found the equilibrium measure, μ_{eq}, of the ensemble of random matrices

$$H' = bM + b_1 \begin{pmatrix} M_1 & 0 \\ 0 & 0 \end{pmatrix} - x_1 \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix}, \quad M \sim GOE^N, \; M_1 \sim GOE^{\kappa N}. \tag{4.55}$$

The Coulomb gas approximation gives us a method of computing $\mathbb{E}|\det(H' - x)|$:

$$\mathbb{E}|\det(H' - x)| \approx \exp\left\{N \int \log|z - x| d\mu_{eq}(z)\right\}. \tag{4.56}$$

We have access to the density of μ_{eq} pointwise (in x and x_1) numerically, and so (4.56) is a matter of one-dimensional quadrature. Recalling (4.14), we then have

$$\mathbb{E}C_N \approx K'_N \iint_B dx dx_1 \; \exp\left\{-(N-2)\left(\frac{1}{2s^2}x^2 + \frac{1}{2s_1^2}(x_1)^2 - \int \log|z - x| d\mu_{eq}(z)\right)\right\} \equiv K'_N \iint_B dx dx_1 \; e^{-(N-2)\Phi(x,x_1)} \tag{4.57}$$

where

$$K_N' = K_N \sqrt{\frac{N-2}{2\pi s_1^2}} \sqrt{\frac{N-2}{2\pi s^2}}. \tag{4.58}$$

Due to Lemma 4.3, the constant term has asymptotic form

$$\begin{aligned} &\frac{1}{N}\log K_N' \\ \sim &\frac{1}{2}\log 2 + \frac{1}{2}\log \pi - \frac{\kappa}{2}\log\left(p + \sigma_z^2 2^{p+q}(p+q)\right) - \frac{\kappa'}{2}\log\left(\sigma_z^2(p+q)2^{p+q}\right) - \frac{\kappa}{2}\log\kappa - \frac{\kappa'}{2}\log\kappa' \\ \equiv &K \end{aligned} \tag{4.59}$$

We then define the desired $\Theta(u_D, u_G)$ as

$$\lim \frac{1}{N}\log \mathbb{E}C_N = \Theta(u_D, u_G) \tag{4.60}$$

and we have

$$\Theta(u_D, u_G) = K - \min_B \Phi. \tag{4.61}$$

Using these numerical methods, we obtain the plot of Φ in B and a plot of Θ for some example p, q, σ_z, κ values, shown in Figures 4.2, 4.3. Numerically obtaining the maximum of Φ on B is not as onerous as it may appear, since $-\Phi$ grows quadratically in $|x|, |x_1|$ at moderate distances from the origin.

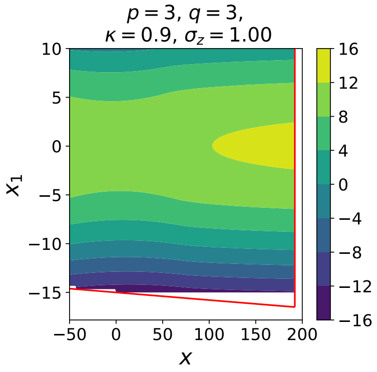


Figure 4.2: Φ for $p = q = 3, \sigma_z = 1, \kappa = 0.9$. Red lines show the boundary of the integration region B.

We numerically verify the legitimacy of this Coulomb point approximation with Monte Carlo integration

$$\mathbb{E}|\det(H' - x)| \approx \frac{1}{n}\sum_{i=1}^{n}\prod_{j=1}^{N}|\lambda_j^{(i)} - x|, \tag{4.62}$$

where $\lambda_j^{(i)}$ is the j-th eigenvalues of the i-th i.i.d. sample from the distribution of H'. The results, comparing $N^{-1}\log\mathbb{E}|\det(H'-x)|$ at $N = 50$ for a variety of x, x_1 are show in Figure 4.4. Note the strong agreement even at such modest N, however to rigorously substantiate the Coulomb gas approximation in (4.56), we must prove a concentration result.

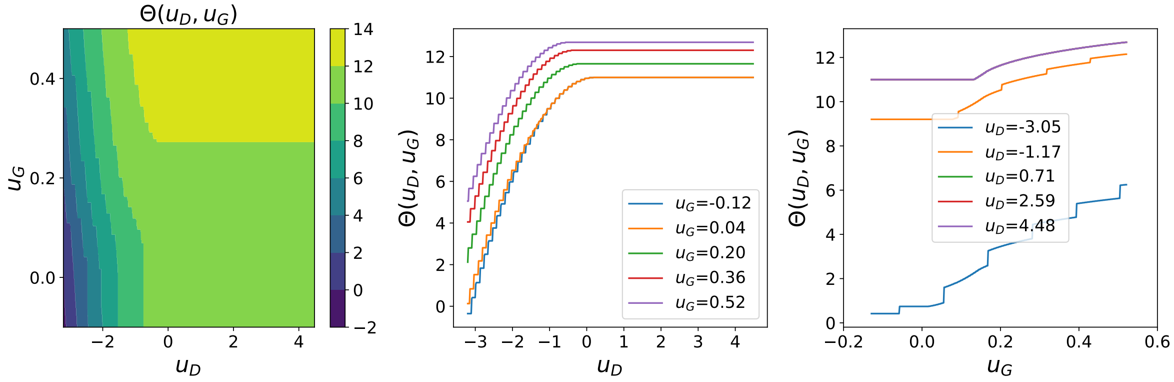


Figure 4.3: Θ and its cross-sections, fixing separately u_D and u_G. Here $p = q = 3, \sigma_z = 1, \kappa = 0.9$.

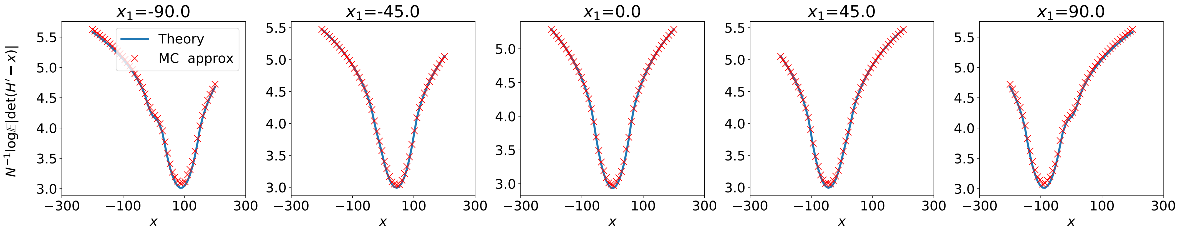


Figure 4.4: Comparison of (4.56) and (4.62), verifying the Coulomb gas approximation numerically. Here $p = q = 3, \sigma_z = 1, \kappa = 0.9$. Sampled matrices for MC approximation are dimension $N = 50$, and $n = 50$ MC samples have been used.

Lemma 4.4. *Let $(H_N)_{N=1}^{\infty}$ be a sequence of random matrices, where for each N*

$$H_N \overset{d}{=} bM + b_1 \begin{pmatrix} M_1 & 0 \\ 0 & 0 \end{pmatrix} - x_1 \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \tag{4.63}$$

and $M \sim GOE^N$, $M_1 \sim GOE^{\kappa N}$. Let μ_N be the empirical spectral measure of H_N and say $\mu_N \to \mu_{eq}$ weakly almost surely. Then for any $(x, x_1) \in \mathbb{R}^2$

$$\mathbb{E}|\det(H_N - xI)| = \exp\left\{N(1+o(1))\int \log|z-x| d\mu_{eq}(z)\right\} \tag{4.64}$$

as $N \to \infty$.

Proof. We begin by establishing an upper bound. Take any $\beta > 0$, then

$$\begin{aligned}
&\int \log|z-x| d\mu_N(z) \\
&= \int \log|z-x| \mathbb{1}\{|x-z| \geqslant e^{\beta}\} d\mu_N(z) + \int \log|z-x| \mathbb{1}\{\log|x-z| < \beta\} d\mu_N(z) \\
&\leqslant \int \log|z-x| \mathbb{1}\{|x-z| \geqslant e^{\beta}\} d\mu_N(z) + \int \min(\log|x-z|, \beta) d\mu_N(z). \qquad (4.65)
\end{aligned}$$

Take also any $\alpha > 0$, then trivially

$$\int \min(\log|x-z|, \beta) d\mu_N(z) \leqslant \int \max(-\alpha, \min(\log|x-z|, \beta)) d\mu_N(z). \tag{4.66}$$

Overall we have, for any $\alpha, \beta > 0$,

$$\begin{aligned}
&\exp\left\{N\int \log|z-x|d\mu_N(z)\right\} \\
\leqslant &\exp\left\{N\int \log|z-x|\mathbb{1}\{|x-z|\geqslant e^{\beta}\}d\mu_N(z)\right\} \\
&\exp\left\{N\int \max(-\alpha, \min(\log|x-z|,\beta))d\mu_N(z)\right\}. \qquad (4.67)
\end{aligned}$$

Thence an application of Hölder's inequality gives

$$\begin{aligned}
\mathbb{E}|\det(H_N-xI)| &= \mathbb{E}\left[\exp\left\{N\int \log|z-x|d\mu_N(z)\right\}\right] \\
&\leqslant \underbrace{\left(\mathbb{E}\left[\exp\left\{2N\int \max\left(-\alpha, \min\left(\log|x-z|,\beta\right)\right)d\mu_N(z)\right\}\right]\right)^{1/2}}_{A_N} \\
&\quad \underbrace{\left(\mathbb{E}\left[\exp\left\{2N\int \log|x-z|\mathbb{1}\{|x-z|\geqslant e^{\beta}\}d\mu_N(z)\right\}\right]\right)^{1/2}}_{B_N}. \qquad (4.68)
\end{aligned}$$

Considering B_N, we have

$$\log|x-z|\mathbb{1}\{|x-z|\geqslant e^{\beta}\} \leqslant |x-z|^{1/2}\mathbb{1}\{|x-z|\geqslant e^{\beta}\} \leqslant e^{-\beta/2}|x-z| \qquad (4.69)$$

and so

$$\begin{aligned}
\mathbb{E}\left[\exp\left\{2N\int \log|x-z|\mathbb{1}\{|x-z|\geqslant e^{\beta}\}\right\}\right] &\leqslant \mathbb{E}\left[\exp\left\{2Ne^{-\beta/2}\frac{\mathrm{Tr}|H_N-xI|}{N}\right\}\right] \\
&= \mathbb{E}\left[\exp\left\{2e^{-\beta/2}\mathrm{Tr}|H_N-xI|\right\}\right]. \qquad (4.70)
\end{aligned}$$

The entries of H_N are Gaussians with variance $\frac{1}{N}b^2, \frac{1}{2N}b^2, \frac{1}{N}(b^2+b_1^2)$ or $\frac{1}{2N}(b^2+b_1^2)$ and all the diagonal and upper diagonal entries are independent. All of these variances are $\mathcal{O}(N^{-1})$, so

$$|H_N-x|_{ij} \leqslant |x|+|x_1|+\mathcal{O}(N^{-1/2})|X_{ij}| \qquad (4.71)$$

where the X_{ij} are i.i.d. standard Gaussians for $i\leqslant j$. It follows that

$$\mathbb{E}\left[\exp\left\{2e^{-\frac{\beta}{2}}\mathrm{Tr}|H_N-xI|\right\}\right] \leqslant e^{2e^{-\frac{\beta}{2}}N(|x|+|x_1|)}\mathbb{E}_{X\sim\mathcal{N}(0,1)}e^{2e^{-\frac{\beta}{2}}\mathcal{O}(N^{1/2})|X|}. \qquad (4.72)$$

Elementary calculations give

$$\mathbb{E}_{X\sim\mathcal{N}(0,1)}e^{c|X|} \leqslant \frac{1}{2}\left(e^{-c^2}+e^{c^2}\right) \leqslant e^{c^2} \qquad (4.73)$$

and so

$$\begin{aligned}
\mathbb{E}\left[\exp\left\{2e^{-\frac{\beta}{2}}\mathrm{Tr}|H_N-xI|\right\}\right] &\leqslant e^{2e^{-\frac{\beta}{2}}N(|x|+|x_1|)}e^{4e^{-\beta}\mathcal{O}(N)} \\
&= \exp\left\{2N\left(e^{-\frac{\beta}{2}}(|x|+|x_1|)+e^{-\beta}\mathcal{O}(1)\right)\right\} \qquad (4.74)
\end{aligned}$$

thus when we take $\beta \to \infty$, we have $B_N \leqslant e^{o(N)}$.

Considering A_N, it is sufficient now to show

$$\mathbb{E}\left[\exp\left\{2N\int f(z)d\mu_N(z)\right\}\right] = \exp\left\{2N\left(\int f(z)d\mu_{eq}(z) + o(1)\right)\right\} \tag{4.75}$$

where $f(z) = 2\max\left(\min(\log|x-z|, \beta), -\alpha\right)$, a continuous and bounded function. For any $\varepsilon > 0$, we have

$$\begin{aligned}&\mathbb{E}\left[\exp\left\{2N\int f(z)d\mu_N(z)\right\}\right]\\ &\leqslant \exp\left\{2N\left(\int f(z)d\mu_{eq}(z) + \varepsilon\right)\right\} + e^{2N\|f\|_\infty}\mathbb{P}\left(\int f(z)d\mu_N(z) \geqslant \int f(z)d\mu_{eq}(z) + \varepsilon\right).\end{aligned} \tag{4.76}$$

The entries of H_N are Gaussian with $\mathcal{O}(N^{-1})$ variance and so obey a log-Sobolev inequality as required by Theorem 1.5 from [GZ+00]. The constant, c, in the inequality is independent of N, x, x_1, so we need not compute it exactly. The theorem from [GZ+00] then gives

$$\mathbb{P}\left(\int f(z)d\mu_N(z) \geqslant \int f(z)d\mu_{eq}(z) + \varepsilon\right) \leqslant \exp\left\{-\frac{N^2}{8c}\varepsilon^2\right\}. \tag{4.77}$$

We have shown

$$\begin{aligned}\mathbb{E}|\det(H_N - xI)| \leqslant A_N B_N &\leqslant \exp\left\{N(1+o(1))\left(\int f(z)d\mu_{eq}(z)\right)\right\}\\ &\leqslant \exp\left\{N(1+o(1))\left(\int \log|x-z|d\mu_{eq}(z)\right)\right\}.\end{aligned} \tag{4.78}$$

We now need to establish a complimentary lower bound to complete the proof. By Jensen's inequality

$$\begin{aligned}\mathbb{E}|\det(H_N - x)| &\geqslant \exp\left(N\mathbb{E}\left[\int \log|z-x|d\mu_N(z)\right]\right)\\ &\geqslant \exp\left(N\mathbb{E}\left[\int \max\left(-\alpha, \log|z-x|\right)d\mu_N(z)\right]\right)\exp\left(N\mathbb{E}\left[\int \log|z-x|\mathbb{1}\{|z-x| \leqslant e^{-\alpha}\}d\mu_N(z)\right]\right)\\ &\geqslant \exp\left(N\mathbb{E}\left[\int \min\left(\beta, \max\left(-\alpha, \log|z-x|\right)\right)d\mu_N(z)\right]\right)\\ &\quad\exp\left(N\mathbb{E}\left[\int \log|z-x|\mathbb{1}\{|z-x| \leqslant e^{-\alpha}\}d\mu_N(z)\right]\right)\end{aligned} \tag{4.79}$$

for any $\alpha, \beta > 0$. Convergence in law of μ_N to μ_{eq} and the dominated convergence theorem give

$$\exp\left(N\mathbb{E}\left[\int \min\left(\beta, \max\left(-\alpha, \log|z-x|\right)\right)d\mu_N(z)\right]\right) \geqslant \exp\left\{N\left(\int \log|x-z|d\mu_{eq}(z) + o(1)\right)\right\} \tag{4.80}$$

for large enough β, because μ_{eq} has compact support. It remains to show that the expectation inside the exponent in the second term of (4.79) converges to zero uniformly in N in the limit $\alpha \to \infty$.

By (4.30), it is sufficient to consider $\langle G_N(z)\rangle$, which is computed via (4.47). Let us define the function Ψ so that

$$\langle G_N(z)\rangle = \frac{2}{b^2}\int d\sigma d\sigma[1](z - i\sigma_{FF})e^{-N\Psi(\sigma,\sigma[1])}. \tag{4.81}$$

Henceforth, $\sigma_{FF}^*, \sigma_{FF}[1]^*, \sigma_{BB}^*, \sigma_{BB}[1]^*$ are the solution to the saddle point equations (4.48-4.51) and $\tilde{\sigma}_{FF}, \tilde{\sigma}_{FF}[1], \tilde{\sigma}_{BB}, \tilde{\sigma}_{BB}[1]$ are integration variables. Around the saddle point

$$z - i\sigma_{FF} = z - i\sigma_{FF}^* - iN^{-\frac{1}{r}}\tilde{\sigma}_{FF} \tag{4.82}$$

for some $r \geqslant 2$. We use the notation $\boldsymbol{\sigma}$ for $(\sigma_{BB}, \sigma_{BB}[1], \sigma_{FF}, \sigma_{FF}[1])$ and similarly $\boldsymbol{\sigma}_{BB}, \boldsymbol{\sigma}_{FF}$. A superscript asterisk on Ψ or any of its derivatives is short hand for evaluation at the saddle point. While the Hessian of Ψ may not in general vanish at the saddle point,

$$\int d\tilde{\sigma} d\tilde{\sigma}[1]\tilde{\sigma}_{FF}e^{-N\tilde{\boldsymbol{\sigma}}^T\nabla^2\Psi^*\tilde{\boldsymbol{\sigma}}} = 0 \tag{4.83}$$

and so we must go to at least the cubic term in the expansion of Ψ around the saddle point, i.e.

$$\langle G_N(z)\rangle = G(z) - \frac{2i}{b^2N^{5/3}}\underbrace{\int_{-\infty}^{\infty} d\tilde{\boldsymbol{\sigma}}_{BB}d\tilde{\boldsymbol{\sigma}}_{FF}\tilde{\sigma}_{FF}e^{-\frac{1}{6}\tilde{\sigma}^i\tilde{\sigma}^j\tilde{\sigma}^k\partial_{ijk}\Psi^*}}_{E(z;x_1)} + \text{exponentially smaller terms.} \tag{4.84}$$

The bosonic (BB) and fermionic (FF) coordinates do not interact, so we can consider derivatives of Φ as block tensors. Simple differentiation gives

$$\begin{aligned}
(\nabla\Psi)_B &= \begin{pmatrix} \frac{2}{b^2}\sigma_{BB} - \kappa\left(\sigma_{BB} + \sigma_{BB}[1] + z + x_1\right)^{-1} - \kappa'\left(\sigma_{BB} + z\right)^{-1} \\ \frac{2}{b_1^2}\sigma_{BB}[1] - \left(\sigma_{BB} + \sigma_{BB}[1] + z + x_1\right)^{-1} \end{pmatrix} \\
\implies (\nabla^2\Psi)_B &= \begin{pmatrix} \kappa\left(\sigma_{BB} + \sigma_{BB}[1] + z + x_1\right)^{-2} + \kappa'\left(\sigma_{BB} + z\right)^{-2} & \kappa\left(\sigma_{BB} + \sigma_{BB}[1] + z + x_1\right)^{-2} \\ \left(\sigma_{BB} + \sigma_{BB}[1] + z + x_1\right)^{-2} & \left(\sigma_{BB} + \sigma_{BB}[1] + z + x_1\right)^{-2} \end{pmatrix}
\end{aligned} \tag{4.85}$$

$$\implies (\nabla^3\Psi)_B^* = \left(\begin{pmatrix} A_B\kappa + B_B\kappa' & A_B\kappa \\ A_B & A_B \end{pmatrix}, A_B\begin{pmatrix} \kappa & \kappa \\ 1 & 1 \end{pmatrix}\right), \tag{4.86}$$

where

$$A_B = -\frac{2}{\left(\sigma_{BB}^* + \sigma_{BB}^*[1] + z + x_1\right)^3}, \quad B_B = -\frac{2}{\left(\sigma_{BB}^* + z\right)^3}. \tag{4.87}$$

$(\nabla^3\Psi)_F^*$ follows similarly with

$$A_F = -\frac{2}{\left(\sigma_{FF}^* + \sigma_{FF}^*[1] - iz - ix_1\right)^3}, \quad B_F = -\frac{2}{\left(\sigma_{FF}^* - iz\right)^3}. \tag{4.88}$$

By the saddle point equations (4.48)-(4.51) we have

$$A_B = 2(\sigma_{BB}[1]^*)^3, \quad B_B = \frac{2}{(\kappa')^3}\left(\frac{2\kappa}{b_1^2}\sigma_{BB}[1]^* - \frac{2}{b^2}\sigma_{BB}^*\right)^3 \tag{4.89}$$

$$A_F = 2(\sigma_{FF}[1]^*)^3, \quad B_F = \frac{2}{(\kappa')^3}\left(\frac{2\kappa}{b_1^2}\sigma_{FF}[1]^* - \frac{2}{b^2}\sigma_{FF}^*\right)^3. \tag{4.90}$$

Let $\xi_1 = \tilde{\sigma}_{BB}, \xi_2 = \tilde{\sigma}_{BB}[1]$. Then

$$\begin{aligned}(\tilde{\sigma}^i\tilde{\sigma}^j\tilde{\sigma}^k\partial_{ijk}\Phi^*)_B &= \left(A_B\kappa + B_B\kappa'\right)\xi_1^3 + A_B(2\kappa+1)\xi_1^2\xi_2[1] + A_B(\kappa+2)\xi_1\xi_2^2 + A_B\xi_2^3 \\ &= A_B\left[\xi_2^3 + (2\kappa+1)\xi_2\xi_1^2 + (2+\kappa)\xi_1\xi_2^2 + C\xi_1^3\right] + \left(B_B\kappa' + A_B\kappa - CA_B\right)\xi_1^3 \end{aligned} \tag{4.91}$$

for any C. Let $\xi_1 = a_1\xi_1'$ and then choose $C = a_1^{-3}$ and $a_1 = (2+\kappa)(2\kappa+1)^{-1}$ to give

$$(\tilde{\sigma}^i\tilde{\sigma}^j\tilde{\sigma}^k\partial_{ijk}\Phi^*)_B = A_B(\xi_1' + \xi_2)^3 + (B_B\kappa' + A_B\kappa - CA_B)a_1^3(\xi_1')^3 \equiv A_B\eta^3 + D_B\xi^3 \tag{4.92}$$

with $\eta = \xi_1' + \xi_2$, $\xi = \xi_1'$, $D_B = B_B\kappa' + A_B\kappa - a_1^{-3}A_B$. The expressions for $(\tilde{\sigma}^i\tilde{\sigma}^j\tilde{\sigma}^k\partial_{ijk}\Phi^*)_F$ follow identically. We thus have

$$E(z;x_1) \propto \left(\int_0^\infty d\xi\ \xi\int_\xi^\infty d\eta\ e^{A_F\eta^3 + D_F\xi^3}\right)\left(\int_0^\infty d\xi\ \int_\xi^\infty d\eta\ e^{A_B\eta^3 + D_B\xi^3}\right) \tag{4.93}$$

or perhaps with the the integration ranges reversed depending on the signs of $\Re A_F, \Re A_B, \Re D_F, \Re D_B$. We have

$$\begin{aligned}|E(z;x_1)| &\leqslant \left|\int_0^\infty d\xi\ \xi\int_\xi^\infty d\eta\ e^{A_F\eta^3 + D_F\xi^3}\right| \cdot \left|\int_0^\infty d\xi\ \int_\xi^\infty d\eta\ e^{A_B\eta^3 + D_B\xi^3}\right| \\ &\leqslant \int_0^\infty d\xi\ \xi\int_\xi^\infty d\eta\ |e^{A_F\eta^3 + D_F\xi^3}| \cdot \int_0^\infty d\xi\ \int_\xi^\infty d\eta\ |e^{A_B\eta^3 + D_B\xi^3}| \\ &\leqslant \int_0^\infty d\xi\ \xi\int_0^\infty d\eta\ |e^{A_F\eta^3 + D_F\xi^3}| \cdot \int_0^\infty d\xi\ \int_0^\infty d\eta\ |e^{A_B\eta^3 + D_B\xi^3}| \\ &\leqslant (|\mathfrak{M}D_F|)^{-2/3}(|\mathfrak{M}A_F|)^{-1/3}(|\mathfrak{M}D_B|)^{-1/3}(|\mathfrak{M}A_B|)^{-1/3}\left(\int_0^\infty e^{-\xi^3}d\xi\right)^3\left(\int_0^\infty \xi e^{-\xi^3}d\xi\right) \end{aligned} \tag{4.94}$$

where we have defined

$$\mathfrak{M}y = \begin{cases}\Re y & \text{if } \Re y \neq 0, \\ \Im y & \text{if } \Re y = 0.\end{cases} \tag{4.95}$$

This last bound follows from a standard Cauchy rotation of integration contour if any of D_F, A_F, D_B, A_B has vanishing real part. (4.94) is valid for $D_B, A_B, D_F, A_F \neq 0$, but if $D_B = 0$ and $A_B \neq 0$, then the preceding calculations are simplified and we still obtain an upper bound but proportional to $(|\mathfrak{M}A_B|)^{-1/3}$. Similarly with $A_B = 0$ and $D_B \neq 0$ and similarly for A_F, D_F. The only remaining cases are $A_B = D_B = 0$ or $A_F = D_F = 0$. But recall (4.90) and (4.50)-(4.51). We immediately see that $A_F = D_F$ if and only if $\sigma_{FF} = \sigma_{FF}[1] = 0$, which occurs for no finite z, x_1. Therefore, for *fixed* $(x, x_1) \in \mathbb{R}^2$, $\alpha > 0$ and any $z \in (x - e^{-\alpha}, x + e^{-\alpha})$

$$|\mathbb{E}\mu_N(z) - \mu_{eq}(z;x_1)| \lesssim N^{-5/3}C(x_1, |x| + e^{-\alpha}) \tag{4.96}$$

where $C(|x_1|, |x| + e^{-\alpha})$ is positive and is decreasing in α. Since μ_{eq} is bounded, it follows that $\mathbb{E}\mu_N$ is bounded, and therefore

$$\mathbb{E}\int \log|z - x|\mathbb{1}\{|z-x| \leqslant e^{-\alpha}\}d\mu_N(z) \to 0 \tag{4.97}$$

as $\alpha \to \infty$ uniformly in N, and so the lower bound is completed. ■

Equipped with this result, we can now prove the legitimacy of the Coulomb gas approximation in our complexity calculation. The proof will require an elementary intermediate result which has undoubtedly appeared in various places before, but we prove it here anyway for the avoidance of doubt.

Lemma 4.5. *Let M_N be a random $N \times N$ symmetric real matrix with independent centred Gaussian upper-diagonal and diagonal entries. Suppose that the variances of the entries are bounded above by cN^{-1} for some constant $c > 0$. Then there exists some constant c_e such that*

$$\mathbb{E}||M_N||_{max}^N \lesssim e^{c_e N}. \tag{4.98}$$

Proof. Let σ_{ij}^2 denote the variance of M_{ij}. Then

$$\begin{aligned}\mathbb{E}||M||_{max}^N &\leqslant \sum_{i,j} \mathbb{E}|M_{i,j}|^N \\ &= \sum_{i,j} \mathbb{E}|\mathcal{N}(0,\sigma_{ij}^2)|^N \\ &= \sum_{i,j} \sigma_{ij}^N \mathbb{E}|\mathcal{N}(0,1)|^N \\ &\leqslant N^2 c^{N/2} N^{-N/2} \mathbb{E}|\mathcal{N}(0,1)|^N. \end{aligned} \tag{4.99}$$

Simple integration with a change of variables gives

$$\mathbb{E}|\mathcal{N}(0,1)|^N = 2^{\frac{N+1}{2}} \Gamma\left(\frac{N+1}{2}\right) \tag{4.100}$$

and then, for large enough N, Stirling's formula gives

$$\begin{aligned}\mathbb{E}|\mathcal{N}(0,1)|^N &\sim 2^{\frac{N+1}{2}} \sqrt{\pi(N+1)} \left(\frac{N+1}{2e}\right)^{\frac{N-1}{2}} \\ &\sim 2\sqrt{\pi} e^{-\frac{N-1}{2}} N^{N/2} \left(\frac{N+1}{N}\right)^{N/2} \\ &\sim 2\sqrt{\pi e} N^{N/2}. \end{aligned} \tag{4.101}$$

So finally

$$\mathbb{E}||M||_{max}^N \lesssim N^2 c^{N/2} = e^{\frac{1}{2}N\log c + 2\log N} \leqslant e^{\left(\frac{1}{2}\log c + 2\right)N}, \tag{4.102}$$

so defining $c_e = \frac{1}{2}\log 2 + 2$ gives the result. ■

Theorem 4.1. *For any $x_1 \in \mathbb{R}$, let H_N be a random $N \times N$ matrix distributed as in the statement of Lemma 4.4. Then as $N \to \infty$*

$$\begin{aligned}&\iint_B dx dx_1 \ \exp\left\{-N\left(\frac{1}{2s^2}x^2 + \frac{1}{2s_1^2}(x_1)^2\right)\right\} \mathbb{E}|\det(H_N(x_1) - x)| \\ =& \iint_B dx dx_1 \ \exp\left\{-N\left(\frac{1}{2s^2}x^2 + \frac{1}{2s_1^2}(x_1)^2 - \int \log|z - x| d\mu_{eq}(z) + o(1)\right)\right\} + o(1). \end{aligned} \tag{4.103}$$

Proof. Let $R > 0$ be some constant, independent of N. Introduce the notation $B_{\leqslant R} = B \cap \{\boldsymbol{z} \in \mathbb{R}^2 \mid |z| \leqslant R\}$, and then

$$\begin{aligned}&\Bigg|\iint_B dxdx_1\ \exp\left\{-N\left(\frac{1}{2s^2}x^2 + \frac{1}{2s_1^2}(x_1)^2\right)\right\}\mathbb{E}|\det(H_N(x_1) - x)|\\ &- \iint_{B_{\leqslant R}} dxdx_1\ \exp\left\{-N\left(\frac{1}{2s^2}x^2 + \frac{1}{2s_1^2}(x_1)^2\right)\right\}\mathbb{E}|\det(H_N(x_1) - x)|\Bigg|\\ &\leqslant \iint_{||\boldsymbol{x}||\geqslant R} dxdx_1\ \exp\left\{-N\left(\frac{1}{2s^2}x^2 + \frac{1}{2s_1^2}(x_1)^2\right)\right\}\mathbb{E}|\det(H_N(x_1) - x)|. \end{aligned} \tag{4.104}$$

We have the upper bound (4.78) of Lemma 4.4 but this cannot be directly applied to (4.104) since the bound relies on uniformity in x, x_1 which can only be established for bounded x, x_1. We use a much cruder bound instead. First, let

$$J_N = H_N + x_1 \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \tag{4.105}$$

and then

$$|\det(H_N - xI)| \leqslant ||J_N||_{\max}^N \max\{|x|, |x_1|\}^N = ||J_N||_{\max}^N \exp\left(N \max\{\log|x|, \log|x_1|\}\right). \tag{4.106}$$

J_N has centred Gaussian entries with variance $\mathcal{O}(N^{-1})$, so Lemma 4.5 applies, and we find

$$\mathbb{E}|\det(H_N - xI)| \lesssim \exp\left(N \max\{\log|x|, \log|x_1|\}\right) e^{c_e N} \tag{4.107}$$

for some constant $c_e > 0$ which is independent of x, x_1 and N, but we need not compute it.

Now we have

$$\begin{aligned}&\Bigg|\iint_B dxdx_1\ \exp\left\{-N\left(\frac{1}{2s^2}x^2 + \frac{1}{2s_1^2}(x_1)^2\right)\right\}\mathbb{E}|\det(H_N(x_1) - x)|\\ &- \iint_{B_{\leqslant R}} dxdx_1\ \exp\left\{-N\left(\frac{1}{2s^2}x^2 + \frac{1}{2s_1^2}(x_1)^2\right)\right\}\mathbb{E}|\det(H_N(x_1) - x)|\Bigg|\\ &\lesssim \iint_{||\boldsymbol{x}||\geqslant R} dxdx_1\ \exp\left\{-N\left(\frac{1}{2s^2}x^2 + \frac{1}{2s_1^2}(x_1)^2 - \max\{\log|x|, \log|x_1|\} - c_e\right)\right\}. \end{aligned} \tag{4.108}$$

But, since μ_{eq} is bounded and has compact support, we can choose R large enough (independent of N) so that

$$\frac{1}{2s^2}x^2 + \frac{1}{2s_1^2}(x_1)^2 - \max\{\log|x|, \log|x_1|\} - c_e > L > 0 \tag{4.109}$$

for all (x, x_1) with $\sqrt{x^2 + x_1^2} > R$ and for some fixed L independent of N. Whence

$$\begin{aligned}&\Bigg|\iint_B dxdx_1\ \exp\left\{-N\left(\frac{1}{2s^2}x^2 + \frac{1}{2s_1^2}(x_1)^2\right)\right\}\mathbb{E}|\det(H_N(x_1) - x)|\\ &- \iint_{B_{\leqslant R}} dxdx_1\ \exp\left\{-N\left(\frac{1}{2s^2}x^2 + \frac{1}{2s_1^2}(x_1)^2\right)\right\}\mathbb{E}|\det(H_N(x_1) - x)|\Bigg|\\ &\lesssim N^{-1} e^{-NL} \to 0 \end{aligned} \tag{4.110}$$

as $N \to \infty$. Finally, for x, x_1 in $B_{\leqslant R}$, the result of the Lemma 4.4 holds uniformly in x, x_1, so

$$\begin{aligned}&\iint_{B_{\leqslant R}} dx dx_1 \, \exp\left\{-N\left(\frac{1}{2s^2}x^2 + \frac{1}{2s_1^2}(x_1)^2\right)\right\}\mathbb{E}|\det(H_N(x_1) - x)| \\ =& \iint_{B_{\leqslant R}} dx dx_1 \, \exp\left\{-N\left(\frac{1}{2s^2}x^2 + \frac{1}{2s_1^2}(x_1)^2 - \int \log|z-x| d\mu_{eq}(z; x_1) + o(1)\right)\right\}. \qquad (4.111)\end{aligned}$$

The result follows from (4.110), (4.111) and the triangle inequality. ■

4.4.1 Asymptotic complexity with prescribed Hessian index

Recall the complexity defined in (4.7):

$$\begin{aligned}C_{N,k_D,k_G} = \Bigg|\Bigg\{&\boldsymbol{w}^{(D)} \in S^{N_D}, \boldsymbol{w}^{(G)} \in S^{N_G} \ : \ \nabla_D L^{(D)} = 0, \nabla_G L^{(G)} = 0, L^{(D)} \in B_D, L^{(G)} \in B_G \\ & i(\nabla_D^2 L^{(D)}) = k_D, \ i(\nabla_G^2 L^{(G)}) = k_G\Bigg\}\Bigg|. \qquad (4.7)\end{aligned}$$

The extra Hessian signature conditions in (4.7) enforce that both generator and discriminator are at low-index saddle points. Our method for computing the complexity C_N in the previous subsection relies on the Coulomb gas approximation applied to the spectrum of H'. However, the Hessian index constraints are formulated in the natural Hessian matrix (4.16), but our spectral calculations proceed from the rewritten form (4.21). We find however that we can indeed proceed much as in Chapter 3. Recall the key Hessian matrix $\tilde{H}$ given in (4.16) by

$$\begin{aligned}\tilde{H} = &\begin{pmatrix} \sqrt{2(N_D-1)}\sqrt{b^2+b_1^2}M^{(D)} & -bG \\ bG^T & \sqrt{2(N_G-1)}bM^{(G)}\end{pmatrix} \\ & - \sqrt{N-2}x\begin{pmatrix} -I_{N_D} & 0 \\ 0 & I_{N_G}\end{pmatrix} + \sqrt{N-2}x_1\begin{pmatrix} I_{N_D} & 0 \\ 0 & 0\end{pmatrix} \qquad (4.112)\end{aligned}$$

where $M^{(D)} \sim GOE^{N_D-1}$, $M^{(G)} \sim GOE^{N_G-1}$, G is $N_D - 1 \times N_G - 1$ Ginibre, and all are independent. Note that we have used (4.23) to slightly rewrite (4.16). We must address the problem of computing

$$\mathbb{E}|\det\tilde{H}|\mathbb{1}\left\{i\left(\sqrt{\kappa}(1+\mathcal{O}(N^{-1}))\sqrt{b^2+b_1^2}M_D + \frac{x+x_1}{\sqrt{2}}\right) = k_D, \ i\left(\sqrt{\kappa'}(1+\mathcal{O}(N^{-1}))bM_G - \frac{x}{\sqrt{2}}\right) = k_G\right\}. \qquad (4.113)$$

Indeed, we introduce integration variables $\boldsymbol{y}_1, \boldsymbol{y}_2, \zeta_1, \zeta_1^*, \zeta_2, \zeta_2^*$, being $(N-2)$-vectors of commuting and anti-commuting variables respectively. Use $[t]$ notation to split all vectors into the first $\kappa N - 1$ and last $\kappa' N - 1$ components. Let

$$A[t] = \boldsymbol{y}_1\boldsymbol{y}_1^T + \boldsymbol{y}_2\boldsymbol{y}_2^T + \zeta_1\zeta_1^\dagger + \zeta_2\zeta_2^\dagger. \qquad (4.114)$$

With these definitions, we have (recalling Chapter 3)

$$|\det\tilde{H}| = (2(N-2))^{\frac{N-2}{2}} \lim_{\epsilon\searrow 0}\int d\Xi \exp\left\{-i\sqrt{\kappa}(1+\mathcal{O}(N^{-1}))\sqrt{b^2+b_1^2}\mathrm{Tr}M^{(D)}A[1]\right.$$
$$\left.-i\sqrt{\kappa'}(1+\mathcal{O}(N^{-1}))b\mathrm{Tr}M^{(G)}A[2]\right\}$$
$$\exp\{\mathcal{O}(\epsilon)\}\exp\{\ldots\} \tag{4.115}$$

where $d\Xi$ is the normalised measure of the $\boldsymbol{y}_1,\boldsymbol{y}_2,\zeta_1,\zeta_1^*,\zeta_2,\zeta_2^*$ and the ellipsis represents terms with no dependence on $M^{(D)}$ or $M^{(G)}$, which we need not write down. The crux of the matter is that we must compute

$$\mathbb{E}_{M^{(D)}}e^{-i\sqrt{\kappa}\sqrt{b^2+b_1^2}\mathrm{Tr}M^{(D)}A[1]}\mathbb{1}\left\{i\left(M_D+\frac{x+x_1}{\sqrt{\kappa}\sqrt{b^2+b_1^2}}(1+\mathcal{O}(N^{-1}))\right)=k_D\right\}, \tag{4.116}$$

$$\mathbb{E}_{M^{(G)}}e^{-i\sqrt{\kappa'}b\mathrm{Tr}M^{(G)}A[2]}\mathbb{1}\left\{i\left(M_G-\frac{x}{\sqrt{\kappa'}b}(1+\mathcal{O}(N^{-1}))\right)=k_G\right\}, \tag{4.117}$$

but in Chapter 3 we performed exactly these calculations (see around (3.242)) and so there exist constants $K_U^{(D)},K_L^{(D)},K_U^{(G)},K_L^{(G)}$ such that

$$K_L^{(D)}e^{-Nk_D\kappa(1+o(1))I_1(\hat{x}_D;\sqrt{2})}e^{-\frac{1}{2N}(b^2+b_1^2)\mathrm{Tr}A[1]^2}$$
$$\leqslant\Re\mathbb{E}_{M^{(D)}}e^{-i\sqrt{\kappa}\sqrt{b^2+b_1^2}\mathrm{Tr}M^{(D)}A[1]}\mathbb{1}\left\{i\left(M_D+\frac{x+x_1}{\sqrt{\kappa}\sqrt{b^2+b_1^2}}(1+\mathcal{O}(N^{-1}))\right)=k_D\right\}$$
$$\leqslant K_U^{(D)}e^{-Nk_D\kappa(1+o(1))I_1(\hat{x}_D;\sqrt{2})}e^{-\frac{1}{2N}(b^2+b_1^2)\mathrm{Tr}A[1]^2} \tag{4.118}$$

and

$$K_L^{(G)}e^{-Nk_G\kappa'(1+o(1))I_1(\hat{x}_G;\sqrt{2})}e^{-\frac{1}{2N}b^2\mathrm{Tr}A[2]^2}$$
$$\leqslant\Re\mathbb{E}_{M^{(G)}}e^{-i\sqrt{\kappa'}b\mathrm{Tr}M^{(G)}A[2]}\mathbb{1}\left\{i\left(M_G-\frac{x}{\sqrt{\kappa'}b}(1+\mathcal{O}(N^{-1}))\right)=k_G\right\}$$
$$\leqslant K_U^{(G)}e^{-Nk_G\kappa'(1+o(1))I_1(\hat{x}_G;\sqrt{2})}e^{-\frac{1}{2N}b^2\mathrm{Tr}A[2]^2} \tag{4.119}$$

where

$$\hat{x}_D=-\frac{x+x_1}{\sqrt{\kappa}\sqrt{b^2+b_1^2}},\quad \hat{x}_G=\frac{x}{\sqrt{\kappa'}b}. \tag{4.120}$$

Here I_1 is the rate function of the largest eigenvalue of the GOE as obtained in [ADG01] and used in [AAC13] and Chapter 3:

$$I_1(u;E)=\begin{cases}\frac{2}{E^2}\int_u^{-E}\sqrt{z^2-E^2}dz & \text{for } u<-E,\\ \frac{2}{E^2}\int_E^{u}\sqrt{z^2-E^2}dz & \text{for } u>E,\\ \infty & \text{for } |u|<E.\end{cases} \tag{4.121}$$

Note that for $u < -E$

$$I_1(u;E) = -\frac{u}{E}\sqrt{u^2-E^2} - \log\left(-u+\sqrt{u^2-E^2}\right) + \log E \tag{4.122}$$

and for $u > E$ we simply have $I_1(u;E) = I_1(-u;E)$. Note also that $I_1(ru;E) = I_1(u,E/r)$.

We have successfully dealt with the Hessian index indicators inside the expectation, however we need some way of returning to the form of $\tilde{H}$ in (4.21) so the complexity calculations using the Coulomb gas approach can proceed as before. We can achieve this with inverse Fourier transforms:

$$e^{-\frac{1}{2N}(b^2+b_1^2)\mathrm{Tr}A[1]^2} = \mathbb{E}_{M_D} e^{-i\sqrt{\kappa}\sqrt{b^2+b_1^2}\mathrm{Tr}M_D A[1]} \tag{4.123}$$

$$e^{-\frac{1}{2N}b^2\mathrm{Tr}A[2]^2} = \mathbb{E}_{M_G} e^{-i\sqrt{\kappa'}b\mathrm{Tr}M_G A[2]} \tag{4.124}$$

from which we obtain

$$\begin{aligned}
&K_L e^{-Nk_D\kappa(1+o(1))I_1(\hat{x}_D;\sqrt{2})} e^{-Nk_G\kappa'(1+o(1))I_1(\hat{x}_G;\sqrt{2})}\mathbb{E}|\det\tilde{H}| \\
\leqslant &\mathbb{E}|\det\tilde{H}|\mathbb{1}\left\{i\left(\sqrt{\kappa}(1+\mathcal{O}(N^{-1}))\sqrt{b^2+b_1^2}M_D + \frac{x+x_1}{\sqrt{2}}\right) = k_D,\ i\left(\sqrt{\kappa'}(1+\mathcal{O}(N^{-1}))bM_G - \frac{x}{\sqrt{2}}\right) = k_G\right\} \qquad (4.125)\\
\leqslant &K_U e^{-Nk_D\kappa(1+o(1))I_1(\hat{x}_D;\sqrt{2})} e^{-Nk_G\kappa'(1+o(1))I_1(\hat{x}_G;\sqrt{2})}\mathbb{E}|\det\tilde{H}|. \qquad (4.126)
\end{aligned}$$

It follows that

$$\begin{aligned}
&K'_N \iint_B dx dx_1 e^{-(N-2)\left[\Phi(x,x_1)+k_G\kappa' I_1(x;\sqrt{2\kappa'}b)+k_D\kappa I_1\left((-(x+x_1);\sqrt{2\kappa(b^2+b_1^2)}\right)\right](1+o(1))} \\
\lesssim &C_{N,k_D,k_G} \\
\lesssim &K'_N \iint_B dx dx_1 e^{-(N-2)\left[\Phi(x,x_1)+k_G\kappa' I_1(x;\sqrt{2\kappa'}b)+k_D\kappa I_1\left((-(x+x_1);\sqrt{2\kappa(b^2+b_1^2)}\right)\right](1+o(1))}. \qquad (4.127)
\end{aligned}$$

So we see that the relevant exponent in this case is the same as for C_N but with additional GOE eigenvalue large deviation terms, giving the complexity limit

$$\begin{aligned}
\lim \frac{1}{N}\log\mathbb{E}C_{N,k_D,k_G} &= \Theta_{k_D,k_G}(u_D,u_G) \\
&= K - \min_B\left\{\Phi + k_G\kappa' I_1(x;\sqrt{2\kappa'}b) + k_D\kappa I_1\left(-(x+x_1);\sqrt{2\kappa(b^2+b_1^2)}\right)\right\}. \qquad (4.128)
\end{aligned}$$

Plots of Θ_{k_D,k_G} for a few values of k_D, k_G are shown in Figure 4.5.

Remark 4.2. Recall that the limiting spectral measure of the Hessian displays a transition as the support splits from one component to two, as shown in Figure 4.1. Let us comment on the relevance of this feature to the complexity. The spectral measure appears in one place in the above complexity calculations: the Coulomb gas integral $\int d\mu_{eq}(z)\log|z-x|$. The effect of integrating against the measure μ_{eq} is to smooth out the transition point. In other words, if μ_{eq} has two components or is at the transition point, one expects to be able to construct another measure ν supported on a single component such that $\int d\nu(z)\log|z-x| = \int d\mu_{eq}(z)\log|z-x|$. We interpret this to mean that the Coulomb gas integral term does not display any features that can be unambiguously attributed to the transition behaviour of the spectral measure.

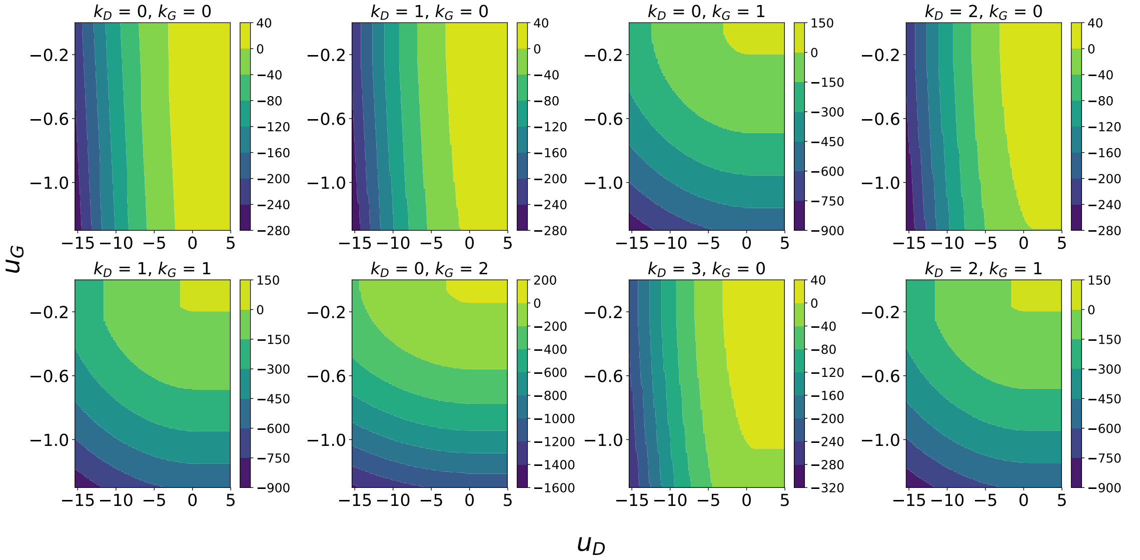


Figure 4.5: Contour plots of Θ_{k_D,k_G} for a few values of k_D, k_G. Here $p = q = 3, \sigma_z = 1, \kappa = 0.9$.

4.5 Implications

4.5.1 Structure of low-index critical points

We examine the fine structure of the low-index critical points for both spin glasses. [Cho+15] used the 'banded structure' of low-index critical points to explain the effectiveness of gradient descent in large multi-layer perceptron neural networks. We undertake to uncover the analogous structure in our dual spin-glass model and thence offer explanations for GAN training dynamics with gradient descent. For a range of (k_D, k_G) values, starting at $(0,0)$, we compute Θ_{k_D,k_G} on an appropriate domain. In the (u_D, u_G) plane, we then find the maximum k_D, and separately k_G, such that $\Theta_{k_D,k_G}(u_D, u_G) > 0$. In the large N limit, this procedure reveals the regions in the (u_D, u_G) plane where critical points of each index of the two spin glasses are found. Figure 4.6 plots these maximum k_D, k_G values as contours on a shared (u_D, u_G) plane. The grey region in the plot clearly shows the 'ground state' boundary beyond which no critical points exist. We use some fixed values of the various parameters: $p = q = 3, \sigma_z = 1, \kappa = 0.9$.

These plots reveal, unsurprisingly perhaps, that something resembling the banded structure of [Cho+15] is present, with the higher index critical points being limited to higher loss values for each network. The 2-dimensional analogues of the E_∞ boundary of [Cho+15] are evident in the bunching of the k_D, k_G contours at higher values. There is, however further structure not present in the single spin-glass multi-layer perceptron model. Consider the contour of $k_D = 0$ at the bottom of the full contour plot in Figure 4.6. Imagine traversing a path near this contour from right to left (decreasing u_D values); an example path is approximately indicated by a black arrow on the figure. At all points along such a path, the only critical points present are exact local minima for both networks, however the losses range over

(i) low generator loss, high discriminator loss;

(ii) some balance between generator and discriminator loss;

(iii) high generator loss, low discriminator loss.

These three states correspond qualitatively to known GAN phenomena:

(i) discriminator collapses to predicting 'real' for all items;

(ii) successfully trained model;

(iii) generator collapses to producing garbage samples which the discriminator trivially identifies.

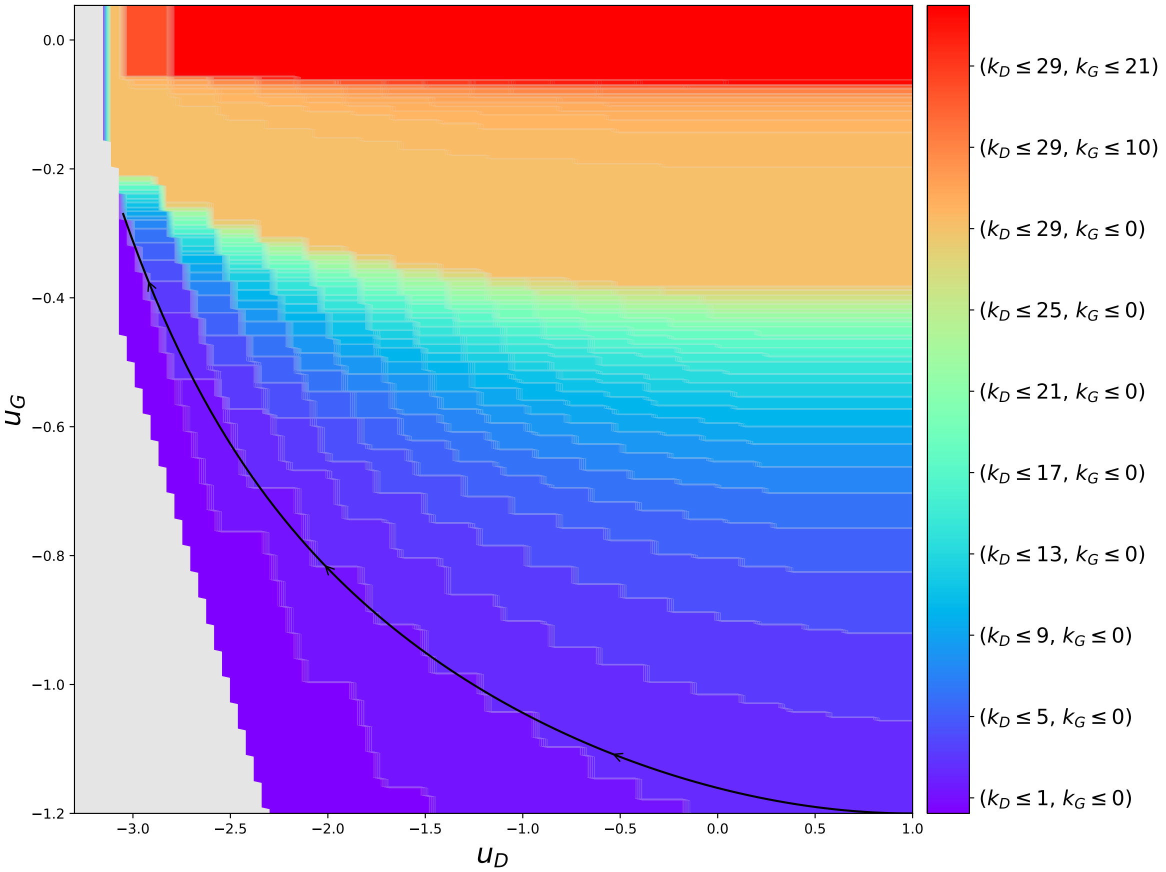


Figure 4.6: Contours in the (u_D, u_G) plane of the maximum k_D and k_G such that $\Theta_{k_D,k_G}(u_D,u_G) > 0$. k_D results shown with a red colour red scheme, and k_G with blue/green. The grey region on the left lies outside the domain of definition of Θ_{k_D,k_G}. Here $p = q = 3, \sigma_z = 1, \kappa = 0.9$. The arrow indicates the approximate location of the contour discussed in the main text.

Overall, the analysis of our model reveals a loss surface that favours convergence to states of low loss for *at least one of the networks*, but not necessarily both. Moreover, our plots of Θ and Θ_{k_D,k_G} in Figures 4.3, 4.5 demonstrate clearly the competition between the two networks, with the minimum attainable discriminator loss increasing as the generator loss decreases and vice-versa. We thus have a qualitative similarity between the minimax dynamics of real GANs and our model, but also a new two-dimensional banded critical points structure. We can further illuminate the structure by plotting, for each (u_D, u_G), the approximate proportion of minima with both $L_D \leqslant u_D$ and $L_G \leqslant u_G$ out of all points where at at least one of those conditions holds. The expression is

$$\Theta(u_D, u_G) - \max\{\Theta(u_D, \infty), \Theta(\infty, u_G)\} \tag{4.129}$$

which gives the log of the ratio in units of N. We show the plot in Figure 4.7. Note that, for large N, any region of the plot away from a value of zero contains exponentially more bad minima – where one of the networks has collapsed – than good minima, with equilibrium between the networks. The model therefore predicts the existence of good local minima (in the bottom left of Figure 4.7) that are effectively inaccessible due to their being exponentially outnumbered by bad local minima.

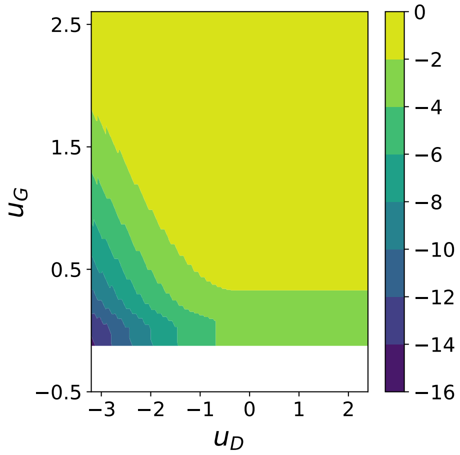


Figure 4.7: Contour plot of the log ratio quantity given in (4.129). This is the approximate proportion of minima with both $L_D \leqslant u_D$ and $L_G \leqslant u_G$ out of all points where at at least one of those conditions holds.

The structure revealed by our analysis offers the following explanation of large GAN training dynamics with gradient descent:

1. As with single feed-forward networks, the loss surface geometry encourages convergence to globally low values of at least one of the network losses.
2. The same favourable geometry encourages convergence to successful states, where both networks achieve reasonably low loss, but also encourages convergence to failure states, where the generator's samples are too easily distinguished by the discriminator, or the discriminator has entirely failed thus providing no useful training signal to the generator.

Remark 4.3. A natural question in the context of our analysis of low-index critical points is: do such points reflect the points typically reached by gradient descent algorithms used to train real GANs? There has been much discussion in the literature of the analogous question for single networks and spin glasses [Cho+15; Bai+19; FFR19]. It is not clear how to settle this question in our case, but we believe our model and its low-index critical points give a description of the baseline properties to be expected of high-dimensional adversarial optimisation problems late in the optimisation procedure. In addition, the unstructured random noise present in spin glasses may be more appropriate in our model for GANs than it is for single spin-glass models of single networks, as GAN generators do genuinely contain unstructured latent noise, rather than just the highly-structured data distributions seen on real data.

Remark 4.4. The issue of meta-stability is also worth mentioning. In single spin glasses, the boundary E_∞ between fixed index and unbounded index critical points is meta-stable [CS95; KPV93]. From the random matrix theory perspective, the E_∞ boundary corresponds to the left edge of the Wigner semi-circle [AAC13]. There are $O(N)$ eigenvalues in any finite interval at the left of the Wigner semi-circle, corresponding to $O(N)$ Hessian eigenvalues in any neighbourhood around zero. The 2D analogue of the E_∞ boundary in our double spin-glass model is expected to possess the same meta-stability: the Wigner semi-circle is replaced by the measure studied in Section 4.3, to which the preceding arguments apply. In the context of deep neural networks, there is a related discussion concerning “wide and flat local optima” of the loss surface, i.e. local optima for which many of the Hessian eigenvalues are close to zero. There are strong indications that deep neural networks converge under gradient-based optimisation to such optima [HS97a; Cha+19; Kes+17; KLY18; Bal+21; BPZ20] and that they are perhaps better for generalisation (i.e. test set loss) than other local optima, however some authors have challenged this view [Din+17a; HHS17; KKB20; HHY19; Gra20b]. It is beyond the scope of the present work to analyse the role of meta-stability further, however we note that the indications from machine learning are that it is most significant when considering generalisation, however our work simplifies to the case of a single loss rather than separately considering training and test loss.

4.5.2 Hyperparameter effects

Our proposed model for GANs includes a few fixed hyperparameters that we expect to control features of the model, namely σ_z and κ. Based on the results of [AAC13; Cho+15] and Chapter 3, and the form of our analytical results above, we do not expect p and q (the number of layers in the discriminator and generator) to have any interesting effect beyond $p, q \geqslant 3$; this is clearly a limitation of the model. We would expect there to exist an optimal value of σ_z that would result in minimum loss, in some sense. The effect of κ is less clear, though we guess that, in the studied $N \to \infty$ limit, all $\kappa \in (0,1)$ are effectively equivalent. Intuitively, choosing $\kappa = 0, 1$ corresponds to one network having a negligible number of parameters when compared with the other and we would expect the much larger network to prevail in the minimax game, however our theoretical results above are valid strictly for $\kappa \in (0,1)$.

In the following two subsections we examine effect of σ_z and κ in our theoretical and in real experiments with a DCGAN [RMC15]. Additional supporting plots are given in the appendix.

4.5.2.1 Effect of variance ratio

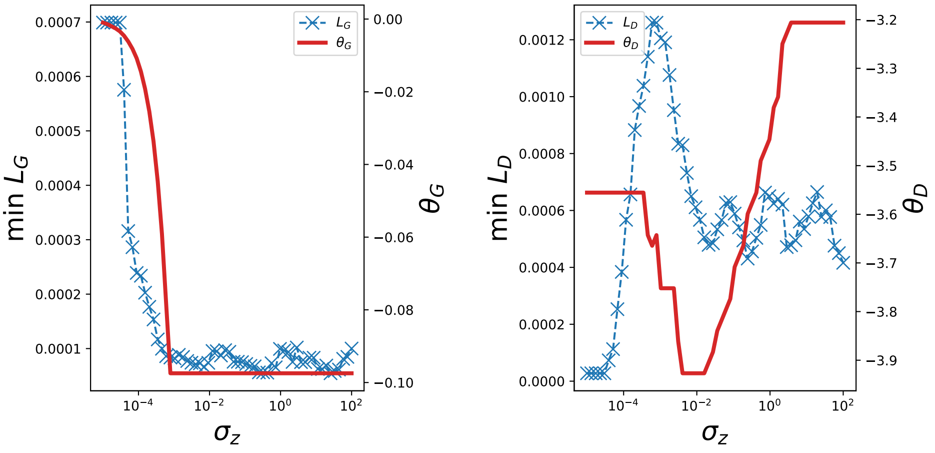
In the definition of complexity, u_D and u_G are upper bounds on the loss of the discriminator and generator, respectively. We are interested in the region of the u_D, u_G plane such that $\Theta(u_D, u_G) > 0$, this being the region where gradient descent algorithms are expected to become trapped. We therefore investigate the minimum loss such that $\Theta > 0$, this being, for a given σ_z, the theoretical minimum loss attainable by the GAN. We consider two natural notions of loss:

1. $\vartheta_D = \min\{u_D \in \mathbb{R} \mid \exists u_G \in \mathbb{R} \ : \ \Theta(u_D, u_G) > 0\}$;
2. $\vartheta_G = \min\{u_G \in \mathbb{R} \mid \exists u_D \in \mathbb{R} \ : \ \Theta(u_D, u_G) > 0\}$.

We vary σ_z over a range of values in $(10^{-5}, 10^{2})$ and compute ϑ_D, ϑ_G.

To compare the theoretical predictions of the effect of σ_z to real GANs, we perform a simple set of experiments. We use a DCGAN architecture [RMC15] with 5 layers in each network, using the reference PyTorch implementation from [18], however we introduce the generator noise scale σ_z. That is, the latent input noise vector $\boldsymbol{z}$ for the generator is sampled from $\mathcal{N}(0, \sigma_z^2 I)$. For a given σ_z, we train the GANs for 10 epochs on CIFAR10 [KH+09] and record the generator and discriminator losses. For each σ_z, we repeat the experiment 30 times and average the minimum attained generator and discriminator losses to account for random variations between runs with the same σ_z. We note that the sample variances of the loss were typically very high, despite the PyTorch random seed being fixed across all runs. We plot the sample means, smoothed with rolling averaging over a short window, in the interest of clearly visualising whatever trends are present. The results are shown in Figure 4.8.

There is a striking similarity between the generator plots, with a sharp decline between $\sigma_z = 10^{-5}$ and around 10^{-3}, after which the minimum loss is approximately constant. The picture for the discriminator is less clear. Focusing on the sections $\sigma_z > 10^{-3}$, both plots show a clear minimum, at around $\sigma_z = 10^{-1}$ in experiments and $\sigma_z = 10^{-2}$ in theory. Note that the scales on the y-axes of these plots should not be considered meaningful. Though there is not precise correspondence between the discriminator curves, we claim that both theory and experiment tell the same qualitative story: increasing σ_z to at least around 10^{-3} gives the lowest theoretical generator loss, and then further increasing to, tentatively, some value in $(10^{-2}, 10^{-1})$ gives the lowest possible discriminator loss at no detriment to the generator.



(a) Generator

(b) Discriminator

Figure 4.8: The effect of σ_z. Comparison of theoretical predictions of minimum possible discriminator and generator losses to observed minimum losses when training DCGAN on CIFAR10. The blue cross-dashed lines show the experimental DCGAN results, and solid red lines show the theoretical results θ_G, θ_D. $p = q = 5$ and $\kappa = 0.5$ are used in the theoretical calculations, to best match the DCGAN architecture. σ_z is shown on a log-scale.

We are not aware of σ_z tuning being widely used in practice for real GANs, rather it is typically taken to be unity. We have chosen this parameter, as it can be directly paralleled in our spin glass model, therefore allowing for the above experimental comparison. Naturally there are other parameters of real GANs that one might wish to study (such as learning rates and batch sizes) however these are much less readily mirrored in the spin glass model and complexity analysis, precluding comparisons between theory and experiment. Nevertheless, the experimental results in Figure 4.8 do demonstrate that tuning σ_z in real GANs could be of benefit, as $\sigma_z = 1$ does not appear to be the optimal value.

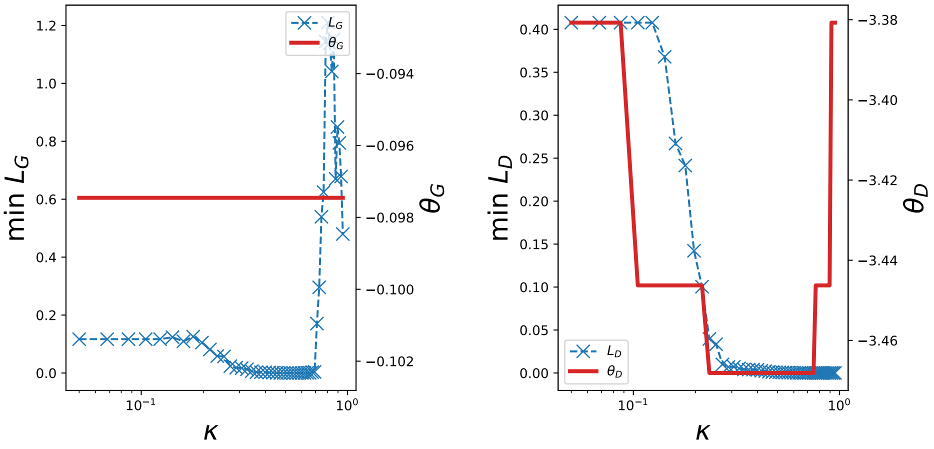
4.5.2.2 Effect of size ratio

Similarly to the previous section, we can investigate the effect of κ using ϑ_D, ϑ_G while varying κ over $(0, 1)$. To achieve this variation in the DCGAN, we vary the number of convolutional filters in each network. The generator and discriminator are essentially mirror images of each other and the number of filters in each intermediate layer are defined as increasing functions[2] of some positive integers n_G, n_D. We fix $n_D + n_G = 128$ and vary n_D to obtain a range of κ values, with $\kappa = \frac{n_d}{n_d + n_g}$. The results are shown in Figure 4.9.

The theoretical model predicts a a broad range of equivalently optimal κ values centred on $\kappa = 0.5$ from the perspective of the discriminator loss, and no effect of κ on the generator loss. The experimental results similarly show a broad range of equivalently optimal κ centred around $\kappa = 0.5$,

[2] Number of filters in a layer is either proportional to n_D or n_D^2 depending on the layer (and similarly with n_G).

however there appear to be deficiencies in our model, particularly for higher κ values. The results of the experiments are intuitively sensible: the generator loss deteriorates for κ closer to 1, i.e. when the discriminator has very many more parameters than the generator, and vice-versa for small κ.



(a) Generator

(b) Discriminator

Figure 4.9: The effect of κ. Comparison of theoretical predictions of minimum possible discriminator and generator losses to observed minimum losses when training DCGAN on CIFAR10. The blue cross-dashed lines show the experimental DCGAN results, and the solid red show the theoretical results ϑ_G, ϑ_D. $p = q = 5$ and $\sigma_z = 1$ are used in the theoretical calculations, to best match the DCGAN architecture.

4.6 Gaussian Hessian calculations

In this section we give the full details of the Gaussian calculations for the distribution of the Hessian:

$$\begin{pmatrix} \nabla^2_D L^{(D)} & \nabla_{GD} L^{(D)} \\ \nabla_{DG} L^{(G)} & \nabla^2_G L^{(G)} \end{pmatrix} \Bigg| \nabla_G L^{(G)} = 0, \nabla_D L^{(D)} = 0, L^{(D)} \in B_D, L^{(G)} \in B_G. \tag{4.130}$$

These calculations are routine and consist of repeated application of standard results for conditioning multivariate Gaussians, but the details are nevertheless intricate.

Recall the definitions

$$L^{(D)}(\boldsymbol{w}^{(D)}, \boldsymbol{w}^{(G)}) = \ell^{(D)}(\boldsymbol{w}^{(D)}) - \sigma_z \ell^{(G)}(\boldsymbol{w}^{(D)}, \boldsymbol{w}^{(G)})$$
$$L^{(G)}(\boldsymbol{w}^{(D)}, \boldsymbol{w}^{(G)}) = \sigma_z \ell^{(G)}(\boldsymbol{w}^{(D)}, \boldsymbol{w}^{(G)})$$

and

$$\ell^{(D)}(\boldsymbol{w}^{(D)}) = \sum_{i_1,\dots,i_p=1}^{N_D} X_{i_1,\dots,i_p} \prod_{k=1}^{p} w^{(D)}_{i_k}$$
$$\ell^{(G)}(\boldsymbol{w}^{(D)}, \boldsymbol{w}^{(G)}) = \sum_{i_1,\dots,i_{p+q}=1}^{N_D+N_G} Z_{i_1,\dots,i_{p+q}} \prod_{k=1}^{p+q} w_{i_k}$$

for i.i.d. Gaussian X and Z, where $\boldsymbol{w}^T = (\boldsymbol{w}^{(D)^T}, \boldsymbol{w}^{(G)^T})$. As mentioned in the main text, we have spherical symmetry in both $\boldsymbol{w}^{(D)}$ and $\boldsymbol{w}^{(G)}$, so it sufficient to consider the distribution (4.130) around some fixed specific points on the spheres S^{N_D} and S^{N_G}. Following [AAC13], we choose the north poles. We can select a coordinate basis around both poles, e.g. with

$$\boldsymbol{w}^{(D)} = (\sqrt{1-\boldsymbol{u}^2}, \boldsymbol{u}), \quad \boldsymbol{w}^{(G)} = (\sqrt{1-\boldsymbol{v}^2}, \boldsymbol{v}),$$

for $\boldsymbol{u} \in \mathbb{R}^{N_D-1}, \boldsymbol{v} \in \mathbb{R}^{N_G-1}$ with $\boldsymbol{u}^2 \leqslant 1, \boldsymbol{v}^2 \leqslant 1$.

We need the joint distributions

$$\left(\ell^{(D)}, \partial_i^{(D)}\ell^{(D)}, \partial_{jk}^{(D)}\ell^{(D)}\right), \quad \left(\ell^{(G)}, \partial_i^{(G)}\ell^{(G)}, \partial_{jk}^{(G)}\ell^{(G)}, \partial_l^{(D)}\ell^{(G)}, \partial_{mn}^{(D)}\ell^{(G)}\right)$$

where the two groups are independent from of each other. *The derivatives $\partial^{(D)}, \partial^{(G)}$ are now Euclidean derivatives with respect to the coordinates $\boldsymbol{u}, \boldsymbol{v}$.* $\ell^{(D)}$ behaves just like a single spin glass, and so we have [AAC13]:

$$Var(\ell^{(D)}) = 1, \tag{4.131}$$
$$Cov(\partial_i^{(D)}\ell^{(D)}, \partial_{jk}^{(D)}\ell^{(D)}) = 0, \tag{4.132}$$
$$\partial_{ij}^{(D)}\ell^{(D)} \mid \{\ell^{(D)} = x_D\} \sim \sqrt{(N_D-1)p(p-1)}GOE^{N_D-1} - x_D p I. \tag{4.133}$$

To find the joint and thence conditional distributions for $\ell^{(G)}$, we first note that $\ell^{(G)}$ is simply a spin glass on a partitioned vector $\boldsymbol{w}^T = (\boldsymbol{w}^{(D)^T}, \boldsymbol{w}^{(G)^T})$, so

$$Cov(\ell^{(G)}(\boldsymbol{w}^{(D)}, \boldsymbol{w}^{(G)}), \ell^{(G)}(\boldsymbol{w}^{(D)'}, \boldsymbol{w}^{(G)'})) = \left(\boldsymbol{w}^{(D)} \cdot \boldsymbol{w}^{(D)'} + \boldsymbol{w}^{(G)} \cdot \boldsymbol{w}^{(G)'}\right)^{p+q} \tag{4.134}$$

from which, by comparing with [AAC13], one can obtain the necessary expressions, at the north poles in a coordinate basis. Practically, one writes $\boldsymbol{w}^{(D)^T} = (\sqrt{1-\sum_j u_j^2}, u_1, \ldots, u_{N_D-1})$, and similarly for $\boldsymbol{w}^{(G)}$. Then one takes derivatives of (4.134) with respect to these new variables around the north poles. Finally, one sets $\boldsymbol{w}^{(D)} = \boldsymbol{w}^{(D)'}$ and takes $u_j = 0 \ \forall j$, and similarly for $\boldsymbol{w}^{(G)}$. The resulting expressions are largely familiar from the standard spin glass in [AAC13], except there are extra cross terms between $\boldsymbol{w}^{(D)}$ and $\boldsymbol{w}^{(G)}$:

$$Var(\ell^{(G)}) = 2^{p+q}, \tag{4.135}$$

$$Cov(\partial_{ij}^{(G)}\ell^{(G)}, \ell^{(G)}) = -(p+q)2^{p+q}\delta_{ij}, \tag{4.136}$$

$$Cov(\partial_{ij}^{(D)}\ell^{(G)}, \ell^{(G)}) = -(p+q)2^{p+q}\delta_{ij}, \tag{4.137}$$

$$Cov(\partial_{ij}^{(G)}\ell^{(G)}, \partial_{kl}^{(G)}\ell^{(G)}) = 2^{p+q}\left[(p+q)(p+q-1)\left(\delta_{ik}\delta_{jl}+\delta_{il}\delta_{jk}\right)+(p+q)^2\delta_{ij}\delta_{kl}\right], \tag{4.138}$$

$$Cov(\partial_{ij}^{(G)}\ell^{(G)}, \partial_{kl}^{(D)}\ell^{(G)}) = 2^{p+q}(p+q)^2\delta_{ij}\delta_{kl}, \tag{4.139}$$

$$Cov(\partial_{i}^{(G)}\partial_{j}^{(D)}\ell^{(G)}, \partial_{k}^{(G)}\partial_{l}^{(D)}\ell^{(G)}) = 2^{p+q}(p+q)(p+q-1)\delta_{ik}\delta_{jl}, \tag{4.140}$$

$$Cov(\partial_{ij}^{(G)}\ell^{(G)}, \partial_{k}^{(G)}\partial_{l}^{(D)}\ell^{(G)}) = 0 \tag{4.141}$$

$$Cov(\partial_{ij}^{(D)}\ell^{(G)}, \partial_{k}^{(D)}\partial_{l}^{(G)}\ell^{(G)}) = 0, \tag{4.142}$$

$$Cov(\partial_{i}^{(D)}\partial_{j}^{(G)}\ell^{(G)}, \ell^{(G)}) = 0. \tag{4.143}$$

Also, all first derivatives of $\ell^{(G)}$ are clearly independent of $\ell^{(G)}$ and its second derivatives by the same reasoning as in [AAC13]. Note that

$$Cov(\partial_{i}^{(D)}L^{(D)}, \partial_{j}^{(D)}L^{(D)}) = (p+\sigma_z^2 2^{p+q}(p+q))\delta_{ij} \tag{4.144}$$

$$Cov(\partial_{i}^{(G)}L^{(G)}, \partial_{j}^{(G)}L^{(G)}) = \sigma_z^2 2^{p+q}(p+q)\delta_{ij} \tag{4.145}$$

$$Cov(\partial_{i}^{(D)}L^{(D)}, \partial_{j}^{(G)}L^{(G)}) = 0 \tag{4.146}$$

and so

$$\varphi_{(\nabla_D L^{(D)}, \nabla_G L^{(G)})}(0) = (2\pi)^{-\frac{N-2}{2}}\left(p+\sigma_z^2 2^{p+1}(p+q)\right)^{-\frac{N_D-1}{2}}\left(\sigma_z^2 2^{p+q}(p+q)\right)^{-\frac{N_G-1}{2}}. \tag{4.147}$$

We need now to calculate the joint distribution of $(\partial_{ij}^{(D)}\ell^{(G)}, \partial_{kl}^{(G)}\ell^{(G)})$ conditional on $\{\ell^{(G)} = x_G\}$. Denote the covariance matrix for $(\partial_{ij}^{(D)}\ell^{(G)}, \partial_{kl}^{(G)}\ell^{(G)}, \ell^{(G)})$ by

$$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} \tag{4.148}$$

where

$$\Sigma_{11} = 2^{p+q}\begin{pmatrix} (p+1)(p+q-1)(1+\delta_{ij})+(p+q)^2\delta_{ij} & (p+q)^2\delta_{ij}\delta_{kl} \\ (p+q)^2\delta_{ij}\delta_{kl} & (p+1)(p+q-1)(1+\delta_{kl})+(p+q)^2\delta_{kl} \end{pmatrix}, \tag{4.149}$$

$$\Sigma_{12} = -2^{p+q}(p+q)\begin{pmatrix} \delta_{ij} \\ \delta_{kl} \end{pmatrix}, \tag{4.150}$$

$$\Sigma_{21} = -2^{p+q}(p+q)\begin{pmatrix} \delta_{ij} & \delta_{kl} \end{pmatrix}, \tag{4.151}$$

$$\Sigma_{22} = 2^{p+q}. \tag{4.152}$$

The conditional covariance is then

$$\bar{\Sigma} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} = 2^{p+q}(p+1)(p+q-1)\begin{pmatrix} 1+\delta_{ij} & 0 \\ 0 & 1+\delta_{kl} \end{pmatrix}. \tag{4.153}$$

Identical reasoning applied to $(\partial_{ij}^{(G)}\ell^{(G)}, \partial_{kl}^{(G)}\ell^{(G)}, \ell^{(G)})$ and $(\partial_{ij}^{(D)}\ell^{(G)}, \partial_{kl}^{(D)}\ell^{(G)}, \ell^{(G)})$ shows that, conditional on $\{\ell^{(G)} = x_G\}$, $\nabla_G^2\ell^{(G)}$ and $\nabla_D^2\ell^{(G)}$ have independent entries up-to symmetry, so 4.153 demonstrates they are independent GOEs and we have:

$$\begin{aligned} \begin{pmatrix} -\nabla_D^2\ell^{(G)} & -\nabla_G\nabla_D\ell^{(G)} \\ \nabla_D\nabla_G\ell^{(G)} & \nabla^2\ell^{(G)} \end{pmatrix} \mid \{\ell^{(G)} = x_G\} \overset{d}{=} \sqrt{2^{p+q+1}(p+q)(p+q-1)}\begin{pmatrix} \sqrt{N_D-1}M_1^{(D)} & -2^{-1/2}G \\ 2^{-1/2}G^T & \sqrt{N_G-1}M^{(G)} \end{pmatrix} \\ -(p+q)x_G 2^{p+1}\begin{pmatrix} -I_{N_D} & 0 \\ 0 & I_{N_G} \end{pmatrix} \end{aligned} \tag{4.154}$$

where $M_1^{(D)} \sim GOE^{N_D-1}$ and $M^{(G)} \sim GOE^{N_G-1}$ are independent GOEs and G is an independent $N_D - 1 \times N_G - 1$ Ginibre matrix with entries of unit variance.

4.7 Conclusion

We have contributed a novel model for the study of large neural network gradient descent dynamics with statistical physics techniques, namely an interacting spin-glass model for generative adversarial neural networks. We believe this is the first attempt in the literature to incorporate advanced architectural features of modern neural networks, beyond basic single network multi-layer perceptrons, into such statistical physics style models. We have conducted an asymptotic complexity analysis via Kac-Rice formulae and Random Matrix Theory calculations of the energy surface of this model, acting as a proxy for GAN training loss surfaces of large networks. Our analysis has revealed a banded critical point structure as seen previously for simpler models, explaining the surprising success of gradient descent in such complicated loss surfaces, but with added structural features that offer explanations for the greater difficulty of training GANs compared to single networks. We have used our model to study the effect of some elementary GAN hyper-parameters and compared with experiments training real GANs on a standard computer vision dataset. We believe that the interesting features of our model, and their correspondence with real GANs, are yet further compelling evidence for the role of statistical physics effects in deep learning and the value of studying such models as proxies for real deep learning models, and in particular the value of concocting more sophisticated models that reflect aspects of modern neural network design and practice.

Our analysis has focused on the annealed complexity of our spin glass model (i.e. taking the logarithm after the expectation) rather than the quenched complexity (i.e. taking the expectation after the logarithm). Ideally one would compute both, as the quenched complexity is often considered to reflect the typical number of stationary points and is bounded above by the annealed complexity. Computing the quenched complexity is typically more challenging than the annealed and such a calculation for our model could be the subject of a further work requiring considerable technical innovations. Even the elegant and very general methods presented recently in [ABM21a] are restricted only to the annealed case. Agreement between annealed and quenched is known only in a few special cases closely related to spherical spin glasses [Sub17; AG20; ASZ20] and is not expected in general [Ros+19]. It is conceivable that quenched and annealed complexity agree in the case of our model, as it closely related to spin glasses and possesses no distinguished directions (i.e. spikes) such as are present in [Ros+19]. Establishing agreement by existing methods requires analysis of pairs of correlated GOE-like matrices. Such an approach for our model may well require analysis of at least 4 correlated matrices (2 per diagonal block), and quite possibly more, including correlations between blocks. We leave this considerable challenge for future work.

From a mathematical perspective, we have extensively studied the limiting spectral density of a novel random matrix ensemble using supersymmetric methods. During the initial explorations for this work, we made considerable efforts to complete the average absolute value determinant calculations directly using a supersymmetric representation, as seen in Chapter 3, however this was found to be analytically intractable (as expected), but also extremely troublesome numerically (essentially due to analytically intractable and highly complicated Riemann sheet structure in $\mathbb{C}^2$). We were able to sidestep these issues by instead using a Coulomb gas approximation, whose validity we have rigorously proved using a novel combination of concentration arguments and supersymmetric asymptotic expansions. We have verified with numerical simulations our derived mean spectral density for the relevant Random Matrix Theory ensemble and also the accuracy of the Coulomb gas approximation.

We hope that future work will be inspired to further study models of neural networks such as we have considered here. Practically, it would be exciting to explore the possibility of using our insights into GAN loss surfaces to devise algorithmic methods of avoiding training failure. Mathematically, the local spectral statistics of our random matrix ensemble may be interesting to study, particularly around the cusp where the two disjoint components of the limiting spectral density merge.

Chapter 5

Generalised loss surface models and implications

The content of this chapter was published first as a pre-print in July 2021 (`https://arxiv.org/abs/2003.01247v5`) and was accepted in January 2023 as an article in *Journal of Machine Learning Research*: "Iterate Averaging in the quest for best test error", Diego Granziol, **Nicholas P. Baskerville**, Xingchen Wan, Samuel Albanie and Stephen Roberts.

The experimental ideas behind this paper were conceived and explored by the other authors before **NPB** joined the project. **NPB** developed much of the mathematical theory, including constructing all the proofs. In this chapter, we include only the mathematical sections of direct relevance to this thesis, all of which are overwhelmingly **NPB**'s work.

5.1 Introduction

The iterate average [PJ92] is the arithmetic mean of the model parameters over the optimisation trajectory $\boldsymbol{w}_{\text{avg}} = \frac{1}{n}\sum_i^n \boldsymbol{w}_i$. It is a classical variance reducing technique in optimisation and offers optimal asymptotic convergence rates and greater robustness to the choice of learning rate [KY03]. Indeed, popular regret bounds that form the basis of gradient-based convergence proofs [DHS11; RKK19] often consider convergence for the iterate average [Duc18]. Further, theoretical extensions have shown that the rate of convergence can be improved by a factor of $\log T$ (where T is the iteration number) by *suffix averaging* [RSS11], which considers a fraction of the last iterates, *polynomial decay averaging* [SZ13] which decays the influence of the previous iterates, or *weighted averaging* [LSB12] which weights the iterate by its iteration number. That the final iterate of SGD is sub-optimal in terms of its convergence rate, by this logarithmic factor, has been proved by [Har+19]. For networks with batch normalisation [IS15], a naïve application of IA (in which we simply average the batch normalisation statistics) is known to lead to poor results [DB19]. However, by computing the batch normalisation statistics for the iterate average using a forward pass of the data at the IA point, [Izm+18] show that the performance of small-scale image experiments such as CIFAR-10/100 and pretrained ImageNet can be significantly improved. Even for small experiments this computation is expensive, so they further approximate IA by taking the average at the end of each epoch instead of each iteration, referred to as *stochastic weight averaging* (SWA).

In this chapter we examine the variance reducing effect of IA in the context of a quadratic approximation to the true loss combined with additive perturbation models for the batch training loss. The theory we present is high-dimensional (i.e. large number of parameters, P) and considers the small batch size (small B) regime, which we term the "deep learning limit". Intuitively, any given example from the training set $j \in \mathcal{D}$, will contain *general features*, which hold over the data generating distribution and *instance specific features* (which are relevant only to the training sample in question). For example, for a training image of a dog, we may have that:

$$\overbrace{\underbrace{\nabla L_{\text{sample}}(\boldsymbol{w})}_{\text{training set example}}}^{\text{dog } j} = \overbrace{\underbrace{\nabla L_{\text{true}}(\boldsymbol{w})}_{\text{general features}}}^{\text{4 legs, snout}} + \overbrace{\underbrace{\varepsilon(\boldsymbol{w})}_{\text{instance-specific features}}.}^{\text{black pixel in top corner, green grass}} \tag{5.1}$$

Under a quadratic approximation to the *true loss*[1] $L_{\text{true}}(\boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{H} \boldsymbol{w}$, where $\boldsymbol{H} = \nabla^2 L$ is the Hessian of the true loss with respect to the weights and we sample a mini-batch gradient of size B at point $\boldsymbol{w} \in \mathbb{R}^{P \times 1}$. The observed gradient is perturbed by $\varepsilon(\boldsymbol{w})$ from the true loss gradient (due to instance specific features). Under this model the component of the $\boldsymbol{w}_t$'th iterate along the j'th eigenvector ϕ_j of the true loss when running SGD with learning rate α can be written:

$$\boldsymbol{w}_t^T \phi_j = (1 - \alpha\lambda_j)^t \boldsymbol{w}_0^T \phi_j - \alpha(1 - \alpha\lambda_j)^{t-1} \varepsilon(\boldsymbol{w}_1)^T \phi_j \cdots, \tag{5.2}$$

in which λ_j are the eigenvalues of $\boldsymbol{H}$. The simplest tractable model for the gradient noise $\varepsilon(\boldsymbol{w}_t)$ is to assume samples from i.i.d. an isotropic, multivariate Normal. In particular, this assumption removes any dependence on $\boldsymbol{w}_t$ and precludes the existence of any distinguished directions in the gradient noise. Using this assumption, we obtain Theorem 5.1 below, which relies on an intermediate result, found in [Ver18].

Lemma 5.1 ([Ver18] Theorem 6.3.2)**.** *Let R be an $m \times n$ matrix, and let $X = (X_1, \ldots, X_n) \in \mathbb{R}^n$ be a random vector with independent mean-zero unit-variance sub-Gaussian coordinates. Then*

$$\mathbb{P}\left(|\|RX\|_2 - \|R\|_F| > t\right) \leqslant 2\exp\left(-\frac{ct^2}{K^4\|R\|^2}\right)$$

where $K = \max_i \|X_i\|_{\psi_2}$ and $c > 0$ is a constant.

Theorem 5.1. *Assume the quadratic loss model $L_{true}(\boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{H} \boldsymbol{w}$, where $\boldsymbol{H}$ has eigenvalues $\{\lambda_i\}_{i=1}^P$ and assume the $\{\varepsilon_t\}_{t=0}^n$ are all i.i.d. Gaussian vectors in $\mathbb{R}^P$ with distribution $\mathcal{N}(0, \sigma^2 B^{-1} I)$ where B is the batch*

[1] The loss under the expectation of the data generating distribution, rather than the loss over the dataset $L_{\text{emp}}(\boldsymbol{w}_k)$.

size. Assume the weights are updated according to the rule from (5.2)

$$\boldsymbol{w}_t^T \phi_j = (1-\alpha\lambda_j)^t \boldsymbol{w}_0^T \phi_j - \alpha(1-\alpha\lambda_j)^{t-1}\varepsilon(\boldsymbol{w}_1)^T\phi_j. \tag{5.3}$$

Assume further that $\alpha\lambda_i \ll 1$ *for all* i *and* $\lambda_i > 0$ *for all* i*. Then there exists a constant* $c > 0$ *such that for all* $\xi > 0$*, as* $n \to \infty$

$$\begin{aligned} &\mathbb{P}\left(\left|\sqrt{\sum_i^P \left(w_{n,i} - w_{0,i}\mathrm{e}^{-n\alpha\lambda_i}(1+o(1))\right)^2} - \sqrt{P\frac{\alpha\sigma^2}{B}\left\langle\frac{1}{\lambda(2-\alpha\lambda)}\right\rangle}\right| \geqslant \xi\right) \leqslant \nu(\xi), \\ &\mathbb{P}\left(\left|\sqrt{\sum_i^P \left(w_{\text{avg},i} - \frac{w_{0,i}}{\lambda_i n\alpha}(1+o(1))\right)^2} - \sqrt{\frac{P\sigma^2}{Bn}\left\langle\frac{1}{\lambda}\right\rangle}\right| \geqslant \xi\right) \leqslant \nu(\xi), \end{aligned} \tag{5.4}$$

where $\nu(\xi) = 2\exp(-c\xi^2)$*.*

Proof. Let $Y = (Y_1, \ldots, Y_P)$ be a random sub-Gaussian vector with independent components. Let

$$X_i = \frac{Y_i - \mathbb{E}Y_i}{\sqrt{\operatorname{Var}Y_i}}, \quad R = \operatorname{diag}(\sqrt{\operatorname{Var}Y_1}, \ldots, \sqrt{\operatorname{Var}Y_P}).$$

Lemma 5.1 then applies, to give

$$\mathbb{P}\left(\left|\|Y - \mathbb{E}Y\|_2 - \sqrt{\sum_{i=1}^P \operatorname{Var}Y_i}\right| > \xi\right) \leqslant 2\exp\left(-\frac{c\xi^2}{K^4\|R\|^2}\right).$$

We have $K \leqslant C\max_i \operatorname{Var}Y_i$ for some constant $C > 0$ ([Ver18], exercise 2.5.8), and $\|R\|^2 = (\max_i \sqrt{\operatorname{Var}Y_i})^2 = \max_i \operatorname{Var}Y_i$. Hence we obtain

$$\mathbb{P}\left(\left|\|Y - \mathbb{E}Y\|_2 - \sqrt{\sum_{i=1}^P \operatorname{Var}Y_i}\right| > \xi\right) \leqslant 2\exp\left(-\frac{c\xi^2}{(\max_i \operatorname{Var}Y_i)^2}\right) \tag{5.5}$$

for some new constant $c > 0$. The proof is then completed if we compute the means and variances of $\boldsymbol{w}_n$ and $\boldsymbol{w}_{\text{avg}}$. To that end, with $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_P)$, the update rule (5.3) gives

$$\boldsymbol{w}_t = (1-\alpha\boldsymbol{\Lambda})^n \boldsymbol{w}_0 + \alpha\sum_{i=0}^{t-1}(1-\alpha\boldsymbol{\Lambda})^{t-i-1}\varepsilon_i, \tag{5.6}$$

for any $1 \leqslant t \geqslant n$. Since $\boldsymbol{\Lambda}$ is diagonal, each component of $\boldsymbol{w}_n$ can be treated independently when we sum to obtain $\boldsymbol{w}_{avg}$, so for any vector $\boldsymbol{v}$

$$\sum_{t=1}^n (1-\alpha\boldsymbol{\Lambda})^t \boldsymbol{v} = \frac{1-(1-\alpha\boldsymbol{\Lambda})^t}{\alpha}\boldsymbol{\Lambda}^{-1}(1-\alpha\boldsymbol{\Lambda})\boldsymbol{v} \tag{5.7}$$

So averaging (5.6) over t gives

$$\boldsymbol{w}_{avg} = \frac{1-(1-\alpha\boldsymbol{\Lambda})^n}{\alpha n}\boldsymbol{\Lambda}^{-1}(1-\alpha\boldsymbol{\Lambda})\boldsymbol{w}_0 + \sum_{t=0}^{n-1}\frac{1-(1-\alpha\boldsymbol{\Lambda})^{n-t}}{n}\boldsymbol{\Lambda}^{-1}\varepsilon_t. \tag{5.8}$$

Since the ε_i are all i.i.d. centred Gaussians, obtaining the distributions of $\boldsymbol{w}_n$ and $\boldsymbol{w}_{avg}$ amounts to computing the covariances

$$\begin{aligned}\text{Cov}\left(\alpha\sum_{i=0}^{n-1}(1-\alpha\boldsymbol{\Lambda})^{n-i-1}\varepsilon_i\right) =&\sigma^2 B^{-1} I\sum_{i=1}^{n-1}\alpha^2(1-\alpha\boldsymbol{\Lambda})^{2(n-i-1)}\\ =&\sigma^2 B^{-1} I\alpha^2(1-(1-\alpha\boldsymbol{\Lambda})^{2n})\left(1-(1-\alpha\boldsymbol{\Lambda})^2\right)^{-1}\end{aligned} \tag{5.9}$$

and similarly

$$\begin{aligned}&\text{Cov}\left(\sum_{t=0}^{n-1}\frac{1-(1-\alpha\boldsymbol{\Lambda})^{n-t}}{n}\boldsymbol{\Lambda}^{-1}\varepsilon_t\right)\\ =&\sum_{t=0}^{n-1}\left(\frac{1-(1-\alpha\boldsymbol{\Lambda})^{n-t}}{n}\boldsymbol{\Lambda}^{-1}\right)^2\\ =&\frac{\boldsymbol{\Lambda}^{-2}}{n^2}\left(n-\frac{2(1-(1-\alpha\boldsymbol{\Lambda})^n)}{\alpha}\boldsymbol{\Lambda}^{-1}+\left(1-(1-\alpha\boldsymbol{\Lambda})^{2n}\right)\left(1-(1-\alpha\boldsymbol{\Lambda})^2\right)^{-1}\right).\end{aligned} \tag{5.10}$$

Now using $\alpha\lambda_i<1$ for all $i=1,2\ldots,P$, and taking $n\to\infty$, (5.6) and (5.9) give

$$\text{Cov}(\boldsymbol{w}_n)\sim\sigma^2\alpha^2B^{-1}\left(1-(1-\alpha\boldsymbol{\Lambda})^2\right)^{-1}=\sigma^2\alpha B^{-1}\left(2\boldsymbol{\Lambda}-\alpha\boldsymbol{\Lambda}^2\right)^{-1} \tag{5.11}$$

and similarly (5.8) and (5.10) give

$$\text{Cov}(\boldsymbol{w}_{avg})\sim\frac{1}{n}\boldsymbol{\Lambda}^{-2}. \tag{5.12}$$

Thus it follows from (5.6) and (5.11) that

$$\mathbb{E}w_{n,i}=(1-\alpha\lambda_i)^n w_{0,i}\sim e^{-n\alpha\lambda_i}w_{0,i},\ \ \text{Var}(w_{n,i})\sim\frac{\sigma^2}{B}\frac{\alpha}{2\lambda_i(1-\alpha\lambda_i)} \tag{5.13}$$

and from (5.8) and (5.12) it follows

$$\mathbb{E}w_{avg,i}\sim\frac{w_{0,i}}{\lambda_i\alpha n},\quad \text{Var}(w_{avg,i})=\frac{\sigma^2}{B}\frac{1}{n\lambda_i^2} \tag{5.14}$$

where in both cases we have used $\alpha\lambda_i\ll 1$ to simplify the expected values for large n. To complete the proof for $\boldsymbol{w}_n$, we apply (5.5) using (5.13) and noting that

$$\sqrt{\sum_{i=1}^{P}Var(w_{n,i})}\sim\frac{\sigma^2P\alpha}{2B}\left\langle\frac{1}{\lambda(1-\alpha\lambda)}\right\rangle \tag{5.15}$$

and $0<\max_i w_{n,i}<\infty$ since $\lambda_i>0$ and $\alpha\lambda_i<1$. The results for $\boldsymbol{w}_{avg}$ follows similarly by using (5.5) with (5.14). This produces two different constants $c>0$ in the statement of (5.4), but we can simply take the smaller of the two constants to produce the desired statement.

■

The final iterate attains exponential convergence in the mean of $\boldsymbol{w}_n$, but does not control the variance term. Whereas for $\boldsymbol{w}_{\text{avg}}$, although the convergence in the mean is worse (linear), the variance vanishes asymptotically – this motivates *tail averaging*, to get the best of both worlds. Another key implication of Theorem 5.1 lies in its dependence on P. P is a gauge of the model size and appears as a simple linear multiplier of the variances of $\boldsymbol{w}_n$ and $\boldsymbol{w}_{avg}$, so increasing over-parametrisation implies increasing variance of the final iterate and the IA, however IA provides a counterbalancing variance reduction effect that is entirely absent from the final iterate. This implies that in more complex, over-parameterised models, we expect the benefit of IA over the final iterate to be greater, as IA provides a mechanism to control the weight variance even as it grows with P.

5.2 A dependent model for the perturbation

We proceed now to propose a relaxation of the gradient perturbation independence assumption. (5.1) can be written equivalently as

$$L_{\text{batch}}(\boldsymbol{w}) = L_{\text{true}}(\boldsymbol{w}) + \eta(\boldsymbol{w}) \tag{5.16}$$

where η is a scalar field with $\nabla\eta = \varepsilon$. Note that we have neglected an irrelevant arbitrary constant in Equation (5.16) and also that we have L_{batch} rather than L_{sample}, but this amounts to scaling the per-sample noise variance σ^2 by the inverse batch size B^{-1}. We model η as a Gaussian process $\mathcal{GP}(m,k)$, where k is some kernel function $\mathbb{R}^P \times \mathbb{R}^P \to \mathbb{R}$ and m is some mean function[2] $\mathbb{R}^P \to \mathbb{R}$. As an example, taking $k(\boldsymbol{w},\boldsymbol{w}') \propto (\boldsymbol{w}^T\boldsymbol{w}')^p$ and restricting $\boldsymbol{w}$ to a hypersphere results in ε taking the exact form of a spherical p-spin glass, studied previously for DNNs [Cho+15; GD88; MPV87; Ros+19; Man+19a] and in Chapters 3 and 4 [Bas+21; Bas+22a]. *We are not* proposing to model the loss surface (batch or true) as a spin glass (or more generally, a Gaussian process), rather we are modelling the perturbation between the loss surfaces in this way. We emphasise that this model is a strict generalisation of the i.i.d. assumption above, and presents a rich, but tractable, model of isotropic Gaussian gradient perturbations in which the noise for different iterates is neither independent nor identically distributed.

Following from our Gaussian process definition, the covariance of gradient perturbations can be computed using a well-known result (see [AT09] equation 5.5.4):

$$\mathrm{Cov}(\varepsilon_i(\boldsymbol{w}), \varepsilon_j(\boldsymbol{w}')) = \partial_{w_i}\partial_{w'_j} k(\boldsymbol{w},\boldsymbol{w}'). \tag{5.17}$$

Further assuming a stationary kernel $k(\boldsymbol{w},\boldsymbol{w}') = k\left(-\frac{1}{2}||\boldsymbol{w}-\boldsymbol{w}'||_2^2\right)$

$$\mathrm{Cov}(\varepsilon_i(\boldsymbol{w}), \varepsilon_j(\boldsymbol{w}')) = (w_i - w'_i)(w'_j - w_j)k''\left(-\frac{1}{2}||\boldsymbol{w}-\boldsymbol{w}'||_2^2\right) + \delta_{ij}k'\left(-\frac{1}{2}||\boldsymbol{w}-\boldsymbol{w}'||_2^2\right). \tag{5.18}$$

[2] It is natural to take $m = 0$ in a model for the sample perturbation, however retaining fully general m does not affect our arguments.

Thus we have a non-trivial covariance between gradient perturbation at different points in weight-space. This covariance structure can be used to prove the upcoming variance reduction result, but first we require some intermediate lemmas.

5.2.1 Intermediate results

In this section we establish some intermediate lemmas that will be required later in the chapter.

Lemma 5.2. *Define the function*

$$r(a;x) = \frac{\gamma(a;x)}{\Gamma(a)}, \tag{5.19}$$

where γ is the lower incomplete gamma function. Assume that $x \ll a$, where x may or may not diverge with a, then as $a \to \infty$, $r(a;x) \to 0$, and more precisely

$$r(a;x) \sim \frac{1}{\sqrt{2\pi}} \exp\left(-x + a\log x - a - a\log a - \frac{1}{2}\log a\right). \tag{5.20}$$

Proof. We have $\gamma(a;x) = a^{-1}x^a {}_1F_1(a;1+a;-x)$, where ${}_1F_1$ is the confluent hypergeometric function of the first kind [AAR99]. Then

$$r(a;x) = \frac{a^{-1}x^a {}_1F_1(a;1+a;-x)}{\Gamma(a)} = \frac{a^{-1}x^a\Gamma(a+1)}{\Gamma(a)^2}\int_0^1 e^{xt}t^{a-1}dt \tag{5.21}$$

where we have used a result of [ASR88]. The integral in (5.21) can be evaluated asymptotically in the limit $x \to \infty$ with $x \ll a$. Writing the integrand as $e^{xt+(a-1)\log t}$ it is plainly seen to have no saddle points in $[0,1]$ given the condition $x \ll a$. The leading order term therefore originates at the right edge $t = 1$. A simple application of Laplace's method leads to

$$\begin{aligned} r(a;x) &\sim \frac{a^{-1}x^a\Gamma(a+1)e^{-x}}{\Gamma(a)^2(a-1-x)} \\ &\sim \frac{x^a e^{-x}}{a\Gamma(a)} \\ &\sim \frac{x^a e^{-x}}{a\sqrt{2\pi a^{-1}}(ae^{-1})^a} \\ &= \frac{1}{\sqrt{2\pi}}\exp\left(-x + a\log x - a - a\log a - \frac{1}{2}\log a\right) \end{aligned}$$

where the penultimate line makes uses of Stirling's approximation [AAR99]. Since $a \gg x$,

$$-x + a\log x - a - a\log a - \frac{1}{2}\log a \sim -a\log a \to -\infty$$

which completes the proof. ■

Lemma 5.3. *Take any $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{n-1} \in \mathbb{R}^P$ let $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, for any $\boldsymbol{\mu} \in \mathbb{R}^P$ and Σ such that $\det\Sigma \geqslant A\sigma^{2P}$ for some constants $A, \sigma > 0$. Consider $P \to \infty$ with $P \gg \log n$ and let $\delta > 0$ be $o(P^{\frac{1}{2}})$ (note that δ and n need not diverge with P, but they can). Define*

$$B_i = \{\boldsymbol{x} \in \mathbb{R}^P \mid ||\boldsymbol{x} - \boldsymbol{x}_i|| < \delta\},$$

then as $P \to \infty$

$$\mathbb{P}\left(\boldsymbol{X} \in \bigcup_i B_i\right) \to 0 \tag{5.22}$$

and moreover as $P, n \to \infty$

$$n^l \mathbb{P}\left(\boldsymbol{X} \in \bigcup_i B_i\right) \to 0, \tag{5.23}$$

for any fixed $l > 0$.

Proof. With the Euclidean volume measure, we have

$$Vol\left(\bigcup_i B_i\right) \leqslant nV_P\delta^P = V_P(\delta n^{1/P})^P$$

where V_P is the volume of the unit sphere in P dimensions. Therefore a sphere of radius $\delta n^{1/P}$ is large enough to enclose all of the B_i and so the probability that $\boldsymbol{X}$ lies in any of the B_i is bounded above by the probability that it lies inside the sphere of radius $\delta n^{1/P}$ centred on its mean $\boldsymbol{\mu}$. Note that with $\hat{\sigma}^2 = (\det \Sigma)^{1/P}$, changing variables $\boldsymbol{x} = \hat{\sigma}^{-1}\Sigma^{1/2}\boldsymbol{y}$ gives

$$\int_{\mathbb{R}^P} d\boldsymbol{x} e^{-\frac{\boldsymbol{x}^T\Sigma^{-1}\boldsymbol{x}}{2}} = \int_{\mathbb{R}^P} d\boldsymbol{y} e^{-\frac{\boldsymbol{y}^2}{2\hat{\sigma}^2}}$$

since the Jacobian is 1. Thus we can reduce to a single dimensional Gaussian integral

$$\begin{aligned}
\mathbb{P}\left(\boldsymbol{X} \in \bigcup_i B_i\right) &\leqslant \frac{1}{(2\pi\hat{\sigma}^2)^{\frac{P}{2}}} \frac{2\pi^{\frac{P}{2}}}{\Gamma(\frac{P}{2})} \int_0^{\delta n^{\frac{1}{P}}} dr\ e^{-\frac{r^2}{2\hat{\sigma}^2}} r^{P-1} \\
&= \frac{2}{\Gamma(\frac{P}{2})} \int_0^{\frac{\delta n^{\frac{1}{P}}}{\sqrt{2}\hat{\sigma}}} dr\ e^{-r^2} r^{P-1} \\
&= \frac{1}{\Gamma(\frac{P}{2})} \int_0^{\frac{\delta n^{\frac{2}{P}}}{2\hat{\sigma}^2}} dr\ e^{-r} r^{\frac{P}{2}-1} \\
&\leqslant \frac{1}{\Gamma(\frac{P}{2})} \int_0^{\frac{\delta n^{\frac{2}{P}}}{2A^{1/P}\sigma^2}} dr\ e^{-r} r^{\frac{P}{2}-1} && \text{(using } \hat{\sigma}^2 \geqslant A^{1/P}\sigma^2\text{)} \\
&\leqslant \frac{1}{\Gamma(\frac{P}{2})} \int_0^{\frac{\delta n^{\frac{2}{P}}}{2\alpha\sigma^2}} dr\ e^{-r} r^{\frac{P}{2}-1} && \text{(with } \alpha \equiv \inf_P A^{1/P} > 0\text{)} \\
&\equiv \frac{1}{\Gamma(\frac{P}{2})} \gamma\left(\frac{P}{2}; \frac{n^{\frac{2}{P}}\delta^2}{2\sigma^2\alpha}\right) && (5.24)
\end{aligned}$$

where γ is the lower incomplete gamma function. Since $P \gg \log n$ and $\delta = o(P^{\frac{1}{2}})$, it follows that

$$x \equiv \frac{n^{\frac{2}{P}}\delta^2}{2\sigma^2\alpha} = o(P)$$

and so Lemma 5.2 can be applied to yield the result. Indeed, recalling that $n \ll e^P$, we have

$$n^l \mathbb{P}\left(\boldsymbol{X} \in \bigcup_i B_i\right) \leqslant e^{lP} r(P/2, x) \sim \frac{1}{\sqrt{2\pi}} \exp\left(lP - x + \frac{P}{2}\log x - \frac{P}{2} - \frac{P}{2}\log\frac{P}{2} - \frac{1}{2}\log\frac{P}{2}\right)$$

for any $l > 0$. But $x = o(P)$ so for P large enough, the term inside the exponential is negative and diverging with P, as required. ■

The previous two lemmas are required to prove the next lemma, which will form the foundation of our argument in the next section.

Lemma 5.4. *Let $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ be a sequence of jointly multivariate Gaussian random variables in $\mathbb{R}^P$ such that*

$$\boldsymbol{X}_i \mid \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{i-1}\} \sim \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)$$

where there exists a $\sigma > 0$ and a constant $A > 0$ such that $\det \Sigma_i \geqslant A\sigma^P$ for all P and i. Let also $\boldsymbol{X}_0$ be any deterministic element of $\mathbb{R}^P$. For $1 \leqslant m \leqslant n$, define the events

$$A_m(\delta) = \{||\boldsymbol{X}_i - \boldsymbol{X}_j||_2 > \delta \mid 0 \leqslant i < j \leqslant m\}.$$

Consider $P \to \infty$ with $P \gg \log n$ and let $\delta > 0$ be $o(P^{\frac{1}{2}})$ (note that δ and n need not diverge with P, but they can). Then $\mathbb{P}(A_n(\delta)) \to 1$ as $P \to \infty$.

Proof. Let us use the definitions of B_i from Lemma 5.3, i.e. let

$$B_i = \{\boldsymbol{x} \in \mathbb{R}^P \mid ||\boldsymbol{x} - \boldsymbol{X}_i|| < \delta\}$$

for $0 < j < n$. Since $A_i(\delta) \subset A_{i-1}(\delta)$ for any i, the chain rule of probability gives

$$\mathbb{P}(A_n(\delta)) = \mathbb{P}\left(\bigcap_{i \leqslant n} A_i(\delta)\right) = \mathbb{P}(A_1(\delta)) \prod_{i=2}^{n-1} \mathbb{P}(A_i \mid A_{i-1})$$

but

$$\mathbb{P}(A_i(\delta) \mid A_{i-1}(\delta)) = 1 - \mathbb{P}\left(\boldsymbol{X}_i \in \bigcup_{j<i} B_j\right)$$

and so

$$\mathbb{P}(A_n(\delta)) = \mathbb{P}(A_1(\delta)) \prod_{i=2}^{n-1}\left(1 - \mathbb{P}\left(\boldsymbol{X}_i \in \bigcup_{j<i} B_j\right)\right) \tag{5.25}$$

$$= \mathbb{P}\left(\boldsymbol{X}_1 \in B_0\right) \prod_{i=2}^{n-1}\left(1 - \mathbb{P}\left(\boldsymbol{X}_i \in \bigcup_{j<i} B_j\right)\right). \tag{5.26}$$

For fixed n, the result is now immediate from (5.23) in Lemma 5.3, since all the probabilities in (5.25) converge to 1 as $P \to \infty$ and there are only a finite number of terms.

Now consider the case that n also diverges. For any n define

$$s_n = \sup_{2 \leqslant i \leqslant n} \mathbb{P}\left(\boldsymbol{X}_i \in \bigcup_{j<i} B_j \right),$$

and then

$$\mathbb{P}(A_n(\delta)) \geqslant \mathbb{P}(\boldsymbol{X}_1 \in B_0) \prod_{i=2}^{n-1} (1 - s_{i-2}).$$

But, by Lemma 5.3 we can write $s_n = (n+1)^{-2} f_{n,P}$ where $f_{n,P} \to 0$ as $P \to \infty$, say, hence

$$\mathbb{P}(A_n(\delta)) \geqslant \mathbb{P}(\boldsymbol{X}_1 \in B_0) \prod_{i=2}^{n-1} \left(1 - (i-1)^{-2} f_{i-2,P}\right) \geqslant \mathbb{P}(\boldsymbol{X}_1 \in B_0) \prod_{i=2}^{\infty} \left(1 - (i-1)^{-2} f_{i-2,P}\right)$$

for large n, since $|f_{n-2,P}| < 1$ and all the extra terms added are strictly between 0 and 1. But

$$\log \prod_{i=2}^{\infty} \left(1 - (i-1)^{-2} f_{i-2,P}\right) \geqslant -\sum_{i=2}^{\infty} (i-1)^{-2} f_{i-2,P} \geqslant -\sup_j f_{j-2,P} \sum_{i=2}^{\infty} (i-1)^{-2} = -\frac{\pi^2}{6} \sup_j f_{j-2,P}$$

and so

$$\mathbb{P}(A_n(\delta)) \geqslant e^{-\sup_j f_{j-2,P} \pi^2/6} \mathbb{P}(\boldsymbol{X}_1 \in B_0)$$

but $f_{j-2,P} \to 0$ for any j, so as $P \to \infty$, $\mathbb{P}(A_n(\delta))$ is lower bounded by a term converging to $\mathbb{P}(\boldsymbol{X}_1 \in B_0)$ which, in turn, converges to 1 by Lemma 5.3.

■

Recall the Gaussian process covariance structure from above (5.18):

$$\operatorname{Cov}(\varepsilon_i(\boldsymbol{w}), \varepsilon_j(\boldsymbol{w}')) = (w_i - w_i')(w_j' - w_j) k''\left(-\frac{1}{2}\|\boldsymbol{w} - \boldsymbol{w}'\|_2^2\right) + \delta_{ij} k'\left(-\frac{1}{2}\|\boldsymbol{w} - \boldsymbol{w}'\|_2^2\right) \tag{5.18}$$

Lemma 5.5. *Assume the covariance structure (5.18). Take any $a_i \in \mathbb{R}$ and define $\bar{\varepsilon} = \sum_{i=1}^n a_i \varepsilon_i$. Then*

$$Tr\, \operatorname{Cov}(\bar{\varepsilon}) = k'(0) P \sum_{i=1}^n a_i^2 + 2P \sum_{1 \leqslant i < j \leqslant n} a_i a_j \left[k'(-\frac{d_{ij}^2}{2}) + P^{-1} k''(-\frac{d_{ij}^2}{2}) d_{ij}^2 \right] \tag{5.27}$$

where we define $d_{ij} = \|\boldsymbol{w}_i - \boldsymbol{w}_j\|_2$.

Proof. Each of the ε_i is Gaussian distributed with covariance matrix $\operatorname{Cov}(\varepsilon_i)$ given by (5.18) and the covariance between different gradients $\operatorname{Cov}(\varepsilon_i, \varepsilon_j)$ is similarly given by (5.18). By standard multivariate Gaussian properties

$$\operatorname{Cov}(\bar{\varepsilon}) = \sum_{i=1}^n a_i^2 \operatorname{Cov}(\varepsilon_i) + \sum_{i \neq j} a_i a_j \operatorname{Cov}(\varepsilon_i, \varepsilon_j), \tag{5.28}$$

then taking the trace

$$\text{Tr Cov}(\bar{\varepsilon}) = \sum_{i=1}^{n} a_i^2 \text{Tr}(\text{Cov}(\varepsilon_i)) + 2 \sum_{1\leqslant i<j\leqslant n} a_i a_j \text{Tr}(\text{Cov}(\varepsilon_i, \varepsilon_j)). \tag{5.29}$$

Using the covariance structure from (5.18) gives

$$\begin{aligned}\text{Tr Cov}(\bar{\varepsilon}) = k'(0)\sum_{i=1}^{n} a_i^2 \text{Tr} I + 2 \sum_{1\leqslant i<j\leqslant n} a_i a_j \Bigg[& k'(-\frac{d_{ij}^2}{2})\text{Tr} I \\ & + k''(-\frac{d_{ij}^2}{2})\text{Tr}(\boldsymbol{w}_i - \boldsymbol{w}_j)(\boldsymbol{w}_j - \boldsymbol{w}_i)^T \Bigg]\end{aligned} \tag{5.30}$$

from which the result follows. ■

5.2.2 Main results for dependent noise models

Theorem 5.2. *Let $\boldsymbol{w}_n$ and $\boldsymbol{w}_{avg}$ be defined as in Theorem 5.1 and let the gradient perturbation be given by the covariance structure in (5.17). Assume that the kernel function k is such that $k(-x)$ and its derivatives decay at least as fast as $|x|^M e^{-x}$, for some $M > 0$, as $x \to \infty$ and define $\sigma^2 B^{-1} = k'(0)$. Assume further that $P^{1-\theta} \gg \log n$ for some $\theta \in (0,1)$. Let $\delta = o(P^{1/2})$. Then $\boldsymbol{w}_n$ and $\boldsymbol{w}_{avg}$ are multivariate Gaussian random variables and, with probability which approaches unity as $P, n \to \infty$ the iterates $\boldsymbol{w}_t$ are all mutually at least δ apart and*

$$\mathbb{E}w_{n,i} \sim e^{-\alpha\lambda_i n} w_{0,i}, \qquad \frac{1}{P}Tr\text{Cov}(\boldsymbol{w}_n) \sim \frac{\alpha\sigma^2}{B}\left\langle \frac{1}{\lambda(2-\alpha\lambda)} \right\rangle, \tag{5.31}$$

$$\mathbb{E}w_{avg,i} \sim \frac{1-\alpha\lambda_i}{\alpha\lambda_i n} w_{0,i}, \frac{1}{P}Tr\text{Cov}(\boldsymbol{w}_{avg}) \leqslant \frac{\sigma^2}{Bn}\left\langle \frac{1}{\lambda} \right\rangle + \mathcal{O}(1)\left(k'(-\frac{\delta^2}{2}) + P^{-1}\delta^2 k''(-\frac{\delta^2}{2})\right). \tag{5.32}$$

Proof. We will prove the result in the case $\lambda_i = \lambda\ \forall i$ for the sake of clarity. The same reasoning can be repeated in the more general case; where one gets $P^{-1}f(\lambda)\text{Tr}I$ below, one need only replace it with $\langle f(\lambda)\rangle$, exploiting linearity of the trace. We will also vacuously replace $\sigma^2 B^{-1}$ with σ^2 to save on notation. For weight iterates $\boldsymbol{w}_i$, we have the recurrence

$$\boldsymbol{w}_i = (1-\alpha\lambda)\boldsymbol{w}_{i-1} + \alpha\varepsilon(\boldsymbol{w}_{i-1})$$

which leads to

$$\boldsymbol{w}_n = (1-\alpha\lambda)^n \boldsymbol{w}_0 + \alpha\sum_{i=0}^{n-1}(1-\alpha\lambda_i)^{n-i-1}\varepsilon(\boldsymbol{w}_i) \tag{5.33}$$

and then

$$\boldsymbol{w}_{avg} = \frac{1-(1-\alpha\lambda)^n}{\alpha\lambda n}(1-\alpha\lambda)\boldsymbol{w}_0 + \sum_{i=0}^{n-1}\varepsilon(\boldsymbol{w}_i)\frac{1-(1-\alpha\lambda)^{n-i}}{\lambda n}. \tag{5.34}$$

As above, define $\varepsilon_i = \varepsilon(\boldsymbol{w}_i)$, for convenience. Now define

$$a_i = \alpha(1-\alpha\lambda)^{n-1-i}, \quad \bar{a}_i = \frac{1-(1-\alpha\lambda)^{n-i}}{\lambda n}.$$

Next we will apply Lemma 5.5 and utilise Lemma 5.4 to bound the variance of $\boldsymbol{w}_{avg}$ and $\boldsymbol{w}_n$. We first gather the following facts, which were also computed and used in the proof of Theorem 1:

$$\sum_{i=1}^{n-1} a_i^2 = \frac{\alpha^2(1-(1-\alpha\lambda)^{2n})}{1-(1-\alpha\lambda)^2} \tag{5.35}$$

$$\sum_{i<j} a_i a_j = \frac{\alpha}{\lambda}\left(\frac{1-(1-\alpha\lambda)^n}{\alpha\lambda} - \frac{1-(1-\alpha\lambda)^{2n}}{1-(1-\alpha\lambda)^2}\right). \tag{5.36}$$

The sum of squares for the $\bar{a}_i$ is simple to obtain similarly

$$\sum_{i=0}^{n-1} \bar{a}_i^2 = \frac{1}{\lambda^2 n^2}\left(n - \frac{2(1-(1-\alpha\lambda)^n)}{\alpha\lambda} + \frac{1-(1-\alpha\lambda)^{2n}}{1-(1-\alpha\lambda)^2}\right). \tag{5.37}$$

We now use the assumption that $0 < \alpha\lambda < 1$ (required for the convergence of gradient descent) which gives, as $n \to \infty$,

$$\sum_{i=1}^{n-1} a_i^2 \sim \frac{\alpha^2}{1-(1-\alpha\lambda)^2} \tag{5.38}$$

$$\sum_{i<j} a_i a_j \sim \frac{\alpha}{\lambda}\left(\frac{1}{\alpha\lambda} - \frac{1}{1-(1-\alpha\lambda)^2}\right) \tag{5.39}$$

$$\sum_{i=1}^{n-1} \bar{a}_i^2 \sim \frac{1}{\lambda^2 n} \tag{5.40}$$

Summing $\sum_{i<j} \bar{a}_i \bar{a}_j$ explicitly is possible but unhelpfully complicated. Instead, some elementary bounds give

$$\sum_{i<j} \bar{a}_i \bar{a}_j \leqslant \left(\sum_{i=0}^{n-1} \bar{a}_i\right)^2 = \frac{1}{\lambda^2 n^2}\left(n - \frac{1-(1-\alpha\lambda)^n}{\alpha\lambda}\right)^2 \sim \frac{1}{\lambda^2}$$

and

$$\sum_{i<j} \bar{a}_i \bar{a}_j \geqslant \sum_{i<j}\left(\frac{1-(1-\alpha\lambda)^{n-1}}{\lambda n}\right)^2 \sim \frac{1}{2\lambda^2}$$

so in particular $\sum_{i<j} \bar{a}_i \bar{a}_j = \mathcal{O}(1)$. Now define the events $A_n(\delta)$ as in Lemma 5.4 using ε_i in place of $\boldsymbol{X}_i$. Further, choose δ large enough so that $k'(-\frac{x^2}{2})$ and $x^2 k''(-\frac{x^2}{2})$ are decreasing for $x > \delta$. Define $k'(0) = \sigma^2$. Lemma 5.5 gives

$$\frac{1}{P}\mathrm{TrCov}(\boldsymbol{w}_n) \mid A_n(\delta) \leqslant \sigma^2 \sum_{i=1}^{n} a_i^2 + 2\sum_{i<j} a_i a_j\left(k'(-\frac{\delta^2}{2}) + P^{-1}\delta^2 k''(-\frac{\delta^2}{2})\right) \tag{5.41}$$

where we note that we have only upper-bounded the second term in (5.41), so using (5.38) and (5.39) and taking δ large enough we obtain

$$\frac{1}{P}\mathrm{TrCov}(\boldsymbol{w}_n) \mid A_n(\delta) = \frac{\sigma^2\alpha^2}{1-(1-\alpha\lambda)^2} + o(1). \tag{5.42}$$

Turning now to $\boldsymbol{w}_{avg}$ we similarly obtain

$$\frac{1}{P}\mathrm{TrCov}(\boldsymbol{w}_{avg}) \mid A_n(\delta) \leqslant \frac{\sigma^2}{n}\frac{1}{\lambda^2} + \mathcal{O}(1)\left(k'(-\frac{\delta^2}{2}) + P^{-1}\delta^2 k''(-\frac{\delta^2}{2})\right) \tag{5.43}$$

and, as before, taking δ large enough we can obtain

$$\frac{1}{P}\mathrm{TrCov}(\boldsymbol{w}_{avg}) \mid A_n(\delta) = o(1). \tag{5.44}$$

Finally recalling (5.33) and (5.34) and writing $(1-\alpha\lambda)^n = e^{-\alpha\lambda n} + o(1)$ for large n, we obtain the results in the statement of the theorem but conditional on the event $A_n(\delta)$. To complete the proof, we need only to establish that $\mathbb{P}(A_n(\delta)) \to 1$ $P, n \to \infty$, which we will do with an application of Lemma 5.4. Since the loss noise term is a Gaussian process, the $\varepsilon(\boldsymbol{w}_i)$ are all jointly Gaussian with the covariance structure (5.18), but to apply Lemma 5.4 we must further establish a lower bound on the covariance of the conditional ε_i. Let Σ_n be the $P\times P$ covariance matrix of $\varepsilon_n \mid \{\varepsilon_1,\ldots,\varepsilon_{n-1}\}$, then we are required to show that there exists some n-independent $A, \sigma > 0$ such that $\det\Sigma_n > A\sigma^{2P}$ for all n (subject to $\log n \ll P$). Define S_n to be the $nP\times nP$ covariance matrix of all of the $\{\varepsilon_i\}_{i=1}^n$, i.e.

$$(S_n)_{iP+j,kP+l} = \mathrm{Cov}\big(\varepsilon_j(\boldsymbol{w}_j), \varepsilon_l(\boldsymbol{w}_k)\big), \quad 0 \leqslant i,k < n, \; 1 \leqslant j,l \leqslant P,$$

and for convenience define $k'(0) = s^2$. The rules of standard Gaussian conditioning give

$$\Sigma_n = s^2 I - X_n S_{n-1}^{-1} X_n^T,$$

where X_n is the $P\times(n-1)P$ matrix such that S_n has the following block structure

$$S_n = \left(\begin{array}{c|c} S_{n-1} & X_n^T \\ \hline X_n & s^2 I_P \end{array}\right), \tag{5.45}$$

so, concretely, from (5.18)

$$(X_n)_{i,Pj+l} = \big((\boldsymbol{w}_n)_i - (\boldsymbol{w}_j)_i\big)\big((\boldsymbol{w}_j)_l - (\boldsymbol{w}_n)_l\big)\, k''\left(-\frac{1}{2}d_{jn}^2\right) + \delta_{il}k'\left(-\frac{1}{2}d_{jn}^2\right), \tag{5.46}$$

for $1 \leqslant i,l \leqslant P$, $0 \leqslant j < n-1$. We can now Taylor expand the determinant

$$\begin{aligned}\det\Sigma_n &= s^{2P}\det\big(1 - s^{-2}X_n S_{n-1}^{-1} X_n^T\big)\\ &= s^{2P}\big(1 - s^{-2}\mathrm{Tr}X_n S_{n-1}^{-1}X_n^T\big) + \ldots\end{aligned}$$

which is valid provided that the trace term is small compared with 1. We have

$$|\mathrm{Tr} X_n S_{n-1}^{-1} X_n^T| \leqslant \mathrm{Tr} X_n X_n^T \|S_{n-1}^{-1}\|_{op} = \|X_n\|_F \|S_{n-1}^{-1}\|_{op}$$

where $\|\cdot\|_F, \|\cdot\|_{op}$ are the Frobenius and operator matrix norms respectively. Hence, it suffices to prove n, P-independent bounds $\|S_{n-1}^{-1}\|_{op} < q$ for some $q > 0$ and $\|X_n\|_F < c$ for some $0 < c < s^2/10$, say, valid for all n large enough, to thence obtain $\det\Sigma_n \geqslant c' s^{2P}$ for some constant $c' > 0$. Strictly speaking, one must use a bounded form of the remainder in Taylor's theorem to make precise all of these constants, but in reality we will see that we can make c as small as necessary, so that certainly $c' > 0$ exists and the bound $\det\Sigma_n \geqslant c' s^{2P}$ holds. Proceeding directly

$$\begin{aligned}\|X_n\|_F = \mathrm{Tr} X_n X_n^T &= \sum_{i,l=1}^{P} \sum_{j=0}^{n-2} (X_n)_{i,Pj+l}^2 \\ &= \sum_{j=0}^{n-2} \left\{ Pk'\left(-\frac{d_{jn}^2}{2}\right) - 2d_{jn}^2 k'\left(-\frac{d_{jn}^2}{2}\right) k''\left(-\frac{d_{jn}^2}{2}\right) + \left[d_{jn}^2 k''\left(-\frac{d_{jn}^2}{2}\right)\right]^2 \right\} \\ &\leqslant (n-1)\left(Pk'\left(-\frac{\delta^2}{2}\right) - 2\delta^2 k'\left(-\frac{\delta^2}{2}\right) k''\left(-\frac{\delta^2}{2}\right) + \left[\delta^2 k''\left(-\frac{\delta^2}{2}\right)\right]^2 \right),\end{aligned}$$

but recall that we require $\delta = o(P^{1/2})$, so take for example $\delta = aP^{1/2-\varphi/2}$ for some $0 < \varphi < 1$, so

$$\|X_n\|_F \leqslant (n-1)\left(Pk'\left(-\frac{P^{1-\varphi}}{2}\right) - 2P^{1-\varphi} k'\left(-\frac{P^{1-\varphi}}{2}\right) k''\left(-\frac{P^{1-\varphi}}{2}\right) + \left[P^{1-\varphi} k''\left(-\frac{P^{1-\varphi}}{2}\right)\right]^2 \right).$$

Now recall that $xk'(-x)$ and $xk''(-x)$ are decaying for large enough x, and $\log n \ll P^{1-\theta}$, hence

$$\|X_n\|_F \leqslant (n-1)\left(2\log^{\frac{1}{1-\theta}} n k'\left(-\frac{\log^{\frac{1-\varphi}{1-\theta}} n}{2}\right) + \left[\log^{\frac{1-\varphi}{1-\theta}} n \;\; k''\left(-\frac{\log^{\frac{1-\varphi}{1-\theta}} n}{2}\right)\right]^2 \right).$$

Since $\theta > 0$, we can take some $0 < \varphi < \theta$ so that there exists $\chi \in (0,1)$ such that

$$\log^{\frac{1-\varphi}{1-\theta}} n > \log^{1+\chi} n \tag{5.47}$$

for large enough n, and so

$$\|X_n\|_F \leqslant (n-1)\left(2\log^{\frac{1}{1-\theta}} n k'\left(-\frac{\log^{1+\chi} n}{2}\right) + \left[\log^{1+\chi} n \;\; k''\left(-\frac{\log^{1+\chi} n}{2}\right)\right]^2 \right).$$

We assume that $k'(x), k''(x)$ decay at least as fast as $x^M e^{-x}$ for some $M > 0$ as $x \to \infty$, i.e. $k'(x)x^{-M}e^x \to 0$ (and similarly $k''(x)$). Writing $n-1 \leqslant n = e^{\log n}$, we have

$$\|X_n\|_F \leqslant 2\log^{\frac{1}{1-\theta}} n k'\left(\log n - \frac{\log^{1+\chi} n}{2}\right) + \left[\log^{1+\chi} n \;\; k''\left(\log n - \frac{\log^{1+\chi} n}{2}\right)\right]^2,$$

but for large n, $\log^{1+\chi} n \gg \log n$ and so this last expression clearly converges to 0 as $n \to \infty$. Indeed, $e^{-\log^{1+\chi} n/2}$ decays faster than any fixed power of n, so the same is true of $\|X_n\|_F$. Hence we can find the constant $c > 0$ such that, for large enough $n > n_0$, say, $\|X_n\|_F < c$, as required. Now we turn to bounding $\|S_{n-1}^{-1}\|_{op}$, which is done by induction on n. Define the upper bounds $\|S_n^{-1}\|_{op} \leqslant q_n$ for all n. Recalling the block structure (5.45), we get the inverse

$$S_n^{-1} = \left(\begin{array}{cc} (S_{n-1} - s^{-2}X_n^T X_n)^{-1} & 0 \\ 0 & \Sigma_n^{-1} \end{array}\right)\left(\begin{array}{cc} I & -s^{-2}X_n^T \\ -s^{-2}X_n & I \end{array}\right) \equiv YZ.$$

$\|S_n^{-1}\|_{op}$ is bounded above by $\|X\|_{op}, \|Y\|_{op}$ and so we now bound these norms in turn. Since the off diagonals are zero, we have

$$\|Y\|_{op} \leqslant \max\{\|\Sigma_n^{-1}\|_{op}, \|(S_{n-1} - s^{-2}X_n^T X_n)^{-1}\|_{op}\}.$$

Recalling the expression for Σ_n above and expanding the matrix inverse

$$\begin{aligned} \|\Sigma_n^{-1}\|_{op} &= s^{-2}\|(I - s^{-2}X_n S_{n-1}^{-1} X_n^T)^{-1}\|_{op} \\ &= s^{-2}\|(I + s^{-2}X_n S_{n-1}^{-1}X_n^T + s^{-4}(X_n S_{n-1}^{-1}X_n^T)^2 + \ldots\|_{op} \\ &\leqslant s^{-2}\left(1 + s^{-2}\|X_n S_{n-1}^{-1}X_n^T\|_{op} + s^{-4}\|_{op}(X_n S_{n-1}^{-1} X_n^T)^2\|_{op} + \ldots\right) \\ &\leqslant s^{-2}\left(1 + s^{-2}\|X_n\|_F\|S_{n-1}^{-1}\|_{op} + s^{-4}\|X_n\|_F^2\|S_{n-1}^{-1}\|_{op}^2 + \ldots\right) \\ &\leqslant s^{-2}\left(1 + s^{-2}\|X_n\|_F\|q_{n-1} + s^{-4}\|X_n\|_F^2\|q_{n-1}^2 + \ldots\right) \\ &\leqslant s^{-2}(1 + \alpha s^{-2} q_{n-1}\|X_n\|_F) \end{aligned}$$

for some constant $\alpha > 0$, since we have already demonstrated that $\|X_n\|_F \to 0$ as $n \to \infty$. For the other term

$$\|(S_{n-1} - s^{-2}X_n^T X_n)^{-1}\|_{op} \leqslant \|S_{n-1}^{-1}\|_{op}\|(I - s^{-2}S_{n-1}^{-1}X_n^T X_n)^{-1}\|_{op}$$

from which point, one proceeds just as for $\|\Sigma_n^{-1}\|_{op}$ to obtain

$$\|(S_{n-1} - s^{-2}X_n^T X_n)^{-1}\|_{op} \leqslant q_{n-1}(1 + \alpha s^{-2} q\|X_n\|_F),$$

hence overall

$$\|Y\|_{op} \leqslant \max\{s^{-2}(1 + \alpha s^{-2}q_{n-1}\|X_n\|_F), q_{n-1}(1 + \alpha s^{-2} q_{n-1}\|X_n\|_F)\}.$$

We can always relax the bound on $\|S_{n-1}^{-1}\|_{op}$ so that $q_{n-1} > s^{-2}$, so we simply have $\|Y\|_{op} \leqslant q_{n-1}(1 + \alpha s^{-2} q_{n-1}\|X_n\|_F)$. To bound $\|Z\|_{op}$, we split it into a sum of two matrices

$$\|Z\|_{op} = \left\|\left(\begin{array}{cc} I & 0 \\ 0 & I \end{array}\right) + \left(\begin{array}{cc} 0 & -s^{-2}X_n^T \\ -s^{-2}X_n & 0 \end{array}\right)\right\|_{op} \leqslant 1 + 2s^{-2}\|X\|_{op} \leqslant 1 + 2s^{-2}\|X_n\|_F,$$

but $\|X_n\|_F \to 0$ as $n \to \infty$, so overall we can say

$$\|S_n^{-1}\|_{op} \leqslant q_{n-1}(1 + r_n), \quad r_n \equiv s^{-2}\|X_n\|_F(\alpha q_{n-1} + 2 + 2\alpha q_{n-1}\|X_n\|_F),$$

which we can simplify to

$$\|S_n^{-1}\|_{op} \leqslant q_{n-1}(1+r'_n), \quad r'_n \equiv s^{-2}\|X_n\|_F(\alpha' q_{n-1}+2)$$

and so can say

$$q_n = q_{n-1} + 2s^{-2}\|X_n\|_F q_{n-1} + s^{-2}\alpha'\|X_n\|_F q_{n-1}^2.$$

For large enough n, we seek a stability solution to this recurrence, i.e. using the ansatz $q_n = q + h_n$ for h_n small

$$q + h_n = q + h_{n-1} + 2s^{-2}\|X_n\|_F q + 2s^{-2}\|X_n\|_F h_{n-1} + s^{-2}\alpha'\|X_n\|_F(q^2 + 2qh_{n-1} + h_{n-1}^2). \tag{5.48}$$

Gathering the leading order terms gives

$$\begin{aligned} h_n &= h_{n-1} + 2s^{-2}q\|X_n\|_F + s^{-2}\alpha'\|X_n\|_F q^2 \\ \implies h_n &= h_{n_0} + s^{-2}q(2+q\alpha')\sum_{j=n_0+1}^{n}\|X_j\|_F. \end{aligned}$$

Recall that $\|X_n\|_F$ decays faster than any fixed power of n, so the sum $\sum_{j\geqslant 2}\|X_j\|_F$ converges, hence for $\varepsilon > 0$ we can take some fixed n_0 large enough so that $\sum_{j=n_0+1}^{n}\|X_j\|_F < \varepsilon$ for all $n > n_0$. We are free to choose $h_{n_0} = 0$ and then for large enough n_0, we can guarantee $|h_n| < 1$, say, thus

$$q_n \leqslant \max\left\{\max_{1\leqslant m\leqslant n_0} q_m, q_{n_0}+1\right\} \equiv q^*.$$

Hence we have succeeded in bounding $\|S_n^{-1}\|_{op} \leqslant q^*$ for all n. Combined with the earlier bound on $\|X_n\|_F$, we have now established the bound $\det\Sigma_n \geqslant c's^{2P}$, so we have satisfied the conditions of Lemma 5.4 and completed the proof. ■

Note that Theorem 5.2 is a generalisation of Theorem 5.1 to the context of our dependent perturbation model. Let us make some clarifying remarks about the theorem and its proof:

1. The bound (5.32) in the statement of the theorem relies on *all* iterates being separated by a distance at least δ. Moreover, the bound is only useful if δ is large enough to ensure the k' and k'' terms are small.

2. Just as in the independent case of Theorem 5.1, the first term in the bound in (5.32) decays only in the case that the number of iterates $n \to \infty$.

3. The remaining conditions on P, n, δ are required for the high-dimensional probability argument which we use to ensure that all iterates are separated by at least δ.

4. $P \gg \log n$ is a perfectly reasonable condition in the context of deep learning. E.g. for a ResNet-50 with $P \approx 25 \times 10^6$, violation of this condition would require $n > 10^{10^7}$. A typical ResNet schedule on ImageNet has $< 10^6$ total steps.

Consequently, our result points to the importance of good separation between weight iterates in IA to retain the independence benefit and variance reduction in a non-independent noise setting, hence one would expect large learning rates to play a crucial role in successful IA. At the same time, our result is particularly adapted to the *deep learning limit* of very many model parameters ($P \to \infty$), since this is the only regime in which we can argue probabilistically for good separation of weight iterates (otherwise one may simply have to assume such separation). Furthermore, the importance of $P \gg \log n$ indicates that perhaps averaging less frequently than every iteration could be beneficial to generalisation. The following corollary makes this intuition precise.

Corollary 5.1. *Let $\boldsymbol{w}_{avg}$ now be a strided iterate average with stride κ, i.e.*

$$\boldsymbol{w}_{avg} = \frac{\kappa}{n}\sum_{i=1}^{\lfloor n/\kappa \rfloor} \boldsymbol{w}_i. \tag{5.49}$$

Then, under the same conditions as Theorem 5.2

$$\mathbb{E}w_{avg,i} = \frac{\kappa(1-\alpha\lambda_i)^{\kappa}}{n(1-(1-\alpha\lambda_i)^{\kappa})}(1+o(1))\,w_{0,i}, \tag{5.50}$$

$$\frac{1}{P}Tr\text{Cov}(\boldsymbol{w}_{avg}) \leqslant \frac{\sigma^2\alpha^2\kappa}{Bn}\left\langle \frac{1}{(1-(1-\alpha\lambda)^{\kappa})^2}\frac{1-(1-\alpha\lambda)^{2\kappa}}{1-(1-\alpha\lambda)^2}\right\rangle + \mathcal{O}(1)\left(k'(-\frac{\delta^2}{2}) + P^{-1}\delta^2 k''(-\frac{\delta^2}{2})\right) \tag{5.51}$$

where the constant $\mathcal{O}(1)$ coefficient of the second term in (5.51) is independent of κ.

Proof. The proof is just as in Theorem 2 (or Theorems 3 or 4), differing only in the values of the $\bar{a}_i$. Indeed, a little thought reveals that the generalisation of $\bar{a}_i$ to the case $\kappa > 1$ is

$$\bar{a}_i = \frac{\alpha\kappa}{n}(1-\alpha\lambda)^{\kappa\left(1+\lfloor\frac{i}{\kappa}\rfloor\right)-1-i}\frac{1-(1-\alpha\lambda)^{\kappa\left(\lfloor\frac{n}{\kappa}\rfloor-\lfloor\frac{i}{\kappa}\rfloor\right)}}{1-(1-\alpha\lambda)^{\kappa}}. \tag{5.52}$$

Note that $\kappa\left\lfloor\frac{i}{\kappa}\right\rfloor - i$ is just the (negative) remainder after division of i by κ. Then for large n

$$\begin{aligned}
\sum_i \bar{a}_i^2 &\sim \frac{\alpha^2\kappa^2}{n^2}\frac{(1-\alpha\lambda)^{2(\kappa-1)}}{(1-(1-\alpha\lambda)^{\kappa})^2}\left\lfloor\frac{n}{\kappa}\right\rfloor\sum_{i=0}^{\kappa-1}(1-\alpha\lambda)^{-2i}\\
&\leqslant \frac{\alpha^2\kappa}{n}\frac{(1-\alpha\lambda)^{2(\kappa-1)}}{(1-(1-\alpha\lambda)^{\kappa})^2}\sum_{i=0}^{\kappa-1}(1-\alpha\lambda)^{-2i}\\
&= \frac{\alpha^2\kappa}{n}\frac{(1-\alpha\lambda)^{2(\kappa-1)}}{(1-(1-\alpha\lambda)^{\kappa})^2}\frac{1-(1-\alpha\lambda)^{-2\kappa}}{1-(1-\alpha\lambda)^{-2}}\\
&= \frac{\alpha^2\kappa}{n}\frac{1}{(1-(1-\alpha\lambda)^{\kappa})^2}\frac{1-(1-\alpha\lambda)^{2\kappa}}{1-(1-\alpha\lambda)^2}.
\end{aligned}$$

and similarly

$$\sum_{i<j}\bar{a}_i\bar{a}_j \sim \frac{\alpha^2\kappa^2}{n^2}\frac{(1-\alpha\lambda)^{2(\kappa-1)}}{(1-(1-\alpha\lambda)^{\kappa})^2}\sum_{i<j}(1-\alpha\lambda)^{\kappa\lfloor i/\kappa\rfloor-i+\kappa\lfloor j/\kappa\rfloor-j} \tag{5.53}$$

$$\sim \frac{\alpha^2\kappa^2}{n^2}\frac{(1-\alpha\lambda)^{2(\kappa-1)}}{(1-(1-\alpha\lambda)^{\kappa})^2}\sum_{j}(1-\alpha\lambda)^{\kappa\lfloor j/\kappa\rfloor-j}\left\lfloor\frac{j}{\kappa}\right\rfloor\frac{1-(1-\alpha\lambda)^{-\kappa}}{1-(1-\alpha\lambda)^{-1}} \tag{5.54}$$

$$\sim \frac{\alpha^2\kappa^2}{n^2}\frac{(1-\alpha\lambda)^{2(\kappa-1)}}{(1-(1-\alpha\lambda)^{\kappa})^2}\left(\frac{1-(1-\alpha\lambda)^{-\kappa}}{1-(1-\alpha\lambda)^{-1}}\right)^2\sum_{j=0}^{\lfloor n/\kappa\rfloor} j \tag{5.55}$$

$$\sim \frac{\alpha^2}{2}\frac{(1-\alpha\lambda)^{2(\kappa-1)}}{(1-(1-\alpha\lambda)^{\kappa})^2}\left(\frac{1-(1-\alpha\lambda)^{-\kappa}}{1-(1-\alpha\lambda)^{-1}}\right)^2 \tag{5.56}$$

$$= \frac{\alpha^2}{2}\frac{(1-\alpha\lambda)^{-2}}{(1-(1-\alpha\lambda)^{-1})^2}. \tag{5.57}$$

■

Intuitively, the first term in the covariance in (5.32) is an "independence term", i.e. it is common between Theorems 5.1 and 5.2 and represents the simple variance reducing effect of averaging. The second variance term in (5.32) comes from dependence between the iterate gradient perturbations. We see from the corollary that an independent model for gradient perturbation would predict an unambiguous inflationary effect of strided IA on variance (the first term in (5.51)). However introducing dependence in the manner that we have predicts a more nuanced picture, where increased distance between weight iterates can counteract the simple "independent term" inflationary effect of striding, leaving open the possibility for striding to improve on standard IA for the purposes of generalisation.

5.3 Extension of theoretical framework to weight decay and adaptive methods

To make a closer connection with the new optimisation algorithms proposed in this work we consider decoupled weight decay (strength γ) and gradient preconditioning:

$$\boldsymbol{w}_t = (1-\alpha\gamma)\boldsymbol{w}_{t-1} - \alpha\tilde{\boldsymbol{H}}_t^{-1}\nabla L_{batch}(\boldsymbol{w}_{t-1}) \tag{5.58}$$

where $\tilde{\boldsymbol{H}}_t^{-1}$ is some approximation to the true loss Hessian used at iteration t. In the presence of weight decay, we move the true loss minimum away from the origin for the analysis, i.e. $L_{\text{true}}(\boldsymbol{w}) = (\boldsymbol{w}-\boldsymbol{w}^*)^T\boldsymbol{H}(\boldsymbol{w}-\boldsymbol{w}^*)$. The update rule is then

$$\boldsymbol{w}_t = \left(1-\alpha\gamma-\alpha\tilde{\boldsymbol{H}}_t^{-1}\boldsymbol{H}\right)\boldsymbol{w}_{t-1} + \alpha\boldsymbol{H}\boldsymbol{w}^* - \alpha\varepsilon(\boldsymbol{w}_{t-1}). \tag{5.59}$$

We take $\tilde{\boldsymbol{H}}_t^{-1}$ to be diagonal in the eigenbasis of $\boldsymbol{H}$, with eigenvalues $\tilde{\lambda}_i^{(t)}+\epsilon$, where ϵ is the standard tolerance parameter [KB14]. One could try to construct the $\tilde{\boldsymbol{H}}_t^{-1}$ from the Gaussian process loss model, so making them stochastic and covarying with the gradient noise, however we do not believe this is tractable. Instead, let us heuristically assume that, with high probability, $\tilde{\lambda}_i^{(t)}$ is close to λ_i, say within a distance ζ, for large enough t and all i. If we take a large enough ζ this is true even for SGD and we expect Adam to better approximate the local curvature matrix than SGD, since this is precisely what it is designed to do. This results in the following theorem.

Theorem 5.3. *Fix some $\zeta > 0$ and assume that $|\tilde{\lambda}_i^{(t)} - \lambda_i| < \zeta$ for all $t \geqslant n_0$, for some fixed $n_0(\zeta)$, with high probability. Use the update rule (5.59). Assume that the λ_i are bounded away from zero and $\min_i \lambda_i > \zeta$. Further assume $c(\gamma + \varepsilon + \zeta) < 1$, where c is a constant independent of $\varepsilon, \zeta, \gamma$ and is defined in the proof. Let everything else be as in Theorem 5.2. Then there exist constants $c_1, c_2, c_3, c_4 > 0$ such that, with high probability,*

$$|\mathbb{E}w_{n,i} - w_i^*| \leqslant e^{-\alpha(1+\gamma-c(\varepsilon+\zeta))n} w_{0,i} + c_1(\varepsilon + \zeta + \gamma) \tag{5.60}$$

$$\left|\frac{1}{P} Tr\text{Cov}(\boldsymbol{w}_n) - \frac{\alpha\sigma^2}{B(2-\alpha)}\right| \leqslant c_2(\varepsilon + \zeta + \gamma) + o(1), \tag{5.61}$$

$$|\mathbb{E}w_{avg,i} - w_i^*| \leqslant \frac{1 - \alpha(1+\gamma - c(\varepsilon+\zeta))}{\alpha(1+\gamma-c(\varepsilon+\zeta))n}(1 + o(1)) w_{0,i} + c_3(\varepsilon + \zeta + \gamma) \tag{5.62}$$

$$\left|\frac{1}{P} Tr\text{Cov}(\boldsymbol{w}_{avg}) - \frac{\sigma^2}{Bn} - \mathcal{O}(1)\left(k'(-\frac{\delta^2}{2}) + P^{-1}\delta^2 k''(-\frac{\delta^2}{2})\right)\right| \leqslant c_4(\gamma, +\zeta + \varepsilon). \tag{5.63}$$

Proof. We begin with the equivalent of (5.33) for update rule (5.59):

$$\begin{aligned} \boldsymbol{w}_n = \prod_{i=0}^{n-1}\left(1 - \alpha\gamma - \alpha\tilde{\boldsymbol{H}}_i^{-1}\Lambda\right)\boldsymbol{w}_0 + \sum_{i=}^{n-1} \alpha\tilde{\boldsymbol{H}}_i^{-1}\Lambda \prod_{j=i+1}^{n-1}\left(1 - \alpha\gamma - \alpha\tilde{\boldsymbol{H}}_j^{-1}\Lambda\right)\boldsymbol{w}^* \\ - \sum_{i=}^{n-1} \alpha\tilde{\boldsymbol{H}}_i^{-1}\Lambda\left[\prod_{j=i+1}^{n-1}\left(1 - \alpha\gamma - \alpha\tilde{\boldsymbol{H}}_j^{-1}\Lambda\right)\right]\varepsilon(\boldsymbol{w}_i). \end{aligned} \tag{5.64}$$

To make progress, we need the following bounds valid for all $t \geqslant n_0$

$$\frac{\lambda_i}{\tilde{\lambda}_i^{(t)} + \varepsilon} = \frac{\lambda_i}{\lambda_i + \tilde{\lambda}_i^{(t)} - \lambda_i + \varepsilon} < \frac{\lambda_i}{\lambda_i + \varepsilon - \zeta} < 1 + |\varepsilon - \zeta|\lambda_i^{-1}$$

and

$$\frac{\lambda_i}{\tilde{\lambda}_i^{(t)} + \varepsilon} = \frac{\lambda_i}{\lambda_i + \tilde{\lambda}_i^{(t)} - \lambda_i + \varepsilon} > \frac{\lambda_i}{\lambda_i + \varepsilon + \zeta} > 1 - (\varepsilon + \zeta)\lambda_i^{-1}$$

where the final inequality in each case can be derived from Taylor's theorem with Lagrange's form of the remainder [SV14]. Since the λ_i are bounded away from zero, we have established

$$\left|\frac{\lambda_i}{\tilde{\lambda}_i^{(t)} + \varepsilon} - 1\right| < c(\varepsilon + \zeta) \tag{5.65}$$

where the constant $c = 1 + (\min_j\{\lambda_j\})^{-1}$, say. From this bound we can in turn obtain

$$\begin{aligned} & 1 - \alpha(\gamma + 1 + c(\varepsilon + \zeta)) < 1 - \alpha(\gamma + (\tilde{\lambda}_i^{(t)} + \varepsilon)^{-1}\lambda_i) < 1 - \alpha(\gamma + 1 - c(\varepsilon + \zeta)) \\ \implies & 1 - \alpha(1 + c(\varepsilon + \zeta + \gamma)) < 1 - \alpha(\gamma + (\tilde{\lambda}_i^{(t)} + \varepsilon)^{-1}\lambda_i) < 1 - \alpha(1 - c(\varepsilon + \zeta + \gamma)) \end{aligned} \tag{5.66}$$

where the second line exploits the assumption $c(\gamma+\varepsilon+\zeta)<1$ and our choice $c>1$. Thus

$$\begin{aligned}\sum_{t=0}^{n-1}\alpha\frac{\lambda_k}{\tilde{\lambda}_k^{(t)}}\prod_{j=t+1}^{n-1}\left(1-\alpha\gamma-\alpha(\tilde{\lambda}_k^{(j)}+\varepsilon)\lambda_k\right) &< \sum_{t=0}^{n-1}\alpha(1+c(\varepsilon+\zeta))\left(1-\alpha(\gamma+1-c(\varepsilon+\zeta))\right)^{n-1-t}\\ &< 1+c_1(\zeta+\varepsilon+\gamma) \qquad (5.67)\end{aligned}$$

where the second inequality follows, for large n, by summing the geometric series and again using Lagrange's form of the remainder in Taylor's theorem. c_1 is some constant, derived from c that we need not determine explicitly. A complementary lower bound is obtained similarly (for large n). We have thus shown that

$$|\mathbb{E}w_{n,i}-w_i^*|<c_1(\varepsilon+\zeta+\gamma)+\prod_{t=0}^{n-1}\left(1-\alpha\gamma-\alpha(\tilde{\lambda}_i^{(t)}+\varepsilon)^{-1}\lambda_i\right)w_{0,i}. \qquad (5.68)$$

Reusing the bound (5.65) then yields (5.60). The remaining results, (5.61)-(5.63) follow similarly using the same bounds and ideas as above, but applied to the corresponding steps from the proof of Theorem 2. ■

Theorem 5.3 demonstrates the same IA variance reduction as seen previously, but in the more general context of weight decay and adaptive optimisation. As expected, improved estimation of the true Hessian eigenvalues (i.e. smaller ζ) reduces the error in recovery of $\boldsymbol{w}^*$. Moreover, increasing the weight decay strength γ decreases the leading order error bounds in (5.60) and (5.62), but only up to a point, as the other error terms are valid and small only if γ is not too large.

5.4 Conclusion

We have proposed a Gaussian Process perturbation between the batch and true risk surfaces and derive the phenomenon of improved generalisation for large learning rates and larger weight decay when combined with iterate averaging observed in practice. We have extended this formalism to include adaptive methods and showed that we expect further improvement when using adaptive algorithms.

Chapter 6

A random matrix approach to damping in deep learning

The content of this chapter was published first as a pre-print in March 2022 (`https://arxiv.org/abs/2011.08181v5`) and later as a journal article: "A random matrix theory approach to damping in deep learning". Diego Granziol, **Nicholas P Baskerville**. *Journal of Physics: Complexity*, 3.2 (2022): 024001.

DG conceived of the main idea behind this work and published it as a pre-print, along with other collaborators, before **NPB** joined the project. **NPB** introduced the random matrix model and derived the adaptive damping algorithm. **NPB** also overhauled the existing mathematical content, only some of which is included in this chapter. All the experiments in this chapter were actually executed by DG but **NPB** contributed equally to their design, analysis and write-up.

6.1 The Spiked Model for the Hessian of the Loss

We conjecture that a key driver of the adaptive generalisation gap is the fact that adaptive methods fail to account for the greater levels of noise associated with their estimates of flat directions in the loss landscape. The fundamental principle underpinning this conjecture, that sharp directions contain information from the underlying process and that flat directions are largely dominated by noise, is theoretically motivated from the spiked covariance model [BS04]. This model has been successfully applied in Principal Component Analysis (PCA), covariance matrix estimation and finance [Blo+16; ER00; BBP17; BBP16]. We revisit this idea in the context of deep neural network optimisation.

In particular, we consider a spiked additive signal-plus-noise random matrix model for the batch Hessian of deep neural network loss surfaces. In this model, results from random matrix theory suggest several practical implications for adaptive optimisation. We use linear shrinkage theory

[BBP16; Bun+16; BBP17] to illuminate the role of damping in adaptive optimisers and use our insights to construct an adaptive damping scheme that greatly accelerates optimisation. We further demonstrate that typical hyper-parameter settings for adaptive methods produce a systematic bias in favour flat directions in the loss landscape and that the adaptive generalisation gap can be closed by redressing the balance in favour of sharp directions. To track the bias towards flat vs sharp directions we define

$$\mathcal{R}_{\text{est-curv}} := \frac{\alpha_{\text{flat}}}{\alpha_{\text{sharp}}}, \tag{6.1}$$

where α_{flat} and α_{sharp} are the learning rates along the flat and sharp directions, respectively and this ratio encapsulates the noise-to-signal ratio as motivated by our conjecture (the terms *flat* and *sharp* are defined more precisely below).

6.1.1 Sharp directions from the true loss surface survive, others wash out

We can rewrite the (random) batch hessian $\boldsymbol{H}_{\text{batch}}$ as the combination of the (deterministic) true hessian $\boldsymbol{H}_{\text{true}}$ plus some fluctuations matrix:

$$\boldsymbol{H}_{\text{batch}}(\boldsymbol{w}) = \boldsymbol{H}_{\text{true}}(\boldsymbol{w}) + \boldsymbol{X}(\boldsymbol{w}). \tag{6.2}$$

In [GZR20] the authors consider the difference between the batch and empirical Hessian, although this is not of interest for generalisation, the framework can be extended to consider the true Hessian. The authors further show, under the assumptions of Lipschitz loss continuity, almost everywhere double differentiable loss and that the data are drawn i.i.d from the data generating distribution that the elements of $\boldsymbol{X}(\boldsymbol{w})$ converge to normal random variables[1]. Under the assumptions of limited dependence between and limited variation in the variance of the elements of the fluctuations matrix, the spectrum of the fluctuations matrix converges to the Wigner semi-circle law [GZR20; Wig93], i.e. weakly almost surely

$$\frac{1}{P}\sum_{i=1}^{P}\delta_{\lambda_i(\boldsymbol{X})} \to \mu_{SC}, \tag{6.3}$$

where the $\lambda_i(\boldsymbol{X})$ are the eigenvalues of $\boldsymbol{X}$ and $d\mu_{SC}(x) \propto \sqrt{2P^2 - x^2}dx$. The key intuition in this chapter is that sharp directions of the true loss surfaces, that is directions in which the true Hessian has its largest eigenvalues, are more reliably estimated by the batch loss than are the flat directions (those with small Hessian eigenvalues). This intuition is natural in random matrix theory and is supported by results such as the following.

Theorem 6.1. *Let $\{\boldsymbol{\theta}_i\}_{i=1}^P$, $\{\phi\}_{i=1}^P$ be the orthonormal eigenbasis of the true Hessian $\nabla^2 L_{\text{true}}$ and batch Hessian $\nabla^2 L_{\text{batch}}$ respectively. Let also $\nu \geqslant \ldots \geqslant \nu_P$ be the eigenvalues of $\nabla^2 L_{\text{true}}$. Assume that $\nu_i = 0$ for all*

[1] Note that although a given batch Hessian is a fixed deterministic property, we are interested in generic properties of batches drawn at random from the data generating distribution for which we make statements and can hence model the fluctuations matrix as a random matrix.

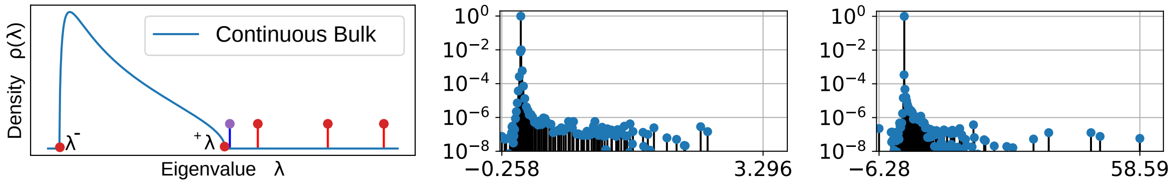
$i > r$, for some fixed r. Assume that $\boldsymbol{X}$ is a generalised Wigner matrix. Then as $P \to \infty$ the following limit holds almost surely

$$|\boldsymbol{\theta}_i^T \phi_i|^2 \to \begin{cases} 1 - \frac{P\sigma^2}{B\nu i^2} & \text{if } |\nu_i| > \sqrt{\frac{P}{B}}\sigma, \\ 0 & \text{otherwise,} \end{cases} \tag{6.4}$$

where σ is the sampling noise per Hessian element.

Proof. This is a direct application of a result of [CD16] which is given more explicitly in the case of GOE Wigner matrices by [BN11]. In particular, we use a scaling of $\boldsymbol{X}$ such that the right edge of the support of its spectral semi-circle is roughly at $P^{1/2}B^{-1/2}\sigma$. The expression in Section 3.1 of [BN11] can then be applied to $P^{-1/2}\boldsymbol{H}_{\text{batch}}$ and re-scaled in $\sqrt{P}$ to give the result. Note that the substantiation of the expression from [BN11] in the case of quite general Wigner matrices is given by Theorem 16 of [CD16]. ■

Results like Theorem 6.1 are available for matrix models other than Wigner, such as rotationally invariant models [Bel+17], and are conjectured to hold for quite general[2] models [BN11]. Convergence of the spectral measure of $P^{-1/2}\boldsymbol{X}$ to the semi-circle is necessary to obtain (6.4), but not sufficient. The technicalities to rigorously prove Theorem 6.1 without assuming a Wigner matrix for $\boldsymbol{X}$ are out of scope for the present work, requiring as they would something like an optimal local semi-circle law for $\boldsymbol{X}$ [EY17b]. We require only the general heuristic principle from random matrix theory encoded in (6.4), namely that *only sharp directions retain information from the true loss surface*. It is expected that this principle will hold for a much wider class of random matrices than those for which it has been rigorously proven. This is acutely important for adaptive methods which rely on curvature estimation, either explicitly for stochastic second order methods or implicitly for adaptive gradient methods.



(a) Hypothetical $\rho(\lambda)$ (b) Val Acc= 94.3, SGD (c) Val Acc= 95.1, Adam

Figure 6.1: (a) Hypothetical spectral density plot with a sharply supported continuous bulk region, a finite size fluctuation shown in blue corresponding to the Tracy-Widom region and three well-separated outliers shown in red. (b,c) VGG-16 Hessian on the CIFAR-10 dataset at epoch 300 for SGD and Adam respectively. Note the "sharper" solution has better validation accuracy.

The spectrum of the noise matrix occupies a continuous region that is sharp in the asymptotic limit [BBP17] known as *bulk* supported between $[\lambda_-, \lambda_+]$ [BBP17; BBP16; Bun+16] and observed in DNNs [Gra+19a; Pap18; Sag+17]. Within this bulk eigenvectors are uniformly distributed on

[2] Roughly speaking, models for which a local law can be established [EY17b].

the unit sphere [BN11] and all information about the original eigenvalue/eigenvector pairs is lost [BAP05]. Hence from a theoretical perspective it makes no sense to estimate these directions and move along them accordingly. An eigenvalue, λ_i, corresponds to a *flat* direction if $\lambda_i \leqslant \lambda_+$. For finite-size samples and network size, there exists a region beyond the predicted asymptotic support of the noise matrix, called the Tracy–Widom region [TW94; El +07], where there may be isolated eigenvalues which are part of the noise matrix spectrum (also shown in Figure 6.1a). The width of the Tracy–Widom region is very much less than that of the bulk. Anything beyond the Tracy–Widom region $\lambda_i \gg \lambda_+$, $\lambda_i \ll \lambda_-$ is considered an outlier and corresponds to a *sharp* direction. *Such directions represent underlying structure from the data*. The eigenvectors corresponding to these eigenvalues can be shown to lie in a cone around their true values [BN11] (see Theorem 6.1). In Figure 6.1b, we show the Hessian of a VGG-16 network at the 300th epoch on CIFAR-100. Here, similar to our hypothetical example, we see a continuous region, followed by a number of eigenvalues which are close to (but not within) the bulk, and finally, several clear outliers.

6.2 Detailed experimental investigation of Hessian directions

In this section we seek to validate our conjecture that movements in the sharp direction of the loss landscape are inherently vital to generalisation by studying a convex non-stochastic example. For such a landscape there is only a single global minimum and hence discussions of bad minima are not pertinent. We implement a second-order optimiser based on the Lanczos iterative algorithm [MS06] (LanczosOPT) against a gradient descent (GD) baseline.

Note on Lanczos The Lanczos algorithm is an iterative algorithm for learning approximations to the eigenvalues/eigenvectors of any Hermitian matrix, requiring only matrix–vector products. The values and vectors learned by Lanczos are known as Ritz values/vectors, which are related to the eigenvalue/eigenvector pairs of the matrix. For example, when using a random vector in the matrix vector product, the Ritz values with a weight given by the first element squared of the corresponding Ritz vector, can be shown to give a moment matched approximation to the spectral density of the underlying matrix. In the same way that the power iteration algorithm converges to the largest eigenvalue (with a rate of convergence depending on the size of the spectral gap $\frac{\lambda_1-\lambda_2}{\lambda_1}$) the Lanczos Ritz values converge to well separated outliers[3]. Similar to the power iteration algorithm, this convergence is irrespective of the original seed vector as long as it is not orthogonal to the associated eigenvectors.

We employ a training set of 1K MNIST [LeC98] examples using logistic regression and validate on a held out test set of 10K examples. Each optimiser is run for 500 epochs. Since the number of well-separated outliers from the spectral bulk is at most the number of classes [Pap18] (which is

[3] Intuitively once the largest outlier has been learned, since Lanczos maintains an orthogonal search space, it converges to the next largest outlier

$n_c = 10$ for this dataset), we expect the Lanczos algorithm to pick out these well-separated outliers when the number of iterations $k \gg n_c$ [Gra+19a; MS06] and therefore use $k = 50$. To investigate the impact of scaling steps in the Krylov subspace given by the sharpest directions, we consider the update $\boldsymbol{w}_{k+1}$ of the form:

$$\boldsymbol{w}_k - \alpha\Big(\frac{1}{\eta}\sum_{i=1}^{k}\frac{1}{\lambda_i+\delta}\phi_i\phi_i^T\nabla L(\boldsymbol{w}_k) + \sum_{i=k+1}^{P}\frac{1}{\delta}\phi_i\phi_i^T\nabla L(\boldsymbol{w}_k)\Big) \tag{6.5}$$

where $P = 7850$ (the number of model parameters) and hence the vast majority of flat directions remain unperturbed. Note that in the case that $k = P = 7850$ we would have a fully second order method, whereas in the case where $k = 0$, by resolution of the identity, we would have gradient descent with learning rate $\frac{\alpha}{\delta}$. Hence equation 6.5 can be seen as scaling the k Ritz eigenvectors by their respective Ritz values, whilst leaving the remaining directions (which by the previous argument are typically the "flatter" directions) unchanged from their gradient descent counterpart. Whilst Equation 6.5 would naively require $\mathcal{O}(P^3)$ operations, i.e a full eigendecomposition, it can in fact equivalently be implemented in the following manner

$$\boldsymbol{w}_k - \alpha\Big(\frac{1}{\eta}\sum_{i=1}^{k}\Big[\frac{1}{\lambda_i+\delta} - \frac{1}{\delta}\Big]\phi_i\phi_i^T\nabla L(\boldsymbol{w}_k) + \frac{1}{\delta}\nabla L(\boldsymbol{w}_k)\Big), \tag{6.6}$$

which requires only k Hessian vector products and hence is of computational complexity $\mathcal{O}(kP)$.

To explore the effect of the sharp directions explicitly as opposed to implicitly, we have introduced perturbations to the optimiser (denoted LOPT[η]), in which we reduce the first term in the parenthesis of Equation 6.5 by a factor of η (we explore scaling factors of 3 and 10). This reduces movement in sharp directions, consequently increases reliance on flat directions during the optimisation trajectory (we increase $\mathcal{R}_{\text{est-curv}}$). This differs from simply increasing δ, which while reducing the movement in all directions, actually relatively increases movement in the sharper directions (decreases $\mathcal{R}_{\text{est-curv}}$). To see this consider the case where $\lambda_i \gg \delta$, in such an instance, increasing δ does not appreciably change movement in the sharp directions, whereas it massively decreases movement in flat directions. For a fixed α, δ controls the $\mathcal{R}_{\text{est-curv}}$.

Experimental Results We show the training and validation curves for various values of damping δ and specific sharpness reduction factor η in Figures 6.2 and 6.3. For ease of exposition we only show curves of adjacent values of damping and in order to focus on the speed of convergence we only show the first 100 epochs of training. We have the full 500 epochs of training, along with all curves colour coded on the same graph in E. We use colours to distinguish δ values and dashing/opacity to indicate η values (dashed is larger than solid, and dashed with lower opacity is larger still). Note that as given by our central hypothesis, increasing δ increases generalisation (we decrease $\mathcal{R}_{\text{est-curv}}$), whereas increasing η decreases generalisation (we increase $\mathcal{R}_{\text{est-curv}}$).

We see in Figure 6.2 that despite an initial instability in training for $\eta = 3, 10$, the red lines with lowest value of damping $\delta = 0.001$, all converge quickly to 0 training error (See E). However the generalisation as measured by the validation error decreases as we increase η. This can be seen as the lighter dashed lines (denoting a decrease in movement in the sharpest directions only) increase in validation error. For the blue lines with $\delta = 0.01$, whilst increasing δ decreases the rate of convergence, $\eta = 3$ attains a final training error of 0, yet differs markedly in validation error for its $\eta = 1$ counterpart. Similarly so the change in validation error for $\eta = 10$ from $\eta = 3$ is much larger than the change in training error. For larger values of δ as shown in Figure 6.3, whilst we see an effect on both training and validation, the effect on validation is much more stark. To show this in an intuitive way, in Figure 6.5, we use a heat map to show the difference from the best training and testing error as a function of δ and η. The best training error was 0 and attained at $\eta = 1, \delta = 0.001$, whereas the best testing error was 0.13 and attained at $\eta = 1, \delta = 1.0$. It is the difference from these values that is shown in Figure 6.5 (so the top left square is 0 for training and similarly the bottom left for testing). As we increase $\mathcal{R}_{\text{est-curv}}$ (by decreasing the value of δ for a fixed α value of 0.01),

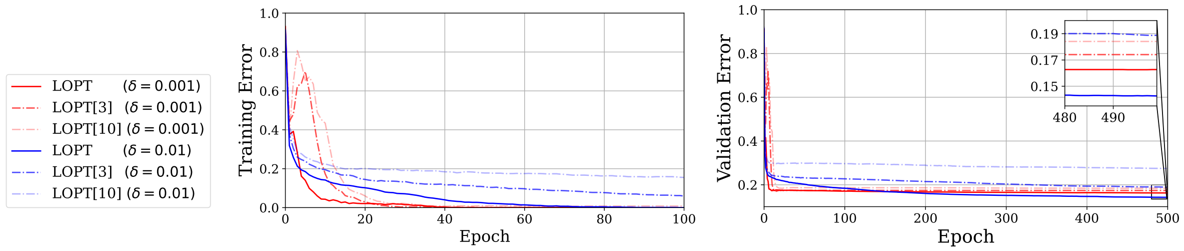


Figure 6.2: Training/test error of LanczosOPT (LOPT) optimisers for logistic regression on the MNIST dataset with fixed learning rate $\alpha = 0.01$ across different damping values, δ. LOPT[η] denotes a modification to the LOPT algorithm that perturbs a subset of update directions by a factor of η. Best viewed in colour.

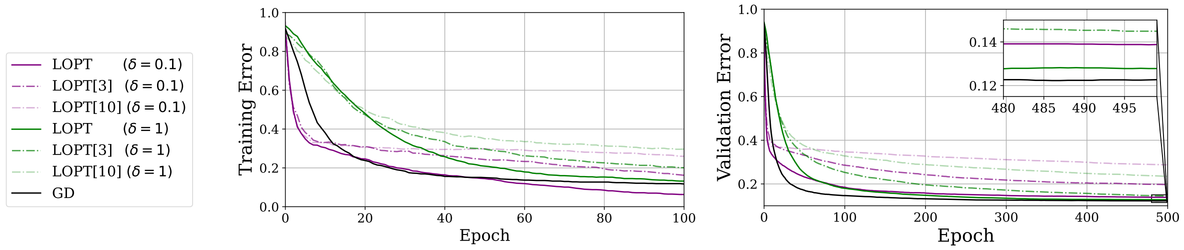
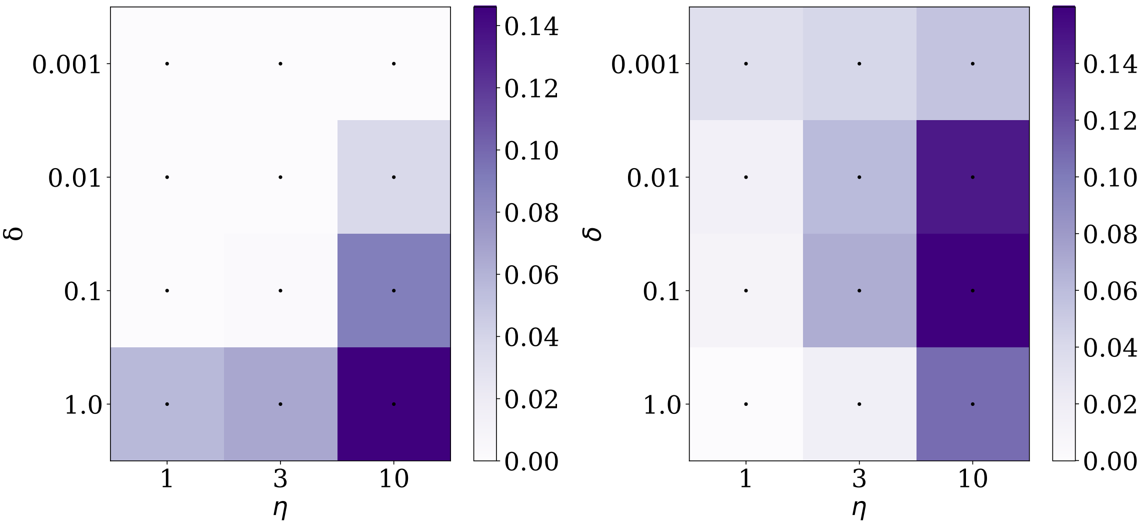


Figure 6.3: Training/test error of LanczosOPT/Gradient Descent (LOPT/GD) optimisers for logistic regression on the MNIST dataset with fixed learning rate $\alpha = 0.01$ across different damping values, δ. LOPT[η] denotes a modification to the LOPT algorithm that perturbs a subset of update directions by a factor of η. Best viewed in colour.

the generalisation of the model suffers correspondingly. For each fixed value of δ, we see clearly that perturbations of greater magnitude cause greater harm to generalisation than training. We also note that for larger values of δ the perturbed optimisers suffer more gravely in terms of the effect on both training and validation. It is of course possible that for such large values of δ we have not converged even after 500 epochs. We show the full training curves in Figure E.1. We observe that



(a) $\Delta(\delta,\eta)$ Training

(b) $\Delta(\delta,\eta)$ Testing

Figure 6.5: Error change with damping/sharp direction perturbation δ,η in LanczosOPT, relative to the single best run. Darker regions indicate higher error. The lowest attained training error (0) and validation error (0.13) are used as reference points of zero.

the generalisation of all algorithms is worsened by explicit limitation of movement in the sharp directions (and an increase of $\mathcal{R}_{\text{est-curv}}$), however for extremely low damping measures (which are typical in adaptive optimiser settings) there is no or very minimal impact in training performance (upper region of Figure 6.5(a). A consequence of this which is already employed in practical machine learning is the use of δ tuning. Essentially using larger than default values of δ (decreasing $\mathcal{R}_{\text{est-curv}}$) so as to not simply avoid problems of numerical stability but also generalise better.

Fashion MNIST: We repeat the experimental procedure for the FashionMNIST dataset [XRV17], which paints an identical picture (at slightly higher testing error) The full training curves are given in Figure E.2.

6.3 The role of damping

Consider a general iterative optimiser that seeks to minimise the scalar loss $L(\boldsymbol{w})$ for a set of model parameters $\boldsymbol{w} \in \mathbb{R}^P$. Recall the $k+1$-th iteration of such an optimiser can be written[4] as follows:

$$\boldsymbol{w}_{k+1} \leftarrow \boldsymbol{w}_k - \alpha_k \boldsymbol{B}^{-1} \nabla L_{\text{batch}}(\boldsymbol{w}_k) \tag{6.7}$$

where α_k is the global learning rate. For SGD, $\boldsymbol{B} = \boldsymbol{I}$ whereas for adaptive methods, $\boldsymbol{B}$ typically comprises some form of approximation to the Hessian i.e. $\boldsymbol{B} \approx \nabla^2 L_{\text{batch}}(\boldsymbol{w}_k)$. Writing this update

[4] Ignoring additional features such as momentum and explicit regularisations.

in the eigenbasis of the Hessian[5] $\nabla^2 L_{\text{batch}}(\boldsymbol{w}_k) = \sum_i^P \lambda_i \phi_i \phi_i^T \in \mathbb{R}^{P\times P}$, where $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_P \geqslant 0$ represent the ordered scalar eigenvalues, the parameter step takes the form:

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \sum_{i=1}^{P} \frac{\alpha}{\lambda_i + \delta} \phi_i \phi_i^T \nabla L_{\text{batch}}(\boldsymbol{w}_k). \tag{6.8}$$

Here, δ is a damping (or numerical stability) term. This damping term (which is typically grid searched [Dau+14] or adapted during training [MG15]) can be interpreted as a trust region [Dau+14] that is required to stop the optimiser moving too far in directions deemed flat ($\lambda_i \approx 0$), known to dominate the spectrum in practice [GZR20; Pap18; GKX19], and hence diverging. In the common adaptive optimiser Adam [KB14], it is set to 10^{-8}. For small values of δ, α must also be small to avoid optimisation instability, hence global learning rates and damping are coupled in adaptive optimisers.

6.3.1 Adaptive updates and damping

The learning rate in the flattest ($\lambda \approx 0$) directions is approximately $\frac{\alpha}{\delta}$, which is larger than the learning rate in the sharpest ($\lambda_i \gg \delta$) directions $\frac{\alpha}{\delta+\lambda_i}$. This difference in per direction effective learning rate makes the best possible (damped) training loss reduction under the assumption that the loss function can be effectively modelled by a quadratic [Mar16b]. Crucially, however, it is agnostic to how accurately each eigenvector component of the update estimates the true underlying loss surface, which is described in Theorem 6.1. Assuming that the smallest eigenvalue $\lambda_P \ll \delta$, we see that $\mathcal{R}_{\text{est-curv}} = 1 + \frac{\lambda_1 - \lambda_P}{\delta}$. This is in contrast to SGD where $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \sum_{i=1}^P \alpha \phi_i \phi_i^T \nabla L_{\text{batch}}(\boldsymbol{w}_k)$ and hence $\mathcal{R}_{\text{est-curv}} = 1$. Note that we can ignore the effect of the overlap between the gradient and the eigenvectors of the batch Hessian because we can rewrite the SGD update in the basis of the batch Hessian eigenvectors and hence reduce the problem to one of the relative learning rates.

The crucial point to note here is that the difference in $\mathcal{R}_{\text{est-curv}}$ is primarily controlled by the damping parameter: smaller values yield a larger $\mathcal{R}_{\text{est-curv}}$, skewing the parameter updates towards flatter directions.

To further explore our central conjecture for modern deep learning architectures (where a large number of matrix–vector products is infeasible) we employ the KFAC [MG15] and Adam [KB14] optimisers on the VGG-16 [SZ14] network on the CIFAR-100 [KH+09] dataset. The VGG-16 allows us to isolate the effect of $\mathcal{R}_{\text{est-curv}}$, as opposed to the effect of different regularisation implementations for adaptive and non-adaptive methods as discussed by [LH18; Zha+18a].

6.3.2 VGG16: a laboratory for adaptive optimisation

The deep learning literature contains very many architectural variants of deep neural networks and a large number of engineering “tricks” which are employed to obtain state of the art results on a great variety of different tasks. The theory supporting the efficacy of such tricks and architectural designs is often wanting and sometimes entirely absent. Our primary objective in this work is to illuminate

[5] We assume this to be positive definite or that we are working with a positive definite approximation thereof.

some theoretical aspects of adaptive optimisers such as appropriate damping and Hessian estimation, so we require a simple and clean experimental environment free from, where possible, interference from as many different competing effects. To this end, the VGG architecture [SZ14] for computer vision is particularly appropriate. With 16 layers, the VGG has over 16 million parameters and is capable of achieving competitive test error on a variety of standard computer vision datasets while being trained without batch normalisation [IS15] or weight decay. Indeed, features such as weight decay and batch normalisation obscure the effect of learning rate and damping, meaning that even quite poor choices can ultimately give reasonable results given sufficient training iterations[GZR20]. In contrast the VGG clearly exposes the effects of learning rate and damping, with training being liable to fail completely or diverge if inappropriate values are used. Furthermore as shown in [GZR20] the VGG is highly unstable if too large a learning rate is used. This allows us to very explicitly test whether amendments provided by theory are helpful in certain contexts, such as training stability, as unstable training very quickly leads to divergence.

Learning Rate Schedule For all experiments unless specified, we use the following learning rate schedule for the learning rate at the t-th epoch:

$$\alpha_t = \begin{cases} \alpha_0, & \text{if } \frac{t}{T} \leqslant 0.5 \\ \alpha_0[1 - \frac{(1-r)(\frac{t}{T}-0.5)}{0.4}] & \text{if } 0.5 < \frac{t}{T} \leqslant 0.9 \\ \alpha_0 r, & \text{otherwise} \end{cases} \tag{6.9}$$

where α_0 is the initial learning rate. T is the total number of epochs budgeted for all CIFAR experiments. We set $r = 0.01$ for all experiments.

	α	
δ	0.0004	0.001
1e-7	**53.1**(62.9)	
4e-4	**21.1**(64.5)	
1e-3	**9.9**(63.5)	**20.8**(64.4)
5e-3		**9.1**(66.2)
8e-3		**2.4**(65.8)

	α		
δ	0.1	0.001	0.0001
1e-1	**6.7**(65.0)		
1e-2		**20.8**(64.8)	
1e-3			**48.2**(62.2)
3e-4			**527.2**(60.2)
1e-4			**711.3**(56.0)

Table 6.1: **Spectral norms and generalisation**. We report the spectral norm λ_1 at the end of training in **bold**, with corresponding validation accuracy (in parentheses) for learning rate/damping α, δ using Adam (left) and KFAC (right) to train a VGG-16 network on CIFAR-100.

6.3.3 KFAC with VGG-16 on CIFAR-100:

By decreasing the global learning rate α whilst keeping the damping-to-learning-rate ratio $\kappa = \frac{\delta}{\alpha}$ constant, we increase the $\mathcal{R}_{\text{est-curv}}$, $\mathcal{R}_{\text{est-curv}}$, which is determined by $\frac{\lambda_i}{\kappa\alpha} + 1$. As shown in Tab. 6.1

and in Figure 6.6 we observe that as we increase $\mathcal{R}_{\text{est-curv}}$ the training performance is effectively unchanged, but generalisation suffers ($35\% \rightarrow 37.8\%$). Whilst decreasing the damping results in poor training for large learning rates, for very low learning rates the network efficiently trains with a lower damping coefficient. Such regimes further increase $\mathcal{R}_{\text{est-curv}}$ and we observe that they generalise more poorly. For $\alpha = 0.0001$ dropping the damping coefficient δ from 0.0003 to 0.0001 drops the generalisation further to 60.2% and then 56% respectively. Similar to logistic regression, for both cases the drop in generalisation is significantly larger than the drop in training accuracy.

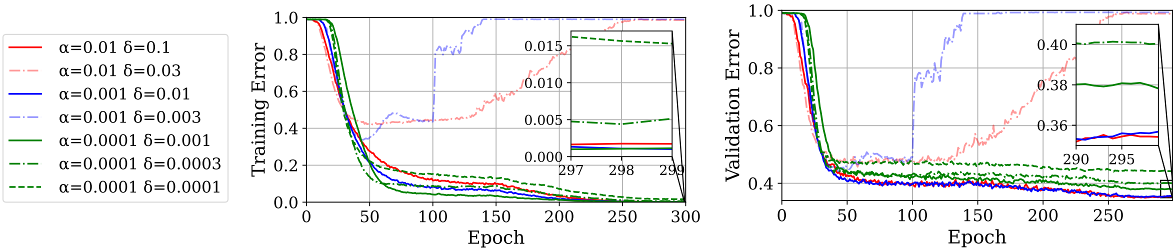


Figure 6.6: Training/validation error of the KFAC optimiser for VGG-16 on the CIFAR-100 dataset with various learning rates α and damping values, δ. The same colour denotes the same learning rate, increasing levels of dashedness denote an ever decreasing damping value.

Adam with VGG-16 on CIFAR-100: We employ Adam with a variety of learning rate and damping coefficients with results as shown in Tab. 6.1 and in Figure 6.7 and compare against a baseline SGD with $\alpha = 0.01$ (corresponding to optimal performance). For the largest learning rate with which Adam trains ($\alpha = 0.0004$) with the standard damping coefficient $\delta = 10^{-8}$, we see that Adam under-performs SGD, but that this gap is reduced by simply increasing the damping coefficient without harming performance. Over-damping decreases the performance. For larger global learning rates enabled by a significantly larger than default damping parameter, when the damping is set too low, the training is unstable (corresponding to the dotted lines). Nevertheless, many of these curves with poor training out-perform the traditional setting on testing. We find that for larger damping coefficients $\delta = 0.005, 0.0075$ Adam is able to match or even beat the SGD baseline, whilst converging faster. We show that this effect is statistically significant in Tab. 6.2. This provides further evidence that for real problems of interest, adaptive methods are not worse than their non-adaptive counterparts as argued by [Wil+17]. We note as shown in Tab. 6.1, that whilst increasing δ always leads to smaller spectral norm, this does not always coincide with better generalisation performance. We extend this experimental setup to include both batch normalisation [IS15] and decoupled weight decay [LH18]. We use a learning rate of 0.001 and a decoupled weight decay of $[0, 0.25]$. For this experiment using a larger damping constant slightly assists training and improves generalisation, both with and without weight decay.

ResNet-50 ImageNet. As shown in Figure 6.9a,6.9b, these procedures have practical impact on large scale problems. Here we show that under a typical 90 epoch ImageNet setup [He+16], with

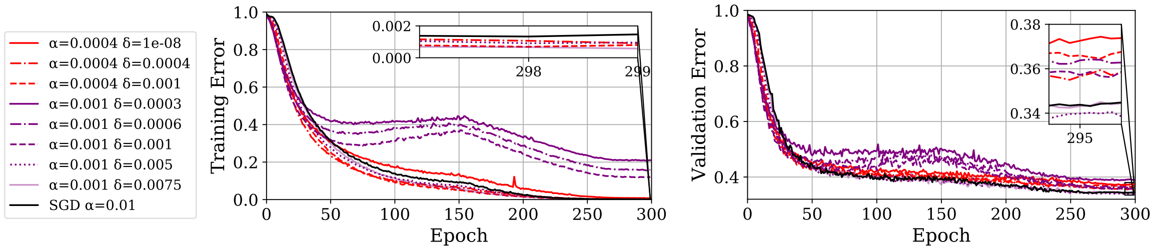


Figure 6.7: Training/validation error of the Adam optimiser for VGG-16 on the CIFAR-100 dataset with various learning rates α and damping values, δ. The same colour denotes the same learning rate, increasing levels of dashedness denote an ever increasing damping value.

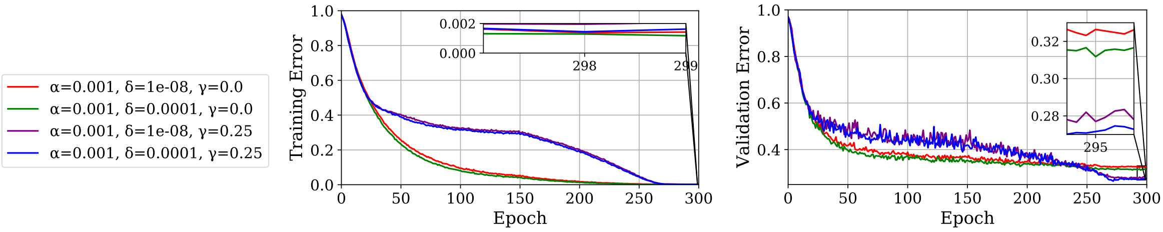
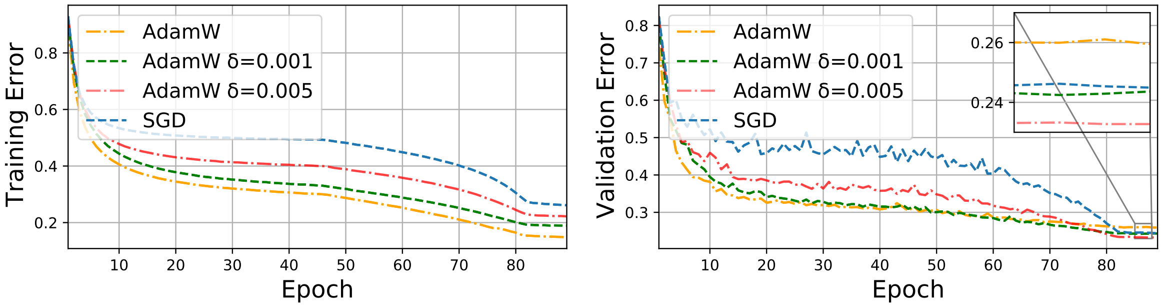


Figure 6.8: Training/validation error of the Adam optimiser for VGG-16BN using Batch Normalisation and Decoupled Weight Decay on the CIFAR-100 dataset with various learning rates α and damping values, δ.



(a) ResNet-50 Training Error

(b) ResNet-50 Testing Error

Figure 6.9: (a-b) The influence of δ on the generalisation gap. Train/Val curves for ResNet-50 on ImageNet. The generalisation gap is completely closed with an appropriate choice of δ.

decoupled weight decay 0.01 for AdamW and 0.0001 for SGD, that by increasing the numerical stability constant δ the generalisation performance can match and even surpass that of SGD, which is considered state-of-the-art and beats AdamW without δ tuning by a significant margin.

Dataset	**Classes**	**Model Architecture**	**SGD**	**Adam-D**	**Adam**
CIFAR-100	100	VGG16	65.3 ± 0.6	65.5 ± 0.7	61.9 ± 0.4
ImageNet	1000	ResNet50	75.7 ± 0.1	76.6 ± 0.1	74.04*

Table 6.2: Statistical Significance. Comparison of test accuracy across CIFAR 100 (5 seeds) and ImageNet (3 seeds). **Adam-D** denotes Adam with increased damping ($\delta = 5e^{-3}$ for CIFAR-100, $\delta = 1e^{-4}$ for ImageNet). *Since it is well established that Vanilla Adam does not generalise well for ImageNet, we do not run this experiment for multiple seeds, we simply report a single seed result for completeness. A more complete discussion for Adam and its generalisation in vanilla form can be found in [GWR20].

6.4 Optimal adaptive damping from random matrix theory

Recall the scaling applied in the direction of the i^{th} eigenvector in (6.8). We make the following observation

$$\frac{1}{\lambda_i + \delta} = \frac{1}{\beta\lambda_i + (1-\beta)} \cdot \frac{1}{\kappa} \tag{6.10}$$

where $\kappa = \beta^{-1}, \beta = (1+\delta)^{-1}$. Hence, using a damping δ is formally equivalent to applying linear shrinkage with factor $\beta = (1+\delta)^{-1}$ to the estimated Hessian and using a learning rate of $\alpha\beta$. Shrinkage estimators are widely used in finance and data science, with linear shrinkage being a common simple method applied to improve covariance matrix estimation [LW04]. The practice of shrinking the eigenvalues while leaving the eigenvectors unchanged is well-established in the fields of sparse component analysis and finance [BBP17]. In the shrinkage literature, the typically considered models are additive and multiplicative [PB20], i.e.

$$\boldsymbol{E} = \boldsymbol{C} + \boldsymbol{X}, \quad \boldsymbol{E} = \boldsymbol{C}^{1/2}\boldsymbol{X}\boldsymbol{C}^{1/2}$$

where $\boldsymbol{E}$ is the observed matrix, $\boldsymbol{C}$ is the non-corrupted (or signal) matrix, and $\boldsymbol{X}$ is the noise matrix. White Wishart $\boldsymbol{X}$ is the simplest example in the multiplicative case, and Wigner matrices are the simplest choice in the additive case. In generality, shrinkage estimators are estimators of $\boldsymbol{C}$ given $\boldsymbol{E}$, and it is common to consider rotationally invariant (or, more precisely, equivariant) estimators which reduce the problem to computing the eigenvalues and eigenvectors of $\boldsymbol{E}$ and then correcting, or *shrinking*, the eigenvalues while keeping the eigenvectors fixed to obtain improved estimation of $\boldsymbol{C}$. Optimal[6] estimators are constructed in [Bun+16; LP; LW12] and most recently [LW20]. We note in passing that such estimators are only possible in the large matrix limit, where functions of the inaccessible matrix $\boldsymbol{C}$ can be replaced by equivalent quantities depending only on $\boldsymbol{E}$. The optimal shrinkage estimators are generally non-linear functions of the eigenvalues of $\boldsymbol{E}$ and depend on integral transforms of the limiting spectral measure of $\boldsymbol{E}$ and also on the

[6] Optimality is commonly defined in terms of Frobenius norm, but some authors have considered the *minimum variance* loss [LW20].

noise matrix $\boldsymbol{X}$. In some very special cases, the optimal shrinkage estimators simplify greatly, for example, in the multiplicative case, if $\boldsymbol{C}$ is an inverse Wishart matrix, the linear shrinkage estimator $\tilde{\boldsymbol{H}} = \beta\boldsymbol{H} + (1-\beta)\boldsymbol{I} = \operatorname{argmin}_{\boldsymbol{H}^*} ||\boldsymbol{H}^* - \boldsymbol{H}_{\text{true}}||_2$ is optimal and an explicit expression for the optimal β is found depending only on the dimensionality of the model and the noise variance [LW04; Bun+16].

In our optimisation context, the additive noise model is perhaps the most natural with $\boldsymbol{C}$ being the true loss Hessian and $\boldsymbol{E}$ the batch loss Hessian, however we cannot expect any special forms on $\boldsymbol{X}$ or $\boldsymbol{C}$ that will produce closed form expressions for the optimal rotational invariant estimator and the linear shrinkage estimator is almost certainly not optimal. We suggest that there is no particular reason to break with rotational invariance in this work, as intuitively any distinguished directions of H_{batch} are those of H_{true}. However linear shrinkage has the great advantage of being simple to integrate into existing adaptive optimisers and it acts intuitively to reduce the movement of the optimiser in pure-noise directions. In fact, it is known that general non-linear shrinkage estimators retain the property of increasing the smallest eigenvalues and decreasing the largest [LW12]. Our interpretation reveals that the damping parameter should not be viewed as a mere numerical convenience to mollify the effect of very small estimate eigenvalues, but rather that an optimal δ should be expected, representing the best linear approximation to the true Hessian and an optimal balancing of variance (the empirical Hessian) and bias (the identity matrix). This optimal choice of δ will produce an optimiser that more accurately descends the directions of the true loss.

The linear shrinkage interpretation given by (6.10) is an elementary algebraic relation but does not by itself establish any meaningful link between damping of adaptive optimisers and linear shrinkage estimators. To that end, we return to the random matrix model (6.2) for the estimated Hessian: Let us write the Hessian as

$$\boldsymbol{H}_{\text{batch}} = \boldsymbol{H}_{\text{true}} + \boldsymbol{X}$$

where $\boldsymbol{X}$ is a random matrix with $\mathbb{E}\boldsymbol{X} = 0$. Note that this model is entirely general, we have simply defined $\boldsymbol{X} = \boldsymbol{H}_{\text{batch}} - \mathbb{E}\boldsymbol{H}_{\text{batch}}$ and $\mathbb{E}\boldsymbol{H}_{\text{batch}} = \boldsymbol{H}_{\text{true}}$. We then seek a linear shrinkage estimator $\tilde{\boldsymbol{H}}(\beta) = \beta\boldsymbol{H}_{\text{batch}} + (1-\beta)\boldsymbol{I}$ such that $E(\beta) = P^{-1}\text{Tr}(\tilde{\boldsymbol{H}} - \boldsymbol{H}_{\text{true}})^2$ is minimised. Note that this is the same objective optimised by [Bun+16] to obtain optimal estimators for various models. In this context, we are not finding the optimal estimator for $\boldsymbol{H}_{\text{true}}$ but rather the optimal *linear shrinkage* estimator. We have

$$E(\beta) = \frac{1}{P}\text{Tr}\left[(\beta-1)\boldsymbol{H}_{\text{true}} + \beta\boldsymbol{X} + (1-\beta)\boldsymbol{I}\right]^2 \equiv \frac{1}{P}\text{Tr}\left[(\beta-1)\boldsymbol{H}_{\text{true}} + \boldsymbol{Y}_\beta\right]^2$$

where $\boldsymbol{Y}_\beta = \beta\boldsymbol{X} + (1-\beta)\boldsymbol{I}$.

A natural assumption in the case of deep learning is that $\boldsymbol{H}_{\text{true}}$ is low-rank, i.e. for $P \to \infty$ either $\text{rank}(\boldsymbol{H}_{\text{true}}) = r$ is fixed or $\text{rank}(\boldsymbol{H}_{\text{true}}) = o(P)$. Empirical evidence for this assumption is found in [GZR20; SBL16; Sag+17; Pap18; GKX19]. In this case the bulk of the spectrum of $\boldsymbol{Y}_\beta$ is the same as that of $(\beta-1)\boldsymbol{H}_{\text{true}} + \boldsymbol{Y}_\beta$ [BN11; CD16; Bel+17]. We will also assume that $\boldsymbol{X}$ admits a deterministic limiting spectral measure μ_X such that

$$\frac{1}{P}\sum_{j=1}^{P}\delta_{\lambda(X)_i} \to \mu \tag{6.11}$$

weakly almost surely. Say $\omega_X(x)dx = d\mu(x)$. Then $\boldsymbol{Y}_\beta$ has limiting spectral density

$$\omega_Y(y) = \beta^{-1}\omega_X(\beta^{-1}(y-1+\beta)).$$

Then for large P

$$\begin{aligned} E(\beta) \approx \beta^{-1}\int y^2\omega_X(\beta^{-1}(y-1+\beta))\ dy = \int(\beta x + 1 - \beta)^2\omega_X(x)\ dx \\ = \beta^2\mu_X(x^2) + (1-\beta)^2 \end{aligned}$$

as the centred assumption on $\boldsymbol{X}$ means that $\int x\omega_X(x)\ dx = 0$. $\mu_X(x^2)$ is shorthand for $\int x^2\omega_X(x)\ dx$. $E(\beta)$ is thus minimised to leading order at $\beta = (1+\mu_X(x^2))^{-1}$. Recalling that $\beta^{-1} = (1+\delta)^{-1}$, this yields $\delta = \mu_X(x^2)$ i.e. the optimal level of damping at large finite P is approximately

$$\delta = P^{-1}\mathrm{Tr}\boldsymbol{X}^2. \tag{6.12}$$

Note that the value (6.12) is a very natural measure of the Hessian noise variance. Therefore if the random matrix model described above is appropriate and the linear shrinkage interpretation (6.10) is meaningful we should expect it to result in close to optimal performance of a given adaptive optimiser. The purpose of adaptive optimisers is to accelerate training, in part by allowing for larger stable learning rates. As discussed throughout this chapter, such optimisation speed often comes at the cost of degraded generalisation. In this context, 'optimal performance' of adaptive optimisers should be taken to mean fast training and good generalisation. As we have discussed above, very large values of δ recover simple non-adaptive SGD, so using (6.12) we should be able to obtain generalisation performance at least as good as SGD and faster optimisation than any choice of δ including the default very small values often used and the larger values considered in Section 6.3.

The value of (6.12) can be easily learned by estimating the variance of the Hessian. The Hessian itself cannot be computed exactly, as it is far too large for $P \geqslant O(10^7)$, however one can compute $\boldsymbol{Hv}$ (and hence $\boldsymbol{H}^2\boldsymbol{v}$) for any vector $\boldsymbol{v}$, using $\nabla^2 L\boldsymbol{v} = \nabla(\boldsymbol{v}^T\nabla L)$. The full approach is given in Algorithm 1.

Algorithm 1 Algorithm to estimate the Hessian variance

```
Input: Sample Hessians $\boldsymbol{H}_i \in \mathbb{R}^{P\times P}$, $1 \leqslant i < N$
Output: Hessian Variance $\sigma^2$
$\boldsymbol{v} \in \mathbb{R}^{1\times P} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
Initialise $\sigma^2 = 0, i = 0, \boldsymbol{v} \leftarrow \boldsymbol{v}/||\boldsymbol{v}||$
for $i < N$ do
    $\sigma^2 \leftarrow \sigma^2 + \boldsymbol{v}^T\boldsymbol{H}_i^2\boldsymbol{v}$
    $i \leftarrow i + 1$
end for
$\sigma^2 \leftarrow \sigma^2 - [\boldsymbol{v}^T(1/N\sum_{j=1}^{N}\boldsymbol{H}_j)\boldsymbol{v}]^2$
```

Extension to non-linear shrinkage. If, as we demonstrate below, our interpretation of damping as linear shrinkage is meaningful, it is natural to ask if we can replace linear shrinkage with more general non-linear shrinkage, effectively defining new adaptive optimisers that replace $\lambda_i + \delta$ in (6.8) by $f(\lambda_i)$ for some non-linear f. Indeed, non-linear shrinkage is known to outperform linear shrinkage in general [LW12; LW20], so we should expect to see further improvements beyond our optimal damping approach, but there are substantial obstacles to progress in this direction. Absent the strongly simplifying assumptions that lead to linear shrinkage, one must handle integral transforms of the spectral density of $\boldsymbol{H}_{\text{batch}}$ to compute general non-linear shrinkage estimators. There are various approaches sin the literature that make use of parametric and kernel estimation fits to these transforms or the spectral density itself [PB20; LW12; LW20] and there are simpler approaches that use cross-validation to construct improved estimators of the true eigenvalues-eigenvector pairs [ADŽ14]. It is, however, observed by Ledoit and Wolf [LW20] that these methods are infeasible for matrices larger than around 1000×1000. Ledoit and Wolf [LW20] propose a new, non-parametric non-linear shrinkage estimator that is quite conceptually simple to implement and can scale to larger matrices, but careful inspection reveals that the required computation time for each shrinkage evaluation is nevertheless $O(P^2)$, where in our case P is on the order of 10^7, so even this approach is infeasible.

6.4.1 Experimental Design and Implementation Details

In order to test our hypothesis for the derived optimal δ (6.12), we run the classical VGG network [SZ14] with 16 layers on the CIFAR-100 dataset, without weight decay or batch normalisation. This gives us maximal sensitivity to the choice of learning rate and appropriate damping.

Now in practice the damping coefficient is typically grid searched over several runs [Dau+14] or there are heuristics such as the Levenberg–Marquardt to adapt the damping coefficient [MG15], which however we find does not give stable training for the VGG. We hence compare against a fixed set damping value δ and a learned damping value as given by our equation (6.12). We find that the variance of the Hessian (6.12) at a random point in weight space (such as at initialisation) or once network divergence has occurred is zero, hence the initial starting value cannot be learned as, with a damping of near zero, the network entirely fails to train (no change in training loss from random). This is to be expected, as in this case the local quadratic approximation to the loss inherent in adaptive methods breaks down. Hence we initialise the learning algorithm with some starting value δ^*, which is then updated every 100 training iterations using equation (6.12). Strictly speaking we should update every iteration, but the value of 100 is chosen arbitrarily as a computational efficiency. Since we are using the variance of the Hessian, which is expensive to compute compared to a simple gradient calculation, we do not want to compute this quantity too often if it can be helped. We run our experiments on a logarithmic grid search in near factors of 3. So learning rates and damping rates, either flat or learned are on the grid of $0.0001, 0.0003, 0.001....$

We find under this setup that the time taken per epoch against the flat damping schedule is only doubled. We get identical results for using a damping gap of 10 and so do not consider this to be a very relevant hyper-parameter. We further calculate the variance of the Hessian over a sub-sample of 10000 examples and do not calculate the variance sample by sample, but over batches of 128 to speed up the implementation. Under the assumption that the data is drawn i.i.d from the dataset the variance is simply reduced by a factor $(\frac{1}{B}-\frac{1}{N}) \approx \frac{1}{B}$ for a small batch size. We do not consider the impact of using only a sub-sample of the data for estimation, but we expect similar results to hold compared to the entire dataset as long as the sub-sample size $S \gg B$. This should allow such a method to be used even for very large datasets, such as ImageNet (with 1-million images), for which a pass of the entire dataset is extremely costly. In theory the sub-sample size and mini-batch size for Hessian variance estimation could be two hyper-parameters which are tuned by considering the effect of reduction on training set or validation set loss metrics with the trade off for computational cost. We do not conduct such analysis here.

We also incorporate an exponential moving average into the learned damping with a co-efficient of 0.7[7] to increase the stability of the learned damping.

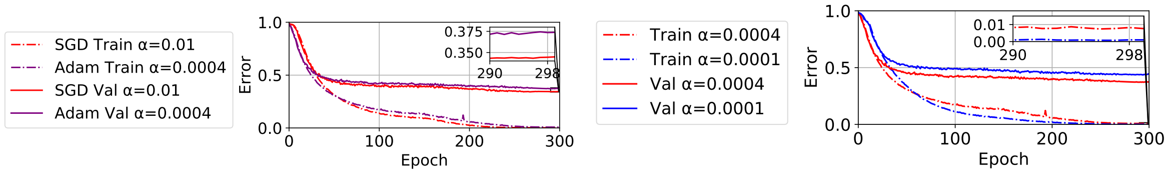
6.4.2 Experiment on CIFAR-100 using KFAC to validate the optimal linear shrinkage

For large damping values δ we simply revert to SGD with learning rate α/δ, so we follow the typical practice of second order methods and use a small learning rate and correspondingly small damping coefficient. However as shown in Figure 6.10 the generalisation and optimisation are heavily dependent on the global learning rate, with larger learning rates often optimising less well but generalising better and vice versa for smaller learning rates. We hence investigate the impact of our damping learner on learning rates one order of magnitude apart. Where in the very low learning rate regime, we show that our method achieves significantly improved training stability with low starting damping and fast convergence and for the large learning rate regime that we even exceed the SGD validation set result.

6.5 Previous Work

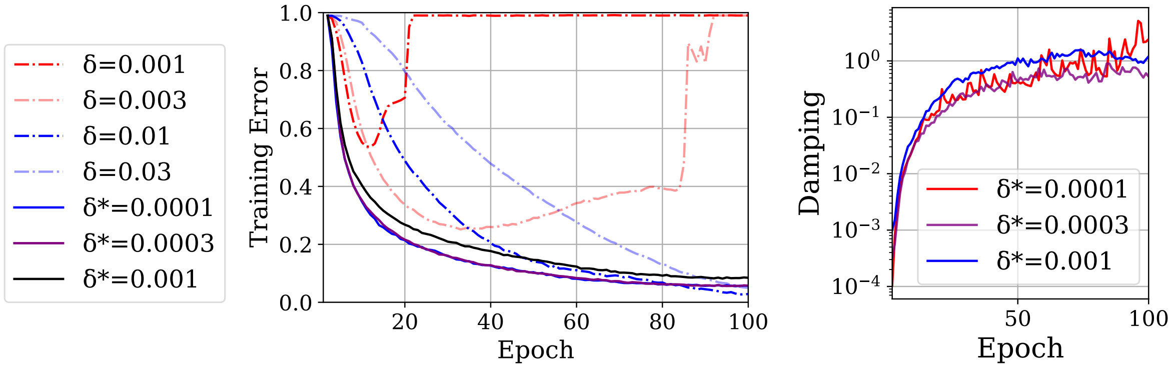
Training KFAC with Auto-Damping: We show the results for a global learning rate of 0.0001 in Figure 6.11a. We see that for the flat damping methods with low values of damping, that training becomes unstable and diverges, despite an initially fast start. Higher damped methods converge, but slowly. In stark contrast, our adaptive damping method is relatively insensitive to their chosen initial values. We show here $\delta^* = \alpha, 3\alpha, 10\alpha$ and all converge and moreover significantly faster than all flat damping methods. The smaller initial damping coefficients $\delta = \alpha, 3\alpha$ converge faster than the larger and, interestingly, follow very similar damping trajectories throughout until the very end of training, as shown in Figure 6.11b.

[7] This value is not tuned and in fact from our plots it may be advisable to consider higher values for greater stability



(a) SGD quickly outgeneralises Adam (b) Adam Train/Val for Learning Rates $\{\alpha_i\}$

Figure 6.10: **Adaptive Generalisation Gap and its extent are clearly visible without regularisation.** Train/Val Error on CIFAR-100 using VGG-16 without batch normalisation and weight decay.

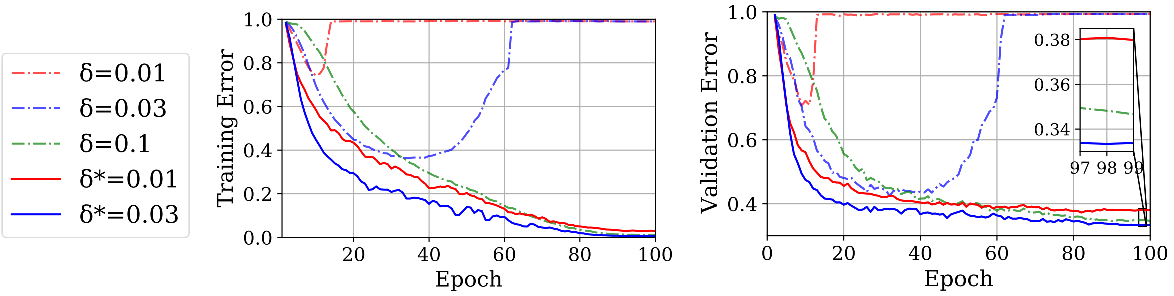


(a) Training Error as a function of Epoch (b) Damping per Epoch

Figure 6.11: VGG-16 on CIFAR-100 dataset using the KFAC optimiser with $\gamma = 0$ (no weight decay) for a learning rate of $\alpha = 0.0001$, batch size $B = 128$ and damping set by δ. For adaptive damping methods the damping is given an initial floor value of δ^* and is then updated using the variance of the Hessian every 100 steps.

Getting Great Generalisation with KFAC and Auto-Damping: We similarly train KFAC on the VGG-16 with a larger learning rate of 0.001, in order to achieve better generalisation. Here we see in Figure 6.12a that relatively low values of flat damping such as 0.01 and 0.03 very quickly diverge, whereas a large value of 0.1 converges slowly to a reasonable test error. The corresponding learned damping curves of 0.01 and 0.03 however converge quickly and the 0.03 initialised damping curve even beats the generalisation performance of the large flat damped version and the test result of SGD on 3x as many training epochs.

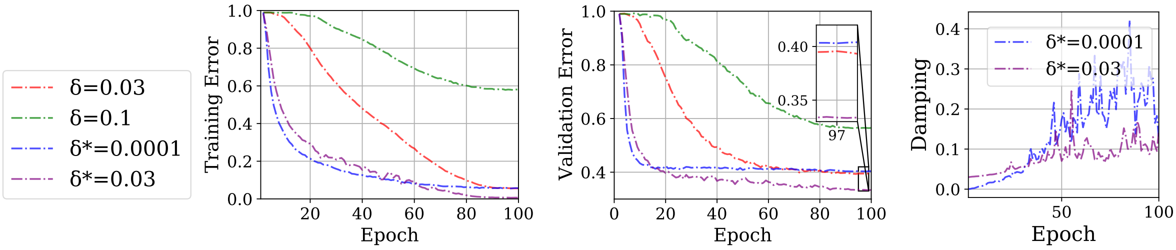
A further look at the value of adaptive damping To elucidate the impact and workings of the adaptive damping further, we consider a select set of curves the learning rate of $\alpha = 0.0001$, shown in Figure 6.13. here we see that starting with an initial damping of $\delta = \alpha$, the adaptive method reaches a comparable generalisation score to the flat damping of $\delta = 0.03$ but at a much faster convergence rate. The initial damping of $\delta = 0.03$ converges not quite as quickly but trains and generalises better than its lower starting damping counterpart. Note from Figure 6.13c that even though the damping of this curve reaches ≈ 0.1 that starting with a flat damping of 0.1 never achieves a comparable generalisation (or even trains well). This implies as expected that it is important to adjust damping



(a) Training Error

(b) Val Error

Figure 6.12: VGG-16 on CIFAR-100 dataset using the KFAC optimiser with $\gamma = 0$ (no weight decay) for a learning rate of $\alpha = 0.001$, batch size $B = 128$ and damping set by δ. For adaptive damping methods the damping is given an initial floor value of δ^* and is then updated using the variance of the Hessian every 100 steps.



(a) Training Error

(b) Val Error

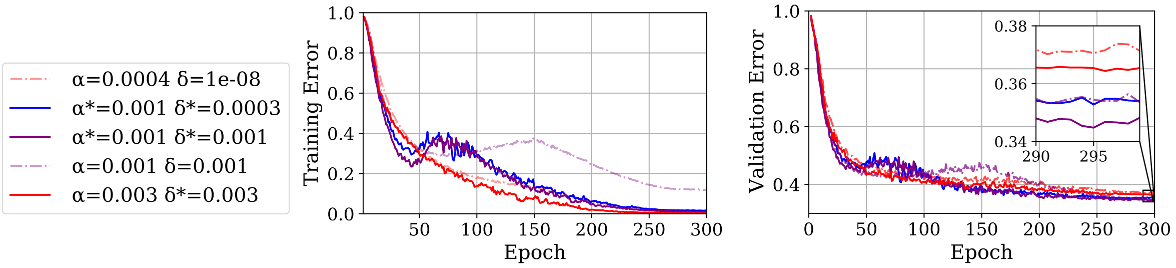
(c) Damping

Figure 6.13: VGG-16 on CIFAR-100 dataset using the KFAC optimiser with $\gamma = 0$ (no weight decay) for a learning rate of $\alpha = 0.0001$, batch size $B = 128$ and damping set by δ. For adaptive damping methods the damping is given an initial floor value of δ^* and is then updated using the variance of the Hessian every 100 steps.

during training.

6.5.1 Adam with Auto-Damping

Given that Adam does not employ an obvious curvature matrix, it is curious to consider whether our learned damping estimator can be of practical value for this optimiser. As discussed in the previous section, Adam's implied curvature can be considered a diagonal approximation to the square root of the gradient covariance. The covariance of the gradients has been investigated to have similarities to the Hessian [Jas+20]. However the nature of the square root, derived from the regret bound in [DHS11] presents an interesting dilemma. In the case of very very small eigenvalues of $\boldsymbol{B}$, the square root actually reduces their impact on the optimisation trajectory, hence it is very plausible that the learned damping could be too harsh (as it is expected to work optimally for the eigenvalues of $\boldsymbol{H}$ and not $\sqrt{\boldsymbol{H}}$). This is actually exactly what we see in Figure 6.14. Whilst an increase in learning rate and damping, along with auto-damping improves both the convergence and validation result over the standard baseline (where the damping is kept at the default value and maximal learning



(a) Training Error

(b) Val Error

Figure 6.14: VGG-16 on CIFAR-100 dataset using the Adam optimiser with $\gamma = 0$ (no weight decay) for a learning rate of α, batch size $B = 128$ and damping set by δ. For adaptive damping methods the damping is given an initial floor value of δ^* and is then updated using the variance of the Hessian every 100 steps. α^* refers to the use of an alternative ramp up schedule where the base learning rate is increased by a factor of 5 at the start of training before being decreased.

rate is found which stably trains) the improvements are small and do not make up the gap with SGD. More specifically they are not better than just using a larger learning rate in combination with a larger flat damping, defeating the purpose of learning the damping factor online.

To alleviate the effect of overly harsh damping, we consider an alternate learning rate schedule where the base learning rate is increased by a factor of 5 early in training and then subsequently decreased. The constant 5 is not tuned but simply a place-holder to consider a more aggressive learning rate schedule to counter-act the effect of the damping learner. These curves are marked with α^* in Figure 6.14.

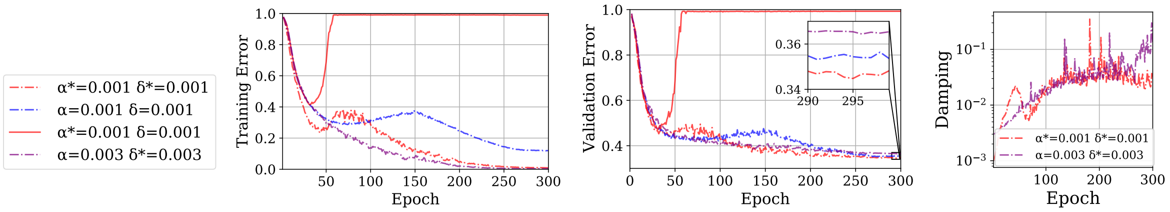
Warm up Learning Rate Schedule For all experiments unless specified, we use the following learning rate schedule for the learning rate at the t-th epoch:

$$\alpha_t = \begin{cases} \alpha_0, & \text{if } \frac{t}{T} \leqslant 0.1 \\ \alpha_0[1 + \frac{(\kappa-1)(\frac{t}{T}-0.1)}{0.2}, & \text{if } \frac{t}{T} \leqslant 0.3 \\ \alpha_0[\kappa - \frac{(\kappa-r)(\frac{t}{T}-0.3)}{0.6}] & \text{if } 0.3 < \frac{t}{T} \leqslant 0.9 \\ \alpha_0 r, & \text{otherwise} \end{cases} \tag{6.13}$$

where α_0 is the initial learning rate. T is the total number of epochs budgeted for all CIFAR experiments. We set $r = 0.01$ and $\kappa = 5$.

While this introduces some slight training instability early in training, which could potentially be managed by altering the schedule, we find that such a schedule boosts the validation performance, particularly so for auto-damped methods, as shown by the blue curve in Figure 6.14b, which surpasses the generalisation of SGD (shown in Figure 6.10).

To more clearly expose the combined impact of adaptive damping and this alternative learning schedule we consider the variations in Figure 6.15 for a learning rate and damping both equal to 0.0001. Here we see that the aggressive learning rate schedule with flat damping diverges, whereas the autodamping stabilises training, allowing for convergence to a solution with excellent generalisation. We see here in Figure 6.15c that the damping coefficient reacts to this large learning rate increase by increasing its rate of damping early, stabilising training. We also show for reference that the typical linear decay schedule, with a larger learning rate and initial damping does not supersede the validation result of smaller learning rate and flat damping counter-part (it does however train better). This demonstrates the necessity of an alternative learning rate schedule to bring out the value of the adaptive damping. We remark however that optimal results in deep learning almost always require some degree of hand-crafted tuning of the learning rate. Our adaptive damping method is not proposed as a panacea, but just an optimal method of setting the damping coefficient. Since changing the damping coefficient effectively changes to geometry of the loss surface, it is entirely reasonable that the learning rate may have to be tweaked to give best results.



(a) Training Error

(b) Val Error

(c) Damping

Figure 6.15: VGG-16 on CIFAR-100 dataset using the Adam optimiser with $\gamma = 0$ (no weight decay) for a learning rate of α, batch size $B = 128$ and damping set by δ. For adaptive damping methods the damping is given an initial floor value of δ^* and is then updated using the variance of the Hessian every 100 steps. α^* refers to the use of an alternative ramp up schedule where the base learning rate is increased by a factor of 5 at the start of training before being decreased.

6.6 Conclusion

In this chapter we have showed using a spiked random matrix model for the batch loss of deep neural networks that we expect sharp directions of loss surface to retain more information about the true loss surface compared to flatter directions. For adaptive methods, which attempt to minimise an implicit local quadratic of the sampled loss surface, this leads to sub-optimal steps with worse generalisation performance. We further investigate the effect of damping on the solution sharpness and find that increasing damping always decreases the solution sharpness, linking to prior work in this area. We find that for large neural networks an increase in damping both assists training and is even able to best the SGD test baseline. An interesting consequence of this finding is that it suggests that damping should be considered an essential hyper-parameter in adaptive gradient methods as it already is in stochastic second order methods. Moreover, our random matrix theory model motivates a novel interpretation of damping as linear shrinkage estimation of the Hessian. We establish the validity of this interpretation by using shrinkage estimation theory to derive an optimal adaptive damping scheme which we show experimentally to dramatically improve optimisation speed with adaptive methods *and* closes the adaptive generalisation gap.

Our work leaves open several directions for further investigation and extension. Mathematically, there is the considerable challenge of determining optimal assumptions on the network, loss function and data distribution such that the key outlier overlap result in Theorem 6.1, or sufficiently similar analogues thereof, can be obtained. On the experimental side, we have restricted ourselves to computer vision datasets and a small number of appropriate standard network architectures. These choices helped to maintain clarity on the key points of investigation, however they are clearly limiting. In particular, it would be natural to reconsider our investigations in situations for which adaptive optimisers typically obtain state of the art results, such as modern natural language processing [Dev+18]. Practically speaking, we have proposed a novel, theoretically motivated and effective adaptive damping method, but it is reliant on relatively expensive Hessian variance estimates throughout training. Future work could focus on cheaper methods of obtaining the required variance estimates.

Chapter 7

Appearance of local random matrix statistics

The content of this chapter was published first as a pre-print in February 2021 (`https://arxiv.org/abs/2102.06740`) and later as a journal article: “Appearance of random matrix theory in deep learning”. **Nicholas P Baskerville**, Diego Granziol and Jonathan P Keating. *Physica A: Statistical Mechanics and its Applications*, 590:126742, 2022.

NPB performed the calculations, designed, coded and ran most of the experiments and wrote most of the random matrix theory aspects of the paper. DG assisted with writing code, ran the training of a few of the neural networks and wrote some of the more machine learning oriented sections of the paper. JPK proposed the research idea, advised throughout and contributed several sections to the paper. Anonymous reviewers spotted some minor errors, advised on changes of presentation and extra experiments and provided useful references.

7.1 Preliminaries

Consider a neural network with weights $\boldsymbol{w} \in \mathbb{R}^P$ and a dataset with distribution $\mathbb{P}_{\text{data}}$. For the purposes of our discussion, a neural network, $f_{\boldsymbol{w}}$ say, is just a non-linear function from some $\mathbb{R}^d$ to some $\mathbb{R}^c$, parametrised by $\boldsymbol{w}$. Neural networks can be defined in many different ways in terms of their weights (the architecture of the network), but these details will not play role in our discussion. What will be important is that the number of weights P will be large, i.e. approaching 10,000 even in the simplest of cases. Let $L(\boldsymbol{w},\boldsymbol{x})$ be the loss of the network for a single datum $\boldsymbol{x}$ and let $\mathcal{D}$ denote any finite sample of data points from $\mathbb{P}_{\text{data}}$. A simple example of L is the squared error $L(\boldsymbol{w},(\boldsymbol{x},\boldsymbol{y})) = ||f_{\boldsymbol{w}}(\boldsymbol{x}) - \boldsymbol{y}||_2^2$, where $\mathbb{P}_{\text{data}}$ is a distribution on tuples of features $\boldsymbol{x}$ and labels $\boldsymbol{y}$. The *true loss* is given by

$$\mathcal{L}_{true}(\boldsymbol{w}) = \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_{\text{data}}} L(\boldsymbol{w},\boldsymbol{x}) \tag{7.1}$$

and the *empirical loss* (or training loss) is given by

$$\mathcal{L}_{emp}(\boldsymbol{w},\mathcal{D}) = \frac{1}{|\mathcal{D}|}\sum_{\boldsymbol{x}\in\mathcal{D}} L(\boldsymbol{w},\boldsymbol{x}). \tag{7.2}$$

Where $\mathcal{D}$ denotes the dataset. The true loss is a deterministic function of the weights, while the empirical loss is a random function with the randomness coming from the random sampling of the finite dataset $\mathcal{D}$. The empirical Hessian $\boldsymbol{H}_{emp}(\boldsymbol{w}) = \nabla^2\mathcal{L}_{emp}(\boldsymbol{w})$, describes the loss curvature at the point $\boldsymbol{w}$ in weight space. By the spectral theorem, the Hessian can be written in terms of its eigenvalue/eigenvector pairs $\boldsymbol{H}_{emp} = \sum_i^P \lambda_i \phi_i \phi_i^T$, where the dependence on $\boldsymbol{w}$ has been dropped to keep the notation simple. The eigenvalues of the Hessian are particularly important, being explicitly required in second-order optimisation methods, and characterising the stationary points of the loss as local minima, local maxima or generally saddle points of some other index.

For a matrix drawn from a probability distribution, its eigenvalues are random variables. The eigenvalue distribution is described by the joint probability density function (j.p.d.f) $p(\lambda_1,\lambda_2,\ldots,\lambda_P)$, also known as the P-point correlation function. The simplest example is the *empirical spectral density (ESD)*, $\rho^{(P)}(\lambda) = \frac{1}{P}\sum_i^P \delta(\lambda-\lambda_i)$. Integrating $\rho^{(P)}(\lambda)$ over an interval with respect to λ gives the fraction of the eigenvalues in that interval. Taking an expectation over the random matrix ensemble, we obtain the *mean spectral density* $\mathbb{E}\rho^{(P)}(\lambda)$, which is a deterministic probability distribution on $\mathbb{R}$. Alternatively, taking the $P\to\infty$ limit, assuming it exists, gives the *limiting spectral density (LSD)* ρ, another deterministic probability distribution on $\mathbb{R}$. A key feature of many random matrix ensembles is *self-averaging* or *ergodicity*, meaning that the leading order term (for large P) in $\mathbb{E}\rho^{(P)}$ agrees with ρ. Given the j.p.d.f, one can obtain the mean spectral density, known as the 1-point correlation function (or any other k-point correlation function) by marginalisation

$$\mathbb{E}\rho^{(P)}(\lambda) = \int p(\lambda,\lambda_2,\ldots,\lambda_P)d\lambda_2\ldots d\lambda_P. \tag{7.3}$$

A GOE matrix is an example of a *Wigner random matrix*, namely a real-symmetric (or complex-Hermitian) matrix with otherwise i.i.d. entries and off-diagonal variance σ^2.[1] The mean spectral density for Wigner matrices is known to be Wigner's semicircle [Meh04]

$$\rho_{SC}(\lambda) = \frac{1}{2\pi\sigma^2 P}\sqrt{4P\sigma^2-\lambda^2}\,\mathbb{1}_{|\lambda|\leqslant 2\sigma\sqrt{P}}. \tag{7.4}$$

The radius of the semicircle[2] is proportional to $\sqrt{P}\sigma$, hence scaling Wigner matrices by $1/\sqrt{P}$ leads to a limit distribution when $P\to\infty$. This is the LSD. With this scaling, there are, on average, $\mathcal{O}(P)$ eigenvalues in any open subset of the compact spectral support. In this sense, the mean (or limiting) spectral density is *macroscopic*, meaning that, as $P\to\infty$, one ceases to see individual eigenvalues, but rather a continuum with some given density.

[1] The GOE corresponds to taking the independent matrix entries to be normal random variables.

[2] Using the Frobenius norm identity $\sum_i^P \lambda_i^2 = P^2\sigma^2$

7.2 Motivation: Microscopic Universality

Random Matrix Theory was first developed in physics to explain the statistical properties of nuclear energy levels, and later used to describe the spectral statistics in atomic spectra, condensed matter systems, quantum chaotic systems etc; see, for example [WM08; Bee97; Ber+87; Boh91]. *None of these physical systems exhibits a semicircular empirical spectral density*. However they all generically show agreement with RMT at the level of the mean eigenvalue spacing when local spectral statistics are compared. Our point is that while neither multi-layer perceptron (MLP) nor Softmax Regression Hessians are described by the Wigner semicircle law which holds for GOE matrices (c.f. Figure 1a) – their spectra contain outliers, large peaks near the origin and the remaining components of the histogram also do not match the semicircle – nevertheless Random Matrix Theory can still (and we shall demonstrate does) describe spectral fluctuations on the scale of their mean eigenvalue spacing.

It is worth noting in passing that possibilities other than random-matrix statistics exist and occur. For example, in systems that are classically integrable, one finds instead Poisson statistics [BT77; Ber+87]; similarly, Poisson statistics also occur in disordered systems in the regime of strong Anderson localisation [Efe99]; and for systems close to integrable one finds a superposition of random-matrix and Poisson statistics [BR84]. So showing that Random Matrix Theory applies is far from being a trivial observation. Indeed it remains one of the outstanding challenges of mathematical physics to prove that the spectral statistics of any individual Hamiltonian system are described by it in the semiclassical limit.

Physics RMT calculations re-scale the eigenvalues to have a mean level spacing of 1 and then typically look at the *nearest neighbour spacings distribution* (NNSD), i.e. the distribution of the distances between adjacent pairs of eigenvalues. One theoretical motivation for considering the NNSD is that it is independent of the Gaussianity assumption and reflects the symmetry of the underlying system. It is the NNSD that is universal (for systems of the same symmetry class) and not the average spectral density, which is best viewed as a parameter of the system. The aforementioned transformation to give mean spacing 1 is done precisely to remove the effect of the average spectral density on the pair correlations leaving behind only the universal correlations. To the best of our knowledge no prior work has evaluated the NNSD of artificial neural networks and this is a central focus of this chapter.

In contrast to the LSD, other k-point correlation functions are also normalised such that the mean spacing between adjacent eigenvalues is unity. At this *microscopic* scale, the LSD is locally constant and equal to 1 meaning that its effect on the eigenvalues' distribution has been removed and only microscopic correlations remain. In the case of Wigner random matrices, for which the LSD varies slowly across the support of the eigenvalue distribution, this corresponds to scaling by $\sqrt{P}$. On this scale the limiting eigenvalue correlations when $P \to \infty$ are *universal*; that is, they are the same for wide classes of random matrices, depending only on symmetry [GMW98]. For example, this universality is exhibited by the NNSD. Consider a 2×2 GOE matrix, in which case the j.p.d.f has a simple form:

$$p(\lambda_1, \lambda_2) \propto |\lambda_1 - \lambda_2| e^{-\frac{1}{2}(\lambda_1^2+\lambda_2^2)}. \tag{7.5}$$

Making the change of variables $\nu_1 = \lambda_1 - \lambda_2, \nu_2 = \lambda_1 + \lambda_2$, integrating out ν_2 and setting $s = |\nu_1|$ results in a density $\rho_{Wigner}(s) = \frac{\pi s}{2} e^{-\frac{\pi}{4}s^2}$, known as the *Wigner surmise* (see Figure 7.1). For larger matrices, the j.p.d.f must include an indicator function $\mathbb{1}\{\lambda_1 \leqslant \lambda_2 \leqslant \ldots \lambda_P\}$ before marginalisation so that one is studying pairs of *adjacent* eigenvalues. While the Wigner surmise can only be proved exactly, as above, for the 2×2 GOE, it holds to high accuracy for the NNSD of GOE matrices of any size provided that the eigenvalues have been scaled to give mean spacing 1.[3] The Wigner surmise density vanishes at 0, capturing 'repulsion' between eigenvalues that is characteristic of RMT statistics, in contrast to the distribution of entirely independent eigenvalues given by the *Poisson law* $\rho_{Poisson}(s) = e^{-s}$. The Wigner surmise is universal in that the same density formula applies to all real-symmetric random matrices, not just the GOE or Wigner random matrices.

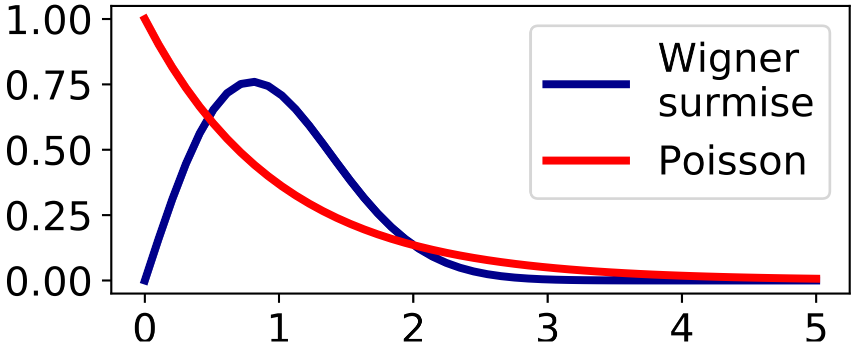


Figure 7.1: The density of the Wigner surmise.

7.3 Methodology

Prior work [Gra+19a; Pap18; GKX19] focusing on the Hessian empirical spectral density has utilised fast Hessian vector products [Pea94] in conjunction with Lanczos [MS06] methods. However, these methods approximate only macroscopic quantities like the spectral density, not microscopic statistics such as nearest neighbour spectral spacings. For modern neural networks, the $\mathcal{O}(P^3)$ Hessian eigendecomposition cost will be prohibitive, e.g. for a Residual Network (Resnet) [He+16] with 34 layers $P = 10^7$. Hence, We restrict to models small enough to perform exact full Hessian computation and eigendecomposition.

We consider single layer neural networks for classification (softmax regression), 2-hidden-layer MLPs[4] and 3 hidden-layer MLPs[5]. On MNIST [Den12], the Hessians are of size 7850×7850 for

[3] An exact formula for the NNSD of GOE matrices of any size, and one that holds in the large P limit, can be found in [Meh04].

[4] Hidden layer widths: 10, 100.

[5] Hidden layer widths: 10, 100, 100.

logistic regression, 9860×9860 for the small MLP and 20060×20060 for the larger 3 hidden-layer MLP, so can be computed exactly by simply applying automatic differentiation twice, and the eigenvalues can be computed exactly in a reasonable amount of time. We also consider a single layer applied to CIFAR-10 [KH+09] classification with pre-trained Resnet-34 embedding features [He+16; PyT21]. While we cannot at present study the full Hessian of, for example, a Resnet-34, we can study the common transfer learning use-case of training only the final layer on some particular task [Sha+14]. The Hessians can be computed at any data point or over any collection of data points. We consider Hessians computed over the entire datasets in question, and over batches of size 64. We separately consider test and train sets.

In order to extend the relevance of our analysis to beyond logistic regression and MLP, we consider one of the simplest convolutional neural networks (CNN) of the form of LeNet [LeC98] on CIFAR-10. Compared to the standard LeNet (which has over 50000 parameters) we reduce the number of neurons in the first fully connected layer from 120 to 35 and the second from 84 to 50. Note that the resulting architecture contains a bottleneck in the intermediate layer, in contrast to the "hour-glass" shapes that are necessary to maintain manageable parameter numbers with full MLP architectures. Despite reducing the total number of parameters by a factor of 3 we find the total validation accuracy drop to be no more than 2%. The total validation accuracy of 69% is significantly below state of the art ≈ 95%, but we are clearly in the regime where significant learning can and does take place, which we consider sufficient for the purposes of this manuscript. We also extend our experiments beyond the cross entropy loss function, by considering a regression problem (L_2 loss) and beyond the high-dimensional feature setting of computer vision with the Bike dataset[6] which has only 13-dimensional feature vectors and a single-dimensional regressand (see Appendix C.2.3 for details of our data pre-processing). The architecture in this case widens considerably in the first layer (from 13 inputs to 100 neurons) and that gradually tapers to the single output. The final test loss (i.e. mean squared error) of the trained model is 0.044 which is competitive with baseline results [Wan+19][7]

Training details: All networks were trained using SGD for 300 epochs with initial learning rate 0.003, linear learning rate decay to 0.00003 between epoch 150 and 270, momentum 0.9 and weight decay 5×10^{-4}. We use a PyTorch [Pas+17] implementation. Full code to reproduce our results is made available [8]. Full descriptions of all network architectures are given in the Appendix C.2.

[6] https://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset (accessed 14/10/21)

[7] [Wan+19] report an RMSE of 0.220 on Bike (which corresponds to 0.048 mean squared error) using a Gaussian process regression model with exact inference.

[8] `https://github.com/npbaskerville/dnn-rmt-spacings`

7.4 Spectral spacing statistics in RMT

Consider a random $P \times P$ matrix M_P with ordered $\lambda_1 \leqslant \lambda_2 \leqslant \ldots \leqslant \lambda_P$. Let I_{ave} be the mean spectral cumulative density function for the random matrix ensemble from which M_P is drawn. The *unfolded spectrum* is defined as

$$l_i = I_{ave}(\lambda_i). \tag{7.6}$$

The unfolded spacings are then defined as

$$s_i = l_i - l_{i-1}, \quad i = 2, \ldots, P. \tag{7.7}$$

With this definition, the mean of the s_i is unity, which means that this transformation has brought the eigenvalues on to the microscopic scale on which universal spectral spacing statistics emerge. We are investigating the presence of Random Matrix Theory statistics in neural networks by considering the nearest neighbour spectral spacings of their Hessians. Within the Random Matrix Theory literature, it has been repeatedly observed [Boh91; Ber+87] that the unfolded spacings of a matrix with RMT pair correlations follow universal distributions determined only by the symmetry class of the M_P. Hessians are real symmetric, so the relevant universality class is GOE and therefore the unfolded neural network spacings should be compared to the Wigner surmise

$$\rho_{Wigner}(s) = \frac{\pi s}{2} e^{-\frac{\pi}{4}s^2}. \tag{7.8}$$

A collection of unfolded spacings $s_2, \ldots, s_P$ from a matrix with GOE spacing statistics should look like a sample of i.i.d. draws from the Wigner surmise density (7.8). For some known random matrix distributions, I_{ave} may be available explicitly, or at least via highly accurate quadrature methods from a known mean spectral density. For example, for the $P \times P$ GOE [AA12] $I^{GOE}_{ave}(\lambda)$ is given by:

$$P\left[\frac{1}{2} + \frac{\lambda}{2\pi P}\sqrt{2P - \lambda^2} + \frac{1}{\pi}\arctan\left(\frac{\lambda}{\sqrt{2P - \lambda^2}}\right)\right]. \tag{7.9}$$

However, when dealing with experimental data where the mean spectral density is unknown, one must resort to using an approximation to I_{ave}. Various approaches are used in the literature, including polynomial spline interpolation [AA12]. The approach of [SWB14; Sch15] is most appropriate in our case, since computing Hessians over many mini-batches of data results in a large pool of spectra which can be used to accurately approximate I_{ave} simply by the empirical cumulative density. Suppose that we have m samples $(M_P^{(i)})_{i=1}^m$ from a random matrix distribution over symmetric $P \times P$ matrices. Fix some integers $m_1, m_2 > 0$ such that $m_1 + m_2 = m$. The spectra of the matrices $(M_P^{(i)})_{i=1}^{m_1}$ can then be used to construct an approximation to I_{ave}. More precisely, let Λ_1 be the set of all eigenvalues of the $(M_P^{(i)})_{i=1}^{m_1}$, then we define

$$\tilde{I}_{ave}(\lambda) = \frac{1}{|\Lambda_1|}|\{\lambda' \in \Lambda_1 \mid \lambda' < \lambda\}|. \tag{7.10}$$

For each of the matrices $(M_P^{(i)})_{i=m_1+1}^m$, one can then use $\tilde{I}_{ave}$ to construct their unfolded spacings. When the matrix size P is small, one can only study the spectral spacing distribution by looking over multiple matrix samples. However, the same spacing distribution is also present for a single matrix in the large P limit. A clear disadvantage of studying unfolded nearest neighbour spectral spacings with the above methods is the need for a reasonably large number of independent matrix samples. This rules-out studying the unfolded spacings of a single large matrix. Another obvious disadvantage is the introduction of error by the approximation of I_{ave}, giving the opportunity for local spectral statistics to be distorted or destroyed. An alternative statistic is the consecutive spacing ratio of [Ata+13]. In the above notation, the ratios for a single $P \times P$ matrix are defined as

$$r_i = \frac{\lambda_i - \lambda_{i-1}}{\lambda_{i-1} - \lambda_{i-2}}, \quad 2 \leqslant i \leqslant P. \tag{7.11}$$

[Ata+13] proved a 'Wigner-like surmise' for the spacing ratios, which for the GOE is

$$P(r) = \frac{27(r + r^2)}{8(1 + r + r^2)^{5/2}}. \tag{7.12}$$

In our experiments, we can compute the spacing ratios for Hessians computed over entire datasets or over batches, whereas the unfolded spacing ratios can only be computed in the batch setting, in which case a random $\frac{2}{3}$ of the batch Hessians are reserved for computing $\tilde{I}_{ave}$ and the remaining $\frac{1}{3}$ are unfolded and analysed. This split is essentially arbitrary, except that we err on the side of using more to compute $\tilde{I}_{ave}$ since even a single properly unfolded spectrum can demonstrate universal local statistics.

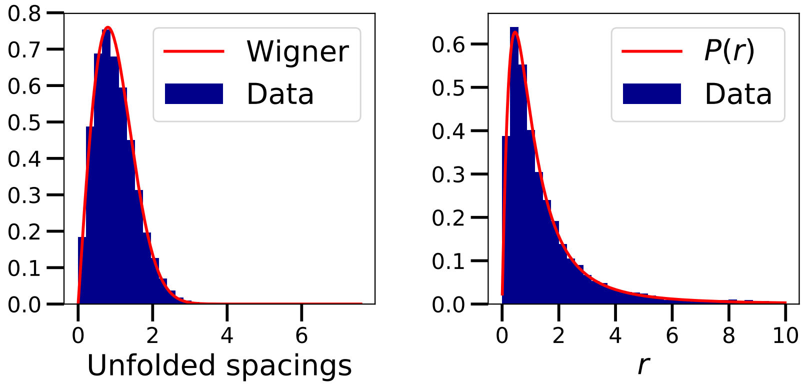
7.5 Results

We display results as histograms of data along with a plot of the Wigner (or the Wigner-like) surmise density. We make a few practical adjustments to the plots. Spacing ratios are truncated above some value, as the presence of a few extreme outliers makes visualisation difficult. We choose a cut-off at 10. Note that around 0.985 of the mass of the Wigner-like surmise is below 10, so this is a reasonable adjustment. The hessians have degenerate spectra. The Wigner surmise is not a good fit to the observed unfolded spectra if the zero eigenvalues are retained. Imposing a lower cut-off of 10^{-20} in magnitude is sufficient to obtain agreement with Wigner.[9] This is below the machine precision, so these omitted eigenvalues are indistinguishable from 0.

7.5.1 MNIST and MLPs

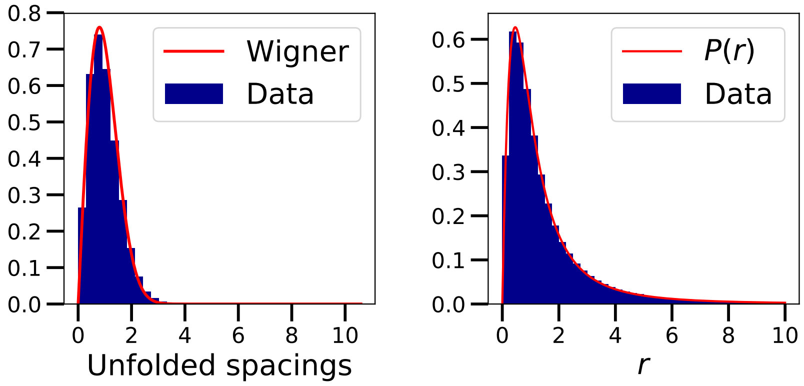
We show results in Figures 7.2 and 7.3, with further plots in the Appendix. We also considered randomly initialised networks and we evaluated the Hessians over train and test datasets separately in

[9] For example, in the case of the 3-hidden-layer MLP on MNIST shown in Figure 7.3, among 157 batch-wise spectra the proportion of eigenvalues below the cut-off was between 0.29 and 0.40.



(a) Unfolded spacings. Batch size 64. (b) Spacing ratios. Entire dataset.

Figure 7.2: Spacing distributions for the Hessian of a logistic regression trained Resnet-34 embeddings of CIFAR10. Hessians computed over the test set.



(a) Unfolded spacings. Batch size 64. (b) Spacing ratios. Batch size 64.

Figure 7.3: Spacing distributions for the Hessian of a 3-hidden-layer MLP trained on MNIST. Hessians computed over the test set.

all cases. Unfolded spacings were computed only for Hessians evaluated on batches of 64 data points, while spacing ratios were computed in batches and over the entire dataset. We observe a striking level of agreement between the observed spectra and the GOE. There was no discernible difference between the train and test conditions, nor between batch and full dataset conditions, nor between trained and untrained models. Note that the presence of GOE statistics for the untrained models is not a foregone conclusion. Of course, the weights of the model are indeed random Gaussian, but the Hessian is still a function of the data set, so it is not the case the Hessian eigenvalue statistics are bound to be GOE a priori. Overall, the very close agreement between Random Matrix Theory predictions and our observations for several different architectures, model sizes and datasets demonstrates a clear presence of RMT statistics in neural networks.

Our results indicate that models for the loss surfaces of large neural networks should include assumptions of GOE local statistics of the Hessian, but ideally avoid such assumptions on the global statistics. To further illustrate this point, consider a Gaussian process $\mathcal{L}_{emp} \sim \mathcal{GP}(0,k)$ where k is some kernel function. Following from our Gaussian process definition, the covariance of derivatives of the empirical loss can be computed using a well-known result (see [AT09] equation 5.5.4), e.g.

$$Cov(\partial_i \mathcal{L}_{emp}(\boldsymbol{w}), \partial_j \mathcal{L}_{emp}(\boldsymbol{w}')) = \partial_{w_i}\partial_{w'_j} k(\boldsymbol{w}, \boldsymbol{w}')$$

and further, assuming a stationary kernel $k(\boldsymbol{w},\boldsymbol{w}') = k\left(-\frac{1}{2}||\boldsymbol{w}-\boldsymbol{w}'||_2^2\right)$ (note abuse of notation)

$$\begin{aligned} &Cov(\partial_i \mathcal{L}_{emp}(\boldsymbol{w}), \partial_j \mathcal{L}_{emp}(\boldsymbol{w}')) \\ &= (w_i - w'_i)(w'_j - w_j) k''\left(-\frac{1}{2}||\boldsymbol{w}-\boldsymbol{w}'||_2^2\right) + \delta_{ij} k'\left(-\frac{1}{2}||\boldsymbol{w}-\boldsymbol{w}'||_2^2\right). \end{aligned} \tag{7.13}$$

Differentiating (7.13) further, we obtain

$$Cov(\partial_{ij}\mathcal{L}_{emp}(\boldsymbol{w}), \partial_{kl}\mathcal{L}_{emp}(\boldsymbol{w})) = k''(0)\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right) + k'(0)^2\delta_{ij}\delta_{kl} \tag{7.14}$$

The Hessian $\boldsymbol{H}_{emp}$ has Gaussian entries with mean zero, so the distribution of $\boldsymbol{H}_{emp}$ is determined entirely by $k'(0)$ and $k''(0)$. Neglecting to choose k explicitly, we vary the values of $k'(0)$ and $k''(0)$ to produce nearest neighbour spectral spacings ratios and spectral densities. The histograms for spectral spacing ratios are indistinguishable and agree very well with the GOE, as shown in Figure 7.5. The spectral densities are shown in Figure 7.4, including examples with rank degeneracy, introduced by defining k only on a lower-dimensional subspace of the input space, and outliers, introduced by adding a fixed diagonal matrix to the Hessian. Figure 7.4 shows varying levels of agreement with the semi-circle law, depending on the choice of $k'(0), k''(0)$.

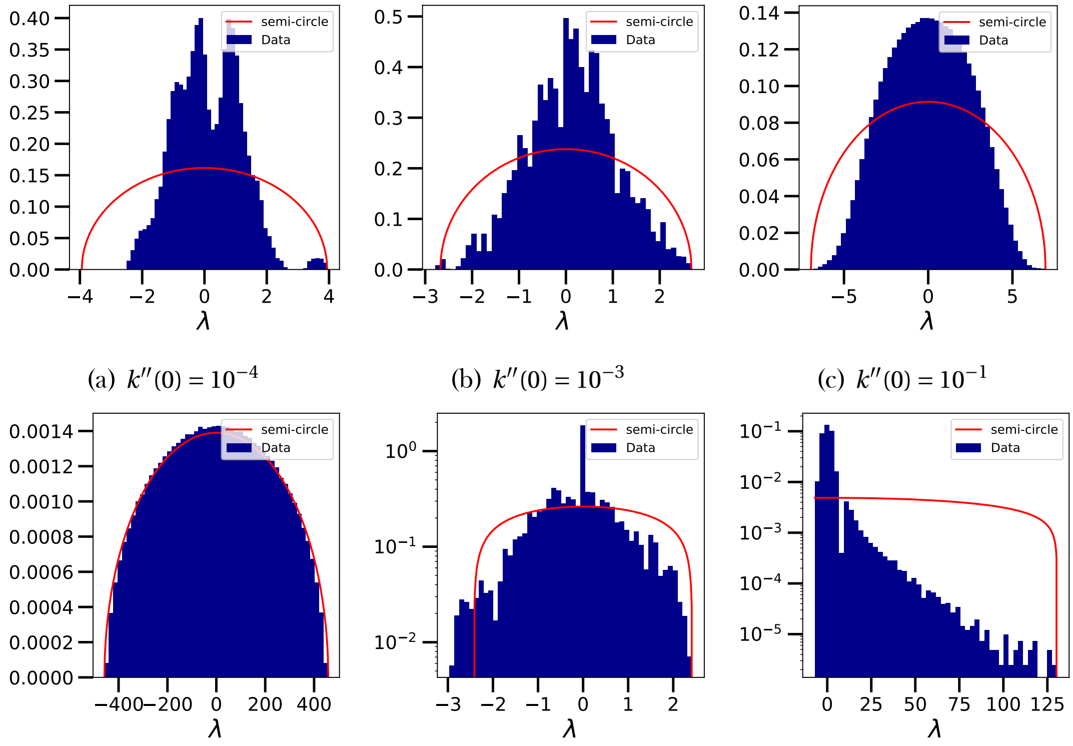
Remark 7.1. The covariance structure in (7.13) is very close to that of a GOE matrix. If $k'(0) = 0$ and $k''(0) \neq= 0$, then the covariance would be exactly that of a GOE matrix. With general values $k'(0) \neq 0 \neq k''(0)$, we see that the second term is non-zero on, and only on, the diagonals of the Hessian, however it does induce dependence between all diagonal elements. We have been unable to compute the limiting spectral density exactly but we suspect it may well be possible.

7.5.2 Beyond the MLP

Figure 7.6 shows the mean spectral density and adjacent spacing ratios for the Hessian of a CNN trained on CIFAR10. As with the MLP networks and MNIST data considered above, we see an obviously non-semicircular mean level density but the adjacent spacing ratios are nevertheless described by the universal GOE law.

7.5.3 Beyond image classification

Figure 7.7 shows the mean spectral density and adjacent spacing ratios for the Hessian of an MLP trained on the Bike dataset. Once again we see an obviously non-semicircular mean level density



(a) $k''(0) = 10^{-4}$

(b) $k''(0) = 10^{-3}$

(c) $k''(0) = 10^{-1}$

(d) $k''(0) = 10$

(e) $k''(0) = 10^{-3}*$

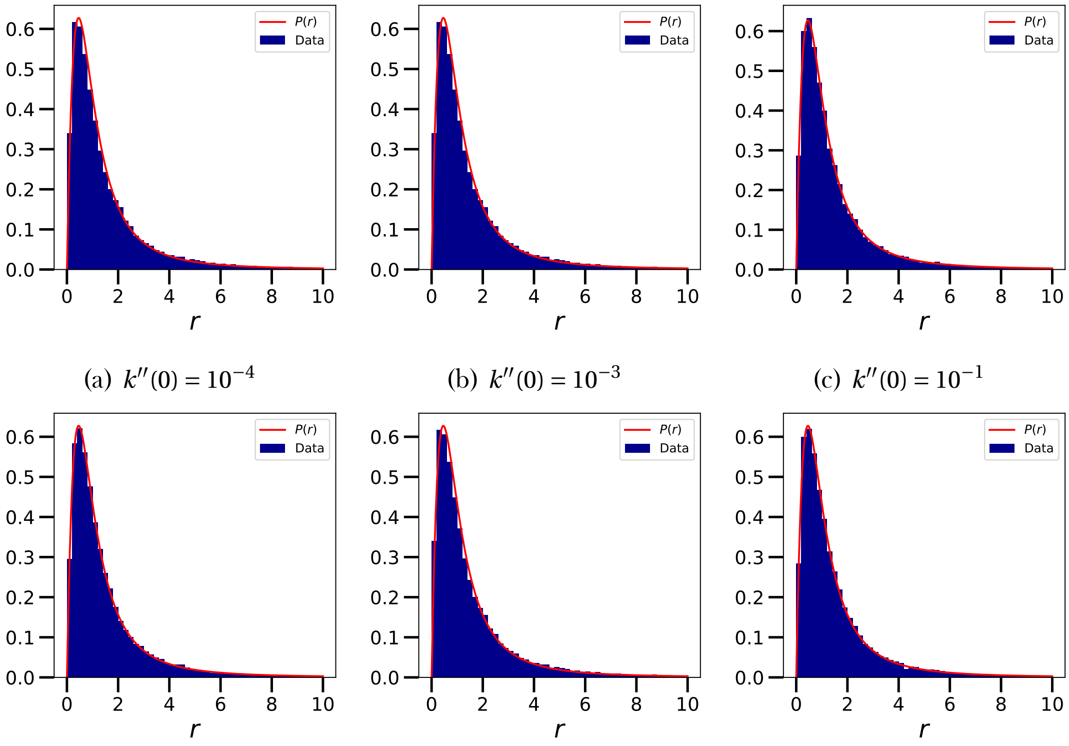
(f) $k''(0) = 0.0001\dagger$

Figure 7.4: Spectral densities of Gaussian process Hessians with various kernel choices. All use $k'(0) = 1$. The dimension is 300 in all cases except (d), in which the Hessian is padded to 400 dimensions with zeros. All histograms are produced with 100 independent Hessian samples. $* = 100$ degenerate directions. $\dagger = 20$ outliers

but the adjacent spacing ratios are nevertheless described by the universal GOE law. This serves to demonstrate that there is nothing special about image data or, more importantly, high input feature dimension, since the Bike dataset has only 13 input features.

7.5.4 Beyond the Hessian

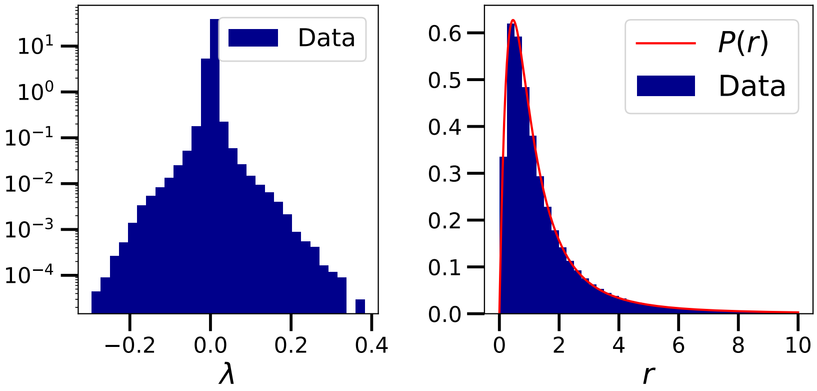
Given that the Hessian is not the only matrix of interest in Machine Learning, it is pertinent to study whether our empirical results hold more generally. There have been lots of investigations for the Gauss-Newton [LD02; PB17], or generalised Gauss-Newton (which is the analogue of the Gauss-Newton when using the cross entropy instead of square loss) matrices, particularly in the fields of optimisation [Dau+14; MS12; MG15; Mar14]. We consider the Gauss-Newton of the network trained on the Bike dataset with square loss. In this case the Gauss Newton $\boldsymbol{G} = \boldsymbol{J}^T\boldsymbol{J}$ shares the same non-null subspace as the Neural Tangent Kernel (NTK) [JGH18; Cai+19], where $\boldsymbol{J}$ denotes the Jacobian, i.e the derivative of the output with respect to the weights, which in this case is simply a vector. The NTK is used for the analysis of trajectories of gradient descent and is particularly interesting for large width networks, where it can be analytically shown that weights remain close to



(a) $k''(0) = 10^{-4}$ (b) $k''(0) = 10^{-3}$ (c) $k''(0) = 10^{-1}$

(d) $k''(0) = 10$ (e) $k''(0) = 10^{-3}*$ (f) $k''(0) = 0.0001\dagger$

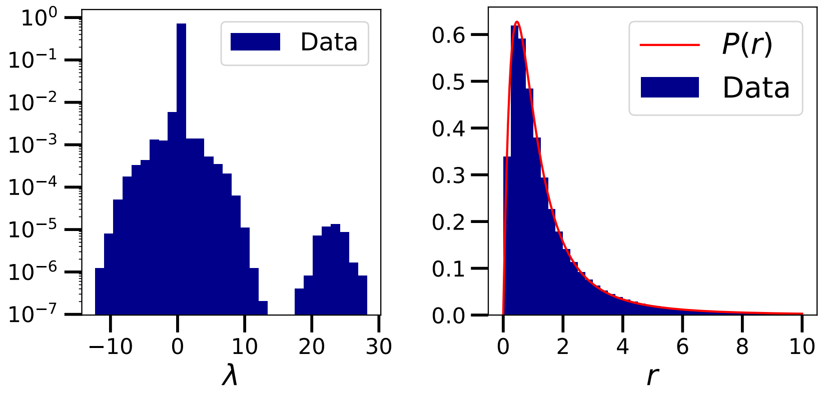
Figure 7.5: Consecutive spacing ratios of Gaussian process Hessians with various kernel choices. All use $k'(0) = 1$. The dimension is 300 in all cases except (d), in which the Hessian is padded to 400 dimensions with zeros. $* = 100$ degenerate directions. $\dagger = 20$ outliers.



(a) Mean spectral density. (b) Spacing ratios.

Figure 7.6: Spectral statistics for the Hessian of a CNN trained on CIFAR10. Hessians computed over batches of size 64 on the test set.

their initialisation and the network is well approximated by its linearisation. Figure 7.8 shows the mean spectral density and adjacent spacing ratios for the Gauss-Newton matrix of an MLP trained

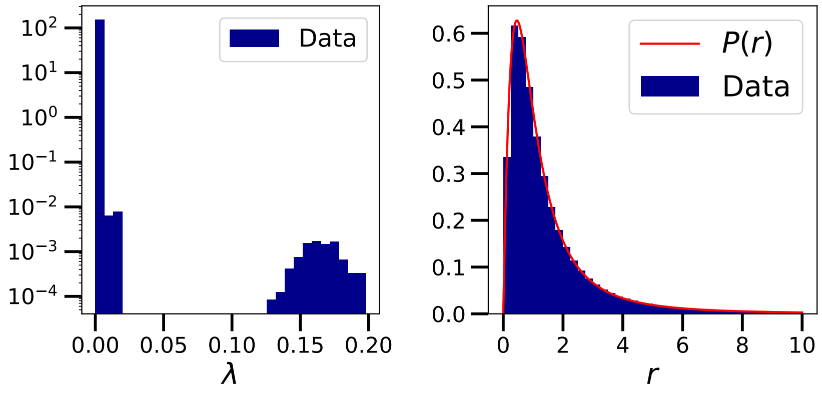


(a) Mean spectral density.

(b) Spacing ratios.

Figure 7.7: Spectral statistics for the Hessian of an MLP trained on the Bike dataset. Hessians computed over batches of size 64 on the test set.

on the Bike dataset. The results are just as for the Hessians above: universal GOE spacings, but the mean density is very much not semicircular. This is an interesting result because even for a different matrix employed in a different context we still see the same universal RMT spacings.



(a) Mean spectral density.

(b) Spacing ratios.

Figure 7.8: Spectral statistics for the Gauss-Newton matrix of an MLP trained on the Bike dataset. Matrices computed over batches of size 64 on the test set.

7.6 Conclusion

We have demonstrated experimentally the existence of random matrix statistics in small neural networks on the scale of the mean eigenvalue separation. This provides the first direct evidence of universal RMT statistics present in neural networks trained on real datasets. Hitherto the role of random matrix theory in deep learning has been unclear. Prior work has studied theoretical models with specific assumptions leading to specific random matrix ensembles. Though certainly insightful, it is not clear to what extent any of these studies are applicable to real neural networks. This work aims to shift the focus by demonstrating the clear presence of universal random matrix behaviour in real neural networks. We expect that future theoretical studies will start from this robust supposition.

When working with a neural network on some dataset, one has information a priori about its Hessian. Its distribution and correlation structure may well be entirely inaccessible, but correlations between Hessian eigenvalues on the local scale can be assumed to be universal and overall the matrix can be rightly viewed as a random matrix possessing universal local statistics.

We focus on small neural networks where Hessian eigendecomposition is feasible. Future research that our work motivates could develop methods to approximate the level spacing distribution of large deep neural networks for which exact Hessian spectra cannot be computed. If the same RMT statistics are found, this would constitute a profound universal property of neural networks models; conversely, a break-down in these RMT statistics would be an indication of some fundamental separation between different network sizes or architectures.

A few recent works [Lou+21; Gol+20; AP20a] considered and used the idea of *Gaussian equivalence* to make theoretical progress in neural network models with fewer assumptions than previously required (e.g. on the data distribution). The principle is that complicated random matrix distributions on non-linear functions of random matrices can be replaced in calculations training and test loss by their Gaussian equivalents, i.e. Gaussian matrices with matching first and second moments. This idea reflects a form of universality and can drastically increase the tractability of calculations. The random matrix universality we have here demonstrated in neural networks may be related, and should be considered as a possible source of other analogous universality simplifications that can render realistic but intractable models tractable.

One intriguing possible avenue is the relation to chaotic systems. Quantum systems with chaotic classical limits are know to display RMT spectral pairwise correlations, whereas Poisson statistics correspond to integrable systems. We suggest that the presence of GOE pairwise correlations in neural network Hessians, as opposed to Poisson, indicates that neural network training dynamics cannot be reduced to some simpler, smaller set of dynamical equations.

Chapter 8

Universal characteristics of loss surfaces

The content of this chapter was published first as a pre-print in May 2022 (`https://arxiv.org/abs/2205.08601`) and later as a journal article in December 2022: “Universal characteristics of neural network loss surfaces from random matrix theory”. **Nicholas P Baskerville**, Jonathan P Keating, Francesco Mezzadri, Joseph Najnudel and Diego Granziol. *Journal of Physics A: Mathematical and Theoretical.*

NPB proposed all three of the main ideas, performed all calculations, proved most of the results, did the vast majority of the write-up and did all analysis of experimental results. DG conducted the neural network training, extracted the empirical Hessian data and contributed to the write-up of sections pertaining to experiments. JN provided several important ideas for the proof in Appendix A. Anonymous reviewers provided helpful feedback on the presentation and spotted several typos.

8.1 General random matrix model for loss surface Hessians

8.1.1 The model

Given a loss function $\mathcal{L} : \mathscr{Y} \times \mathscr{Y} \to \mathbb{R}$, a data generating distribution $\mathbb{P}_{\text{data}}$ supported on $\mathscr{X} \times \mathscr{Y}$ and a neural network $f_{\boldsymbol{w}} : \mathscr{X} \to \mathscr{Y}$ parametrised by $\boldsymbol{w} \in \mathbb{R}^N$, its batch Hessian is given by

$$H_{\text{batch}} = \frac{1}{b} \sum_{i=1}^{b} \frac{\partial^2}{\partial \boldsymbol{w}^2} \mathcal{L}(f_{\boldsymbol{w}}(\boldsymbol{x}_i), y_i), \quad (\boldsymbol{x}_i, y_i) \overset{\text{i.i.d.}}{\sim} \mathbb{P}_{\text{data}} \tag{8.1}$$

and its true Hessian is given by

$$H_{\text{true}} = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathbb{P}_{\text{data}}} \frac{\partial^2}{\partial \boldsymbol{w}^2} \mathcal{L}(f_{\boldsymbol{w}}(\boldsymbol{x}), y). \tag{8.2}$$

Both H_{batch} and H_{true} are $N \times N$ matrix functions of $\boldsymbol{w}$; H_{batch} is random but H_{true} is deterministic. Only in very specific cases and under strong simplifying assumptions can one hope to obtain the distribution of H_{batch} or the value of H_{true} from $\mathcal{L}, \mathbb{P}_{\text{data}}$ and $f_{\boldsymbol{w}}$. Inspired by the success of many random matrix theory applications, e.g. in Physics, we will instead seek to capture the essential features of deep neural network Hessians in a sufficiently general random matrix model.

We introduce the following objects:

- A sequence (in N) of random real symmetric $N \times N$ matrices X. X possesses a limiting spectral probability measure μ, i.e. if $\lambda_1, \ldots, \lambda_N$ are the eigenvalues of X then

$$\frac{1}{N}\sum_{i=1}^{N}\delta_{\lambda_i} \to \mu \tag{8.3}$$

  weakly almost surely. We further assume that μ has compact support and admits a smooth density with respect to Lebesgue measure.

- A sequence (in N) of deterministic real symmetric $N \times N$ matrices A with eigenvalues

$$\theta_1, \ldots, \theta_p, \xi_1, \ldots \xi_{N-p-q}, \theta'_1, \ldots, \theta'_q \tag{8.4}$$

  for fixed integers p, q. We assume the existence of limiting measure ν such that, weakly,

$$\frac{1}{N-p-q}\sum_{i=1}^{N-p-q}\delta_{\xi_i} \to \nu \tag{8.5}$$

  where ν is a compactly supported probability measure. The remaining eigenvalues satisfy

$$\theta_1 > \ldots > \theta_p > \mathtt{r}(\nu), \ \theta'_1 < \ldots < \theta'_q < \mathtt{l}(\nu). \tag{8.6}$$

  ν is also assumed to be of the form $\nu = \varepsilon\eta + (1-\varepsilon)\delta_0$ where η is a compactly supported probability measure which admits a density with respect to Lebesgue measure.

- A decreasing function $\mathtt{s} : \mathbb{N} \to (0, 1)$.

With these definitions, we construct the following model for the Hessian:

$$H_{\text{batch}} \equiv H = \mathtt{s}(b)X + A \tag{8.7}$$

where b is the batch size. We have dropped the subscript on H_{batch} for brevity. Note that H takes the place of the batch Hessian and A taken the place of the true Hessian. $\mathtt{s}(b)X$ takes the place of the random noise introduced by sampling a finite batch at which to evaluate the Hessian. $\mathtt{s}(b)$ is an overall scaling induced in X by the batch-wise averaging.

This model is almost completely general. Note that we allow the distribution of X and the value of A to depend on the position in weight space $\boldsymbol{w}$. The only restrictions imposed by the model are

1. the existence of ν;
2. the position of θ_i, θ'_j relative to the support of ν;

3. ν may only possess an atom at 0;

4. the fixed number of θ_i, θ'_j;

5. the existence of μ;

6. the existence of the scaling $\mathsf{s}(b)$ in batch size.

All of the above restrictions are discussed later in the section. Finally, we must introduce some properties of the noise model X in order to make any progress. We introduce the assumption that the eigenvectors of X obey *quantum unique ergodicity* (QUE) [BY17]. The precise meaning of this assumption and a thorough justification and motivation is given later in this section. For now it suffices to say that QUE roughly means that the eigenvectors of X are *delocalised* or that they behave roughly like the rows (or columns) of a uniform random $N \times N$ orthogonal matrix (i.e. a matrix with Haar measure). QUE is known to hold for standard ensembles in random matrix theory, such as quite general Wigner matrices, Wishart matrices, adjacency matrices of certain random graphs etc. Moreover, as discussed further section 8.1.5 below, QUE can be thought of as a property of quite general random matrix models.

8.1.2 Quantum unique ergodicity

Quantum unique ergodicity was introduced in Chapter 2 but for convenience we recall some details here. It is well known that the eigenvectors of quite general random matrices display a universal property of *delocalisation*, namely

$$|u_k|^2 \sim \frac{1}{N} \tag{8.8}$$

for any component u_k of an eigenvector $\boldsymbol{u}$. Universal delocalisation was conjectured by Wigner along with the Wigner surmise for adjacent eigenvalue spacing. Both of these properties, and the more familiar phenomenon of universal correlation functions on the microscopic scale have since been rigorously established for quite a variety of matrix models e.g. [EY17a; EY12; EKS19]. [BY17] show that the eigenvectors of generalised Wigner matrices obey *Quantum unique ergodicity*, a particular form of delocalisiation, stronger than the above statement. Specifically, they are shown to be approximately Gaussian in the following sense ([BY17] Theorem 1.2):

$$\sup_{\|\boldsymbol{q}\|=1} \sup_{\substack{I\subset[N], \\ |I|=n}} \left| \mathbb{E}P\left((N|\boldsymbol{q}^T\boldsymbol{u}_k|^2)_{k\in I}\right) - \mathbb{E}P\left((|\mathcal{N}_j|^2)_{j=1}^n\right)\right| \leqslant N^{-\varepsilon}, \tag{8.9}$$

for large enough N, where $\mathcal{N}_j$ are i.i.d. standard normal random variables, $(\boldsymbol{u}_k)_{k=1}^N$ are the normalised eigenvectors, P is any polynomial in n variables and $\varepsilon > 0$. Note that the set I in this statement is a subset of $[N]$ of *fixed size* n; n is not permitted to depend on N.

8.1.3 Batch Hessian outliers

Let $\{\lambda_i\}$ be the eigenvalues of H. To set the context of our results, let us first simplify and suppose momentarily that $\mathsf{s} = 1$ and, instead of mere QUE, X has eigenvectors distributed with Haar measure, and A is fixed rank, i.e. $\xi_i = 0\ \forall i$, then the results of [BN11] would apply and give

$$\lambda_j \overset{a.s.}{\to} \begin{cases} g_\mu^{-1}(1/\theta_j) & \text{if } \theta_j > 1/g_\mu(\mathtt{r}(\mu)), \\ \mathtt{r}(\mu) & \text{otherwise}, \end{cases} \tag{8.10}$$

for $j = 1,\dots,p$, and

$$\lambda_{N-j+1} \overset{a.s.}{\to} \begin{cases} g_\mu^{-1}(1/\theta'_j) & \text{if } \theta'_j < 1/g_\mu(\mathtt{l}(\mu)), \\ \mathtt{l}(\mu) & \text{otherwise}, \end{cases} \tag{8.11}$$

for $j = 1,\dots,q$. What follows is our main results for the outliers of H under the general conditions described above.

Theorem 8.1. *Let H be the Hessian matrix model defined in (8.7) and meeting all the conditions in Section 8.1. Then there exist $U_\varepsilon, L_\varepsilon \in \mathbb{R}$ such that, for $j = 1,\dots,p$,*

$$\lambda_j = \begin{cases} \omega^{-1}(\theta_j) & \textit{if } \omega^{-1}(\theta_j) > U_\varepsilon, \\ U_\varepsilon & \textit{otherwise.} \end{cases} \tag{8.12}$$

and for $j = 1,\dots,q$,

$$\lambda_{N-j+1} = \begin{cases} \omega^{-1}(\theta'_j) & \textit{if } \omega^{-1}(\theta_j) < L_\varepsilon, \\ L_\varepsilon & \textit{otherwise,} \end{cases} \tag{8.13}$$

and

$$\omega^{-1}(\theta) = \theta + s(b)R_\mu(s(b)\theta^{-1}) + \varepsilon s(b)^2 d_\eta(\theta)R'_\mu(s(b)\theta^{-1}) + \mathcal{O}(\varepsilon^2) \tag{8.14}$$

where we define $d_\eta(z) = g_\eta(\theta_j) - \theta_j^{-1}$.

An interlude on prior outlier results *It was conjectured in [BN11] that (8.10)-(8.11) still hold when X has delocalised eigenvectors in some sense, rather than strictly Haar. Indeed, a careful consideration of the proof in that work does reveal that something weaker than Haar would suffice, for example QUE. See in particular the proof of the critical Lemma 9.2 therein which can clearly be repeated using QUE. There is a considerable subtlety here, however, which is revealed best by considering more recent results on deformations of general Wigner matrices. [KY17] shows that very general deterministic deformations of general Wigner matrices possess an optimal anisotropic local law, i.e. $Y + B$ for Wigner Y and deterministic symmetric B. It is expected therefore that $Y + B$ has delocalised eigenvectors in the bulk. Consider the case where B is diagonal, and say that B has a fixed number of "spike" eigenvalues $\varphi_1 > \ldots > \varphi_r$ and remaining eigenvalues $\zeta_1,\dots,\zeta_{N-r}$ where* *the empirical measure of the ζ_i converges to some measure τ and $\varphi_r > r(\tau)$. We can then split $B = B_i + B_o$ where B_i contains only the ζ_j and B_o only the φ_j. The previously mentioned results applies to $Y + B_i$ and then we might expect the generalised result of [BN11] to apply to give outliers $g^{-1}_{\mu_{SC}\boxplus\tau}(1/\varphi_i)$ of $Y + B$. This contradicts, however, another result concerning precisely the the outliers of such generally deformed Wigner matrices. It was shown in [CD16] that the outliers of $Y + B$ are $\omega^{-1}(\varphi_j)$ where ω is the subordination function such that $g_{\mu_{SC}\boxplus\tau}(z) = g_\tau(\omega(z))$. These two expressions coincide when*

$$\begin{aligned}
&\omega^{-1}(z) = g^{-1}_{\mu_{SC}\boxplus\tau}(z^{-1})\\
\iff &\omega^{-1}(z) = \omega^{-1}(g_\tau^{-1}(z^{-1}))\\
\iff &g_\tau^{-1}(z^{-1}) = z\\
\iff &g_\tau(z) = z^{-1}\\
\iff &\tau = \delta_0,
\end{aligned} \tag{8.15}$$

i.e. only when B is in fact of negligible rank as $N \to \infty$. This apparent contradiction is resolved by the observation that the proof in [BN11] in fact relies implicitly on an isotropic local law. *Note in particular section 4.1, which translated to our context, would require $\boldsymbol{v}^T G_{Y+B_i}(z)\boldsymbol{v} \approx g_{\mu_{SC}\boxplus\tau}(z)$ with high probability for general unit vectors $\boldsymbol{v}$. Such a result holds if and only if $Y + B_i$ obeys an isotropic local law and is violated if its local law is instead anistropic, as indeed it is, thanks to the deformation.*

Proof of Theorem 8.1 The conditions on X required to invoke Theorem 8.3 from Section 8.2 are satisfied, so we conclude that

$$\hat{g}_H(z) = g_{\mu_b\boxplus\nu}(z) + o(1) = g_\nu(\omega(z)) + o(1) = \hat{g}_A(\omega(z)) + o(1) \tag{8.16}$$

where ω is the subordination function such that $g_{\mu_b\boxplus\nu}(z) = g_\nu(\omega(z))$ and μ_b is the limiting spectral measure of $\mathsf{s}(b)X$. The reasoning found in [CD16] then applies regarding the outliers of H. Indeed, suppose that λ is an outlier of H, i.e. λ is an eigenvalue of H contained in $\mathbb{R}\backslash\text{supp}(\mu\boxplus\nu)$. Necessarily $\hat{g}_H$ possesses a singularity at λ, and so $\hat{g}_A$ must have a singularity at $\omega(\lambda)$. For this singularity to persist for all N, $\omega(\lambda)$ must coincide with one of the outliers of A which, unlike the bulk eigenvalues ξ_j, remain fixed for all N. Therefore we have the following expressions for the outliers of H:

$$\{\omega^{-1}(\theta_j) \mid \omega^{-1}(\theta_j) \in \mathbb{R}\backslash\text{supp}(\mu_b \boxplus \nu)\} \cup \{\omega^{-1}(\theta'_j) \mid \omega^{-1}(\theta'_j) \in \mathbb{R}\backslash\text{supp}(\mu_b \boxplus \nu)\}. \tag{8.17}$$

We now consider ε to be small and analyse these outlier locations as a perturbation in ε. Firstly note that

$$g_{\mu_b}(z) = \int \frac{d\mu_b(x)}{z - x} = \int \frac{d\mu(x/\mathsf{s}(b))}{z - x} = \mathsf{s}(b)\int \frac{d\mu(x)}{z - \mathsf{s}(b)x} = g_\mu(z/\mathsf{s}(b)). \tag{8.18}$$

Also

$$\begin{aligned}
\omega^{-1}(z) &= g^{-1}_{\mu_b\boxplus\nu}(g_\nu(z)) \qquad (8.19)\\
&= R_{\mu_b}(g_\nu(z)) + g_\nu^{-1}(g_\nu(z))\\
&= R_{\mu_b}(g_\nu(z)) + z. \qquad (8.20)
\end{aligned}$$

We now must take care in computing R_{μ_b} from g_{μ_b}. Recall that the R-transform of a measure is defined as a formal power series [AGZ10]

$$R(z) = \sum_{n=0}^{\infty} k_{n+1} z^n \tag{8.21}$$

where k_n is the n-th cumulant of the measure. It is known [AGZ10] that $k_n = C_n$ where the functional inverse of the Stieljtes transform of the measure is given by the formal power series

$$K(z) = \frac{1}{z} + \sum_{n=1} C_n z^{n-1}. \tag{8.22}$$

Now let m_n be the n-th moment of μ and similarly let $m_n^{(b)}$ be the n-th moment of μ_b, so formally

$$g_\mu(z) = \sum_{n\geqslant 0} m_n z^{-(n+1)}, \quad g_{\mu_b}(z) = \sum_{n\geqslant 0} m_n^{(b)} z^{-(n+1)}. \tag{8.23}$$

Also let k_n be the n-th cumulant of μ and $k_n^{(b)}$ be the n-th cumulant of μ_b. Referring to the proof of Lemma 5.3.24 in [AGZ10] we find the relations

$$m_n = \sum_{r=1}^{n} \sum_{\substack{0\leqslant i_1,\dots,i_r \leqslant n-r \\ i_1+\dots+i_r = n-r}} k_r m_{i_1} \dots m_{i_r}, \tag{8.24}$$

$$m_n^{(b)} = \sum_{r=1}^{n} \sum_{\substack{0\leqslant i_1,\dots,i_r \leqslant n-r \\ i_1+\dots+i_r = n-r}} k_r^{(b)} m_{i_1}^{(b)} \dots m_{i_r}^{(b)}. \tag{8.25}$$

Note, in particular, that $m_1 = k_1$. But clearly the moments of μ_b have a simple scaling in $\mathsf{s}(b)$, namely $m_n^{(b)} = \mathsf{s}(b)^n m_n$, hence

$$m_n = \mathsf{s}(b)^{-n} \sum_{r=1}^{n} \sum_{\substack{0\leqslant i_1,\dots,i_r \leqslant n-r \\ i_1+\dots+i_r = n-r}} k_r^{(b)} m_{i_1} \dots m_{i_r} \mathsf{s}(b)^{n-r} \tag{8.26}$$

from which we deduce $k_n^{(b)} = \mathsf{s}(b)^n k_n$, which establishes that $R_{\mu_b}(z) = \mathsf{s}(b) R_\mu(\mathsf{s}(b)z)$. Recalling (8.20) we find

$$\omega^{-1}(z) = \mathsf{s}(b) R_\mu(\mathsf{s}(b) g_\nu(z)) + z. \tag{8.27}$$

The form of ν gives

$$g_\nu(z) = (1-\epsilon)\int \frac{dt}{z-t}\delta_0(t) + \epsilon \int \frac{d\eta(t)}{t-z} = \frac{1-\epsilon}{z} + \epsilon g_\eta(z) = \frac{1}{z} + \epsilon\left(g_\eta(z) - \frac{1}{z}\right) \tag{8.28}$$

and so we can expand to give

$$\begin{aligned}\omega^{-1}(\theta_j) &= \theta_j + \mathsf{s}(b) R_\mu(\mathsf{s}(b)\theta_j^{-1}) + \epsilon \mathsf{s}(b)^2 \left(g_\eta(\theta_j) - \theta_j^{-1}\right) R_\mu'(\mathsf{s}(b)\theta_j^{-1}) + \mathcal{O}(\epsilon^2) \\ &= \theta_j + \mathsf{s}(b) R_\mu(\mathsf{s}(b)\theta_j^{-1}) + \epsilon \mathsf{s}(b)^2 d_\eta(\theta_j) R_\mu'(\mathsf{s}(b)\theta_j^{-1}) + \mathcal{O}(\epsilon^2)\end{aligned} \tag{8.29}$$

where we have defined $d_\eta(z) = g_\eta(\theta_j) - \theta_j^{-1}$. The argument with the lower outliers $\{\theta'_j\}_{j=1}^q$ is identical.

The problem of determining the support of $\mu_b \boxplus \nu$ is difficult and almost certainly analytically intractable, with [BES20] containing the most advanced results in that direction. However overall, we have a model for deep neural network Hessians with a spectrum consisting, with high-probability, of a compactly supported bulk $\mu_b \boxplus \nu$ and a set of outliers given by (8.29) (and similarly for θ'_j) subject to (8.17). The constants $L_\varepsilon, U_\varepsilon$ in the statement (8.12)-(8.13) of the theorem are simply the lower and upper edges of the support of $\text{supp}(\mu_b \boxplus \nu)$. ■

Note that (8.29) reduces to outliers of the form $\theta_j + \mathsf{s}(b)^2 R_\mu(\theta_j^{-1})$ if $\varepsilon = 0$ or $d_\eta = 0$, as expected from [BN11][1].

(8.29) is a generalised form of the result used in [GZR20]. We have the power series

$$R_\mu(\mathsf{s}(b)\theta_j^{-1}) = k_1^{(\mu)} + \frac{k_2^{(\mu)}\mathsf{s}(b)}{\theta_j} + \frac{k_3^{(\mu)}\mathsf{s}(b)^2}{\theta_j^2} + \ldots, \tag{8.30}$$

$$d_\eta(\theta_j) = \frac{m_1^{(\eta)}}{\theta_j^2} + \frac{m_2^{(\eta)}}{\theta_j^3} + \ldots \tag{8.31}$$

where $m_n^{(\eta)}$ are the moments of η and $k_n^{(\mu)}$ are the cumulants of μ. In the case that the spikes θ_j are large enough, we approximate by truncating these power series to give

$$\omega^{-1}(\theta_j) \approx \theta_j + \mathsf{s}(b) m_1^{(\mu)} + \mathsf{s}(b)^2 k_2^{(\mu)} \left(\frac{1}{\theta_j} + \frac{\varepsilon m_1^{(\eta)}}{\theta_j^2} \right) \tag{8.32}$$

where the approximation is more precise for larger b and smaller ε and we have used the fact that the first cumulant of any measure matches the first moment. One could consider for instance a power law for $\mathsf{s}(b)$, i.e.

$$\omega^{-1}(\theta_j) \approx \theta_j + \frac{k_1^{(\mu)}}{b^v} + \frac{k_2^{(\mu)}}{b^{2v}} \left(\frac{1}{\theta_j} + \frac{\varepsilon m_1^{(\eta)}}{\theta_j^2} \right) = \theta_j + \frac{m_1^{(\mu)}}{b^v} + \frac{k_2^{(\mu)}}{b^{2v}} \left(\frac{1}{\theta_j} + \frac{\varepsilon m_1^{(\eta)}}{\theta_j^2} \right) \tag{8.33}$$

for some $v > 0$. In the case that μ is a semicircle, then all cumulants apart from the second vanish, so setting $\varepsilon = 0$ recovers *exactly*

$$\omega^{-1}(\theta_j) = \theta_j + \frac{\sigma^2}{4b^{2v}\theta_j} \tag{8.34}$$

where σ is the radius of the semicircle. To make the link with [GZR20] obvious, we can take $v = 1/2$ and μ to be the semicircle, so giving

$$\omega^{-1}(\theta_j) \approx \theta_j + \frac{\sigma^2}{4b\theta_j} \tag{8.35}$$

where we have truncated $\mathcal{O}(\varepsilon)$ term. We present an argument in favour of the $v = 1/2$ power law below, but we allow for general v when comparing to experimental data.

[1] Note that $d_\eta = 0 \iff \eta = \delta_0$ which is clearly equivalent (in terms of ν) to $\varepsilon = 0$.

Remark 8.1. It is quite possible for μ's density to have a sharp spike at the origin, or even for μ to contain a δ atom at 0, as observed empirically in the spectra of deep neural network Hessians.

8.1.4 Experimental results

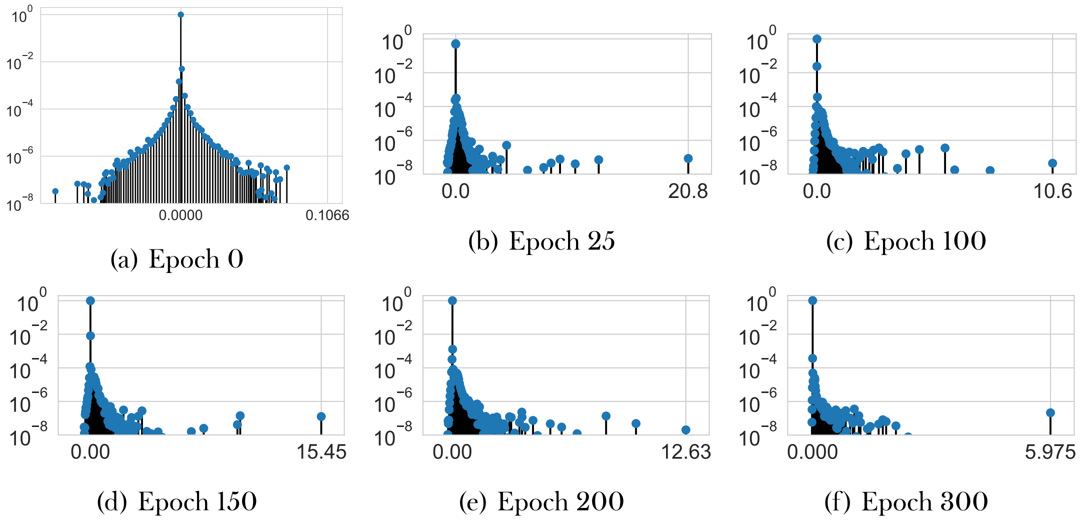
The random matrix Hessian model introduced above is quite general and abstract. Necessarily the measures μ and η must be allowed to be quite general as it is well established experimentally [Pap18; Gra20a; BGK22] that real-world deep neural network Hessians have spectral bulks that are not familiar as being any standard canonical examples from random matrix theory. That being said, the approximate form in (8.33) gives quite a specific form for the Hessian outliers. In particular, the constants $m_1^{(\mu)}, m_1^{(\eta)}$ and $m_2^{(\mu)}, \varepsilon > 0$ are shared between all outliers at all batch sizes. If the form of the Hessian outliers seen in (8.33) is not observed experimentally, it would suggest at least one of the following does not hold:

1. batch sampling induces a simple multiplicative scaling on the Hessian noise (8.7);
2. the true Hessian is approximately low-rank (as measured by ε) and has a finite number of outliers;
3. the Hessian noise model X has QUE.

In view of this third point, agreement with (8.33) provides an indirect test for the presence of universal random matrix statistics in deep neural network Hessians.

We can use Lanczos power methods [MS06] to compute good approximations to the top few outliers in the batch Hessian spectra of deep neural networks [GZR20]. Indeed the so-called Pearlmutter trick [Pea94] enables efficient numerical computation of Hessian-vector products, which is all that one requires for power methods. Over a range of batch sizes, we compute the top 5 outliers of the batch Hessian for 10 different batch seeds. We repeat this procedure at every 25 epochs throughout the training of two standard deep neural networks for computer vision tasks, VGG16 and WideResNet28 × 10, on the CIFAR100 dataset [KH+09] and at every epoch during the training of a simple multi-layer perceptron network on the MNIST dataset [LeC+98]. By the end of training each of the models have high test accuracy, specifically the VGG16 architecture which does not use batch normalisation, has a test accuracy of $\approx 75\%$, whereas the WideResNet28 × 10 has a test accuracy of $\approx 80\%$. The MLP has a test set accuracy of $\approx 95\%$. Full experimental details are given in Appendix D.

Remark 8.2. There is a subtlety with regard to obtaining the top outliers using the Lanczos power method. Indeed, since Lanczos provides, in some sense, an approximation to the whole spectrum of a matrix, truncating at m iterations for a $N \times N$ matrix cannot produce good approximations to all of the m top eigenvalues. In reality, experimental results [Pap18; GWG19] show that, for deep neural networks, and using sufficiently many iterations (m), the top r eigenvalues may be recovered, for $r \ll m$. We display some spectral plots of the full Lanczos results in the Figure 8.1 which demonstrate clearly a large number of outliers, and clearly more than 5. These are not intended to be exhaustive and we recommend references such as [Pap18] for detailed discussion of spectral densities like these. As a result, we can have confidence that our numerical procedure is indeed recovering approximations to the top few eigenvalues required for our experiments.



(a) Epoch 0

(b) Epoch 25

(c) Epoch 100

(d) Epoch 150

(e) Epoch 200

(f) Epoch 300

Figure 8.1: Approximate empirical spectral densities of the Hessian of the VGG16 network trained on MNIST at various stages from initialisation to the end of training. Note the clear presence of large outliers present already at epoch 25.

Let $\lambda_b^{(i,j,e)}$ be the top i-th empirical outlier (so $i = 1$ is the top outlier) for the j-th batch seed and a batch size of b for the model at epoch e. To compare the experimental results to our theoretical model, we propose the following form:

$$\lambda_b^{(i,j,e)} \approx \theta^{(i,e)} + \frac{\alpha^{(e)}}{b^{\upsilon}} + \frac{\beta^{(e)}}{b^{2\upsilon}}\left(\frac{1}{\theta^{(i,e)}} + \frac{\gamma^{(e)}}{(\theta^{(i,e)})^2}\right) \tag{8.36}$$

where $\beta^{(e)} > 0$ (as the second cumulant of a any measure of non-negative) and $\theta^{(i,e)} > \theta^{(i+1,e)} > 0$ for all i, e. The parameters $\alpha^{(e)}, \beta^{(e)}, \gamma^{(e)}$ and $\theta^{(i,e)}$ need to be fit to the data, which could be done with standard black-box optimisation to minimise squared error in (8.36), however we propose an alternative approach which reduces the number of free parameters and hence should regularise the optimisation problem. Observe that (8.36) is linear in the parameters $\alpha^{(e)}, \beta^{(e)}, \gamma^{(e)}$ so, neglecting the positivity constraint on $\beta^{(e)}$, we can in fact solve exactly for optimal values. Firstly let us define $\bar{\lambda}_b^{(i,e)}$ to be the empirical mean of $\lambda_b^{(i,j,e)}$ over the batch seed index j. Each epoch will be treated entirely separately, so let us drop the e superscripts to streamline the notation. We are then seeking to optimise $\alpha, \beta, \gamma, \theta^{(i)}$ to minimise

$$E = \sum_{i,b}\left(\bar{\lambda}_b^{(i)} - \theta^{(i)} - \frac{\alpha}{b^{\upsilon}} - \frac{\beta}{\theta^{(i)} b^{2\upsilon}} - \frac{\beta\gamma}{b^{2\upsilon}(\theta^{(i)})^2}\right)^2. \tag{8.37}$$

Now make the following definitions

$$y_{ib} = \bar{\lambda}_b^{(i)} - \theta^{(i)},\ \boldsymbol{x}_{ib} = \begin{pmatrix} b^{-v} \\ (\theta^{(i)} b)^{-2v} \\ (b^{2v}(\theta^{(i)})^2)^{-1} \end{pmatrix},\ \boldsymbol{w} = \begin{pmatrix} \alpha \\ \beta \\ \beta\gamma \end{pmatrix}, \tag{8.38}$$

so that

$$E = \sum_{i,b} (y_{ib} - \boldsymbol{w}^T \boldsymbol{x}_{ib})^2. \tag{8.39}$$

Finally we can define the n-dimensional vector $\boldsymbol{Y}$ by flattening the matrix $(y_{ib})_{ib}$, and the $3 \times n$ matrix $\boldsymbol{X}$ by stacking the vectors $\boldsymbol{x}_{ib}$ and then flattening of the i, b indices. That done, we have have a standard linear regression problem with design matrix $\boldsymbol{X}$ and parameters $\boldsymbol{w}$. For fixed θ, the global minimum of E is then attained at parameters

$$\boldsymbol{w}^*(\boldsymbol{\theta}) = (\boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X}\boldsymbol{Y} \tag{8.40}$$

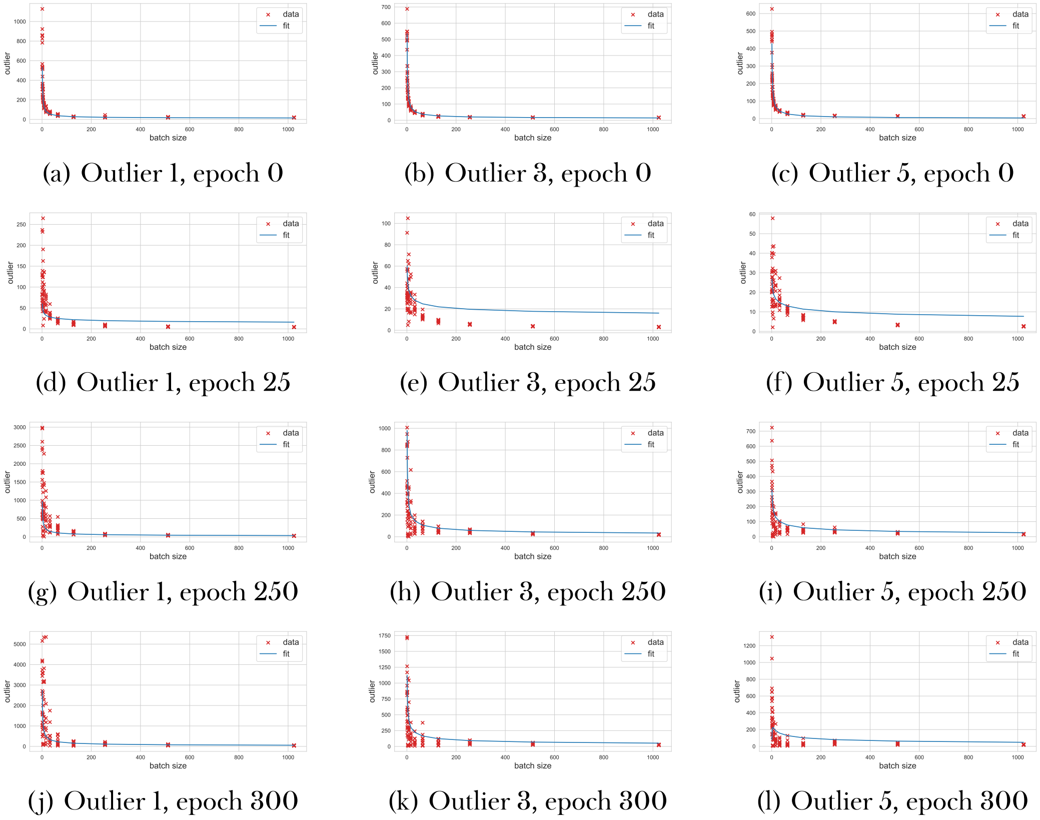
where the dependence on the parameters $\boldsymbol{\theta}$ is through $\boldsymbol{Y}$ and $\boldsymbol{X}$ as above. We thus have

$$\alpha = w_1^*, \beta = w_2^*, \gamma = w_3^* / w_2^*$$

and can plug these values back in to (8.37) to obtain an optimisation problem only over the $\theta^{(i)}$. There is no closed form solution for the optimal $\theta^{(i)}$ for this problem, so we fit them using gradient descent. The various settings and hyperparameters of this optimisation were tuned by hand to give convergence and are detailed in D.3. To address the real constraint $\beta > 0$, we add a penalty term to the loss (8.37) which penalises values of $\theta^{(i)}$ leading to negative values of β. The constraint $\theta^{(i)} > \theta^{(i+1)} > 0$ is implemented using a simple differentiable transformation detailed in D.2.. Finally, the exponent v is selected by fitting the parameters for each v in $\{-0.1, -0.2, \ldots, -0.9\}$ and taking the value with the minimum mean squared error E.

The above process results in 12 fits for VGG and Resnet and 10 for MLP (one per epoch). For each of these, we have a theoretical fit for each of the 5 top outliers as a function of batch size which can be compared graphically to the data, resulting in $(2 \times 12 + 10) \times 5 = 170$ plots. Rather than try to display them all, we will select a small subset that illustrates the key features. Figure 8.2 shows results for the Resnet at epochs 0 (initialisation), 25, 250 and 300 (end of training) and outliers 1, 3 and 5. Between the three models, the Resnet shows consistently the best agreement between the data and the parametric form (8.36). The agreement is excellent at epoch 0 but quickly degrades to that seen in the second row of Figure 8.2, which is representative of the early and middle epochs for the Resnet. Towards the end of training the Resnet returns to good agreement between theory and data, as demonstrated in the third and fourth rows of Figure 8.2 at epochs 250 and 300 respectively.

The VGG16 also has excellent agreement between theory and data at epoch 0, and thereafter is similar to the early epochs of the Resnet, i.e. reasonable, but not excellent, until around epoch 225



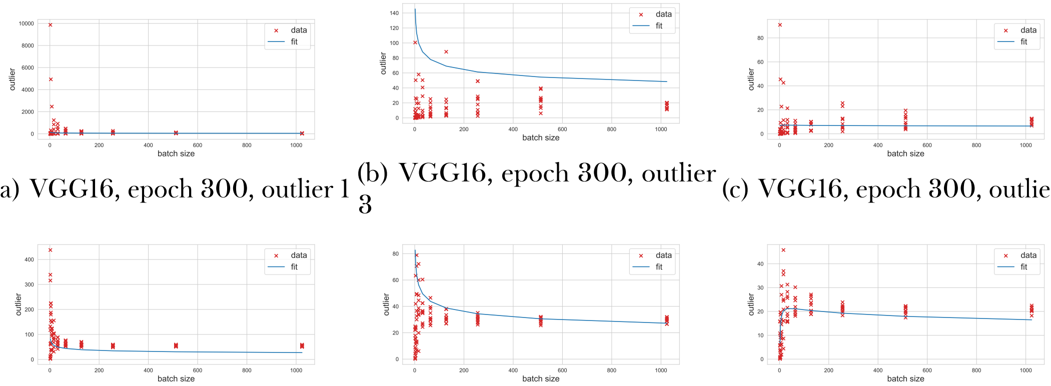
(a) Outlier 1, epoch 0
(b) Outlier 3, epoch 0
(c) Outlier 5, epoch 0
(d) Outlier 1, epoch 25
(e) Outlier 3, epoch 25
(f) Outlier 5, epoch 25
(g) Outlier 1, epoch 250
(h) Outlier 3, epoch 250
(i) Outlier 5, epoch 250
(j) Outlier 1, epoch 300
(k) Outlier 3, epoch 300
(l) Outlier 5, epoch 300

Figure 8.2: The batch-size scaling of the outliers in the spectra of the Hessians of the Resnet loss on CIFAR100. Training epochs increase top-to-bottom from initialisation to final trained model. Left-to-right the outlier index varies (outlier 1 being the largest). Red cross show results from Lanczos approximations over 10 samples (different batches) for each batch size. The blue lines are parametric power law fits of the form (8.36).

where the agreement starts to degrade significantly until the almost complete failure at epoch 300 shown in the first row of Figure 8.3. The MLP has the worst agreement between theory and data, having again excellent agreement at epoch 0, but really quite poor agreement even by epoch 1, as shown in the second row of Figure 8.3.

The experimental results show an ordering Resnet > VGG > MLP, in terms of how well the random matrix theory loss surface predictions explain the Hessian outliers. We conjecture that this relates to the difficulty of the loss surfaces. Resnets are generally believed to have smoother, simpler loss surfaces [Li+18] and be easier to train than other architectures, indeed the residual connections were originally introduced for precisely this reason. The VGG is generally more sensitive to training set-up, requiring well-tuned hyperparameters to avoid unstable or unsuccessful training (see Chapter 6 [GB22]). The MLP is perhaps too small to benefit from high-dimensional highly over-parametrised effects.

The parameter values obtained for all models over all epochs are shown in Figure 8.4, with a



(a) VGG16, epoch 300, outlier 1

(b) VGG16, epoch 300, outlier 3

(c) VGG16, epoch 300, outlier 5

(d) MLP, epoch 1, outlier 1

(e) MLP, epoch 1, outlier 3

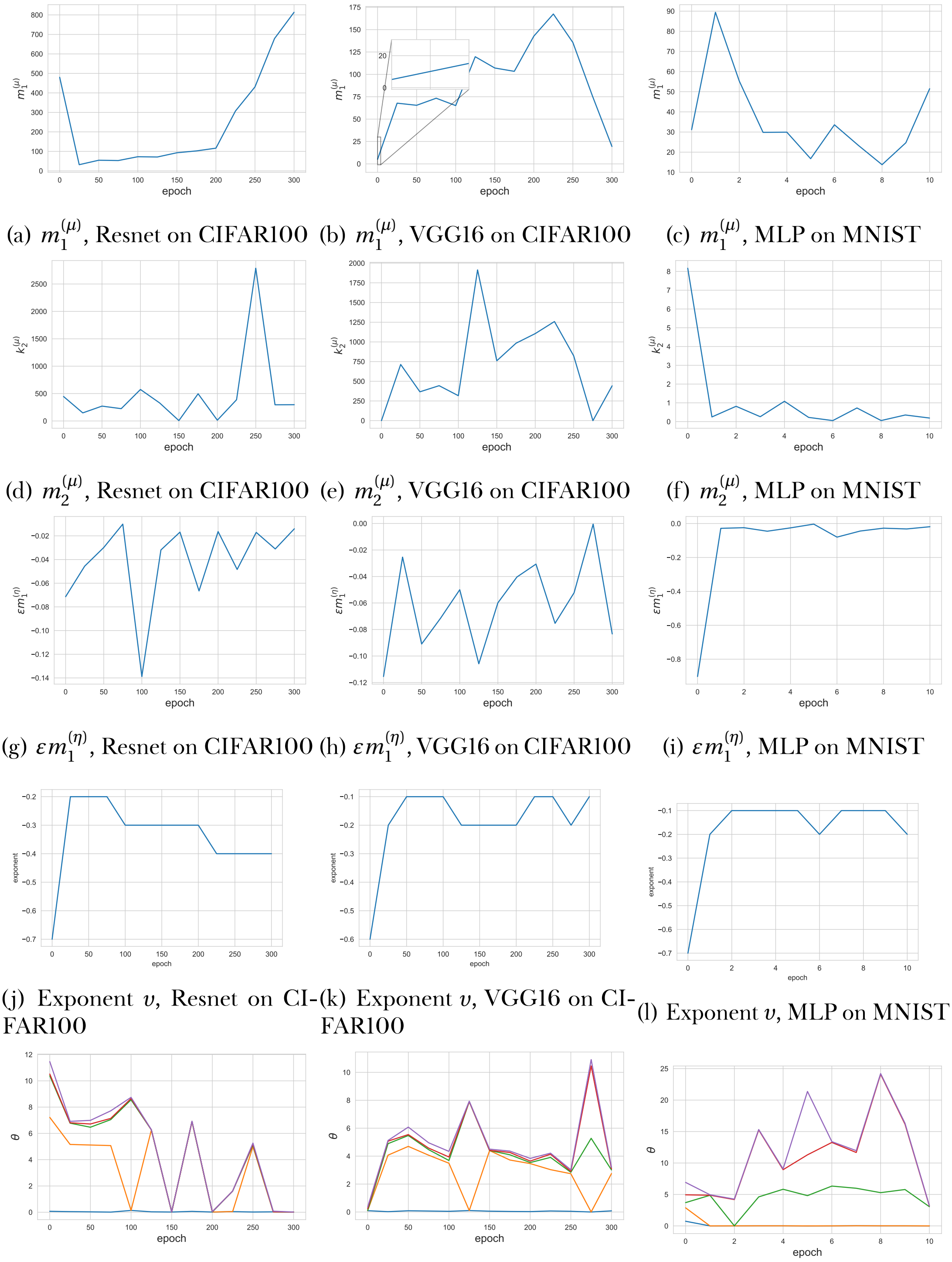
(f) MLP, epoch 1, outlier 5

Figure 8.3: Left-to-right the outlier index varies (outlier 1 being the largest). Red cross show results from Lanczos approximations over 10 samples (different batches) for each batch size. The blue lines are parametric power law fits of the form (8.36). This plot show the final epoch (300) for the VGG16 on CIFAR100 and the first epoch for the MLP on MNIST, both being examples of the parametric fit failing to match the data.

column for each model. There are several interesting features to draw out of these plots, however note that we cannot meaningfully interpret the parameters for the MLP beyond epoch 0, as the agreement with (8.36) is so poor. Firstly consider the parameter $m_1^{(\mu)}$, which is interpreted as the first moment (i.e. mean) of the spectral density of the noise matrix X. $m_1^{(\mu)} = 0$ is significant, as it is seen in the case of the a symmetric measure μ, such as the Wigner semicircle used by [GZR20]. For the VGG, $m_1^{(\mu)}$ starts close to 0 (Figure 8.4b) and generally grows with training epochs (note that the right hand side of this plot is not trustworthy, as we have observed that the agreement with (8.36) does not survive to the end of training). For the Resnet, we see a similar upwards trend (Figure 8.4a), with the notable exception that of initialisation (epoch 0). These two observations together, suggest that training encourages a skew in the spectrum of X away from symmetry around 0, however for some structural reason the Resnet is highly skewed at initialisation.

Note that for all models this parameter starts close to 0 and generally grows with training epochs, noting that the right hand side of Figure 8.4b at the higher epochs should be ignored owing to the bad fit discussed above.

It is interesting also to observe that $\varepsilon m_1^{(\eta)}$ remains small for all epochs particularly compared to $m_1^{(\mu)}, k_2^{(\mu)}$. This is consistent with the derivation of (8.36), which relies on ε being small, however we emphasise that *this was not imposed as a numerical constraint* but arises naturally from the data. Recall that the magnitude of $\varepsilon m_1^{(\eta)}$ measures the extent of the deviation of A from being exactly low rank, so its small but non-zero values suggest that it is indeed important to allow for the true Hessian to have non-zero rank in the $N \to \infty$ limit. Finally, we comment that the best exponent is generally not $\upsilon = 1/2$. Again, the results from the Resnet are the most reliable and they appear to show that the batch scaling, as characterised by υ, is not constant throughout training, particularly comparing epoch 0 and epoch 300, say.



(a) $m_1^{(\mu)}$, Resnet on CIFAR100 (b) $m_1^{(\mu)}$, VGG16 on CIFAR100 (c) $m_1^{(\mu)}$, MLP on MNIST

(d) $m_2^{(\mu)}$, Resnet on CIFAR100 (e) $m_2^{(\mu)}$, VGG16 on CIFAR100 (f) $m_2^{(\mu)}$, MLP on MNIST

(g) $\varepsilon m_1^{(\eta)}$, Resnet on CIFAR100 (h) $\varepsilon m_1^{(\eta)}$, VGG16 on CIFAR100 (i) $\varepsilon m_1^{(\eta)}$, MLP on MNIST

(j) Exponent υ, Resnet on CIFAR100 (k) Exponent υ, VGG16 on CIFAR100 (l) Exponent υ, MLP on MNIST

(m) $\theta_1,\ldots,\theta_5$, Resnet on CIFAR100 (n) $\theta_1,\ldots,\theta_5$, VGG16 on CIFAR100 (o) $\theta_1,\ldots,\theta_5$, MLP on MNIST

Figure 8.4: The parameter values produced when fitting experimental neural network Hessian outlier data to (8.36).

8.1.5 Justification and motivation of QUE

We recall the various types of local law first introduced in section 2.7. All provide high probability control on the error between the (random) matrix Green's function $G(z) = (z - X)^{-1}$ and certain deterministic equivalents. In all cases we use the set

$$\boldsymbol{S} = \{E + i\eta \in \mathbb{C} \mid |E| \leqslant \omega^{-1},\ N^{-1+\omega} \leqslant \eta \leqslant \omega^{-1}\} \tag{8.41}$$

for $\omega \in (0,1)$ and the local law statements holds for all (large) $D > 0$ and (small) $\xi > 0$ and for all large enough N. The *averaged local law* states:

$$\sup_{z\in\boldsymbol{S}} \mathbb{P}\left(\left|\frac{1}{N}\mathrm{Tr}G(z) - g_\mu(z)\right| > N^\xi\left(\frac{1}{N\eta} + \sqrt{\frac{\Im g_\mu(z)}{N\eta}}\right)\right) \leqslant N^{-D}. \tag{8.42}$$

The *isotropic local law* states:

$$\sup_{\|\boldsymbol{u}\|,\|\boldsymbol{v}\|=1,z\in\boldsymbol{S}} \mathbb{P}\left(|\boldsymbol{u}^T G(z)\boldsymbol{v} - g_\mu(z)| > N^\xi\left(\frac{1}{N\eta} + \sqrt{\frac{\Im g_\mu(z)}{N\eta}}\right)\right) \leqslant N^{-D}. \tag{8.43}$$

The *anisotropic local law* states:

$$\sup_{\|\boldsymbol{u}\|,\|\boldsymbol{v}\|=1,z\in\boldsymbol{S}} \mathbb{P}\left(|\boldsymbol{u}^T G(z)\boldsymbol{v} - \boldsymbol{u}^T\Pi(z)\boldsymbol{v}| > N^\xi\left(\frac{1}{N\eta} + \sqrt{\frac{\Im g_\mu(z)}{N\eta}}\right)\right) \leqslant N^{-D} \tag{8.44}$$

where $\Pi(\cdot)$ is an $N \times N$ deterministic matrix function on $\mathbb{C}$. The *entrywise local law* states:

$$\sup_{z\in\boldsymbol{S},1\leqslant i,j\leqslant N} \mathbb{P}\left(|G_{ij}(z) - \Pi_{ij}(z)| > N^\xi\left(\frac{1}{N\eta} + \sqrt{\frac{\Im g_\mu(z)}{N\eta}}\right)\right) \leqslant N^{-D}. \tag{8.45}$$

As mentioned above, quantum unique ergodicity was proved for general Wigner matrices in [BY17]. It appears that the key ingredient in the proof of QUE (8.9) in [BY17] is the isotropic local semicircle law (8.43) for general Wigner matrices. Indeed, all the intermediate results in Sections 4 of [BY17] take only (8.43) and general facts about the Dyson Brownian Motion eigenvector flow given by

$$d\lambda_k = \frac{dB_{kk}}{\sqrt{N}} + \left(\frac{1}{N}\sum_{\ell\neq k}\frac{1}{\lambda_k - \lambda_\ell}\right)dt, \tag{8.46}$$

$$du_k = \frac{1}{\sqrt{N}}\sum_{\ell\neq k}\frac{dB_{kl}}{\lambda_k - \lambda_\ell}u_\ell - \frac{1}{2N}\sum_{\ell\neq k}\frac{dt}{(\lambda_k - \lambda_\ell)^2}u_k. \tag{8.47}$$

This can be generalised to

$$d\lambda_k = \frac{dB_{kk}}{\sqrt{N}} + \left(-V(\lambda_i) + \frac{1}{N}\sum_{\ell\neq k}\frac{1}{\lambda_k - \lambda_\ell}\right)dt, \tag{8.48}$$

$$du_k = \frac{1}{\sqrt{N}}\sum_{\ell\neq k}\frac{dB_{kl}}{\lambda_k - \lambda_\ell}u_\ell - \frac{1}{2N}\sum_{\ell\neq k}\frac{dt}{(\lambda_k - \lambda_\ell)^2}u_k. \tag{8.49}$$

where V is a potential function. Note that the eigenvector dynamics are unaffected by the presence of the potential V, so we expect to be able to generalise the proof of [KY17] to any random matrix ensemble with an isotropic local law by defining the potential V so that the invariant ensemble with distribution $Z^{-1}e^{-N\mathrm{Tr}V(X)}dX$ has equilibrium measure μ (Z is a normalisation constant). We show how to construct such a V from μ in Section 8.3.

The arguments so far suffice to justify a generalisation of the "dynamical step" in the arguments of [BY17], so it remains to consider the "comparison step". The dynamical step establishes QUE for the matrix ensemble with a small Gaussian perturbation, but in the comparison step one must establish that the perturbation can be removed without breaking QUE. To our knowledge no such argument has been articulated beyond generalized Wigner matrices, with the independence of entries and comparable scale of variances being critical to the arguments given by [BY17]. Our guiding intuition is that QUE of the form (8.9) is a general property of random matrices and can reasonably be expected to hold in most, if not all, cases in which there is a local law and universal local eigenvalue statistics are observed. At present, we are not able to state a precise result establishing QUE in sufficient generality to be relevant for this work, so we shall take it as an assumption.

Assumption 8.2. Let X be an ensemble of $N \times N$ real symmetric random matrices. Assume that X admits a limiting spectral measure is μ with Stieljtes transform m. Suppose that the isotropic local law (8.43) holds for X with μ. Then there is some set $\mathbb{T}_N \subset [N]$ with $|\mathbb{T}_N^c| = o(N)$ such that with $|I| = n$, for any polynomial P in n indeterminates, there exists some $\varepsilon(P) > 0$ such that for large enough N we have

$$\sup_{\substack{I\subset\mathbb{T}_N, |I|=n,\\ \|\boldsymbol{q}\|=1}} \left|\mathbb{E}\left(P\left(\left(N(\boldsymbol{q}^T\boldsymbol{u}_k)^2\right)_{k\in I}\right)\right) - \mathbb{E}\left(P\left(\left(|\mathcal{N}_j|^2\right)_{k\in I}\right)\right)\right| \leqslant N^{-\varepsilon}. \tag{8.50}$$

Note that the isotropic local law in Assumption 8.2 can be obtained from the weaker entrywise law (8.45) as in Theorem 2.14 of [Blo+14] provided there exists a $C > 0$ such that $\mathbb{E}|X_{ij}|^2 \leqslant CN^{-1}$ for all i, j and there exists $C_p > 0$ such that $\mathbb{E}|\sqrt{N}X_{ij}|^p \leqslant C_p$ for all i, j and integer $p > 0$.

Remark 8.3. In [BY17] the restriction $I \subset \mathbb{T}_N$ is given for the explicit set

$$\mathbb{T}_N = [N]\backslash\{(N^{1/4}, N^{1-\delta}) \cup (N - N^{1-\delta}, N - N^{1/4})\} \tag{8.51}$$

for some $0 < \delta < 1$. In the case of generalised Wigner matrices, this restriction on the indices has since been shown to be unnecessary [BL22; Ben20; BL21]. In our context, we could simply take as an assumption all results holds with $\mathbb{T}_N = [N]$, however our results can in fact be proved using only the above assumption that $|\mathbb{T}_N^c| = o(N)$, so we shall retain this weaker form of the assumptions.

This section is not intended to prove QUE from explicit known properties of deep neural network Hessians, but rather to provide justification for it as a reasonable modeling assumption in the noise model for Hessians defined in section 8.1.1. We have shown how QUE can be obtained from an isotropic (or entrywise) local law beyond the Wigner case. It is important to go beyond Wigner or any other standard random matrix ensemble, as we have observed above that the standard macroscopic spectral densities of random matrix theory such as the semicircle law are not observed in practice. That said, we are not aware of any results establishing QUE in the more general case of anistropic local laws, and this appears to be a very significant technical challenge. We must finally address why a local law assumption, isotropic or otherwise, may be reasonable for the noise matrix X in our Hessian model. Over the last decade or so, universal local statistics of random matrices in the form of k-point correlation functions on the appropriate microscopic scale have been established for a litany of random matrix ensembles. An immediate consequence of such results is that, on the scale of unit mean eigenvalue spacing, Wigner's surmise holds to a very good approximation, depending only on the symmetry class (orthogonal, unitary or symplecitic). Such universality results are rather older for invariant ensembles [Dei99; EY17a] and can be established with orthogonal polynomial techniques, however the recent progress focusing on non-invariant ensembles, beginning with Wigner matrices [EYY12] and proceeding to much more general ensembles [EKS19], is built on a very general "three step strategy" (though see [EY12] for connections between universality in invariant and non-invariant ensembles). As with the QUE proof discussed above, the key ingredient in these proofs, as part of the three step strategy [EY17a], is establishing a local law. The theoretical picture that has emerged is that, for very general random matrices, when universal local eigenvalue statistics are observed in random matrices, it is due to the mechanism of short time scale relaxation of local statistics under Dyson Brownian Motion made possible by a local law.

In Chapter 7 [BGK22] we observed that universal local eigenvalue statistics do indeed appear to be present in the Hessian of real, albeit quite small, deep neural networks. Given all of this context, we propose that a local law assumption of some kind is reasonable for deep neural network Hessians and not particularly restrictive. As we have shown, if we are willing to make the genuinely restrictive assumption of an isotropic local law for the Hessian noise model, then QUE follows. However an anistropic local law is arguably more plausible as we expect deep neural networks Hessians to contain a good deal of dependence between entries, and such correlations are know to generically lead to anisotropic local laws [EKS19].

8.1.6 Motivation of true Hessian structure

In this section we revisit and motivate the assumptions made about the Hessian in Section 8.1.1. Firstly note that one can always define $A = \mathbb{E}H_{\text{batch}}$ and it is natural then to associate A with the true Hessian H_{true}. In light of (8.1), it is natural to expect some fixed form of the law for $H_{\text{batch}} - A$ for any batch size, but with an overall scaling $\mathsf{s}(b)$, which must naturally be decreasing in b as experimental results show that the overall spectral width of the batch Hessians of neural networks decreases with increasing batch size. Next we address the assumptions made about the spectrum of A. The first assumption one might think to make is that A has fixed rank relative to N, with spectrum consisting only of the spikes θ_i, θ'_j. Indeed, it has been repeatedly observed, in our own experiments and others [Pap18; GZR20], that neural network Hessians contain a number of spectral outliers separated from the spectral bulk. It is natural to conjecture that such outliers arise from some outliers in an underlying structured deterministic matrix of which the batch Hessian is a noisy version, as in the case of BBP style phase transitions in random matrix theory. The outliers in neural network Hessians have been associated with inter-class separation in the case of classification models [Pap19] and it can be observed that spectra lack (or have smaller and fewer) outliers at the start of training, or if they are intentionally trained to give poor (i.e. random) predictive performance. That being said, in almost any experiment with sensibly trained neural networks, spectral outliers are observed, and over a range of batch sizes (and hence noise levels) suggesting that some of the spike eigenvalues in the true Hessian are above the phase transition threshold.

Behind such an assumption is the intuition that the data distribution does not depend on N and so, in the over-parametrised limit $N \to \infty$, the overwhelming majority of directions in weight space are unimportant. The form we take for A in the above is a strict generalisation of the fixed rank assumption; A still has a fixed number of spiked directions, but the parameter ε controls the rank of A. Since any experimental investigation is necessarily limited to $N < \infty$, the generalisation to $\varepsilon > 0$ is particularly important. Compact support of the measures μ and η is consistent with experimental observations of deep neural network Hessian spectra.

8.1.7 The batch size scaling

Our experimental results considered $\mathsf{s}(b) = b^{-\upsilon}$ and $\upsilon = 1/2$ is the value required to give agreement with [GZR20], a choice which we now justify. From (8.1) we have

$$H_{\text{batch}} = \frac{1}{b}\sum_{i=1}^{b}\left(H_{\text{true}} + X^{(i)}\right) \tag{8.52}$$

where $X^{(i)}$ are i.i.d. samples from the law of X. Suppose that the entries X_{ij} were Gaussian, with $\text{Cov}(X_{ij}, X_{kl}) = \Sigma_{ij,kl}$. Then $Z = X_{ij}^{(p)} + X_{ij}^{(q)}$ has

$$\text{Cov}(Z_{ij}, Z_{kl}) = \mathbb{E}X_{ij}^{(p)}X_{kl}^{(p)} + \mathbb{E}X_{ij}^{(q)}X_{kl}^{(q)} - \mathbb{E}X_{ij}^{(p)}\mathbb{E}X_{kl}^{(p)} - \mathbb{E}X_{ij}^{(q)}\mathbb{E}X_{kl}^{(q)} = 2\Sigma_{ij,kl}. \tag{8.53}$$

In the case of centred X, one then obtains

$$\frac{1}{b}\sum_{i=1}^{b} X^{(i)} \overset{d}{=} b^{-1/2}X. \tag{8.54}$$

Note that this does not quite match the case described in Section 8.1.1, since we do not assume there that $\mathbb{E}X = 0$, however we take this a rough justification for $\mathsf{s}(b) = b^{-1/2}$ as an ansatz. Moreover, numerical experimentation with $\mathsf{s}(b) = b^{-\upsilon}$ for values of $\upsilon > 0$ shows that $q = 1/2$ gives a reasonable fit to the data (note that the values shown in Figures 8.4j, 8.4k, 8.4l are those producing the best fit, but $\upsilon = 1/2$ was seen to be not much inferior).

8.2 Spectral free addition from QUE

8.2.1 Intermediate results on QUE

This section establishes some intermediate results that follow from assuming QUE for the eigenvectors of a matrix. They will be crucial for our application in the following section.

Lemma 8.1. *Consider a real orthogonal $N \times N$ matrix U with rows $\{\boldsymbol{u}_i^T\}_{i=1}^N$. Assume that $\{\boldsymbol{u}_i\}_{i=1}^N$ are the eigenvectors of a real random symmetric matrix with QUE. Let P be a fixed $N \times N$ real orthogonal matrix. Let $V = UP$ and denote the rows of V by $\{\boldsymbol{v}_i^T\}_{i=1}^N$. Then $\{\boldsymbol{v}_i\}_{i=1}^N$ also satisfy QUE.*

Proof. Take any unit vector $\boldsymbol{q}$, then for any $k = 1, \ldots, N$

$$\boldsymbol{q}^T \boldsymbol{v}_k = \sum_j q_j V_{kj} = \sum_{j,l} q_j U_{kl} P_{lj} = (P\boldsymbol{q})^T \boldsymbol{u}_k.$$

But $\|P\boldsymbol{q}\|_2 = \|\boldsymbol{q}\|_2 = 1$ since P is orthogonal, so the statement of QUE for $\{\boldsymbol{u}_i\}_{i=1}^N$ transfers directly to $\{\boldsymbol{v}_i\}_{i=1}^N$ thanks to the supremum of all unit $\boldsymbol{q}$. ■

Lemma 8.2. *Consider a real orthogonal $N \times N$ matrix U with rows $\{\boldsymbol{u}_i^T\}_{i=1}^N$. Assume that $\{\boldsymbol{u}_i\}_{i=1}^N$ are the eigenvectors of a real random symmetric matrix with QUE. Let $\ell_0(\boldsymbol{q}) = \sum_i \mathbf{1}\{q_i \neq 0\}$ count the non-zero elements of a vector with respect to a fixed orthonormal basis $\{\boldsymbol{e}_i\}_{i=1}^N$. For any fixed integer $s > 0$, define the set*

$$\mathbb{V}_s = \left\{\boldsymbol{q} \in \mathbb{R}^N \mid \|\boldsymbol{q}\| = 1,\ \ell_0(\boldsymbol{q}) = s,\ q_i = 0\ \forall i \in \mathbb{T}_N^c\right\} \tag{8.55}$$

where, recall the definition

$$\mathbb{T}_N = [N] \backslash \{(N^{1/4}, N^{1-\delta}) \cup (N - N^{1-\delta}, N - N^{1/4})\}.$$

Then the columns $\{\boldsymbol{u}_i'\}_{i=1}^N$ of U satisfy a weaker form of QUE (for any fixed $n, s > 0$):

$$\sup_{\boldsymbol{q} \in \mathbb{V}_s} \sup_{\substack{I \subset \mathbb{T}_N \\ |I| = n}} \left| \mathbb{E}P\left((N|\boldsymbol{q}^T \boldsymbol{u}_k|^2)_{k \in I}\right) - \mathbb{E}P\left((|\mathcal{N}_j|^2)_{j=1}^m\right)\right| \leqslant N^{-\varepsilon}. \tag{8.56}$$

We will denote this form of QUE as $\widehat{\text{QUE}}$.

Proof. Take some $\boldsymbol{q} \in \mathbb{V}_s$. Then there exists some $J \subset \mathbb{T}_N$ with $|J| = s$ and non-zero $\{q_k\}_{k \in J}$ such that

$$\boldsymbol{q}^T \boldsymbol{u}_k' = \sum_{j \in J} q_j \boldsymbol{e}_j^T \boldsymbol{u}_k'.$$

Take $\{\boldsymbol{e}_i\}_{i=1}^N$ to be a standard basis with $(\boldsymbol{e}_i)_j = \delta_{ij}$, then $\boldsymbol{e}_j^T \boldsymbol{u}_k' = U_{jk} = \boldsymbol{e}_k^T \boldsymbol{u}_j$ so

$$\boldsymbol{q}^T \boldsymbol{u}_k' = \sum_{j \in J} q_j \boldsymbol{e}_k^T \boldsymbol{u}_j$$

but then the coefficients q_j can be absorbed into the definition of the general polynomial in the statement (8.9) of QUE for $\{\boldsymbol{u}_i\}_{i=1}^N$, which completes the proof, noting that the sum only includes indices contained in $\mathbb{T}_N$ owing to the definition of $\mathbb{V}_s$. ■

Lemma 8.3. *Fix some real numbers $\{y_i\}_{i=1}^r$. Fix also a diagonal matrix Λ and an orthonormal set of vectors $\{\boldsymbol{v}_i\}_{i=1}^N$ that satisfies $\widehat{QUE}$. Then there exists an $\varepsilon > 0$ and $\boldsymbol{\eta}_i \in \mathbb{C}^N$ with*

$$\eta_{ij}^2 \in [-1,1] \ \forall j \in \mathbb{T}_N, \tag{8.57}$$

$$\eta_{ij}^2 \in [-N^{\varepsilon}, N^{\varepsilon}] \ \forall j \in \mathbb{T}_N^c. \tag{8.58}$$

such that for any integer $l > 0$

$$\mathbb{E}\left(\sum_{i=1}^r y_i \boldsymbol{v}_i^T \Lambda \boldsymbol{v}_i\right)^l - \mathbb{E}\left(\sum_{i=1}^r y_i \frac{1}{N} \boldsymbol{g}_i^T \Lambda \boldsymbol{g}_i\right)^l = N^{-(1+\varepsilon)l}\left(\sum_{i=1}^r y_i \boldsymbol{\eta}_i^T \Lambda \boldsymbol{\eta}_i\right)^l \tag{8.59}$$

where the $\boldsymbol{g}_i$ are i.i.d. Gaussians $N(0, I_N)$.

Proof. Let $\{\boldsymbol{e}_i\}_{i=1}^N$ be the standard orthonormal basis from above. Then

$$\begin{aligned}
\mathbb{E}\left(\sum_{i=1}^r y_i \boldsymbol{v}_i^T \Lambda \boldsymbol{v}_i\right)^l &= \mathbb{E} \sum_{i_1,\dots,i_l=1}^{r} \prod_{k=1}^{l} y_{i_k} \boldsymbol{v}_{i_k}^T \Lambda \boldsymbol{v}_{i_k} \\
&= \mathbb{E} \sum_{i_1,\dots,i_l=1}^{r} \sum_{j_1,\dots,j_l=1}^{N} \prod_{k=1}^{l} y_{i_k} \lambda_{j_k} (\boldsymbol{e}_{j_k}^T \boldsymbol{v}_{i_k})^2
\end{aligned} \tag{8.60}$$

$$\begin{aligned}
\implies \mathbb{E}\left(\sum_{i=1}^r y_i \boldsymbol{v}_i^T \Lambda \boldsymbol{v}_i\right)^l - \mathbb{E}\left(\sum_{i=1}^r y_i \frac{1}{N} \boldsymbol{g}_i^T \Lambda \boldsymbol{g}_i\right)^l &= N^{-l} \sum_{i_1,\dots,i_l=1}^{r} \sum_{j_1,\dots,j_l=1}^{N} \prod_{k=1}^{l} y_{i_k} \lambda_{j_k} \left[N\mathbb{E}(\boldsymbol{e}_{j_k}^T \boldsymbol{v}_{i_k})^2 - \mathbb{E}(\boldsymbol{e}_{j_k}^T \boldsymbol{g}_{i_k})^2\right] \\
&= N^{-l} \sum_{i_1,\dots,i_l=1}^{r} \sum_{j_1,\dots,j_l \in \mathbb{T}_N} \prod_{k=1}^{l} y_{i_k} \lambda_{j_k} \left[N\mathbb{E}(\boldsymbol{e}_{j_k}^T \boldsymbol{v}_{i_k})^2 - \mathbb{E}(\boldsymbol{e}_{j_k}^T \boldsymbol{g}_{i_k})^2\right] \\
&\quad + N^{-l} \sum_{i_1,\dots,i_l=1}^{r} \sum_{\substack{j_1 \in \mathbb{T}_N^c, \\ j_2,\dots,j_l \in \mathbb{T}_N}} \prod_{k=1}^{l} y_{i_k} \lambda_{j_k} \left[N\mathbb{E}(\boldsymbol{e}_{j_k}^T \boldsymbol{v}_{i_k})^2 - \mathbb{E}(\boldsymbol{e}_{j_k}^T \boldsymbol{g}_{i_k})^2\right] \\
&\quad + \dots
\end{aligned} \tag{8.61}$$

The ellipsis represents the similar terms where further of the $j_1, \dots, j_r$ are in $\mathbb{T}_N^c$. For $j \in \mathbb{T}_N^c$ the terms

$$\left[N\mathbb{E}(\boldsymbol{e}_{j_k}^T \boldsymbol{v}_{i_k})^2 - \mathbb{E}(\boldsymbol{e}_{j_k}^T \boldsymbol{g}_{i_k})^2\right] \tag{8.62}$$

are excluded from the statement of $\widehat{\text{QUE}}$, however we can still bound them crudely. Indeed

$$\sum_{j \in \mathbb{T}_N^c} N(\boldsymbol{e}_j^T \boldsymbol{v}_i)^2 = \sum_{j=1}^{N} N(\boldsymbol{e}_j^T \boldsymbol{v}_i)^2 - \sum_{j \in \mathbb{T}_N} N(\boldsymbol{e}_j^T \boldsymbol{v}_i)^2 = N - \sum_{j \in \mathbb{T}_N} N(\boldsymbol{e}_j^T \boldsymbol{v}_i)^2 \tag{8.63}$$

but since the bound of $\widehat{\text{QUE}}$ applies for $j \in \mathbb{T}_N$

$$N\mathbb{E}(\boldsymbol{e}_j^T \boldsymbol{v}_i)^2 = \mathbb{E}(\boldsymbol{e}_j^T \boldsymbol{g})^2 + o(1) = 1 + o(1) \quad \forall j \in \mathbb{T}_N, \tag{8.64}$$

then

$$\sum_{j\in\mathbb{T}_N^c} N(\boldsymbol{e}_j^T\boldsymbol{v}_i)^2 = N - N(1+o(1)) = o(N) \implies \mathbb{E}(\boldsymbol{e}_j^T\boldsymbol{v}_i)^2 = o(1)\ \forall j\in\mathbb{T}_N^c. \tag{8.65}$$

Note that this error term is surely far from optimal, but is sufficient here. Overall we can now say

$$\left|\left[N\mathbb{E}(\boldsymbol{e}_j^T\boldsymbol{v}_i)^2 - \mathbb{E}(\boldsymbol{e}_j^T\boldsymbol{g}_i)^2\right]\right| \leqslant 1 + o(1) \leqslant 2\ \forall j\in\mathbb{T}_N^c. \tag{8.66}$$

We can apply $\widehat{\text{QUE}}$ to the terms in square parentheses to give $\epsilon_1,\ldots,\epsilon_r>0$ such that

$$|N\mathbb{E}(\boldsymbol{e}_{j_k}^T\boldsymbol{v}_{i_k})^2 - \mathbb{E}(\boldsymbol{e}_{j_k}^T\boldsymbol{g}_{i_k})^2| \leqslant N^{-\epsilon_{i_k}} \quad \forall j_k\in\mathbb{T}_N\ \forall i_k=1,\ldots,r. \tag{8.67}$$

We can obtain a single error bound by setting $\epsilon = \min_i \epsilon_i$, where clearly $\epsilon>0$ and then write

$$N\mathbb{E}(\boldsymbol{e}_{j_k}^T\boldsymbol{v}_{i_k})^2 - \mathbb{E}(\boldsymbol{e}_{j_k}^T\boldsymbol{g}_{i_k})^2 = \eta_{i_kj_k}^2 N^{-\epsilon} \tag{8.68}$$

where $\eta_{i_kj_k}^2\in[-1,1]$. To further include the indices $j\in\mathbb{T}_N^c$, we extend the expression (8.68) to all j_k by saying

$$\eta_{i_kj_k}^2 \in [-1,1]\ \forall j_k\in\mathbb{T}_N, \tag{8.69}$$

$$\eta_{i_kj_k}^2 \in [-N^{\epsilon},N^{\epsilon}]\ \forall j_k\in\mathbb{T}_N^c. \tag{8.70}$$

Overall we have

$$\mathbb{E}\left(\sum_{i=1}^r y_i\boldsymbol{v}_i^T\Lambda\boldsymbol{v}_i\right)^l - \mathbb{E}\left(\sum_{i=1}^r y_i\frac{1}{N}\boldsymbol{g}_i^T\Lambda\boldsymbol{g}_i\right)^l = N^{-l(1+\epsilon)}\sum_{i_1,\ldots,i_l=1}^r\sum_{j_1,\ldots,j_l=1}^N\prod_{k=1}^l y_{i_k}\lambda_{j_k}\eta_{i_kj_k}^2 \tag{8.71}$$

but by comparing with (8.60) we can rewrite as

$$\mathbb{E}\left(\sum_{i=1}^r y_i\boldsymbol{v}_i^T\Lambda\boldsymbol{v}_i\right)^l - \mathbb{E}\left(\sum_{i=1}^r y_i\frac{1}{N}\boldsymbol{g}_i^T\Lambda\boldsymbol{g}_i\right)^l = \left(\sum_{i=1}^r N^{-(1+\epsilon)}y_i\boldsymbol{\eta}_i^T\Lambda\boldsymbol{\eta}_i\right)^l \tag{8.72}$$

where $\boldsymbol{\eta}_i^T = (\eta_{i1},\ldots,\eta_{iN})$.

■

8.2.2 Main result

Theorem 8.3. *Let X be an $N\times N$ real symmetric random matrix and let D be an $N\times N$ symmetric matrix (deterministic or random). Let $\hat{\mu}_X,\hat{\mu}_D$ be the empirical spectral measures of the sequence of matrices X,D and assume there exist deterministic limit measures μ_X,μ_D. Assume that X has QUE, i.e. 8.2. Assume also the $\hat{\mu}_X$ concentrates in the sense that*

$$\mathbb{P}(W_1(\hat{\mu}_X,\mu_X)>\delta) \lesssim e^{-N^{\tau}f(\delta)} \tag{8.73}$$

where $\tau>0$ and f is some positive increasing function. Then $H=X+D$ has a limiting spectral measure and it is given by the free convolution $\mu_X\boxplus\mu_D$.

Remark 8.4. A condition like (8.73) is required so that the Laplace method can be applied to the empirical measure $\hat{\mu}_X$. There are of course other ways to formulate such a condition. Consider for example the conditions used in Theorems 1.2 and 4.1 of [ABM21a]. There it is assumed the existence of a sequence of deterministic measures $(\mu_N)_{N\geqslant 1}$ and a constant $\kappa > 0$ such that for large enough N

$$W_1(\mathbb{E}\hat{\mu_X}, \mu_N) \leqslant N^{-\kappa}, \quad W_1(\mu_N, \mu_X) \leqslant N^{-\kappa}, \tag{8.74}$$

which is of course just a deterministic version of (8.73). [ABM21a] introduce the extra condition around concentration of Lipschitz traces:

$$\mathbb{P}\left(\left|\frac{1}{N}\mathrm{Tr}f(H_N) - \frac{1}{N}\mathbb{E}\mathrm{Tr}f(H_N)\right| > \delta\right) \leqslant \exp\left(-\frac{c_\zeta}{N^\zeta}\min\left\{\left(\frac{N\delta}{\|f\|_{Lip}}\right)^2, \left(\frac{N\delta}{\|f\|_{Lip}}\right)^{1+\varepsilon_0}\right\}\right), \tag{8.75}$$

for all $\delta > 0$, Lipschitz f and N large enough, where $\zeta, c_\zeta > 0$ are some constants. As shown in the proof of Theorem 1.2, this condition is sufficient to obtain

$$\mathbb{P}\left(\left|\int |\lambda| d\hat{\mu}_X(\lambda) - \int |\lambda| d\mathbb{E}\hat{\mu}_X(\lambda)\right| \leqslant t\right) \leqslant \exp\left(-\frac{c_\zeta}{N^\zeta}\min\left\{(2Nt\eta)^2, (2Nt\eta)^{1+\varepsilon_0}\right\}\right) \tag{8.76}$$

for any $t > 0$ and for large enough N. Note that [ABM21a] prove this instead for integration against a regularised version of $\log|\lambda|$, but the proof relies only the integrand's being Lipschitz, so it goes through just the same here. (8.76) and (8.74) clearly combine to give (8.73). The reader may ignore this remark if they are content to take (8.73) as an assumption. Alternatively, as we have shown, (8.73) can be replaced by (8.74) and (8.75), conditions which have already been used for quite general results in the random matrix theory literature.

Proof. We shall denote use the notation

$$G_H(z) = \frac{1}{N}\mathrm{Tr}(z - H)^{-1}. \tag{8.77}$$

Recall the supersymmetric approach to calculating the expected trace of the resolvent of a random matrix ensemble:

$$\mathbb{E}_H G_H(z) = \frac{1}{N}\frac{\partial}{\partial \mathrm{j}}\bigg|_{\mathrm{j}=0} \mathbb{E}_H Z_H(\mathrm{j}) \tag{8.78}$$

where

$$Z_H(\mathrm{j}) = \frac{\det(z + \mathrm{j} - H)}{\det(z - H)} = \int d\Psi e^{-i\mathrm{Tr}AH} e^{i\mathrm{Tr}\Psi\Psi^\dagger J}, \tag{8.79}$$

$$A = \phi\phi^\dagger + \chi\chi^\dagger, \tag{8.80}$$

$$J = I_N \otimes \begin{pmatrix} z & 0 \\ 0 & \mathrm{j} + z \end{pmatrix}, \tag{8.81}$$

$$d\Psi = \frac{d\phi d\phi^* d\chi d\chi^*}{-(2\pi)^N i}, \tag{8.82}$$

$$\Psi = \begin{pmatrix} \phi \\ \chi \end{pmatrix} \tag{8.83}$$

with $\phi \in \mathbb{C}^N$ and χ, χ^* being N-long vectors of anti-commuting variables. Independence of X and D gives

$$\begin{aligned}\mathbb{E}_H Z_H(\mathrm{j}) &= \int d\Psi e^{i\mathrm{Tr}\Psi\Psi^\dagger J}\mathbb{E}_{X,D} e^{-i\mathrm{Tr}A(X+D)}\\ &= \int d\Psi e^{i\mathrm{Tr}\Psi\Psi^\dagger J}\mathbb{E}_D e^{-i\mathrm{Tr}AD}\mathbb{E}_X e^{-i\mathrm{Tr}AX}.\end{aligned} \tag{8.84}$$

$\mathbb{E}_D$ simply means integration against a delta-function density if D is deterministic.

Let us introduce some notation: for $N\times N$ matrices K, $\Phi_X(K) = \mathbb{E}_X e^{-i\mathrm{Tr}XK}$, and similarly Φ_D. We also define a new matrix ensemble $\bar{X} \overset{d}{=} O^T \Lambda O$, where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ are equal in distribution to the eigenvalues of X and O is an entirely independent Haar-distributed orthogonal matrix.

Now

$$\begin{aligned}\mathbb{E}_H Z_H(\mathrm{j}) &= \int d\Psi e^{i\mathrm{Tr}\Psi\Psi^\dagger J}\Phi_{\bar{X}}(K)\Phi_D(K) + \int d\Psi e^{i\mathrm{Tr}\Psi\Psi^\dagger J}(\Phi_X(A) - \Phi_{\bar{X}}(A))\Phi_D(A)\\ \implies \mathbb{E}G_{D+X}(z) &= \mathbb{E}G_{D+\bar{X}}(z) + \frac{1}{N}\frac{\partial}{\partial \mathrm{j}}\bigg|_{\mathrm{j}=0}\int d\Psi e^{i\mathrm{Tr}\Psi\Psi^\dagger J}(\Phi_X(A) - \Phi_{\bar{X}}(A))\Phi_D(A) \equiv \mathbb{E}G_{D+\bar{X}}(z) + E(z)\end{aligned} \tag{8.85}$$

and so we need to analyse the error term $E(z)$.

Now consider $X = U^T \Lambda U$ where the rows of U are the eigenvectors $\{\boldsymbol{u}_i\}_i$ of X. Say also that $K = Q^T Y Q$ for diagonal $Y = (y_1, \ldots, y_r, 0, \ldots, 0)$, where we note that K has fixed rank, by construction. Then

$$\mathrm{Tr}XK = Y(UQ^T)^T\Lambda(UQ^T)$$

but Lemma 8.1 establishes that the rows of UQ^T obey QUE, since the rows of U do. Further, Lemma 8.2 then establishes that the columns of UQ^T obey $\widehat{\text{QUE}}$ as required by Lemma 8.3. Let $\{\boldsymbol{v}_i\}$ be those columns, then we have

$$\mathrm{Tr}XK = \sum_{i=1}^r y_i \boldsymbol{v}_i^T \Lambda \boldsymbol{v}_i. \tag{8.86}$$

The expectation over X can be split into eigenvalues and conditional eigenvectors

$$\Phi_X(K) = \mathbb{E}_\Lambda \mathbb{E}_{U|\Lambda}\sum_{l=0}^\infty \frac{1}{l!}(-i)^l\left(\mathrm{Tr}U^T\Lambda UK\right)^l. \tag{8.87}$$

We can simply bound

$$\left|\sum_{l=0}^n \frac{1}{l!}(-i)^l\left(\mathrm{Tr}U^T\Lambda U\right)^l\right| \leqslant e^{|\mathrm{Tr}U^T\Lambda UK|} \tag{8.88}$$

for any n, but clearly

$$\mathbb{E}_{U|\Lambda}e^{|\mathrm{Tr}U^T\Lambda UK|} < \infty \tag{8.89}$$

since, whatever the distribution of $U \mid \Lambda$, the integral is over a compact group (the orthogonal group $O(N)$) and the integrand has no singularities. Therefore, by the dominated convergence theorem

$$\Phi_X(K) = \mathbb{E}_\Lambda \sum_{l=0}^{\infty} \frac{1}{l!}(-i)^l \mathbb{E}_{U|\Lambda} \left(\text{Tr} U^T \Lambda U K\right)^l \tag{8.90}$$

and in precisely the same way

$$\Phi_X(K) = \mathbb{E}_\Lambda \sum_{l=0}^{\infty} \frac{1}{l!}(-i)^l \mathbb{E}_{O\sim\mu_{Haar}} \left(\text{Tr} O^T \Lambda O K\right)^l. \tag{8.91}$$

Recalling (8.86) we now have

$$\Phi_X(K) = \mathbb{E}_\Lambda \sum_{l=0}^{\infty} \frac{1}{l!}(-i)^l \mathbb{E}_{U|\Lambda} \left(\sum_{i=1}^{r} y_i \boldsymbol{v}_i^T \Lambda \boldsymbol{v}_i\right)^l. \tag{8.92}$$

and similarly

$$\Phi_{\bar{X}}(K) = \mathbb{E}_\Lambda \sum_{l=0}^{\infty} \frac{1}{l!}(-i)^l \mathbb{E}_{U|\Lambda} \left(\sum_{i=1}^{r} y_i \bar{\boldsymbol{v}}_i^T \Lambda \bar{\boldsymbol{v}}_i\right)^l. \tag{8.93}$$

where the $\bar{\boldsymbol{v}}_i$ are defined in the obvious way from $\bar{X}$. We would now apply $\widehat{\text{QUE}}$, but to do so we must insist that $\mathbb{E}_\Lambda$ is taken over the *ordered* eigenvalues of X. Having fixed that convention, Lemma 8.3 can be applied to the terms

$$\mathbb{E}_{U|\Lambda} \left(\sum_{i=1}^{r} y_i \boldsymbol{v}_i^T \Lambda \boldsymbol{v}_i\right)^l \tag{8.94}$$

in (8.92). The terms in $\Phi_{\bar{X}}$ can be treated similarly. This results in

$$\begin{aligned} \Phi_X(K) - \Phi_{\bar{X}}(K) = \mathbb{E}_\Lambda \Bigg[\sum_{l=0}^{\infty} \frac{i^l}{l!} & \left\{ \mathbb{E}_{\{\boldsymbol{g}_i\}_{i=1}^r} \left(\sum_{i=1}^{r} y_i \frac{1}{N} \boldsymbol{g}_i^T \Lambda \boldsymbol{g}_i\right)^l + \left(\sum_{i=1}^{r} N^{-(1+\varepsilon)} y_i \boldsymbol{\eta}_i^T \Lambda \boldsymbol{\eta}_i\right)^l \right\} \\ & - \sum_{l=0}^{\infty} \frac{i^l}{l!} \left\{ \mathbb{E}_{\{\boldsymbol{g}_i\}_{i=1}^r} \left(\sum_{i=1}^{r} y_i \frac{1}{N} \boldsymbol{g}_i^T \Lambda \boldsymbol{g}_i\right)^l + \left(\sum_{i=1}^{r} N^{-(1+\varepsilon)} y_i \tilde{\boldsymbol{\eta}}_i^T \Lambda \tilde{\boldsymbol{\eta}}_i\right)^l \right\} \Bigg] \end{aligned} \tag{8.95}$$

The exponential has infinite radius of convergence, so we may re-order the terms in the sums to give cancellation

$$\Phi_X(K) - \Phi_{\bar{X}}(K) = \mathbb{E}_\Lambda \sum_{l=1}^{\infty} \frac{1}{l!} N^{-(1+\varepsilon)l} (-i)^l \left(\sum_{i=1}^{r} y_i \boldsymbol{\eta}_i^T \Lambda \boldsymbol{\eta}_i\right)^l - \mathbb{E}_\Lambda \sum_{l=1}^{\infty} \frac{1}{l!} N^{-(1+\varepsilon)l} (-i)^l \left(\sum_{i=1}^{r} y_i \tilde{\boldsymbol{\eta}}_i^T \Lambda \tilde{\boldsymbol{\eta}}_i\right)^l.$$

Here $\varepsilon > 0$ and $\boldsymbol{\eta}_i, \tilde{\boldsymbol{\eta}}_i \in \mathbb{C}^N$ with

$$-1 \leqslant [(\boldsymbol{\eta}_i)_j]^2, [(\tilde{\boldsymbol{\eta}}_i)_j]^2 \leqslant 1 \quad \forall i = 1, \ldots, r, \ \forall j \in \mathbb{T}_N, \tag{8.96}$$

$$-N^\varepsilon \leqslant [(\boldsymbol{\eta}_i)_j]^2, [(\tilde{\boldsymbol{\eta}}_i)_j]^2 \leqslant N^\varepsilon \quad \forall i = 1, \ldots, r, \ \forall j \in \mathbb{T}_N. \tag{8.97}$$

Simplifying, we obtain

$$\Phi_X(K) - \Phi_{\tilde{X}}(K) = \mathbb{E}_\Lambda \exp\left(-iN^{-(1+\varepsilon)}\sum_{i=1}^r y_i \boldsymbol{\eta}_i^T \Lambda \boldsymbol{\eta}_i\right) - \mathbb{E}_\Lambda \exp\left(-iN^{-(1+\varepsilon)}\sum_{i=1}^r y_i \tilde{\boldsymbol{\eta}}_i^T \Lambda \tilde{\boldsymbol{\eta}}_i\right). \tag{8.98}$$

Since $|\mathbb{T}_N^c| \leqslant 2N^{1-\delta}$ we have

$$\sum_{j\in\mathbb{T}_N^c} |\lambda_j| \leqslant \mathcal{O}(N^{1-d}N^{-1})\text{Tr}|\Lambda| \tag{8.99}$$

and so

$$|\boldsymbol{\eta}_i^T \Lambda \boldsymbol{\eta}_i| \leqslant \text{Tr}|\Lambda|\left(1 + \mathcal{O}(N^{\varepsilon-\delta})\right). \tag{8.100}$$

For any fixed $\delta > 0$, ε can be reduced if necessary so that $\varepsilon < \delta$ and then for sufficiently large N we obtain, say,

$$|\boldsymbol{\eta}_i^T \Lambda \boldsymbol{\eta}_i| \leqslant 2\text{Tr}|\Lambda|. \tag{8.101}$$

Thence we can write $\boldsymbol{\eta}_i^T \Lambda \boldsymbol{\eta}_i = \text{Tr}|\Lambda|\xi_i$ for $\xi_i \in [-2,2]$, and similarly $\tilde{\boldsymbol{\eta}}_i^T \Lambda \tilde{\boldsymbol{\eta}}_i = \text{Tr}|\Lambda|\tilde{\xi}_i$. Now

$$\mathbb{E}_\Lambda \exp\left(-iN^{-(1+\varepsilon)}\sum_{i=1}^r \xi_i y_i \text{Tr}|\Lambda|\right) = \mathbb{E}_\Lambda \exp\left(-iN^{-\varepsilon}\sum_{i=1}^r \xi_i y_i \int d\hat{\mu}_X(\lambda)|\lambda|\right)$$

so we can apply Laplace's method to the empirical spectral measure $\hat{\mu}_X$ to obtain

$$\mathbb{E}_\Lambda \exp\left(-iN^{-(1+\varepsilon)}\sum_{i=1}^r \xi_i y_i \text{Tr}|\Lambda|\right) = \exp\left(-iN^{-\varepsilon}(q + o(1))\sum_{i=1}^r \xi_i y_i\right) + o(1) \tag{8.102}$$

where the $o(1)$ terms do not depend on the y_i and where we have defined

$$q = \int d\mu_X(\lambda)|\lambda|. \tag{8.103}$$

Further, we can write $\sum_{i=1}^r \xi_i y_i = \zeta\text{Tr}K$, where $\zeta \in [\min_i\{\xi_i\}, \max_i\{\xi_i\}] \subset [-1,1]$, and similarly $\sum_{i=1}^r \tilde{\xi}_i y_i = \tilde{\zeta}\text{Tr}K$. Then

$$\Phi_X(K) - \Phi_{\tilde{X}}(K) = e^{-iN^{-\varepsilon}\zeta(q+o(1))\text{Tr}K} - e^{-iN^{-\varepsilon}\tilde{\zeta}(q+o(1))\text{Tr}K} + o(1) \tag{8.104}$$

but

$$\frac{1}{N}\frac{\partial}{\partial \mathrm{j}}\bigg|_{\mathrm{j}=0} \int d\Psi e^{i\text{Tr}\Psi\Psi^\dagger J} e^{-iN^{-\varepsilon}\zeta(q+o(1))\text{Tr}K}\Phi_D(A) = \mathbb{E}G_{D+N^{-\varepsilon}\zeta(q+o(1))I}(z) = \mathbb{E}G_D(z + \mathcal{O}(N^{-\varepsilon}))$$

$$\implies E(z) = \mathbb{E}G_D(z+\mathcal{O}(N^{-\varepsilon})) + o(1) - \mathbb{E}G_D(z+\mathcal{O}(N^{-\varepsilon})) - o(1) = o(1). \tag{8.105}$$

We have thus established that

$$\mathbb{E}G_{D+X}(z) = \mathbb{E}G_{D+\tilde{X}}(z) + o(1) \tag{8.106}$$

from which one deduces that $\mu_{D+X} = \mu_{D+\tilde{X}} = \mu_D \boxplus \mu_{\tilde{X}} = \mu_D \boxplus \mu_X$. ■

Remark 8.5. We have also constructed a non-rigorous argument for Theorem 8.3 where the supersymmetric approach is replaced by the replica method. This approach simplifies some of the analysis but at the expense of being not at all rigorous (indeed there are integral expressions in this argument that are manifestly infinite). The supersymmetric methods used here are not fully rigorous (like most of their applications) but we note that recent work is beginning to elevate supersymmetric random matrix calculations to full rigour [SS17; Shc20].

8.2.3 Experimental validation

Let $U(a,b)$ denote the uniform distribution on the interval (a,b), and $\Gamma(a)$ the Gamma-distribution with scale parameter a. We consider the following matrix ensembles:

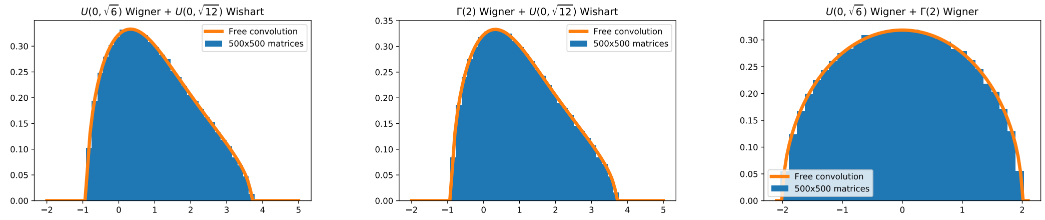
$$\begin{aligned} M \sim GOE^n \ &: \ \mathrm{Var}(M_{ij}) = \frac{1+\delta_{ij}}{2n}, \\ M \sim UWig^n &: \ \sqrt{n}M_{ij} \overset{i.i.d}{\sim} U(0,\sqrt{6}) \ \text{ up to symmetry}, \\ M \sim \Gamma Wig^n &: \ 2\sqrt{n}M_{ij} \overset{i.i.d}{\sim} \Gamma(2) \ \text{ up to symmetry}, \\ M \sim UWish^n &: \ M \overset{d}{=} \frac{1}{m}XX^T, \ X_{ij} \overset{i.i.d}{\sim} U(0,\sqrt{12}) \ \text{ for } X \text{ of size } n\times m, \ \frac{n}{m} \overset{n,m\to\infty}{\to} \alpha, \\ M \sim Wish^n &: M \overset{d}{=} \frac{1}{m}XX^T, \ X_{ij} \overset{i.i.d}{\sim} \mathcal{N}(0,1) \ \text{ for } X \text{ of size } n\times m, \ \frac{n}{m} \overset{n,m\to\infty}{\to} \alpha. \end{aligned}$$

All of the $GOE^n, UWig^n, \Gamma Wig^n$ have the same limiting spectral measure, namely μ_{SC}, the semicircle of radius $\sqrt{2}$. $UWish^n, Wish^n$ have a Marcenko-Pastur limiting spectral measure μ_{MP}, and the constant $\sqrt{12}$ is chosen so that the parameters of the MP measure match those of a Gaussian Wishart matrix $Wish^n$. $GOE^n, Wish^n$ are the only ensembles whose eigenvectors are Haar distributed, but all ensembles obey a local law in the sense above. It is known that the sum of GOE^n and any of the other ensembles will have limiting spectral measure given by the free additive convolution of μ_{SC} and the other ensemble's measure (so either $\mu_{SC} \boxplus \mu_{MP}$ or $\mu_{SC} \boxplus \mu_{SC}$), indeed this free addition property holds for any invariant ensemble [AGZ10]. Our result implies that the same holds for addition of the non-invariant ensembles. Sampling from the above ensembles is simple, so we can easily generate spectral histograms from multiple independent matrix samples for large n. $\mu_{SC} \boxplus \mu_{SC}$ is just another semi-circle measure but with radius 2. $\mu_{SC} \boxplus \mu_{MP}$ can be computed in the usual manner with R-transforms and is given by the solution to the polynomial

$$\frac{\alpha}{2}t^3 - \left(\frac{1}{2} + \alpha z\right)t^2 + (z+\alpha-1)t - 1 = 0.$$

i.e. Say the cubic has roots $\{r_1, r_2 + is_2, r_2 - is_2\}$ for $s_2 \geqslant 0$, then the density of $\mu_{SC} \boxplus \mu_{MP}$ at z is s_2/π. This can all be solved numerically. The resulting plots are in Figure 8.5 and clearly show agreement between the free convolutions and sampled spectral histograms.

We can also test the result in another more complicated case. Consider the case of random d-regular graphs on N vertices. Say $M \sim Reg^{N,d}$ is the distribution of the adjacency matrix of such random graphs. The limiting spectral density of $M \sim Reg^{N,d}$ is known in closed form, as



(a) $UWig^n + UWish^n$ (b) $\Gamma Wig^n + UWish^n$ (c) $UWig^n + \Gamma Wig^n$

Figure 8.5: Comparison of theoretical spectral density and empirical from sampled matrices all of size 500 × 500. We combine 50 independent matrix samples per plot.

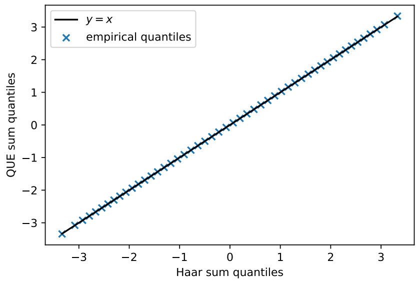


Figure 8.6: q-q plot comparing the spectrum of samples from $Reg^{N,d} + UWig^N$ (y-axis) to samples from $Reg^{N,d} + GOE^N$ (x-axis).

is its Stieljtes transform [BHY19] and [BHY19] established a local law of the kind required for our results. Moreover, there are known efficient algorithms for sampling random d-regular graphs [KV03; SW99] along with implementations [HSS08]. Let $\mu_{KM}^{(d)}$ be the Kesten-McKay law, the limiting spectral measure of d-regular graphs. We could find an explicit degree-6 polynomial for the Stieljtes transform of $\mu_{KM}^{(d)} \boxplus \mu_{SC}$ and compare to spectral histograms as above. Alternatively we can investigate agreement with $\mu_{KM}^{(d)} \boxplus \mu_{SC}$ indirectly by sampling and comparing spectra from say $Reg^{N,d} + UWig^N$ and also from $Reg^{N,d} + GOE^N$. The latter case will certainly yield the distribution $\mu_{KM}^{(d)} \boxplus \mu_{SC}$ since the GOE matrices are freely independent from the adjacency matrices. Figure shows a q-q plot[2] for samples of the spectra from these two matrix distributions and demonstrates near-perfect agreement, thus showing that indeed the spectrum of $Reg^{N,d} + UWig^N$ is indeed described by $\mu_{KM}^{(d)} \boxplus \mu_{SC}$. We reached the same conclusion when repeating the above experiment with $UWish^N + Reg^{N,d}$ and $Wish^n + Reg^{N,d}$.

[2] Recall that a q-q plot shows the quantiles of one distribution on the x axis and another on the y axis. Given two cumulative density functions F_X, F_Y and their percent point functions F_X^{-1}, F_Y^{-1}, the q-q plot is a plot of the parametric curve $(F_X^{-1}(q), F_Y^{-1}(q))$ for $q \in [0,1]$. Given only finite samples from the random variables X and Y, the empirical percent point functions can be estimated and used in the q-q plot.

8.3 Invariant equivalent ensembles

For an invariant ensemble [Dei99] with potential V we have the following integro-differential equation relating the equilibrium measure μ to the potential V [Unt19]:

$$\frac{\beta}{2}\fint \frac{1}{x-y}d\mu(y) = V'(x). \tag{8.107}$$

So in the case of real symmetric matrices we have

$$\frac{1}{2}\bar{g}_\mu(x) = V'(x) \tag{8.108}$$

where g_μ is the Stieljtes transform of μ and the bar over $\bar{g}_\mu$ indicates that the principal value has been taken.

Given a sufficiently nice μ (8.107) defines V up-to a constant of integration on supp(μ), but V is not determined on $\mathbb{R}\backslash\text{supp}(\mu)$, as is made clear by the following lemma, which we prove for completeness but which has appeared before in various works (e.g. [Dei99]).

Lemma 8.4. *For compactly supported probability measure μ on $\mathbb{R}$ and real potential V, define*

$$S_V[\mu](y) = V(y) - \int d\mu(x)\log|y-x|. \tag{8.109}$$

Suppose $S_V[\mu](y) = c$, a constant, for all $y \in supp(\mu)$ and $S_V[\mu](y) \geqslant c$ for all $y \in \mathbb{R}$. Then μ is a minimiser amongst all probability measures on $\mathbb{R}$ of the energy

$$\mathcal{E}_V[\mu] = \int d\mu(x)V(x) - \iint_{x<y} d\mu(x)d\mu(y)\log|x-y|. \tag{8.110}$$

Proof. Consider a probability measure that is close to μ in the sense of W_1 distance, say.

For any such measure, one can find an arbitrarily close probability measure μ' of the form

$$\mu' = \mu + \sum_{i=1}^{r} a_i \mathbf{1}_{[y_i-\delta_i, y_i+\delta_i]} - \sum_{i=1}^{s} b_i \mathbf{1}_{[z_i-\eta_i, z_i+\eta_i]} \tag{8.111}$$

where all $a_i, b_i > 0$ and $\delta_i, \eta_i, a_i, b_i \leqslant \epsilon$ for some small $\epsilon > 0$. To ensure that μ' is again a probability measure we must impose $\sum_i a_i = \sum_j b_j$. The strategy now is to expand $\mathcal{E}_V[\mu']$ about μ to first order in ϵ, but first note the symmetrisation

$$\iint_{x<y} d\mu(x)d\mu(y)\log|x-y| = \frac{1}{2}\iint_{x\neq y} d\mu(x)d\mu(y)\log|x-y|. \tag{8.112}$$

Then

$$\begin{aligned}\mathcal{E}_V[\mu'] - \mathcal{E}_V[\mu] &= \sum_{i=1}^{r} a_i V(y_i) - \sum_{i=1}^{s} b_i V(z_i) - \sum_{i=1}^{r} a_i \int d\mu(x)\log|x-y_i| + \sum_{i=1}^{r} b_i \int d\mu(x)\log|x-z_i| + \mathcal{O}(\epsilon^2)\\ &= \sum_{i=1}^{r} a_i S_V[\mu](y_i) - \sum_{i=1}^{r} b_i S_V[\mu](z_i) + \mathcal{O}(\epsilon^2).\end{aligned} \tag{8.113}$$

Observe that if all $y_i, z_i \in \text{supp}(\mu)$ then $S_V[\mu](y_i) = S_V[\mu](y_i) = c$ and so $\mathcal{E}_V[\mu'] = \mathcal{E}_V[\mu]$. Without loss of generality therefore, we take $y_i \notin \text{supp}(\mu)$ and $z_i \in \text{supp}(\mu)$, whence

$$\mathcal{E}_V[\mu'] - \mathcal{E}_V[\mu] \geqslant c \sum_{i=1}^{r} a_i - c \sum_{i=1}^{s} b_i = 0. \tag{8.114}$$

■

The next lemma establishes that, while not unique, a potential V can always be constructed given a measure μ.

Lemma 8.5. *Consider a probability measure μ on $\mathbb{R}$ with compact support, absolutely continuous with respect to the Lebesgue measure. Then there exists a potential $V : \mathbb{R} \to \mathbb{R}$ which yields a well-defined invariant distribution on real symmetric matrices for which the equilibrium measure is μ.*

Proof. (8.108) can be integrated to obtain V and the condition $S_V[\mu] = c$ (a constant) on $\text{supp}(\mu)$ determines V uniquely on $\text{supp}(\mu)$. Next observe that, for $y \in \mathbb{R}\backslash\text{supp}(\mu)$ there exists some constant $R > 0$ such that $|x - y| \leqslant R + |y|$, since μ is compactly supported, and so $\log|x - y| \leqslant |y| + R$. Therefore

$$S_V[\mu](y) \geqslant V(y) - |y| - R. \tag{8.115}$$

V must be chosen on $\mathbb{R}\backslash\text{supp}(\mu)$ to satisfy $S_V[\mu](y) \geqslant c$, which can be achieved by ensuring

$$V(y) \geqslant |y| + R + c. \tag{8.116}$$

Additionally, V must be defined for large y such that it defines an legitimate invariant ensemble on symmetric real matrices, i.e. V must decay sufficiently quickly at infinity to give an integrable probability density. Finally, V must be sufficiently smooth, and certainly continuous, so there are boundary conditions at the boundary of $\text{supp}(\mu)$. Suppose $\text{supp}(\mu)$ is composed of K disjoint intervals, then there are $2K$ boundary conditions on V, and the bound (8.116) imposes one further condition. Sufficiently fast decay at infinity can be satisfied by any even degree polynomial V of degree at least 2, therefore a degree $2K + 2$ polynomial can be found with sufficiently fast decay at infinity, satisfying all the boundary conditions and (8.116). ■

8.4 Universal complexity of loss surfaces

8.4.1 Extension of a key result and prevalence of minima

Let's recall Theorem 4.5 from [ABM21a]. $H_N(u)$ is our random matrix ensemble with some parametrisation $u \in \mathbb{R}^m$ and its limiting spectral measure is $\mu_\infty(u)$.

Define

$$\mathcal{G}_{-\varepsilon} = \{u \in \mathbb{R}^m \mid \mu_\infty(u)\left((-\infty, 0)\right) \leqslant \varepsilon\}. \tag{8.117}$$

So $\mathcal{G}_{-\varepsilon}$ is the event that $\mu_\infty(u)$ is close to being supported only on $(0, \infty)$. Let $l(u), r(u)$ be the left and right edges respectively of the support of $\mu_\infty(u)$.

Theorem 8.4 ([ABM21a] Theorem 4.5). *Fix some $\mathcal{D} \subset \mathbb{R}^m$ and suppose that $\mathcal{D}$ and the matrices $H_N(u)$ satisfy the following.*

- *For every $R > 0$ and every $\varepsilon > 0$, we have*

$$\lim_{N\to\infty} \frac{1}{N\log N} \log \left[\sup_{u\in B_R} \mathbb{P}\left(d_{BL}(\hat{\mu}_{H_N(u)}, \mu_\infty(u)) > \varepsilon\right)\right] = -\infty. \tag{8.118}$$

- *Several other assumptions detailed in [ABM21a].*

Then for any $\alpha > 0$ and any fixed $p \in \mathbb{N}$, we have

$$\lim_{N\to\infty} \frac{1}{N} \log \int_{\mathcal{D}} e^{-(N+p)\alpha u^2} \mathbb{E}[|\det(H_N(u))|\mathbf{1}\{i(H_N(u)) = 0\}]\, du = \sup_{u\in\mathcal{D}\cap\mathcal{G}} \left\{ \int_{\mathbb{R}} \log|\lambda| d\mu_\infty(u)(\lambda) - \alpha u^2 \right\}. \tag{8.119}$$

We claim the following extension

Corollary 8.1. *Under the same assumptions as the above theorem and for any integer sequence $k(N) > 0$ such that $k/N \to 0$ as $N \to \infty$, we have*

$$\lim_{N\to\infty} \frac{1}{N} \log \int_{\mathcal{D}} e^{-(N+p)\alpha u^2} \mathbb{E}[|\det(H_N(u))|\mathbf{1}\{i(H_N(u)) \leqslant k\}]\, du = \sup_{u\in\mathcal{D}\cap\mathcal{G}} \left\{ \int_{\mathbb{R}} \log|\lambda| d\mu_\infty(u)(\lambda) - \alpha u^2 \right\}. \tag{8.120}$$

Proof. Firstly note that

$$\begin{aligned} &\frac{1}{N} \log \int_{\mathcal{D}} e^{-(N+p)\alpha u^2} \mathbb{E}[|\det(H_N(u))|\mathbf{1}\{i(H_N(u)) \leqslant k\}]\, du \\ \geqslant &\frac{1}{N} \log \int_{\mathcal{D}} e^{-(N+p)\alpha u^2} \mathbb{E}[|\det(H_N(u))|\mathbf{1}\{i(H_N(u)) = 0\}]\, du, \end{aligned} \tag{8.121}$$

so it suffices to establish a complementary upper bound. The proof in of Theorem 4.5 in [ABM21a] establishes an upper bound using

$$\lim_{N\to\infty} \frac{1}{N} \log \int_{(\mathcal{G}_{-\varepsilon})^c} e^{-N\alpha u^2} \mathbb{E}[|\det(H_N(u)|\mathbf{1}\{i(H_N(u)) = 0\}]\, du = -\infty \tag{8.122}$$

which holds for all $\varepsilon > 0$. Indeed, $\mathcal{D} = (\mathcal{D} \cap \mathcal{G}_{-\varepsilon}) \cup (\mathcal{D} \cap (\mathcal{G}_{-\varepsilon})^c)$, so

$$\begin{aligned} &\int_{\mathcal{D}} e^{-(N+p)\alpha u^2} \mathbb{E}[|\det(H_N(u))|\mathbf{1}\{i(H_N(u)) \leqslant k\}]\, du \\ \leqslant &\int_{\mathcal{D}\cap\mathcal{G}_{-\varepsilon}} e^{-(N+p)\alpha u^2} \mathbb{E}[|\det(H_N(u))|\mathbf{1}\{i(H_N(u)) \leqslant k\}]\, du \\ &+ \int_{(\mathcal{G}_{-\varepsilon})^c} e^{-(N+p)\alpha u^2} \mathbb{E}[|\det(H_N(u))|\mathbf{1}\{i(H_N(u)) \leqslant k\}]\, du, \end{aligned} \tag{8.123}$$

so our proof is complete if we can prove the analogous result

$$\lim_{N\to\infty} \frac{1}{N} \log \int_{(\mathcal{G}_{-\varepsilon})^c} e^{-N\alpha u^2} \mathbb{E}[|\det(H_N(u)|\mathbf{1}\{i(H_N(u)) \leqslant k\}]\, du = -\infty. \tag{8.124}$$

As in [ABM21a], let f_ε be some $\frac{1}{2}$-Lipschitz function satisfying $\frac{\varepsilon}{2}\mathbf{1}_{x\leqslant-\varepsilon} \leqslant f_\varepsilon(x) \leqslant \frac{\varepsilon}{2}\mathbf{1}_{x\leqslant 0}$. Suppose $u \in (\mathcal{G}_{-\varepsilon})^c$ and also $i(H_N(u)) \leqslant k$. Then we have

$$0 \leqslant \int d\hat{\mu}_{H_N(u)}(x)\ f_\varepsilon(x) \leqslant \frac{k\varepsilon}{2N} \tag{8.125}$$

and also

$$\frac{\varepsilon^2}{2} \leqslant \int d\mu_\infty(u)(x)\ f_\varepsilon(x) \leqslant \frac{\varepsilon}{2}. \tag{8.126}$$

We have

$$\begin{aligned} d_{BL}(\hat{\mu}_{H_N(u)}, \mu_\infty(u)) &\geqslant \left|\int d\hat{\mu}_{H_N(u)}(x)\ f_\varepsilon(x) - \int d\mu_\infty(u)(x)\ f_\varepsilon(x)\right| \\ &\geqslant \left|\left|\int d\hat{\mu}_{H_N(u)}(x)\ f_\varepsilon(x)\right| - \left|\int d\mu_\infty(u)(x)\ f_\varepsilon(x)\right|\right|, \end{aligned} \tag{8.127}$$

so if we can choose

$$\frac{k\varepsilon}{2N} \leqslant \frac{\varepsilon^2}{2} - \eta \tag{8.128}$$

for some $\eta > 0$, then we obtain $d_{BL}(\hat{\mu}_{H_N(u)}, \mu_\infty(u)) \geqslant \eta$. Then applying (8.118) yields the result (8.124). (8.128) can be satisfied if

$$\varepsilon \geqslant \frac{k}{2N} + \frac{1}{2}\sqrt{\frac{k^2}{N^2} + 8\eta}. \tag{8.129}$$

So, given $\varepsilon > 0$, we can take N large enough such that, say, $\frac{k(N)}{N} < \frac{\varepsilon}{4}$. By taking $\eta < \frac{\varepsilon^2}{128}$ we obtain

$$\frac{k}{2N} + \frac{1}{2}\sqrt{\frac{k^2}{N^2} + 8\eta} < \frac{\varepsilon}{8} + \frac{1}{\sqrt{2}}\max\left(\sqrt{8\eta}, \frac{\varepsilon}{4}\right) < \frac{1+\sqrt{2}}{8}\varepsilon < \varepsilon \tag{8.130}$$

and so (8.129) is satisfied. Now finally (8.118) can be applied (with η in place of ε) and so we conclude (8.124).

Overall we see that the superexponential BL condition (8.118) is actually strong enough to deal with any $o(N)$ index not just index-0. This matches the GOE (or generally invariant ensemble) case, in which the terms with $\mathbf{1}\{i(H_N(u)) = k\}$ are suppressed compared to the exact minima terms $\mathbf{1}\{i(H_N(u)) = 0\}$. ■

Remark 8.6. Note that Corollary 8.1 establishes that, on the exponential scale, the number of critical points of any index $k(N) = o(N)$ is no more than the number of exact local minima.

8.4.2 The dichotomy of rough and smooth regions

Recall the batch loss from Section 8.1.1:

$$\frac{1}{b}\sum_{i=1}^{b} \mathcal{L}(f_{\boldsymbol{w}}(\boldsymbol{x}_i), y_i), \quad (\boldsymbol{x}_i, y_i) \overset{\text{i.i.d.}}{\sim} \mathbb{P}_{data}. \tag{8.131}$$

As with the Hessian in Section 8.1.1, we use the model $L \equiv L_{\text{batch}}(\boldsymbol{w}) = L_{\text{true}}(\boldsymbol{w}) + \mathsf{s}(b)V(\boldsymbol{w})$, where V is a random function $\mathbb{R}^N \to \mathbb{R}$.

Now let us define the complexity for sets $\mathscr{B} \subset \mathbb{R}^N$

$$C_N(\mathscr{B}) = |\{\boldsymbol{w} \in \mathscr{B} \mid \nabla L(\boldsymbol{w}) = 0\}|. \tag{8.132}$$

This is simply the number of stationary points of the training loss in the region $\mathscr{B}$ of weight space. A Kac-Rice formula applied to ∇L gives

$$\mathbb{E}C_N = \int_{\mathscr{B}} d\boldsymbol{w}\ \phi_{\boldsymbol{w}}(-\mathsf{s}(b)^{-1}\nabla L_{\text{true}})\mathbb{E}|\det(A + \mathsf{s}(b)X)| \tag{8.133}$$

where $\phi_{\boldsymbol{w}}$ is the density of ∇V at $\boldsymbol{w}$. A rigorous justification of this integral formula would, for example, have to satisfy the conditions of the results of [AT+07]. This is likely to be extremely difficult in any generality, though is much simplified in the case of Gaussian V (and X) - see [AT+07] Theorem 12.1.1 or Chapter 3, Lemma 3.5 ([Bas+21] Theorem 4.4). Hereafter, we shall take (8.133) as assumed. The next step is to make use of strong self-averaging of the random matrix determinants. Again, we are unable to establish this rigorously at present, but note that this property has been proved in some generality by [ABM21a], although we are unable to satisfy all the conditions of those results in any generality here. Self-averaging and using the addition results above gives

$$\frac{1}{N}\log\mathbb{E}|\det(A + \mathsf{s}(b)X)| = \int d(\mu_b \boxplus \nu)(\lambda)\log|\lambda| + o(1)$$

where μ_b, ν depend in principle on $\boldsymbol{w}$. We are concerned with $N^{-1}\log\mathbb{E}C_N$, and in particular its sign, which determines the complexity of the loss surface in $\mathscr{B}$: positive ↔ exponentially many (in N) critical points, negative ↔ exponentially few (i.e. none). The natural next step is to apply the Laplace method with large parameter N to determine the leading order term in $\mathbb{E}C_N$, however the integral is clearly not of the right form. Extra assumptions on $\phi_{\boldsymbol{w}}$ and ∇L_{true} could be introduced, e.g. that they can be expressed as functions of only a finite number of combinations of coordinates of $\boldsymbol{w}$.

Suppose that $\phi_{\boldsymbol{w}}$ has its mode at 0, for any $\boldsymbol{w}$, which is arguably a natural property, reflecting in a sense that the gradient noise has no preferred direction in $\mathbb{R}^N$. The sharp spike at the origin in the spectral density of deep neural network Hessians suggests that generically

$$\int d(\mu_b \boxplus \nu)(\lambda)\log|\lambda| < 0. \tag{8.134}$$

We claim it is reasonable to expect the gradient (and Hessian) variance to be increasing in $\|\boldsymbol{w}\|_2$. Indeed, consider the general form of the simplest deep neural network, a *multi-layer perceptron*:

$$f_{\boldsymbol{w}}(\boldsymbol{x}) = \sigma(\boldsymbol{b}^{(L)} + W^{(L)}\sigma(\boldsymbol{b}^{(L-1)} + W^{(L-1)}\ldots\sigma(\boldsymbol{b}^{(1)} + W^{(1)}\boldsymbol{x})\ldots)) \tag{8.135}$$

where all of the weight matrices $W^{(l)}$ and bias vectors $\boldsymbol{b}^{(l)}$ combine to give the weight vector $\boldsymbol{w}$. Viewing $\boldsymbol{x}$ as a random variable, making f a random function of $\boldsymbol{w}$, we expect from the above that the variance in $f_{\boldsymbol{w}}$ is generally increasing in $\|\boldsymbol{w}\|_2$, and so therefore similarly with L_{batch}.

Overall it follows that $\phi_{\boldsymbol{w}}(-\mathsf{s}(b)^{-1}\nabla L_{\text{true}})$ is generally decreasing in $\|\nabla L_{\text{true}}\|$, but the maximum value at $\phi_{\boldsymbol{w}}(0)$ is decreasing in $\|\boldsymbol{w}\|_2$. The picture is therefore that the loss surface is simple and without critical points in regions for which ∇L_{true} is far from 0. In neighbourhoods of $\nabla L_{\text{true}} = 0$, the loss surface may become complex, with exponentially many critical points, however if $\|\boldsymbol{w}\|_2$ is too large then the loss surface may still be without critical points. In addition, the effect of larger batch size (and hence larger $\mathsf{s}(b)^{-1}$) is to simplify the surface. These considerations indicate that deep neural network loss surfaces are simplified by over-parametrisation, leading to the spike in the Hessian spectrum and thus (8.134). The simple fact that neural networks' construction leads gradient noise variance to increase with $\|\boldsymbol{w}\|_2$ has the effect of simplifying the loss landscape far from the origin of weight space, and even precluding the existence of any critical points of the batch loss.

8.5 Implications for curvature from local laws

Consider a general stochastic gradient update rule with curvature-adjusted preconditioning:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \alpha B_t^{-1}\nabla L(\boldsymbol{w}_t) \tag{8.136}$$

where recall that $L(\boldsymbol{w})$ is the batch loss, viewed as a random function on weight space. B_t is some preconditioning matrix which in practice would be chosen to somehow approximate the curvature of L. Such methods are discussed at length in [Mar16a] and also describe some of the most successful optimisation algorithms used in practice, such as Adam [KB14]. The most natural choice for B_t is $B_t = \nabla^2 L(\boldsymbol{w}_t)$, namely the Hessian of the loss surface. In practice, it is standard to include a damping parameter $\delta > 0$ in B_t, avoid divergences when inverting. Moreover, typically B_t will be constructed to be some positive semi-definite approximation to the curvature such as the generalised Gauss Newton matrix [Mar16a], or the diagonal gradient variance form used in Adam [KB14]. Let us now suppose that $B_t = B_t(\delta) = \hat{H}_t + \delta$, where $\hat{H}_t$ is some chosen positive semi-definite curvature approximation and $\delta > 0$. We can now identify $B_t(\delta)^{-1}$ as in fact the Green's function of $\hat{H}_t$, i.e.

$$B_t(\delta)^{-1} = -(-\delta - \hat{H}_t)^{-1} = -G_t(-\delta). \tag{8.137}$$

But G_t is precisely the object used in the statement of a local law on for $\hat{H}_t$. Note that $\nabla L(\boldsymbol{w}_t)$ is a random vector and however $\hat{H}_t$ is constructed, it will generally be a random matrix and dependent on $\nabla L(\boldsymbol{w}_t)$ in some manner that is far too complicated to handle analytically. As we have discussed at length hitherto, we conjecture that a local law is reasonable assumption to make on random matrices arising in deep neural networks. In particular in Chapter 7 [BGK22] we demonstrated universal local random matrix theory statistics not just for Hessians of deep networks but also for Generalised Gauss-Newton matrices. Our aim here is to demonstrate how a local law on $\hat{H}_t$ dramatically simplifies the statistics of (8.136). Note that some recent work [WHS22] has also made use of random matrix local laws to simplify the calculation of test loss for neural networks.

A local law on $\hat{H}_t$ takes the precise form (for any $\xi, D > 0$

$$\sup_{\|\boldsymbol{u}\|,\|\boldsymbol{v}\|=1, z\in\boldsymbol{S}} \mathbb{P}\left(|\boldsymbol{u}^T G(z)\boldsymbol{v} - \boldsymbol{u}^T \Pi(z)\boldsymbol{v}| > N^{\xi}\left(\frac{1}{N\eta} + \sqrt{\frac{\Im g_{\mu}(z)}{N\eta}}\right)\right) \leqslant N^{-D} \tag{8.138}$$

where

$$\boldsymbol{S} = \left\{E + i\eta \in \mathbb{C} \mid |E| \leqslant \omega^{-1},\ N^{-1+\omega} \leqslant \eta \leqslant \omega^{-1}\right\} \tag{8.139}$$

μ is the limiting spectral measure of $\hat{H}_t$ and, crucially, Π is a *deterministic* matrix. We will use the following standard notation to re-express (8.138)

$$|\boldsymbol{u}^T G(z)\boldsymbol{v} - \boldsymbol{u}^T \Pi(z)\boldsymbol{v}| \prec \Psi_N(z), \quad \|\boldsymbol{u}\|, \|\boldsymbol{v}\| = 1, z \in \boldsymbol{S}, \tag{8.140}$$

where $\Psi_N(z) = \frac{1}{N\eta} + \sqrt{\frac{\Im g_{\mu}(z)}{N\eta}}$ and the probabilistic statement, valid for all $\xi, D > 0$ is implicit in the symbol $\prec$. In fact, we will need the local law outside the spectral support, i.e. at $z = x + i\eta$ where $x \in \mathbb{R}\backslash\text{supp}(\mu)$. In that case $\Psi_N(z)$ is replaced by $\frac{1}{N(\eta+\kappa)}$ where κ is the distance of x from $\text{supp}(\mu)$ on the real axis, i.e.

$$|\boldsymbol{u}^T G(z)\boldsymbol{v} - \boldsymbol{u}^T \Pi(z)\boldsymbol{v}| \prec \frac{1}{N(\eta+\kappa)}, \quad \|\boldsymbol{u}\|, \|\boldsymbol{v}\| = 1,\ x \in \mathbb{R}\backslash\text{supp}(\mu). \tag{8.141}$$

For $\delta > 0$ this becomes

$$|\boldsymbol{u}^T G(-\delta)\boldsymbol{v} - \boldsymbol{u}^T \Pi(-\delta)\boldsymbol{v}| \prec \frac{1}{N\delta}\|\boldsymbol{u}\|_2\|\boldsymbol{v}\|_2 \tag{8.142}$$

for $\delta > 0$ and now any $\boldsymbol{u}, \boldsymbol{v}$. Applying this to (8.136) gives

$$|\boldsymbol{u}^T B_t^{-1}\nabla L(\boldsymbol{w}_t) - \boldsymbol{u}^T \Pi_t(-\delta)\nabla L(\boldsymbol{w}_t)| \prec \frac{1}{N\delta}\|\boldsymbol{u}\|_2\|\nabla L(\boldsymbol{w}_t)\|_2. \tag{8.143}$$

Consider any $\boldsymbol{u}$ with $\|\boldsymbol{u}\|_2 = \alpha$, then we obtain

$$|\boldsymbol{u}^T B_t^{-1}\nabla L(\boldsymbol{w}_t) - \boldsymbol{u}^T \Pi_t(-\delta)\nabla L(\boldsymbol{w}_t)| \prec \frac{\alpha\|\nabla L(\boldsymbol{w}_t)\|_2}{N\delta}. \tag{8.144}$$

Thus with high probability, for large N, we can replace (8.136) by

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \alpha\Pi_t(-\delta)\nabla L(\boldsymbol{w}_t) \tag{8.145}$$

incurring only a small error, provided that

$$\delta >> \frac{\|\nabla L(\boldsymbol{w}_t)\|_2}{N}\alpha. \tag{8.146}$$

Note that the only random variable in (8.145) is $\nabla L(\boldsymbol{w}_t)$. If we now consider the case $\nabla L(\boldsymbol{w}_t) = \nabla\bar{L}(\boldsymbol{w}_t) + \boldsymbol{g}(\boldsymbol{w}_t)$ for deterministic $\bar{L}$, then

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \alpha\Pi_t(-\delta)\nabla\bar{L}(\boldsymbol{w}_t) - \alpha\Pi_t(-\delta)\boldsymbol{g}(\boldsymbol{w}_t) \tag{8.147}$$

and so the noise in the parameter update is entirely determined by the gradient noise. Moreover note the *linear* dependence on $\boldsymbol{g}$ in (8.147). For example, a Gaussian model for $\boldsymbol{g}$ immediately yields a Gaussian form in (8.147), and e.g. if $\mathbb{E}\boldsymbol{g} = 0$, then

$$\mathbb{E}(\boldsymbol{w}_{t+1} - \boldsymbol{w}_t) = -\alpha\Pi_t(-\delta)\mathbb{E}\nabla L(\boldsymbol{w}_t). \tag{8.148}$$

A common choice in practice for $\hat{H}$ is a diagonal matrix, e.g. the diagonal positive definite curvature approximation employed by Adam [KB14]. In such cases, $\hat{H}$ is best viewed as an approximation to the eigenvalues of some positive definite curvature approximation. The next result establishes that a local law assumption on a general curvature approximation matrix can be expected to transfer to an analogous result on a diagonal matrix of its eigenvalues.

Proposition 8.1. *Suppose that $\hat{H}$ obeys a local law of the form (8.141). Define the diagonal matrix D such that $D_i \overset{d}{=} \lambda_i$ where $\{\lambda_i\}_i$ are the sorted eigenvalues of $\hat{H}$. Let $G_D(z) = (z-D)^{-1}$ be the resolvent of D. Let $\mathfrak{q}_j[\mu]$ be the j-th quantile of μ, the limiting spectral density of $\hat{H}$, i.e.*

$$\int_{-\infty}^{\mathfrak{q}_j[\mu]} d\mu(\lambda) = \frac{j}{N}. \tag{8.149}$$

Then D obeys the local law

$$|(G_D)_{ij} - \delta_{ij}(z - \mathfrak{q}_j[\mu])^{-1}| \prec \frac{1}{N^{2/3}(\kappa+\eta)^2}, \quad z = x + i\eta, \ x \in \mathbb{R}\backslash supp(\mu), \tag{8.150}$$

where κ is the distance of x from $supp(\mu)$. Naturally, we can redefine $D_i = \lambda_{\sigma i}$ for any permutation $\sigma \in S_N$ and the analogous statement replacing $\mathfrak{q}_j[\mu]$ with $\mathfrak{q}_{\sigma(j)}$ will hold.

Proof. As in [EY17a], the local law (8.140), (8.141) is sufficient to obtain rigidity of the eigenvalues in the bulk, i.e. for any $\epsilon, D > 0$

$$\mathbb{P}\left(\exists j \mid |\lambda_j - \mathfrak{q}_j[\mu]| \geqslant N^{\epsilon}\left[\min(j, N-j+1)\right]^{-1/3} N^{-2/3}\right) \leqslant N^{-D}. \tag{8.151}$$

Then we have

$$\left|\frac{1}{z-\lambda_j} - \frac{1}{z - \mathfrak{q}_j[\mu]}\right| = \left|\frac{\lambda_j - \mathfrak{q}_j[\mu]}{(z-\lambda_j)(z-\mathfrak{q}_j[\mu])}\right|. \tag{8.152}$$

For $z = x + i\eta$ and x at a distance $\kappa > 0$ from $\text{supp}(\mu)$

$$|z - \mathfrak{q}_j[\mu]|^2 \geqslant \eta^2 + \kappa^2 \geqslant \frac{1}{2}(\eta+\kappa)^2, \tag{8.153}$$

and the same can be said for $|z - \mathfrak{q}_j[\mu]|^2$ with high probability, by applying the rigidity (8.151). A second application of rigidity to $|\lambda_j - \mathfrak{q}_j[\mu]|$ gives

$$\left|\frac{1}{z-\lambda_j} - \frac{1}{z - \mathfrak{q}_j[\mu]}\right| \prec \frac{1}{N^{2/3}\min(j, N-j+1)^{1/3}(\kappa+\eta)^2} \tag{8.154}$$

which yields the result. ■

With this result in hand, we get the generic update rule akin to (8.147), with high probability

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \alpha \text{ diag}\left(\frac{1}{\pi_j + \delta}\right)\nabla \bar{L}(\boldsymbol{w}_t) - \alpha \text{ diag}\left(\frac{1}{\pi_j + \delta}\right)\boldsymbol{g}(\boldsymbol{w}_t) \tag{8.155}$$

where $\{\pi_j\}_{j=1}^N$ are the eigenvalues of $\Pi_t(0)$ and we emphasise again that the π_j are *deterministic*; the only stochastic term is the gradient noise $\boldsymbol{g}(\boldsymbol{w}_t)$.

Implications for preconditioned stochastic gradient descent The key insight from this section is that generic random matrix theory effects present in preconditioning matrices of large neural networks can be expected to drastically simplify the optimisation dynamics due to high-probability concentration of the pre-conditioning matrices around deterministic equivalents, nullifying the statistical interaction between the pre-conditioning matrices and gradient noise. Moreover, with this interpretation, the damping constant typically added to curvature estimate matrices is more than a simple numerical convenience: it is essential to yield the aforementioned concentration results.

As an example of the kind of analysis that the above makes possible, consider the results of Chapter 5 (or see [Gra+21] for more details) . The authors consider a Gaussian process model for the noise in the loss surface, resulting in tractable analysis for convergence of stochastic gradient descent in the presence of statistical dependence between gradient noise in different iterations. Such a model implies a specific form of the loss surface Hessian and its statistical dependence on the gradient noise. This situation is a generalisation of the spin glass model exploited in various works [Cho+15] and in Chapters 3 and 4, except that in those cases the Hessian can be shown to be independent of the gradients. Absent the very special conditions that lead to independence, one expects the analysis to be intractable, hence why in Chapter 5 we restrict to stochastic gradient descent without preconditioning, or simply assume a high probability concentration on a deterministic equivalent. To make this discussion more concrete, consider a model $L = L_{\text{true}} + V$ where V is a Gaussian process with mean 0 and covariance function

$$K(\boldsymbol{x}, \boldsymbol{x}') = k\left(\frac{1}{2}\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2\right) q\left(\frac{1}{2}(\|\boldsymbol{x}\|_2^2 + \|\boldsymbol{x}'\|_2^2)\right), \tag{8.156}$$

where k is some decreasing function and q some increasing function. The discussion at the end of the previous section suggests that the covariance function for loss noise should not be modelled as stationary, hence the inclusion of the function q in (8.156). For convenience define $\Delta = \frac{1}{2}(\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2)$ and $S = \frac{1}{2}(\|\boldsymbol{x}\|_2^2 + \|\boldsymbol{x}'\|_2^2)$. Then it is a short exercise in differentiation to obtain

$$\begin{aligned}\text{Cov}\left(\partial_i V(\boldsymbol{w}), \partial_j V(\boldsymbol{w})\right) &= \text{Cov}\left(\partial_i V(\boldsymbol{w}), \partial_j V(\boldsymbol{w}')\right)\Big|_{\boldsymbol{w}=\boldsymbol{w}'} \\ &= \frac{\partial^2}{\partial w_i \partial w_j'} K(\boldsymbol{w}, \boldsymbol{w}')\Big|_{\boldsymbol{w}=\boldsymbol{w}'} \\ &= -k'(0) q(\|\boldsymbol{w}\|_2)\delta_{ij} + k(0) q''(\|\boldsymbol{w}\|_2^2) w_i w_j. \end{aligned} \tag{8.157}$$

and moreover

$$\begin{aligned}\operatorname{Cov}\left(\partial_{il}V(\boldsymbol{w}),\partial_{j}V(\boldsymbol{w})\right) &= \operatorname{Cov}\left(\partial_{il}V(\boldsymbol{w}),\partial_{j}V(\boldsymbol{w}')\right)\Big|_{\boldsymbol{w}=\boldsymbol{w}'}\\ &= \frac{\partial^3}{\partial w_i\partial w_l\partial w_j'}K(\boldsymbol{w},\boldsymbol{w}')\Big|_{\boldsymbol{w}=\boldsymbol{w}'}\\ &= -k'(0)q'(\|\boldsymbol{w}\|_2^2)w_l\delta_{ij}+q'''(\|\boldsymbol{w}\|_2^2)k(0)w_iw_lw_j'-k'(0)q'(\|\boldsymbol{x}\|_2)w_i\delta_{jl}.\end{aligned} \tag{8.158}$$

Hence we see that the gradients of L and its Hessian are statistically dependent by virtue of the non-stationary structure of V. Putting aside issues of positive definite pre-conditioning matrices, and taking δ such that $(\nabla^2 L+\delta)^{-1}$ exists (almost surely) for large N, it would appear that the distribution of $(\nabla^2 L+\delta)^{-1}\partial V$ will be complicated and non-Gaussian, assuming no extra information about the statistical interaction between the resolvent matrix and the gradient. This example concretely illustrates our point: even in almost the simplest case, where the gradient noise is Gaussian, the pre-conditioned gradients are generically considerably more complicated and non-Gaussian. Moreover, centred Gaussian noise on gradient is transformed into generically non-centred noise by pre-conditioning. Continuing the differentiation above, it is elementary to obtain the covariance structure of the Hessian $\nabla^2 V$, though the expressions are not instructive. Crucially, however, the Hessian is Gaussian and the covariance of any of its entries is $\mathcal{O}(1)$ (in large N), so the conditions in Example 2.12 of [EKS19] apply to yield an optimal local law on the Hessian, which in turn yields the above high-probability concentration of $(\nabla^2 L+\delta)^{-1}$ provided that δ is large enough. This argument ratifies an intuition from random matrix theory, that for large N the resolvent matrix $(\nabla^2 L+\delta)^{-1}$ is self-averaging and will be close, with high probability, to some deterministic equivalent matrix.

8.6 Conclusion

In this chapter we have considered several aspects of so-called universal random matrix theory behaviour in deep neural networks. Motivated by prior experimental results, we have introduced a model for the Hessians of DNNs that is more general than any previously considered and, we argue, actually flexible enough to capture the Hessians observed in real-world DNNs. Our model is built using random matrix theory assumptions that are more general than those previously considered and may be expected to hold in quite some generality. By proving a new result for the addition of random matrices, using a novel combination of quantum unique ergodicity and the supersymmetric method, we have derived expressions for the spectral outliers of our model. Using Lanczos approximation to the outliers of large, practical DNNs, we have compared our expressions for spectral outliers to data and demonstrated strong agreement for some DNNs. As well as corroborating our model, this analysis presents indirect evidence of the presence of universal local random matrix statistics in DNNs, extending earlier experimental results. Our analysis also highlights a possibly interesting distinction between some DNN architectures, as Resnet architectures appear to better agree with our theory than other architectures and Resnets have been previously observed to have better-behaved loss surfaces than many other architectures.

We also presented quite general arguments regarding the number of local optima of DNN loss surfaces and how ‘rough’ or ‘smooth’ such surfaces are. Our arguments build on a rich history of complexity calculations in the statistical physics and mathematics literature but, rather than performing detailed calculations in some specific, highly simplified toy model, we instead present general insights based on minimal assumptions. Finally we highlight an important area where random matrix local laws, an essential aspect of universality, may very directly influence the performance of certain popular optimisation algorithms for DNNs. Indeed, we explain how numerical damping, combined with random matrix local laws, can act to drastically simplify the training dynamics of large DNNs.

Overall this chapter demonstrates the relevance of random matrix theory to deep neural networks beyond highly simplified toy models. Moreover, we have shown how quite general and universal properties of random matrices can be fruitfully employed to derive practical, observable properties of DNN spectra. This work leaves several challenges for future research. All of our work relies on either local laws for e.g. DNN Hessians, or on matrix determinant self-averaging results. Despite the considerable progress towards establishing local laws for random matrices over the last decade or-so, it appears that establishing any such laws for, say, the Hessians of any DNNs is quite out of reach. We expect that the first progress in this direction will come from considering DNNs with random i.i.d. weights and perhaps simple activation functions. Based on the success of recent works on random DNNs [PS20], we conjecture that the Gram matrices of random DNN Jacobians may be the simplest place to establish a local law, adding to the nascent strand of *nonlinear* random matrix theory [PW17; BP19; PS20]. We also believe that there is more to be gained in further studies of forms of random matrix universality in DNNs. For example, our ideas may lead to tractable analysis of popular optimisation algorithms such as Adam [KB14] as the problem is essentially reduced to deriving a local law for the gradient pre-conditioning matrix and dealing with the gradient noise.

APPENDIX A

NEURAL NETWORKS WITH GENERAL ACTIVATION FUNCTIONS: SUPPLEMENTARY

This appendix provides supporting material for Chapter 3.

A.1 Specific expression for the low-rank perturbation matrix

The the rank-2 $N-1 \times N-1$ matrix S arises throughout the course of Sections 3.2 and 3.3 and Lemma 3.4. The specific value of S is not required at any point during our calculations and, even though its eigenvalues appear in the result of Theorem 3.4, it is not apparent that explicit expressions for its eigenvalues would affect the practical implications of the theorem. These considerations notwithstanding, in this supplementary section we collate all the expressions involved in the development of S from the modeling of the activation function in Section 3.2 through to Lemma 3.4. Beginning at the final expression for S in Lemma 3.4

$$S_{ij} = \frac{1}{\sqrt{2(N-1)H(H-1)}}\left(\xi_3 + \xi_2(\delta_{i1}+\delta_{j1}) + \xi_1\delta_{i1}\delta_{j1}\right), \tag{A.1}$$

where, recalling the re-scaling (3.53),

$$\xi_0 = \sum_{\ell=1}^{H} N^{-\ell/2}\rho_\ell^{(N)} \tag{A.2}$$

$$\xi_1 = \sum_{\ell=1}^{H-2} N^{-\ell/2}\rho_\ell^{(N)}\left[(H-\ell)(H-\ell-1)+1\right] \tag{A.3}$$

$$\xi_2 = \sum_{\ell=1}^{H-2} N^{-\ell/2}\rho_\ell^{(N)}(H-\ell-2) \tag{A.4}$$

$$\xi_3 = \sum_{\ell=1}^{H-2} N^{-\ell/2}\rho_\ell^{(N)} \tag{A.5}$$

The ρ_ℓ were defined originally in (3.36) and re-scaled around (3.43) so that

$$\rho_\ell = \frac{\mathbb{E}A_{i,j}^{(\ell)}}{\mathbb{E}A_{i,j}} \tag{A.6}$$

where $A_{i,j}$ are discrete random variables taking values in

$$\mathcal{A} := \left\{ \prod_{i=1}^{H} \alpha_{j_i} \ : \ j_1, \ldots, j_H \in \{1, \ldots, L\} \right\} \tag{A.7}$$

and $A_{i,j}^{(\ell)}$ take values in

$$\mathcal{A}^{(\ell)} := \left\{ \beta_k \prod_{r=1}^{H-\ell} \alpha_{j_r} \ : \ j_1, \ldots, j_{H-\ell}, k \in \{1, \ldots, L\} \right\} \tag{A.8}$$

but we have not prescribed the mass function of the $A_{i,j}$ or $A_{i,j}^{(\ell)}$. Lastly recall that the α_j, β_j are respectively the slopes and intercepts of the piece-wise linear function chosen to approximate the activation function f.

A.2 Experimental details

In this section we give further details of the experiments presented in Section 3.2.4.

The MLP architecture used consists of hidden layers of sizes $1000, 1000, 500, 250$. The CNN architecture used is a standard LeNet style architecture:

1. 6 filters of size 4×4.
2. Activation.
3. Max pooling of size 2×2 and stride 2.
4. 16 filters of size 4×4.
5. Activation.
6. Max pooling of size 2×2 and stride 2.
7. 120 filters of size 4×4.
8. Activation.
9. Dropout.
10. Fully connected to size 84.
11. Activation

12. Dropout.

13. Fully connected to size 10.

The activation functions used were the ubiquitous `ReLU` defined by

$$\texttt{ReLU}(x) = \max(0, x), \tag{A.9}$$

and `HardTanh` defined by

$$\texttt{HardTanh}(x) = \begin{cases} x & \text{for } x \in (-1, 1), \\ -1 & \text{for } x \leqslant -1, \\ 1 & \text{for } x \geqslant 1, \end{cases} \tag{A.10}$$

and a custom 5 piece function f_5 with gradients $0.01, 0.1, 1, 0.3, 0.03$ on $(-\infty, -2), (-2, -1), (-1, 1), (1, 2), (2, \infty)$ respectively, and $f_5(0) = 0$. We implemented all the networks and experiments in PyTorch [Pas+17] and our code is made available in the form of a Python notebook capable of easily reproducing all plots[1].

[1] `https://github.com/npbaskerville/loss-surfaces-general-activation-functions.`

Appendix B

A spin glass model for generative adversarial networks: supplemetary

This appendix provides supporting material for Chapter 4.

B.1 Bipartite spin-glass formulation

Recalling the expression for $\ell^{(G)}$, one could argue that a more natural formulation would be

$$\ell^{(G)}(\boldsymbol{w}^{(D)}, \boldsymbol{w}^{(G)}) = \sum_{i_1,\dots,i_p=1}^{N_D} \sum_{j_1,\dots,j_q=1}^{N_G} Z_{i_1,\dots,i_p,j_1,\dots,j_q} \prod_{k=1}^{p} w_{i_k}^{(D)} \prod_{l=1}^{q} w_{j_l}^{(G)}$$

for i.i.d. Gaussian Z. In this case, each term in the sum contains exactly p weights from the discriminator network and q weights from the generator. This object is known as a bipartite spin glass. We will now present the Gaussian calculations. We need the joint distributions

$$\left(\ell^{(D)}, \partial_i^{(D)}\ell^{(D)}, \partial_{jk}^{(D)}\ell^{(D)}\right), \quad \left(\ell^{(G)}, \partial_i^{(G)}\ell^{(G)}, \partial_{jk}^{(G)}\ell^{(G)}, \partial_l^{(D)}\ell^{(G)}, \partial_{mn}^{(D)}\ell^{(G)}\right)$$

where the two groups are independent from of each other. As in [AAC13], we will simplify the calculation by evaluating in the region of the north poles on each hyper-sphere. $\ell^{(D)}$ behaves just like a single spin glass, and so we have [AAC13]:

$$Var(\ell^{(D)}) = 1, \tag{B.1}$$

$$Cov(\partial_i^{(D)}\ell^{(D)}, \partial_{jk}^{(D)}\ell^{(D)}) = 0, \tag{B.2}$$

$$\partial_{ij}^{(D)}\ell^{(D)} \mid \{\ell^{(D)} = x_D\} \sim \sqrt{(N_D-1)p(p-1)}GOE^{N_D-1} - x_D pI, \tag{B.3}$$

$$Cov(\partial_i^{(D)}\ell^{(D)}, \partial_j^{(D)}\ell^{(D)}) = p\delta_{ij}. \tag{B.4}$$

To find the joint and thence conditional distributions for $\ell^{(G)}$, we first compute the covariance function, which follows from the independence of the Z:

$$Cov(\ell^{(G)}(\boldsymbol{w}^{(D)},\boldsymbol{w}^{(G)}),\ell^{(G)}(\boldsymbol{w}^{(D)'},\boldsymbol{w}^{(G)'})) \tag{B.5}$$

$$=\sum_{\substack{i_1,\ldots,i_p=1\\ i'_1,\ldots,i'_p=1}}^{N_D}\ \sum_{\substack{j_1,\ldots,j_q=1\\ j'_1,\ldots,j'_q=1}}^{N_G} \mathbb{E}Z_{\boldsymbol{i}\boldsymbol{i}}Z_{\boldsymbol{i}'\boldsymbol{j}'}\prod_{k=1}^{p} w_{i_k}^{(D)}w_{i'_k}^{(D)'}\prod_{l=1}^{q} w_{j_l}^{(G)}w_{j'_l}^{(G)'} \tag{B.6}$$

$$=\sum_{i_1,\ldots,i_p=1}^{N_D}\ \sum_{j_1,\ldots,j_q=1}^{N_G}\prod_{k=1}^{p} w_{i_k}^{(D)}w_{i_k}^{(D)'}\prod_{l=1}^{q} w_{j_l}^{(G)}w_{j_l}^{(G)'} \tag{B.7}$$

$$=(\boldsymbol{w}^{(D)}\cdot\boldsymbol{w}^{(D)'})^p(\boldsymbol{w}^{(G)}\cdot\boldsymbol{w}^{(G)'})^q \tag{B.8}$$

The product structure of the covariance function implies that we can write down the following covariances directly from the simple spin-glass case, as the $\partial^{(D)}$ and $\partial^{(G)}$ derivatives act independently on their respective terms:

$$Var(\ell^{(G)}) = 1, \tag{B.9}$$

$$Cov(\partial_{ij}^{(G)}\ell^{(G)},\ell^{(G)}) = -q\delta_{ij}, \tag{B.10}$$

$$Cov(\partial_{ij}^{(D)}\ell^{(G)},\ell^{(G)}) = -p\delta_{ij}, \tag{B.11}$$

$$Cov(\partial_{ij}^{(G)}\ell^{(G)},\partial_{kl}^{(G)}\ell^{(G)}) = q(q-1)\left(\delta_{ik}\delta_{jl}+\delta_{il}\delta_{jk}\right)+q^2\delta_{ij}\delta_{kl}, \tag{B.12}$$

$$Cov(\partial_{ij}^{(D)}\ell^{(G)},\partial_{kl}^{(D)}\ell^{(G)}) = p(p-1)\left(\delta_{ik}\delta_{jl}+\delta_{il}\delta_{jk}\right)+p^2\delta_{ij}\delta_{kl}, \tag{B.13}$$

$$Cov(\partial_{ij}^{(G)}\ell^{(G)},\partial_{kl}^{(D)}\ell^{(G)}) = pq\delta_{ij}\delta_{kl}, \tag{B.14}$$

$$Cov(\partial_{i}^{(G)}\partial_{j}^{(D)}\ell^{(G)},\partial_{k}^{(G)}\partial_{l}^{(D)}\ell^{(G)}) = pq\delta_{ik}\delta_{jl}, \tag{B.15}$$

$$Cov(\partial_{ij}^{(G)}\ell^{(G)},\partial_{k}^{(G)}\partial_{l}^{(D)}\ell^{(G)}) = 0 \tag{B.16}$$

$$Cov(\partial_{ij}^{(D)}\ell^{(G)},\partial_{k}^{(D)}\partial_{l}^{(G)}\ell^{(G)}) = 0, \tag{B.17}$$

$$Cov(\partial_{i}^{(D)}\partial_{j}^{(G)}\ell^{(G)},\ell^{(G)}) = 0. \tag{B.18}$$

Also, all first derivatives of $\ell^{(G)}$ are clearly independent of $\ell^{(G)}$ and its second derivatives by the same reasoning and

$$Cov(\partial_{i}^{(G)}\ell^{(G)},\partial_{j}^{(G)}\ell^{(G)}) = q\delta_{ij}, \tag{B.19}$$

$$Cov(\partial_{i}^{(D)}\ell^{(G)},\partial_{j}^{(D)}\ell^{(G)}) = p\delta_{ij}, \tag{B.20}$$

$$Cov(\partial_{i}^{(D)}\ell^{(G)},\partial_{j}^{(G)}\ell^{(G)}) = 0. \tag{B.21}$$

We caw deduce the full gradient covariances, recalling that $\ell^{(D)}$ and $\ell^{(G)}$ are independent:

$$Cov(\partial_{i}^{(D)}L^{(D)},\partial_{j}^{(D)}L^{(D)}) = p(1+\sigma_z^2)\delta_{ij} \tag{B.22}$$

$$Cov(\partial_{i}^{(G)}L^{(G)},\partial_{j}^{(G)}L^{(G)}) = \sigma_z^2 q\delta_{ij} \tag{B.23}$$

$$Cov(\partial_{i}^{(D)}L^{(D)},\partial_{j}^{(G)}L^{(G)}) = 0 \tag{B.24}$$

and so

$$\varphi_{(\nabla_D L^{(D)}, \nabla_G L^{(G)})}(0) = (2\pi)^{-\frac{N-2}{2}} \left(p + \sigma_z^2 p\right)^{-\frac{N_D-1}{2}} \left(\sigma_z^2 q\right)^{-\frac{N_G-1}{2}}. \tag{B.25}$$

We need now to calculate the joint distribution of $(\partial_{ij}^{(D)} \ell^{(G)}, \partial_{kl}^{(G)} \ell^{(G)})$ conditional on $\{\ell^{(G)} = x_G\}$. Denote the covariance matrix for $(\partial_{ij}^{(D)} \ell^{(G)}, \partial_{kl}^{(G)} \ell^{(G)}, \ell^{(G)})$ by

$$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} \tag{B.26}$$

where

$$\Sigma_{11} = \begin{pmatrix} p(p-1)(1+\delta_{ij}) + p^2\delta_{ij} & pq\delta_{ij}\delta_{kl} \\ pq\delta_{ij}\delta_{kl} & q(q-1)(1+\delta_{kl}) + q^2\delta_{kl} \end{pmatrix}, \tag{B.27}$$

$$\Sigma_{12} = -\begin{pmatrix} p\delta_{ij} \\ q\delta_{kl} \end{pmatrix}, \tag{B.28}$$

$$\Sigma_{21} = -\begin{pmatrix} p\delta_{ij} & q\delta_{kl} \end{pmatrix}, \tag{B.29}$$

$$\Sigma_{22} = 1. \tag{B.30}$$

The conditional covariance is then

$$\bar{\Sigma} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \tag{B.31}$$

$$= \begin{pmatrix} p(p-1)(1+\delta_{ij}) & 0 \\ 0 & q(q-1)(1+\delta_{kl}) \end{pmatrix}. \tag{B.32}$$

Repeating this calculation for $(\partial_{ij}^{(G)} \ell^{(G)}, \partial_{kl}^{(G)} \ell^{(G)}, \ell^{(G)})$ demonstrates that $\nabla_G^2 \ell^{(G)} \mid \{\ell^{(G)} = x_G\}$ has independent entries, up-to symmetry. The result (B.32) demonstrates that, conditional on $\{\ell^{(G)} = x_G\}$, $\nabla_G^2 \ell^{(G)}$ and $\nabla_D^2 \ell^{(G)}$ are independent GOEs. In summary, from (B.32) and (B.15-B.17) we obtain

$$\begin{pmatrix} -\nabla_D^2 \ell^{(G)} & -\nabla_G \nabla_D \ell^{(G)} \\ \nabla_D \nabla_G \ell^{(G)} & \nabla^2 \ell^{(G)} \end{pmatrix} \mid \{\ell^{(G)} = x_G\} \overset{d}{=} \sqrt{2} \begin{pmatrix} \sqrt{N_D - 1}\sqrt{p(p-1)} M^{(D)} & -2^{-1/2}\sqrt{pq} G \\ 2^{-1/2}\sqrt{pq} G^T & \sqrt{N_G - 1}\sqrt{q(q-1)} M^{(G)} \end{pmatrix}$$
$$- x_G \begin{pmatrix} -pI_{N_D} & 0 \\ 0 & qI_{N_G} \end{pmatrix} \tag{B.33}$$

where $M^{(D)} \sim GOE^{N_D - 1}$ and $M^{(G)} \sim GOE^{N_G - 1}$ are independent GOEs and G is an independent $N_D - 1 \times N_G - 1$ Ginibre matrix with entries of unit variance.

At this point a problem becomes apparent. Suppose that $q \leqslant p$, then the variance of the lower-right block is strictly less than that of the off diagonal blocks. If we proceed with the strategy in the main text, there is no way of decomposing the lower-right block as a sum of two independent smaller variance GOEs with one matching the variance of the off diagonal blocks. Similarly, if

$q > p$, then the final Hessian involving $L^{(D)}, L^{(G)}$ will have lower-variance in the upper-left block than the off-diagonals unless very specific undesirable conditions hold on p, q and σ_z. In either of these cases, we cannot decompose the final Hessian as a sum of a large $N-2 \times N-2$ GOE and some smaller GOEs in the upper-left or lower-right blocks. We would therefore have to truly compute the Ginibre averages in the supersymmetric method, which we believe is intractable.

We could complete the complexity calculation via the methods of chapter 4 supposing that the appropriate conditions hold on p, q and σ_z. It would look much the same as the calculation in the main text, though the resulting polynomial for the spectral density would be different. Since this work was completed, the complexity results for bipartite spin glasses were obtained in [McK21] using an entirely new method developed in the companion paper [ABM21a]. Applying this method arguably presents more technical hurdles than the supersymmetric approach to complexity calculations, however it is much more general and can be applied to the above model for any p, q and σ_z.

B.2 Extra plots

This section contains some extra plots to back up the comparisons between our model's predictions and the experimental DCGAN results in Section 4.5.2. In particular, we produce versions of the plots in Figures 4.8 and 4.9 but for various values of p and q other than $p = q = 5$. Since $p = q = 5$ is the structurally correct choice for the DCGAN, it is natural to ask if any agreement between theory and experiment is most closely obtained with $p = q = 5$. Figure B.4 shows that the model has the same deficiency in κ for all p, q values tested. Figure B.3 shows best agreement for $p = q = 5$, $p = 3, q = 7$ and $p = 7, q = 3$, and similarly in Figure B.2. There is perhaps weak evidence that the role of p and q as representing the number of layers in the networks has some merit experimentally.

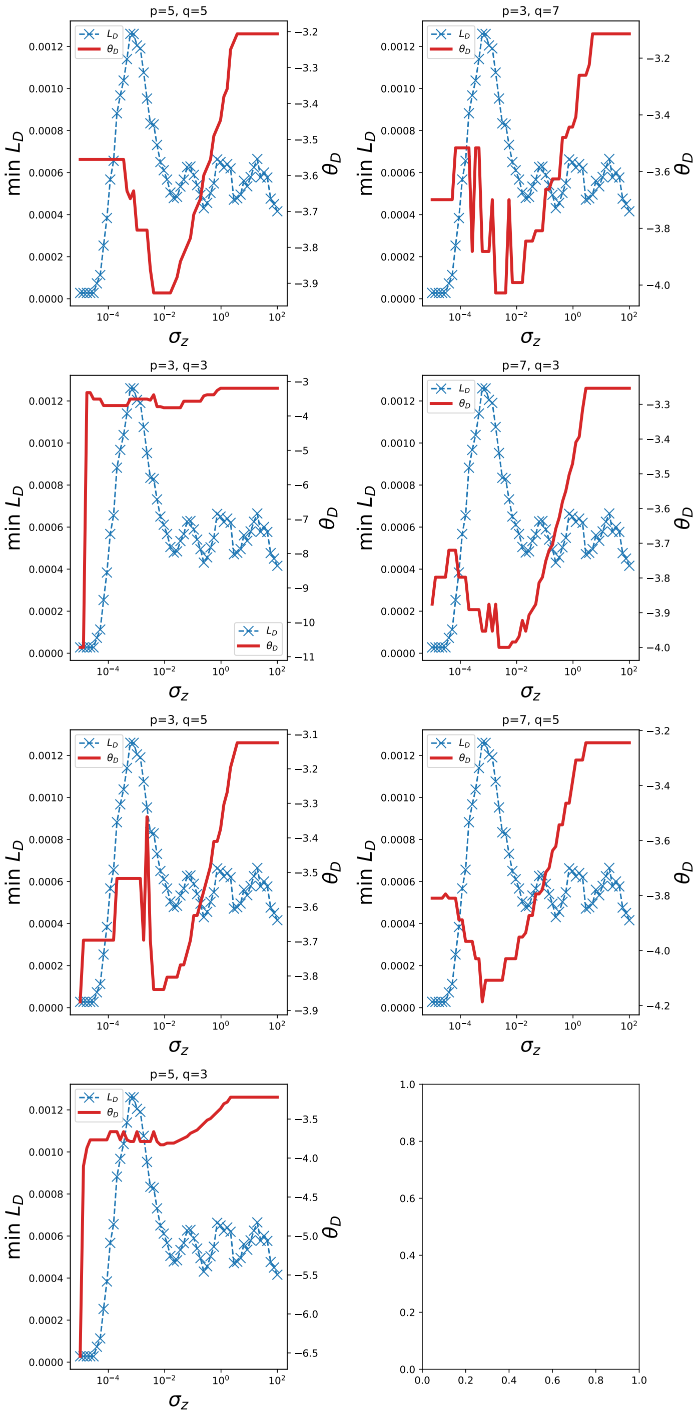


Figure B.1: The effect of σ_z on minimum L_D. Comparison of theoretical predictions of minimum possible discriminator and generator losses to observed minimum losses when training DCGAN on CIFAR10. The blue cross-dashed lines show the experimental DCGAN results, and the solid red show the theoretical results ϑ_G, ϑ_D. $\kappa = 0.5$ is used and p, q are varied.

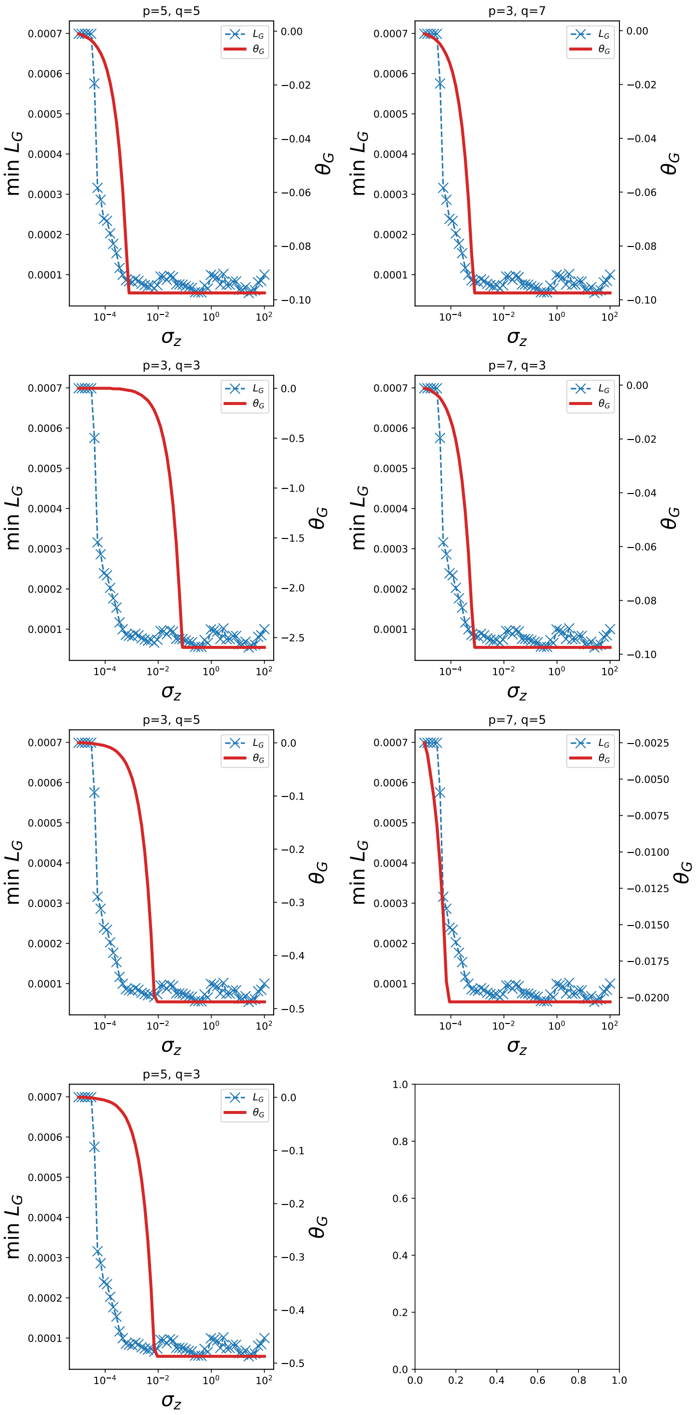


Figure B.2: The effect of σ_z on minimum L_G. Comparison of theoretical predictions of minimum possible discriminator and generator losses to observed minimum losses when training DCGAN on CIFAR10. The blue cross-dashed lines show the experimental DCGAN results, and the solid red show the theoretical results ϑ_G, ϑ_D. $\kappa = 0.5$ is used and p, q are varied.

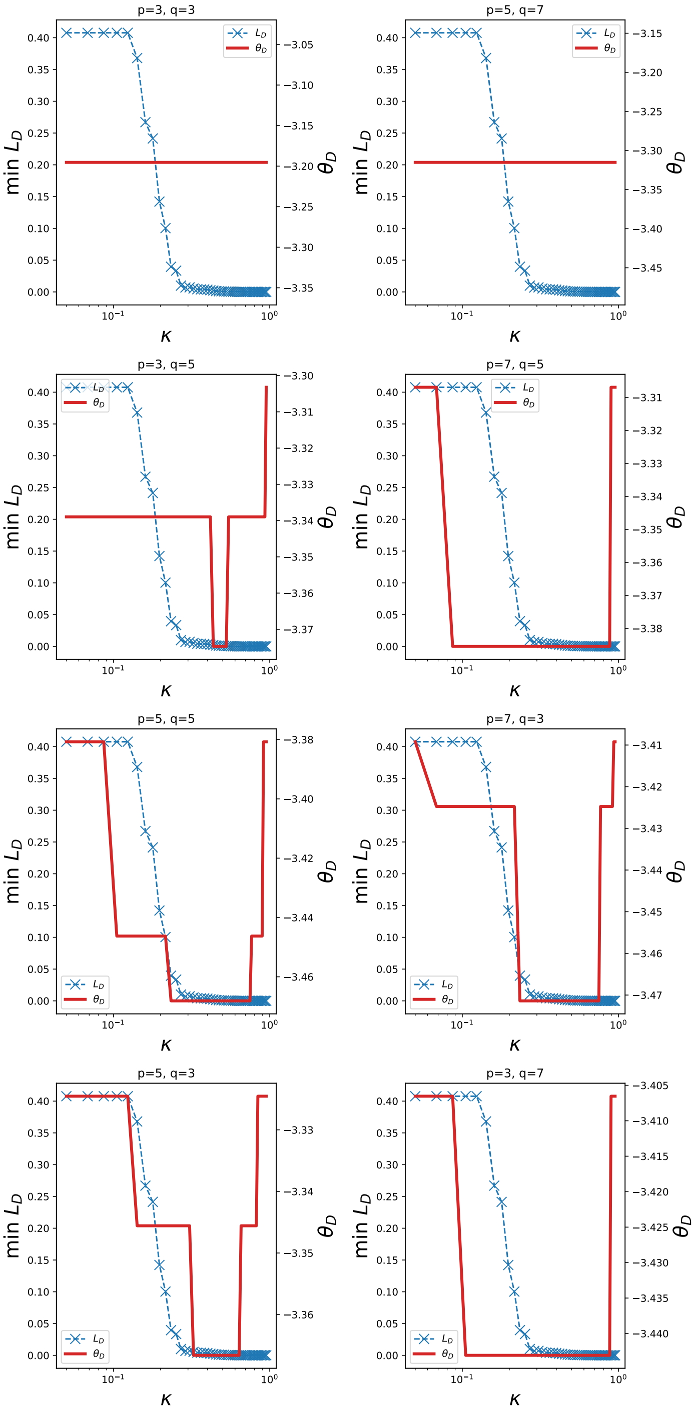


Figure B.3: The effect of κ on minimum L_D. Comparison of theoretical predictions of minimum possible discriminator and generator losses to observed minimum losses when training DCGAN on CIFAR10. The blue cross-dashed lines show the experimental DCGAN results, and the solid red show the theoretical results ϑ_G, ϑ_D. $\sigma_z = 1$ is used and p, q are varied.

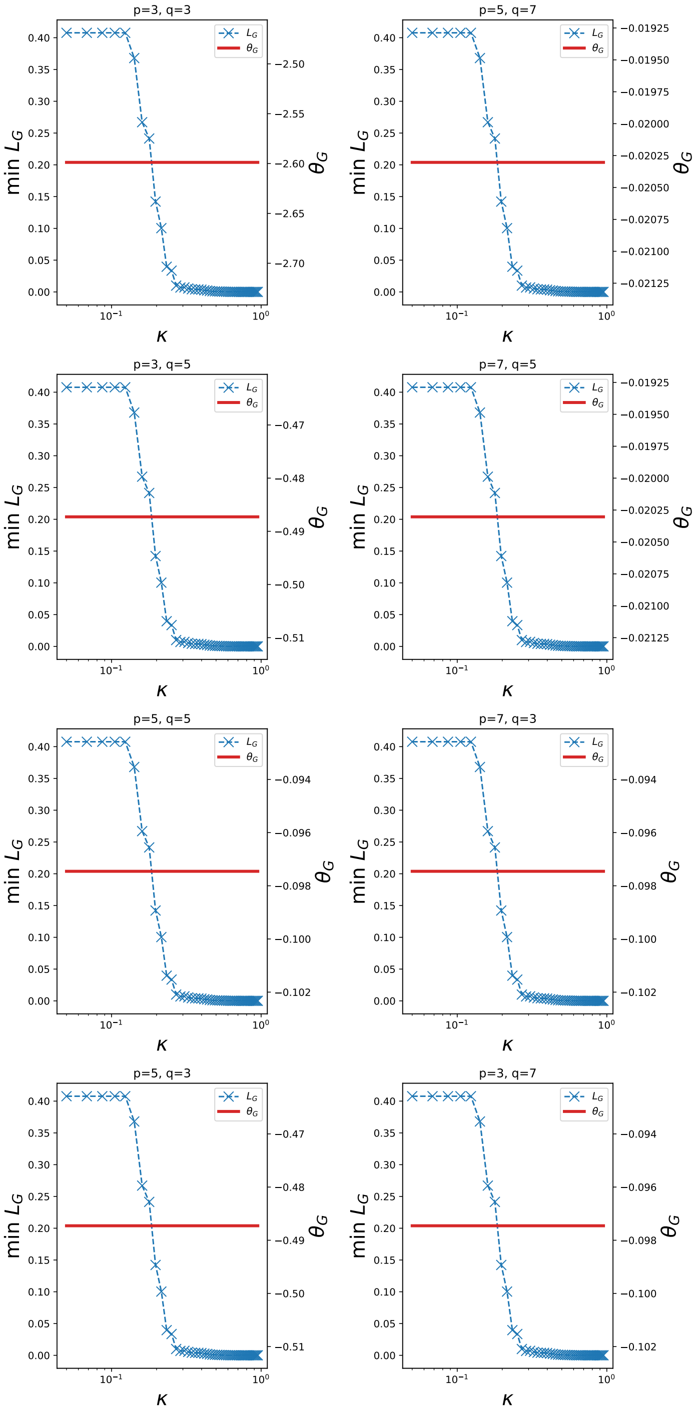


Figure B.4: The effect of κ on minimum L_G. Comparison of theoretical predictions of minimum possible discriminator and generator losses to observed minimum losses when training DCGAN on CIFAR10. The blue cross-dashed lines show the experimental DCGAN results, and the solid red show the theoretical results ϑ_G, ϑ_D. $\sigma_z = 1$ is used and p, q are varied.

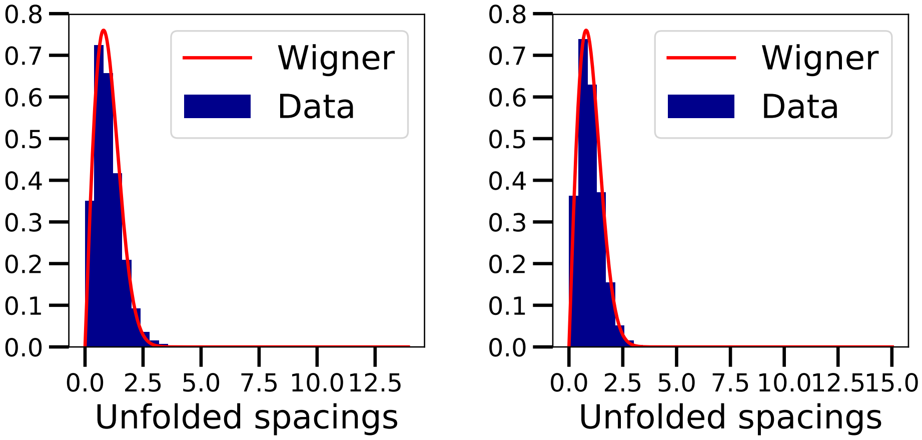
Appendix C

Appearance of local random matrix statistics: supplementary

This appendix provides supporting material for Chapter 7.

C.1 Extra Figures and Degeneracy Investigation

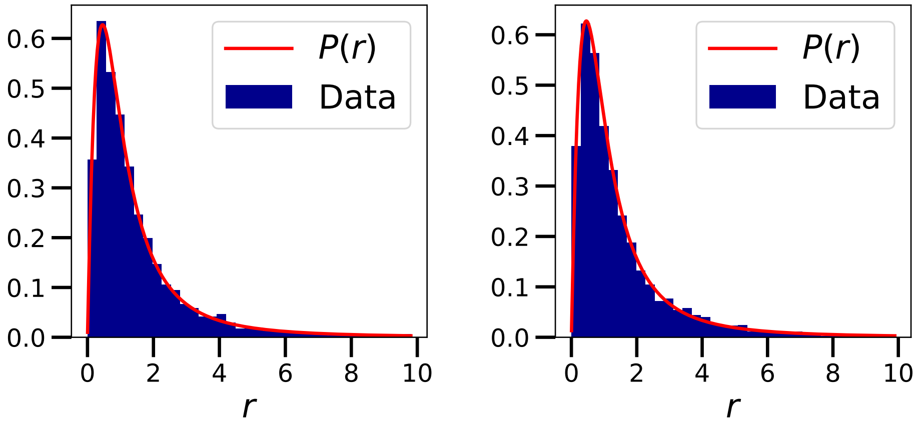
Figure C.5 compares the effect of degeneracy on unfolded spacings in each of the 3 cases considered. We see that the logistic MNIST models (trained and untrained) have a much greater level of degeneracy, whereas the CIFAR10-Resnet34 spectra clearly have GOE spacings even without any cut-off. Figures C.2–C.4 show further unfolded spacing and spacing ratio results like those in the main text.



(a) Batch train

(b) Batch test

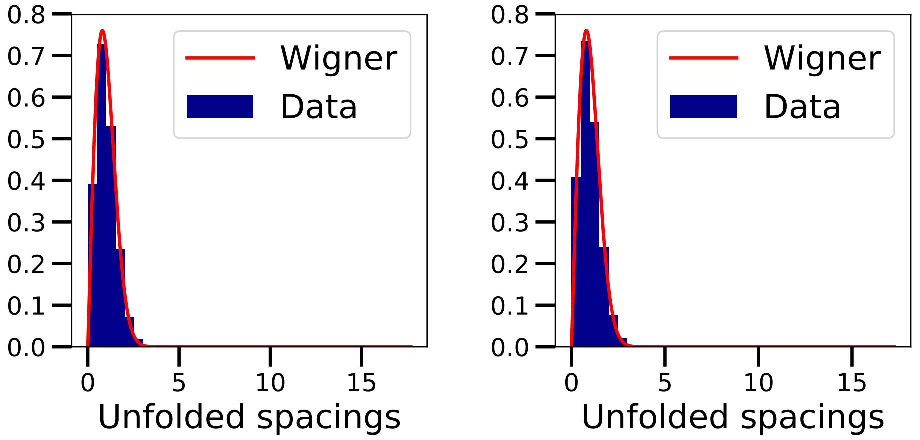
Figure C.1: Unfolded spacings for the Hessian of a logistic regression trained on MNIST. Hessian computed batches of size 64 of the training and test datasets.



(a) All train

(b) All test

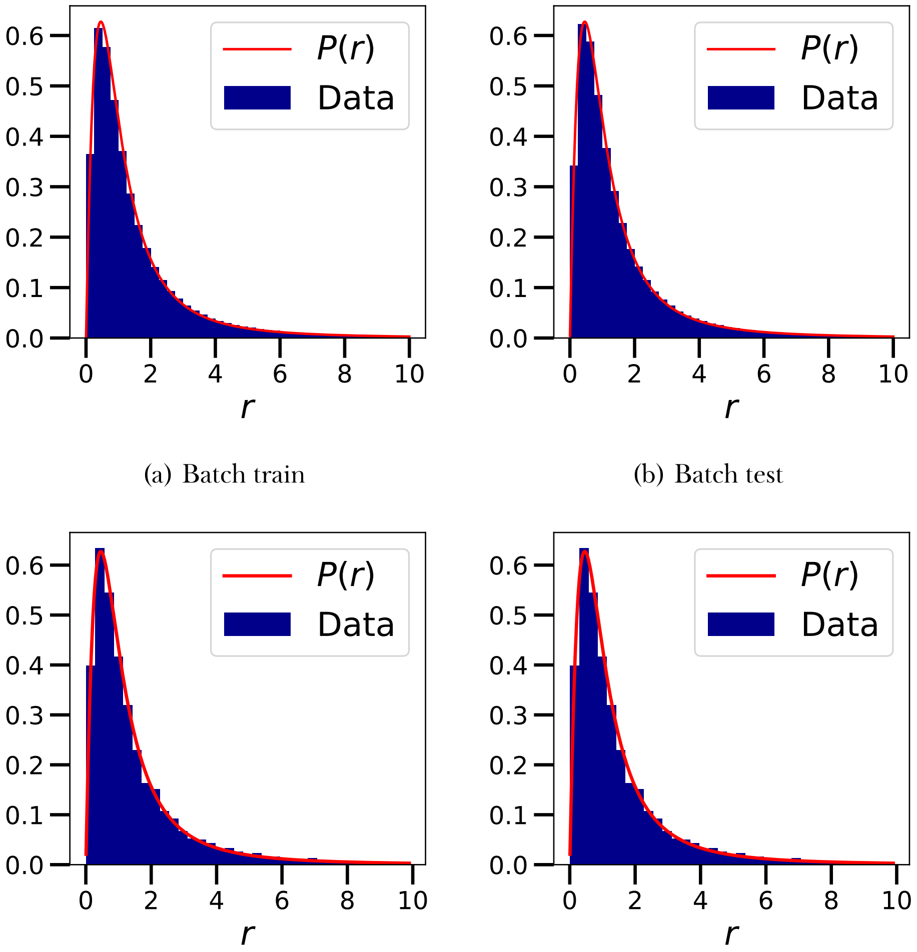
Figure C.2: Consecutive spacing ratios for the Hessian of a logistic regression trained on MNIST. Hessian computed batches of size 64 of the training and test sets, and over the whole train and test sets.



(a) Batch train

(b) Batch test

Figure C.3: Unfolded spacings for the Hessian of a randomly initialised logistic regression for MNIST. Hessian computed batches of size 64 of the training and test datasets.



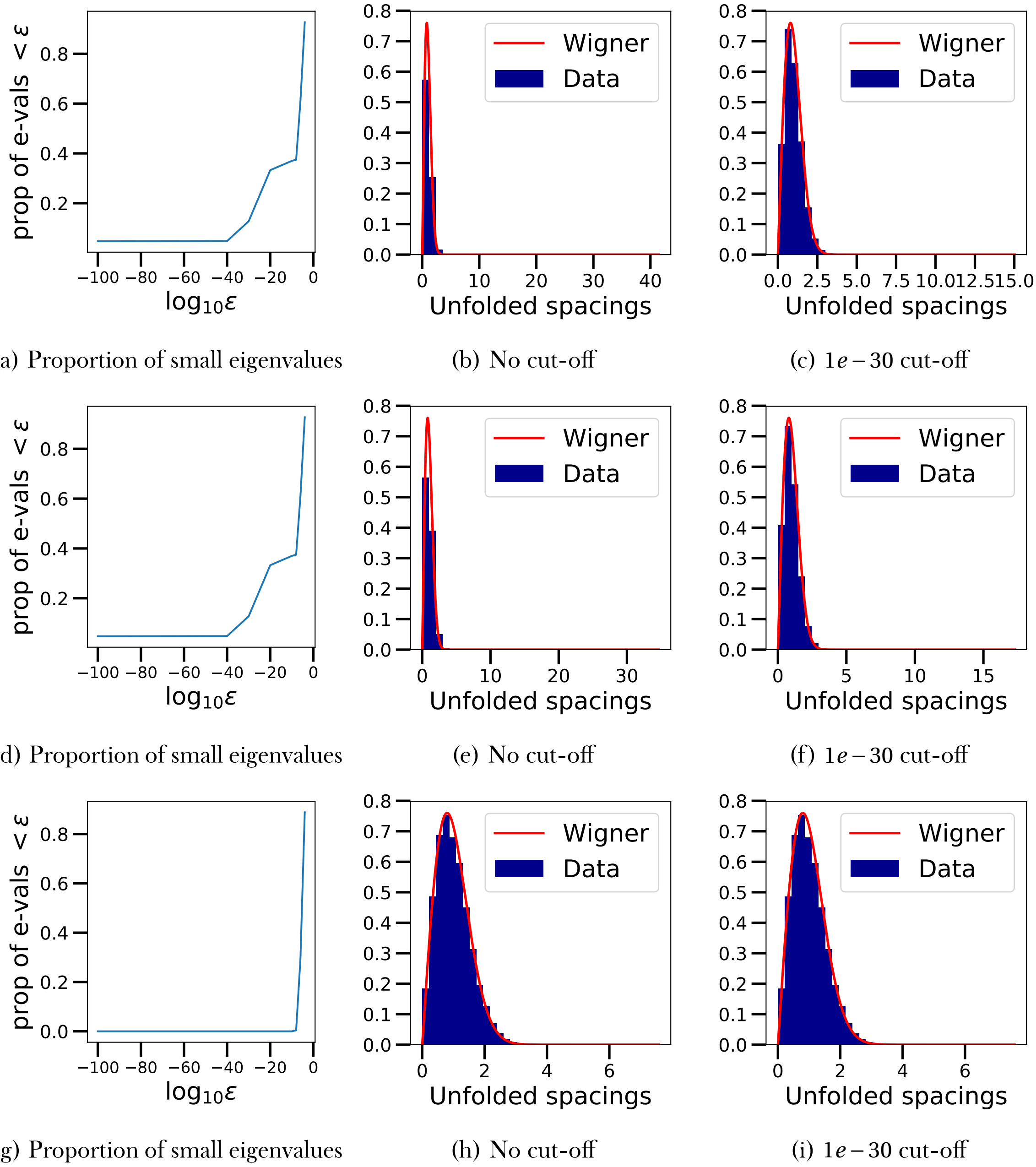
(a) Batch train

(b) Batch test

(c) All train

(d) All test

Figure C.4: Consecutive spacing ratios for the Hessian of a randomly initialised logistic regression for MNIST. Hessian computed batches of size 64 of the training and test sets, and over the whole train and test sets.



(a) Proportion of small eigenvalues

(b) No cut-off

(c) $1e-30$ cut-off

(d) Proportion of small eigenvalues

(e) No cut-off

(f) $1e-30$ cut-off

(g) Proportion of small eigenvalues

(h) No cut-off

(i) $1e-30$ cut-off

Figure C.5: Unfolded spacings for the Hessian of a logistic regression. Showing MNIST (top), untrained MNIST (middle) and Resnet34 embedded CIFAR10 (bottom). Comparing the effect of a cuff-off for very small eigenvalues.

C.2 Experimental details

C.2.1 Network architectures

Logistic regression (MNIST)

1. Input features 784 to 10 output logits.

2-layer MLP (MNIST)

1. Input features 784 to 10 neurons.
2. 10 neurons to 100 neurons.
3. 100 neurons to 10 output logits.

3-layer MLP (MNIST)

1. Input features 784 to 10 neurons.
2. 10 neurons to 100 neurons.
3. 100 neurons to 100 neurons.
4. 100 neurons to 10 output logits.

Logistic regression on ResNet features (CIFAR10)

1. Input features 513 to 10 neurons.

LeNet (CIFAR10)

1. Input features 32x32x3 through 5x5 convolution to 6 output channels.
2. 2x2 max pooling of stride 2.
3. 5x5 convolution to 16 output channels.
4. 2x2 max pooling of stride 2.
5. Fully connection layer from 400 to 120.
6. Fully connection layer from 120 to 84.
7. Fully connection layer from 84 to output 10 logits.

MLP (CIFAR10)

1. 3072 input features to 10 neurons.
2. 10 neurons to 300 neurons.
3. 300 neurons to 100 neurons.

MLP (Bike)

1. 13 input features to 100 neurons.
2. 100 neurons to 100 neurons.
3. 100 neurons to 50 neurons.
4. 50 neurons to 1 regression output.

C.2.2 Other details

All networks use the same (default) initialisation of weights in PyTorch, which is the 'Kaiming uniform' method of [He+15]. All networks used ReLU activation functions.

C.2.3 Data pre-processing

For the image datasets MNIST and CIFAR10 we use standard computer vision pre-processing, namely mean and variance standardisation across channels. We refer to the accompanying code for the precise procedure

The Bike dataset has 17 variables in total, namely: `instant`, `dteday`, `season`, `yr`, `mnth`, `hr`, `holiday`, `weekday`, `workingday`, `weathersit`, `temp`, `atemp`, `hum`, `windspeed`, `casual`, `registered`, `cnt`. All variables are either positive integers or real numbers. It is standard to view `cnt` as the regressand, so one uses some or all of the remaining features to predict `cnt`. This is the approach we take, however we slightly reduce the number of features by dropping `instant`, `casual`, `registered`, since `instant` is just an index and `casual+registered=cnt`, so including those features would render the problem trivial. We map `dteday` to a integer uniquely representing the date and we standardise `cnt` by dividing by its mean.

Appendix D

Universal characteristics of loss surfaces: supplementary

This appendix provides supporting material for Chapter 8 including full details of the experimental set-up and analysis for the outlier experiments.

D.1 Architectures and training of models.

We use the GPU powered Lanczos quadrature algorithm [Gar+18; MS06], with the Pearlmutter trick [Pea94] for Hessian vector products, using the PyTorch [Pas+17] implementation of both Stochastic Lanczos Quadrature and the Pearlmutter. We then train a 16 Layer VGG CNN [SZ14] with $P = 15291300$ parameters and the 28 Layer Wide Residual Network [ZK16; He+16] architectures on the CIFAR-100 dataset [KH+09] (45,000 training samples and 5,000 validation samples) using SGD. We use the following learning rate schedule:

$$\alpha_t = \begin{cases} \alpha_0, & \text{if } \frac{t}{T} \leqslant 0.5 \\ \alpha_0[1 - \frac{(1-r)(\frac{t}{T}-0.5)}{0.4}] & \text{if } 0.5 < \frac{t}{T} \leqslant 0.9 \\ \alpha_0 r, & \text{otherwise.} \end{cases} \tag{D.1}$$

We use a learning rate ratio $r = 0.01$ and a total number of epochs budgeted $T = 300$. We further use momentum set to $\rho = 0.9$, a weight decay coefficient of 0.0005 and data-augmentation on PyTorch [Pas+17].

D.2 Implementation of constraints

As mentioned in the main text, one of the three weights of the linear model fit in the outlier analysis, β, is constrained to be positive, as it corresponds to a second cumulant, i.e. a variance, of a probability measure. Recall that the linear model's parameters are solved exactly as functions of the unknown $\theta^{(i)}$, and these parameters are in turn optimised using gradient descent. β is unconstrained during the linear solve, but its value is determined by the $\theta^{(i)}$, so to impose the constraint $\beta > 0$ we add to the mean squared error loss the term

$$\beta = 1000 \max(0, -\beta) \tag{D.2}$$

which penalises negative β values and is minimised at any non-negative value. The factor 1000 was roughly tuned by hand to give consistently positive values for β.

There is also the constraint that $\theta^{(i)} > \theta^{(i+1)} > 0$ for all i. This is imposed simply using a re-parametrisation. We introduce unconstrained raw value $t^{(i)}$ taking values in $\mathbb{R}$ and define

$$\theta^{(i)} = \sum_{j=1}^{i} \log(1 + \exp(t^{(j)})),$$

then the gradient descent optimisation is simply performed over the $t^{(i)}$.

D.3 Fitting of outlier model

We optimise the mean squared error with respect to the raw parameters $t^{(i)}$ using 200 iterations of Adam [KB14] with a learning rate of 0.2. The learning rate was chosen heuristically by increasing in steps until training became unstable. The number of iterations was chosen heuristically as being comfortably sufficient to obtain convergence of Adam. The raw parameters $t^{(i)}$ were initialised by drawing independently from a standard Gaussian. The $t^{(i)}$ were initialised and trained using the above method 20 times and the values with the lowest mean squared error were chosen.

Appendix E

A random matrix approach to damping in deep learning: supplementary

This appendix provides some extra training plots in support of Chapter 6.

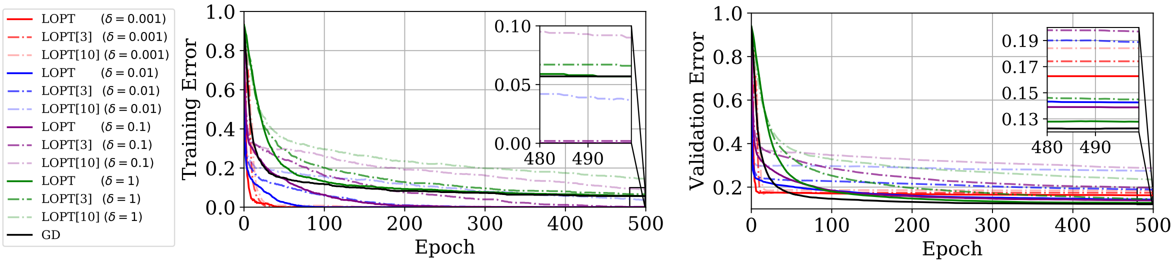


Figure E.1: Training/test error of LanczosOPT/Gradient Descent (LOPT/GD) optimisers for logistic regression on the MNIST dataset with fixed learning rate $\alpha = 0.01$ across different damping values, δ. LOPT[η] denotes a modification to the LOPT algorithm that perturbs a subset of update directions by a factor of η. Best viewed in colour.

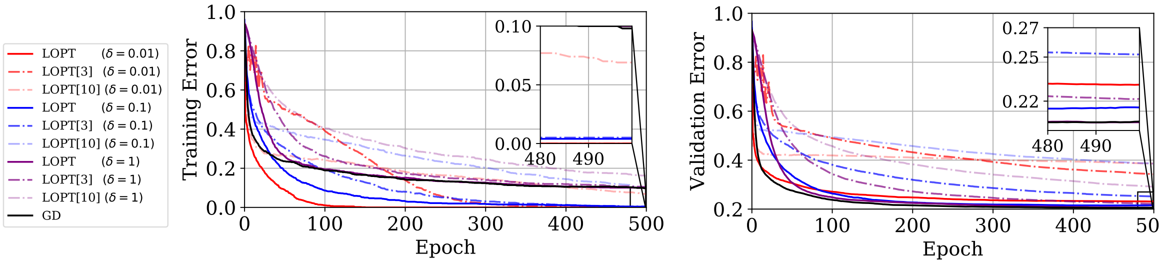


Figure E.2: Training/test error of LanczosOPT/Gradient Descent (LOPT/GD) optimisers for logistic regression on the FashionMNIST dataset with fixed learning rate $\alpha = 0.01$ across different damping values, δ. LOPT[η] denotes a modification to the LOPT algorithm that perturbs a subset of update directions by a factor of η. Best viewed in colour.

Bibliography

[18] *DCGAN Faces Tutorial*. https://github.com/pytorch/tutorials/blob/master/beginner_source/dcgan_faces_tutorial.py. Accessed: 2020-09-30. 2018.

[AA12] Sherif M Abuelenin and Adel Y Abul-Magd. “Effect of unfolding on the spectral statistics of adjacency matrices of complex networks”. In: *Procedia Computer Science* 12 (2012), pp. 69–74.

[AAC13] Antonio Auffinger, Gérard Ben Arous, and Jiri Cerni. “Random matrices and complexity of spin glasses”. In: *Communications on Pure and Applied Mathematics* 66.2 (2013), pp. 165–201.

[AAR99] George E. Andrews, Richard Askey, and Ranjan Roy. *Special Functions*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1999. DOI: 10.1017/CBO9781107325937.

[Aba+16] Martín Abadi et al. “Tensorflow: A system for large-scale machine learning”. In: *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. 2016, pp. 265–283.

[ABM21a] Gerard Ben Arous, Paul Bourgade, and Benjamin McKenna. “Exponential growth of random determinants beyond invariance”. In: *arXiv preprint arXiv:2105.05000* (2021).

[ABM21b] Gérard Ben Arous, Paul Bourgade, and Benjamin McKenna. “Landscape complexity beyond invariance and the elastic manifold”. In: *arXiv preprint arXiv:2105.05051* (2021).

[ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. “Wasserstein generative adversarial networks”. In: *International conference on machine learning*. PMLR. 2017, pp. 214–223.

[ADG01] G Ben Arous, Amir Dembo, and Alice Guionnet. “Aging of spherical spin glasses”. In: *Probability theory and related fields* 120.1 (2001), pp. 1–67.

[ADŽ14] Karim M Abadir, Walter Distaso, and Filip Žikeš. “Design-free estimation of variance matrices”. In: *Journal of Econometrics* 181.2 (2014), pp. 165–180.

[AG20] Antonio Auffinger and Julian Gold. “The number of saddles of the spherical p-spin model”. In: *arXiv preprint arXiv:2007.09269v1.q* (2020).

[AG97] G Ben Arous and Alice Guionnet. "Large deviations for Wigner's law and Voiculescu's non-commutative entropy". In: *Probability theory and related fields* 108.4 (1997), pp. 517–542.

[AGZ10] Greg W Anderson, Alice Guionnet, and Ofer Zeitouni. *An introduction to random matrices*. Cambridge stidies in advanced mathematics 118. Cambridge university press, 2010.

[ALP22] Ben Adlam, Jake A Levinson, and Jeffrey Pennington. "A Random Matrix Perspective on Mixtures of Nonlinearities in High Dimensions". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2022, pp. 3434–3457.

[And62] Theodore Wilbur Anderson. *An introduction to multivariate statistical analysis*. Wiley New York, 1962.

[Ann+03] Alessia Annibale et al. "Supersymmetric complexity in the Sherrington-Kirkpatrick model". In: *Physical Review E* 68.6 (2003), p. 061103.

[AP20a] Ben Adlam and Jeffrey Pennington. "The neural tangent kernel in high dimensions: Triple descent and a multi-scale theory of generalization". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 74–84.

[AP20b] Ben Adlam and Jeffrey Pennington. "Understanding double descent requires a fine-grained bias-variance decomposition". In: *Advances in neural information processing systems* 33 (2020), pp. 11022–11032.

[Aro+19] Gerard Ben Arous et al. "The landscape of the spiked tensor model". In: *Communications on Pure and Applied Mathematics* 72.11 (2019), pp. 2282–2330.

[ASR88] Milton Abramowitz, Irene A Stegun, and Robert H Romer. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. 1988.

[ASZ20] Gérard Ben Arous, Eliran Subag, and Ofer Zeitouni. "Geometry and temperature chaos in mixed spherical spin glasses at low temperature: the perturbative regime". In: *Comm. Pure Appl. Math.* 73.8 (2020), pp. 1732–1828.

[AT+07] Robert J Adler, Jonathan E Taylor, et al. *Random fields and geometry*. Vol. 80. Springer, 2007.

[AT09] Robert J Adler and Jonathan E Taylor. *Random fields and geometry*. Springer Science & Business Media, 2009.

[Ata+13] YY Atas et al. "Distribution of the ratio of consecutive level spacings in random matrix ensembles". In: *Physical review letters* 110.8 (2013), p. 084101.

[Bad+17] Abdul Malik Badshah et al. "Speech emotion recognition from spectrograms with deep convolutional neural network". In: *2017 international conference on platform technology and service (PlatCon)*. IEEE. 2017, pp. 1–5.

[Bai+19] Marco Baity-Jesi et al. "Comparing dynamics: Deep neural networks versus glassy systems". In: *Journal of Statistical Mechanics: Theory and Experiment* 2019.12 (2019), p. 124013.

[Bal+21] Carlo Baldassi et al. "Unveiling the structure of wide flat minima in neural networks". In: *arXiv preprint arXiv:2107.01163* (2021).

[BAP05] Jinho Baik, Gérard Ben Arous, and Sandrine Péché. "Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices". In: *The Annals of Probability* 33.5 (2005), pp. 1643–1697.

[Bar93] Andrew R Barron. "Universal approximation bounds for superpositions of a sigmoidal function". In: *IEEE Transactions on Information theory* 39.3 (1993), pp. 930–945.

[Bas+21] Nicholas P Baskerville et al. "The loss surfaces of neural networks with general activation functions". In: *Journal of Statistical Mechanics: Theory and Experiment* 2021.6 (2021), p. 064001.

[Bas+22a] Nicholas P Baskerville et al. "A Spin Glass Model for the Loss Surfaces of Generative Adversarial Networks". In: *Journal of Statistical Physics* 186.2 (2022), pp. 1–45.

[Bas+22b] Nicholas P Baskerville et al. "Universal characteristics of deep neural network loss surfaces from random matrix theory". In: *Journal of Physics A: Mathematical and Theoretical* 55.49 (Dec. 2022), p. 494002. DOI: 10.1088/1751-8121/aca7f5. URL: https://dx.doi.org/10.1088/1751-8121/aca7f5.

[BBP16] Joël Bun, Jean-Philippe Bouchaud, and Marc Potters. *My beautiful laundrette: Cleaning correlation matrices for portfolio optimization*. 2016.

[BBP17] Joël Bun, Jean-Philippe Bouchaud, and Marc Potters. "Cleaning large correlation matrices: tools from random matrix theory". In: *Physics Reports* 666 (2017), pp. 1–109.

[Bee97] Carlo WJ Beenakker. "Random-matrix theory of quantum transport". In: *Reviews of modern physics* 69.3 (1997), p. 731.

[Bel+17] Serban T Belinschi et al. "Outliers in the spectrum of large deformed unitarily invariant models". In: *The Annals of Probability* 45.6A (2017), pp. 3571–3625.

[Bel+19] Mikhail Belkin et al. "Reconciling modern machine-learning practice and the classical bias–variance trade-off". In: *Proceedings of the National Academy of Sciences* 116.32 (2019), pp. 15849–15854.

[Bel21] Mikhail Belkin. "Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation". In: *Acta Numerica* 30 (2021), pp. 203–248.

[Ben20] Lucas Benigni. "Eigenvectors distribution and quantum unique ergodicity for deformed Wigner matrices". In: *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*. Vol. 56. 4. Institut Henri Poincaré. 2020, pp. 2822–2867.

[Ber+15] Daniel Berjón et al. “Optimal piecewise linear function approximation for GPU-based applications”. In: *IEEE transactions on cybernetics* 46.11 (2015), pp. 2584–2595.

[Ber+87] Michael V Berry et al. “Quantum chaology”. In: *Proc. Roy. Soc. London A* 413 (1987), pp. 183–198.

[Ber02] Michael V Berry. “Statistics of nodal lines and points in chaotic quantum billiards: perimeter corrections, fluctuations, curvature”. In: *Journal of Physics A: Mathematical and General* 35.13 (2002), p. 3025.

[BES20] Zhigang Bao, László Erdős, and Kevin Schnelli. “On the support of the free additive convolution”. In: *Journal d'Analyse Mathématique* 142.1 (2020), pp. 323–348.

[BGK22] Nicholas P Baskerville, Diego Granziol, and Jonathan P Keating. “Appearance of Random Matrix Theory in deep learning”. In: *Physica A: Statistical Mechanics and its Applications* 590 (2022), p. 126742.

[BGM12] Florent Benaych-Georges, Alice Guionnet, and Mylène Maïda. “Large deviations of the extreme eigenvalues of random deformations of matrices”. In: *Probability Theory and Related Fields* 154.3-4 (2012), pp. 703–751.

[BHM18] Mikhail Belkin, Daniel J Hsu, and Partha Mitra. “Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate”. In: *Advances in neural information processing systems* 31 (2018).

[BHY19] Roland Bauerschmidt, Jiaoyang Huang, and Horng-Tzer Yau. “Local Kesten–McKay law for random regular graphs”. In: *Communications in Mathematical Physics* 369.2 (2019), pp. 523–636.

[Bia97] Philippe Biane. “On the free convolution with a semi-circular distribution”. In: *Indiana University Mathematics Journal* (1997), pp. 705–718.

[BL21] Lucas Benigni and Patrick Lopatto. “Fluctuations in local quantum unique ergodicity for generalized Wigner matrices”. In: *arXiv preprint arXiv:2103.12013* (2021).

[BL22] Lucas Benigni and Patrick Lopatto. “Optimal delocalization for generalized Wigner matrices”. In: *Advances in Mathematics* 396 (2022), p. 108109.

[Blo+14] Alex Bloemendal et al. “Isotropic local laws for sample covariance and generalized Wigner matrices”. In: *Electronic Journal of Probability* 19 (2014), pp. 1–53.

[Blo+16] Alex Bloemendal et al. “On the principal components of sample covariance matrices”. In: *Probability theory and related fields* 164.1-2 (2016), pp. 459–552.

[BLO98] Corinne Berzin, José R León, and Joaquín Ortega. “Level crossings and local time for regularized Gaussian processes”. In: *Probability and Mathematical Statistics* 18.1 (1998), pp. 39–81.

[BM80] Alan J Bray and Michael A Moore. “Metastable states in spin glasses”. In: *Journal of Physics C: Solid State Physics* 13.19 (1980), p. L469.

[BM81] AJ Bray and MA Moore. “Metastable states in the solvable spin glass model”. In: *Journal of Physics A: Mathematical and General* 14.9 (1981), p. L377.

[BN11] Florent Benaych-Georges and Raj Rao Nadakuditi. “The eigenvalues and eigenvectors of finite, low rank perturbations of large random matrices”. In: *Advances in Mathematics* 227.1 (2011), pp. 494–521.

[Boh91] Oriol Bohigas. *Random matrix theories and chaotic dynamics*. Tech. rep. Paris-11 Univ., 1991.

[Boj+17] Piotr Bojanowski et al. “Enriching word vectors with subword information”. In: *Transactions of the association for computational linguistics* 5 (2017), pp. 135–146.

[Bot12] Léon Bottou. “Stochastic Gradient Descent Tricks”. In: *Neural Networks: Tricks of the Trade: Second Edition*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 421–436. ISBN: 978-3-642-35289-8.

[BP19] Lucas Benigni and Sandrine Péché. “Eigenvalue distribution of nonlinear models of random matrices”. In: *arXiv preprint arXiv:1904.03090* (2019).

[BPZ20] Carlo Baldassi, Fabrizio Pittorino, and Riccardo Zecchina. “Shaping the learning landscape in neural networks around wide flat minima”. In: *Proceedings of the National Academy of Sciences* 117.1 (2020), pp. 161–170.

[BR84] Michael V Berry and Marko Robnik. “Semiclassical level spacings when regular and chaotic orbits coexist”. In: *Journal of Physics A: Mathematical and General* 17.12 (1984), p. 2413.

[Bro+20] Tom Brown et al. “Language models are few-shot learners”. In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[BRT19] Mikhail Belkin, Alexander Rakhlin, and Alexandre B Tsybakov. “Does data interpolation contradict statistical optimality?” In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 1611–1619.

[BS04] Jinho Baik and Jack W Silverstein. “Eigenvalues of large sample covariance matrices of spiked population models”. In: *arXiv preprint math/0408165* (2004).

[BT77] Michael Victor Berry and Michael Tabor. “Level clustering in the regular spectrum”. In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 356.1686 (1977), pp. 375–394.

[Bun+16] Joël Bun et al. “Rotational Invariant Estimator for General Noisy Matrices.” In: *IEEE Trans. Information Theory* 62.12 (2016), pp. 7475–7490.

[BY17] Paul Bourgade and H-T Yau. "The eigenvector moment flow and local quantum unique ergodicity". In: *Communications in Mathematical Physics* 350.1 (2017), pp. 231–278.

[Cai+19] Tianle Cai et al. "Gram-gauss-newton method: Learning overparameterized neural networks for regression problems". In: *arXiv preprint arXiv:1905.11675* (2019).

[CD16] Mireille Capitaine and Catherine Donati-Martin. "Spectrum of deformed random matrices and free probability". In: *arXiv preprint arXiv:1607.05560* (2016).

[CFV16] Fabio Deelan Cunden, Paolo Facchi, and Pierpaolo Vivo. "A shortcut through the Coulomb gas method for spectral linear statistics on random matrices". In: *Journal of Physics A: Mathematical and Theoretical* 49.13 (2016), p. 135202.

[CG18] Jinghui Chen and Quanquan Gu. "Closing the generalization gap of adaptive gradient methods in training deep neural networks". In: *arXiv preprint arXiv:1806.06763* (2018).

[CGG99] Andrea Cavagna, Juan P Garrahan, and Irene Giardina. "Quenched complexity of the mean-field p-spin spherical model with external magnetic field". In: *Journal of Physics A: Mathematical and General* 32.5 (1999), p. 711.

[Cha+19] Pratik Chaudhari et al. "Entropy-SGD: biasing gradient descent into wide valleys". In: *Journal of Statistical Mechanics: Theory and Experiment* 2019.12 (Dec. 2019), p. 124018. DOI: 10.1088/1742-5468/ab39d9. URL: https://doi.org/10.1088/1742-5468/ab39d9.

[Cho+15] Anna Choromanska et al. "The loss surfaces of multilayer networks". In: *Artificial intelligence and statistics*. PMLR. 2015, pp. 192–204.

[Chu+14] Junyoung Chung et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling". In: *arXiv preprint arXiv:1412.3555* (2014).

[CLA15] Anna Choromanska, Yann LeCun, and Gérard Ben Arous. "Open problem: The landscape of the loss surfaces of multilayer networks". In: *Conference on Learning Theory*. 2015, pp. 1756–1760.

[cod20] Papers with code. *State-of-the-art*. 2020. URL: https://paperswithcode.com/sota (visited on 03/24/2020).

[Con+17] Alexis Conneau et al. "Very Deep Convolutional Networks for Text Classification". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 1107–1116. URL: https://www.aclweb.org/anthology/E17-1104.

[Cou+19] Romain Couillet et al. "Random matrix-improved estimation of covariance matrix distances". In: *Journal of Multivariate Analysis* 174 (2019), p. 104531.

[Cri+03] Andrea Crisanti et al. "Complexity in the Sherrington-Kirkpatrick model in the annealed approximation". In: *Physical Review B* 68.17 (2003), p. 174401.

[CS95] Andrea Crisanti and H-J Sommers. “Thouless-Anderson-Palmer approach to the spherical p-spin spin glass model”. In: *Journal de Physique I* 5.7 (1995), pp. 805–813.

[Cub+19] Ekin D Cubuk et al. “RandAugment: Practical data augmentation with no separate search”. In: *arXiv preprint arXiv:1909.13719* (2019).

[Cyb89] George Cybenko. “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314.

[Dau+14] Yann N Dauphin et al. “Identifying and attacking the saddle point problem in high-dimensional non-convex optimization”. In: *Advances in neural information processing systems*. 2014, pp. 2933–2941.

[Dau+22] Ingrid Daubechies et al. “Nonlinear Approximation and (Deep) ReLU Networks”. In: *Constructive Approximation* 55.1 (2022), pp. 127–172.

[DB19] Aaron Defazio and Léon Bottou. “On the ineffectiveness of variance reduced optimization for deep learning”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 1753–1763.

[Dei99] Percy Deift. *Orthogonal polynomials and random matrices: a Riemann-Hilbert approach*. Vol. 3. American Mathematical Soc., 1999.

[Den12] Li Deng. “The MNIST database of handwritten digit images for machine learning research [best of the web]”. In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 141–142.

[Dev+18] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).

[Dev+19] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: `10.18653/v1/N19-1423`. URL: `https://www.aclweb.org/anthology/N19-1423`.

[DG14] Cicero Dos Santos and Maira Gatti. “Deep convolutional neural networks for sentiment analysis of short texts”. In: *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers*. 2014, pp. 69–78.

[DH02] E. Delabaere and C.J. Howls. “Global asymptotics for multiple integrals with boundaries”. In: *Duke Mathematical Journal* 112.2 (2002), pp. 199–264. URL: `https://eprints.soton.ac.uk/29213/`.

[DHS11] John Duchi, Elad Hazan, and Yoram Singer. “Adaptive subgradient methods for online learning and stochastic optimization”. In: *Journal of machine learning research* 12.Jul (2011), pp. 2121–2159.

[Din+17a] Laurent Dinh et al. “Sharp Minima Can Generalize For Deep Nets”. In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, Aug. 2017, pp. 1019–1028. URL: `https://proceedings.mlr.press/v70/dinh17b.html`.

[Din+17b] Laurent Dinh et al. “Sharp minima can generalize for deep nets”. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 1019–1028.

[Duc18] John C Duchi. “Introductory lectures on stochastic optimization”. In: *The Mathematics of Data* 25 (2018), p. 99.

[Dys62a] Freeman J Dyson. “A Brownian-motion model for the eigenvalues of a random matrix”. In: *Journal of Mathematical Physics* 3.6 (1962), pp. 1191–1198.

[Dys62b] Freeman J Dyson. “Statistical theory of the energy levels of complex systems. I”. In: *Journal of Mathematical Physics* 3.1 (1962), pp. 140–156.

[Dys70] Freeman J Dyson. “Correlations between eigenvalues of a random matrix”. In: *Communications in Mathematical Physics* 19.3 (1970), pp. 235–250.

[E+20] Weinan E et al. *Towards a Mathematical Understanding of Neural Network-Based Machine Learning: what we know and what we don't*. 2020. DOI: `10.48550/ARXIV.2009.10713`. URL: `https://arxiv.org/abs/2009.10713`.

[Efe96] Konstantin Efetov. “Supermathematics”. In: *Supersymmetry in Disorder and Chaos*. Cambridge University Press, 1996, pp. 8–28. DOI: `10.1017/CBO9780511573057.003`.

[Efe99] Konstantin Efetov. *Supersymmetry in disorder and chaos*. Cambridge university press, 1999.

[EKS19] László Erdős, Torben Krüger, and Dominik Schröder. “Random matrices with slow correlation decay”. In: *Forum of Mathematics, Sigma*. Vol. 7. Cambridge University Press. 2019.

[El +07] Noureddine El Karoui et al. “Tracy–Widom limit for the largest eigenvalue of a large class of complex sample covariance matrices”. In: *The Annals of Probability* 35.2 (2007), pp. 663–714.

[ER00] Richard Everson and Stephen Roberts. “Inferring the eigenvalues of covariance matrices from limited, noisy data”. In: *IEEE transactions on signal processing* 48.7 (2000), pp. 2083–2091.

[ES17] László Erdős and Kevin Schnelli. “Universality for random matrix flows with time-dependent density”. In: *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*. Vol. 53. 4. Institut Henri Poincaré. 2017, pp. 1606–1656.

[EV01] Andreas Engel and Christian Van den Broeck. *Statistical mechanics of learning*. Cambridge University Press, 2001.

[EY12] László Erdős and Horng-Tzer Yau. “Universality of local spectral statistics of random matrices”. In: *Bulletin of the American Mathematical Society* 49.3 (2012), pp. 377–414.

[EY17a] László Erdos and Horng-Tzer Yau. “A dynamical approach to random matrix theory”. In: *Courant Lecture Notes in Mathematics* 28 (2017).

[EY17b] László Erdős and Horng-Tzer Yau. *A dynamical approach to random matrix theory*. Vol. 28. American Mathematical Soc., 2017.

[EYY12] László Erdős, Horng-Tzer Yau, and Jun Yin. “Bulk universality for generalized Wigner matrices”. In: *Probability Theory and Related Fields* 154.1-2 (2012), pp. 341–407.

[FFR19] Giampaolo Folena, Silvio Franz, and Federico Ricci-Tersenghi. “Rethinking mean-field glassy dynamics and its relation with the energy landscape: the awkward case of the spherical mixed p-spin model”. In: *arXiv preprint arXiv:1903.01421* (2019).

[FN15] Yan V Fyodorov and André Nock. “On random matrix averages involving half-integer powers of GOE characteristic polynomials”. In: *Journal of Statistical Physics* 159.4 (2015), pp. 731–751.

[For10] Peter J Forrester. *Log-gases and random matrices (LMS-34)*. Princeton University Press, 2010.

[FS02] Yan V Fyodorov and Eugene Strahov. “Characteristic polynomials of random Hermitian matrices and Duistermaat–Heckman localisation on non-compact Kähler manifolds”. In: *Nuclear Physics B* 630.3 (2002), pp. 453–491.

[FW07] Yan V Fyodorov and Ian Williams. “Replica symmetry breaking condition exposed by random matrix calculation of landscape complexity”. In: *Journal of Statistical Physics* 129.5-6 (2007), pp. 1081–1116.

[Fyo04] Yan V Fyodorov. “Complexity of random energy landscapes, glass transition, and absolute value of the spectral determinant of random matrices”. In: *Physical review letters* 92.24 (2004), p. 240601.

[Fyo05] Yan V Fyodorov. “Counting stationary points of random landscapes as a random matrix problem”. In: *Acta Physica Polonica B* 36 (2005), pp. 2699–2707.

[Fyo19] Yan V Fyodorov. “A spin glass model for reconstructing nonlinearly encrypted signals corrupted by noise”. In: *Journal of Statistical Physics* 175 (2019), pp. 789–818.

[Gar+18] Jacob Gardner et al. “Gpytorch: Blackbox matrix-matrix Gaussian Process inference with GPU acceleration”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 7576–7586.

[GB22] Diego Granziol and Nicholas Baskerville. “A random matrix theory approach to damping in deep learning”. In: *Journal of Physics: Complexity* 3.2 (2022), p. 024001.

[GD88] Elizabeth Gardner and Bernard Derrida. “Optimal storage properties of neural network models”. In: *Journal of Physics A: Mathematical and general* 21.1 (1988), p. 271.

[GD98] Matt W Gardner and SR Dorling. “Artificial neural networks (the multilayer perceptron)–a review of applications in the atmospheric sciences”. In: *Atmospheric environment* 32.14-15 (1998), pp. 2627–2636.

[GKX19] Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. “An Investigation into Neural Net Optimization via Hessian Eigenvalue Density”. In: *arXiv preprint arXiv:1901.10159* (2019).

[GM+05] Alice Guionnet, M Maı, et al. “A Fourier view on the R-transform and related asymptotics of spherical integrals”. In: *Journal of functional analysis* 222.2 (2005), pp. 435–490.

[GMW98] Thomas Guhr, Axel Müller–Groeling, and Hans A Weidenmüller. “Random-matrix theories in quantum physics: common concepts”. In: *Physics Reports* 299.4-6 (1998), pp. 189–425.

[Gol+20] Sebastian Goldt et al. “The Gaussian equivalence of generative models for learning with shallow neural networks”. In: *arXiv preprint arXiv:2006.14709* (2020).

[Goo+14a] Ian Goodfellow et al. “Generative Adversarial Nets”. In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani et al. Curran Associates, Inc., 2014, pp. 2672–2680. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[Goo+14b] Ian Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[Goo+16] Ian Goodfellow et al. *Deep learning*. Vol. 1. MIT press Cambridge, 2016.

[Gra+19a] Diego Granziol et al. “MLRG Deep Curvature”. In: *arXiv preprint arXiv:1912.09656* (2019).

[Gra+19b] Diego Granziol et al. “Towards understanding the true loss surface of deep neural networks using random matrix theory and iterative spectral methods”. In: (2019).

[Gra+20] Diego Granziol et al. *Towards understanding the true loss surface of deep neural networks using random matrix theory and iterative spectral methods*. https://openreview.net/forum?id=H1gza2NtwH. 2020.

[Gra+21] Diego Granziol et al. *Iterative Averaging in the Quest for Best Test Error*. 2021. URL: https://arxiv.org/abs/2003.01247.

[Gra20a] Diego Granziol. “Beyond random matrix theory for deep networks”. In: *arXiv preprint arXiv:2006.07721* (2020).

[Gra20b] Diego Granziol. “Flatness is a False Friend”. In: *arXiv preprint arXiv:2006.09091* (2020).

[GSW12] Michael B. Green, John H. Schwarz, and Edward Witten. *Superstring Theory: 25th Anniversary Edition*. Vol. 1. Cambridge Monographs on Mathematical Physics. Cambridge University Press, 2012. DOI: `10.1017/CBO9781139248563`.

[Guh91] Thomas Guhr. “Dyson’s correlation functions and graded symmetry”. In: *Journal of mathematical physics* 32.2 (1991), pp. 336–347.

[Guo+17] Chuan Guo et al. “On Calibration of Modern Neural Networks”. In: *CoRR* abs/1706.04599 (2017). arXiv: `1706.04599`. URL: `http://arxiv.org/abs/1706.04599`.

[GW90] T Guhr and HA Weidenmüller. “Isospin mixing and spectral fluctuation properties”. In: *Annals of Physics* 199.2 (1990), pp. 412–446.

[GWG19] Diego Granziol, Xingchen Wan, and Timur Garipov. “Deep Curvature Suite”. In: *arXiv preprint arXiv:1912.09656* (2019).

[GWR20] Diego Granziol, Xingchen Wan, and Stephen Roberts. “Iterate Averaging Helps: An Alternative Perspective in Deep Learning”. In: *arXiv preprint arXiv:2003.01247* (2020).

[GZ+00] Alice Guionnet, Ofer Zeitouni, et al. “Concentration of the spectral measure for large matrices”. In: *Electronic Communications in Probability* 5 (2000), pp. 119–136.

[GZR20] Diego Granziol, Stefan Zohren, and Stephen Roberts. “Learning Rates as a Function of Batch Size: A Random Matrix Theory Approach to Neural Network Training”. In: *arXiv preprint arXiv:2006.09092* (2020).

[GZR22] Diego Granziol, Stefan Zohren, and Stephen Roberts. “Learning Rates as a Function of Batch Size: A Random Matrix Theory Approach to Neural Network Training”. In: *Journal of Machine Learning Research* 23.173 (2022), pp. 1–65. URL: `http://jmlr.org/papers/v23/20-1258.html`.

[Har+19] Nicholas JA Harvey et al. “Tight analyses for non-smooth stochastic gradient descent”. In: *Conference on Learning Theory*. PMLR. 2019, pp. 1579–1613.

[Has+09] Trevor Hastie et al. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, 2009.

[He+15] Kaiming He et al. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1026–1034.

[He+16] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[HHS17] Elad Hoffer, Itay Hubara, and Daniel Soudry. "Train longer, generalize better: closing the generalization gap in large batch training of neural networks". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: `https://proceedings.neurips.cc/paper/2017/file/a5e0ff62be0b08456fc7f1e88812af3d-Paper.pdf`.

[HHY19] Haowei He, Gao Huang, and Yang Yuan. "Asymmetric Valleys: Beyond Sharp and Flat Local Minima". In: *arXiv preprint arXiv:1902.00744* (2019).

[HS97a] Sepp Hochreiter and Jürgen Schmidhuber. "Flat Minima". In: *Neural Computation* 9.1 (Jan. 1997), pp. 1–42. ISSN: 0899-7667. DOI: `10.1162/neco.1997.9.1.1`. eprint: `https://direct.mit.edu/neco/article-pdf/9/1/1/813385/neco.1997.9.1.1.pdf`. URL: `https://doi.org/10.1162/neco.1997.9.1.1`.

[HS97b] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[HSS08] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. "Exploring Network Structure, Dynamics, and Function using NetworkX". In: *Proceedings of the 7th Python in Science Conference*. Ed. by Gaël Varoquaux, Travis Vaught, and Jarrod Millman. Pasadena, CA USA, 2008, pp. 11–15.

[HSW89] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators". In: *Neural networks* 2.5 (1989), pp. 359–366.

[Inc20] Google Inc. *Machine Learning Glossary*. 2020. URL: `https://developers.google.com/machine-learning/glossary%5C#o` (visited on 02/13/2020).

[IS15] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *arXiv preprint arXiv:1502.03167* (2015).

[Izm+18] Pavel Izmailov et al. "Averaging weights leads to wider optima and better generalization". In: *arXiv preprint arXiv:1803.05407* (2018).

[Jas+20] Stanislaw Jastrzebski et al. "The Break-Even Point on the Optimization Trajectories of Deep Neural Networks". In: *International Conference on Learning Representations*. 2020. URL: `https://openreview.net/forum?id=r1g87C4KwB`.

[JC17] Katarzyna Janocha and Wojciech Marian Czarnecki. "On Loss Functions for Deep Neural Networks in Classification". In: *CoRR* abs/1702.05659 (2017). arXiv: `1702.05659`. URL: `http://arxiv.org/abs/1702.05659`.

[JGH18] Arthur Jacot, Franck Gabriel, and Clément Hongler. "Neural tangent kernel: Convergence and generalization in neural networks". In: *Advances in neural information processing systems* 31 (2018).

[JMM96] Anil K Jain, Jianchang Mao, and K Moidin Mohiuddin. “Artificial neural networks: A tutorial”. In: *Computer* 29.3 (1996), pp. 31–44.

[Kac43] Mark Kac. “On the average number of real roots of a random algebraic equation”. In: *Bulletin of the American Mathematical Society* 49.4 (1943), pp. 314–320.

[KB14] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[Kes+17] Nitish Shirish Keskar et al. “On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima”. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL: `https://openreview.net/forum?id=H1oyRlYgg`.

[KH+09] Alex Krizhevsky, Geoffrey Hinton, et al. “Learning multiple layers of features from tiny images”. In: (2009).

[KKB20] Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. *Generalization in Deep Learning*. 2020. arXiv: `1710.05468 [stat.ML]`.

[KLA19] Tero Karras, Samuli Laine, and Timo Aila. “A style-based generator architecture for generative adversarial networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4401–4410.

[KLA20] Tero Karras, Samuli Laine, and Timo Aila. “A Style-Based Generator Architecture for Generative Adversarial Networks.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), pp. 1–1.

[KLY18] Bobby Kleinberg, Yuanzhi Li, and Yang Yuan. “An alternative view: When does SGD escape local minima?” In: *International Conference on Machine Learning*. PMLR. 2018, pp. 2698–2707.

[KPV93] Jorge Kurchan, Giorgio Parisi, and Miguel Angel Virasoro. “Barriers and metastable states as saddle points in the replica approach”. In: *Journal de Physique I* 3.8 (1993), pp. 1819–1838.

[KS17] Nitish Shirish Keskar and Richard Socher. “Improving generalization performance by switching from Adam to SGD”. In: *arXiv preprint arXiv:1712.07628* (2017).

[KS87] I Kanter and Haim Sompolinsky. “Associative recall of memory without errors”. In: *Physical Review A* 35.1 (1987), p. 380.

[KV03] Jeong Han Kim and Van H Vu. “Generating random regular graphs”. In: *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*. 2003, pp. 213–222.

[KY03] Harold Kushner and G George Yin. *Stochastic approximation and recursive algorithms and applications*. Vol. 35. Springer Science & Business Media, 2003.

[KY17] Antti Knowles and Jun Yin. “Anisotropic local laws for random matrices”. In: *Probability Theory and Related Fields* 169.1 (2017), pp. 257–352.

[Lan50] Cornelius Lanczos. “An iteration method for the solution of the eigenvalue problem of linear differential and integral operators”. In: (1950).

[LB+95] Yann LeCun, Yoshua Bengio, et al. “Convolutional networks for images, speech, and time series”. In: *The handbook of brain theory and neural networks* 3361.10 (1995), p. 1995.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), pp. 436–444.

[LC10] Yann LeCun and Corinna Cortes. “MNIST handwritten digit database”. In: (2010). URL: `http://yann.lecun.com/exdb/mnist/`.

[LD02] Meng Heng Loke and Torleif Dahlin. “A comparison of the Gauss–Newton and quasi-Newton methods in resistivity imaging inversion”. In: *Journal of applied geophysics* 49.3 (2002), pp. 149–162.

[LeC+89] Yann LeCun et al. “Backpropagation applied to handwritten zip code recognition”. In: *Neural computation* 1.4 (1989), pp. 541–551.

[LeC+98] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[LeC98] Yann LeCun. “The MNIST database of handwritten digits”. In: *http://yann. lecun. com/exdb/mnist/* (1998).

[LH18] Ilya Loshchilov and Frank Hutter. “Decoupled weight decay regularization”. In: (2018).

[Li+18] Hao Li et al. “Visualizing the loss landscape of neural nets”. In: vol. 31. 2018.

[LL20] Yulong Lu and Jianfeng Lu. “A universal approximation theorem of deep neural networks for expressing probability distributions”. In: *Advances in neural information processing systems* 33 (2020), pp. 3094–3105.

[LLC+18] Cosme Louart, Zhenyu Liao, Romain Couillet, et al. “A random matrix approach to neural networks”. In: *The Annals of Applied Probability* 28.2 (2018), pp. 1190–1248.

[LNV18] Giacomo Livan, Marcel Novaes, and Pierpaolo Vivo. “Introduction to random matrices theory and practice”. In: *Monograph Award* (2018), p. 63.

[Loj63] Stanislaw Lojasiewicz. “A topological property of real analytic subsets”. In: *Coll. du CNRS, Les équations aux dérivées partielles* 117.87-89 (1963), p. 2.

[Lou+21] Bruno Loureiro et al. “Capturing the learning curves of generic features maps for realistic data sets with a teacher-student model”. In: *arXiv preprint arXiv:2102.08127* (2021).

[LP] Olivier Ledoit and Sandrine Péché. “Eigenvectors of some large sample covariance matrix ensembles”. In: *Probability Theory and Related Fields* 151.1 (), pp. 233–264.

[LSB12] Simon Lacoste-Julien, Mark Schmidt, and Francis Bach. "A simpler approach to obtaining an O (1/t) convergence rate for the projected stochastic subgradient method". In: *arXiv preprint arXiv:1212.2002* (2012).

[LT16] Ming-Yu Liu and Oncel Tuzel. "Coupled Generative Adversarial Networks". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Vol. 29. 2016, pp. 469–477.

[Lu+17] Zhou Lu et al. "The expressive power of neural networks: A view from the width". In: *Advances in neural information processing systems* 30 (2017).

[LW04] Olivier Ledoit and Michael Wolf. "A well-conditioned estimator for large-dimensional covariance matrices". In: *Journal of multivariate analysis* 88.2 (2004), pp. 365–411.

[LW12] Olivier Ledoit and Michael Wolf. "Nonlinear shrinkage estimation of large-dimensional covariance matrices". In: *The Annals of Statistics* 40.2 (2012), pp. 1024–1060.

[LW20] Olivier Ledoit and Michael Wolf. "Analytical nonlinear shrinkage of large-dimensional covariance matrices". In: *The Annals of Statistics* 48.5 (2020), pp. 3043–3065.

[Maj+11] Satya N Majumdar et al. "How many eigenvalues of a Gaussian random matrix are positive?" In: *Physical Review E* 83.4 (2011), p. 041105.

[Man+19a] Stefano Sarao Mannelli et al. "Passed & spurious: Descent algorithms and local minima in spiked matrix-tensor models". In: *arXiv preprint arXiv:1902.00139* (2019).

[Man+19b] Stefano Sarao Mannelli et al. "Who is Afraid of Big Bad Minima? Analysis of gradient-flow in spiked matrix-tensor models". In: *Advances in Neural Information Processing Systems*. 2019, pp. 8676–8686.

[Mar10] James Martens. "Deep learning via Hessian-free optimization." In: *ICML*. Vol. 27. 2010, pp. 735–742.

[Mar14] James Martens. "New insights and perspectives on the natural gradient method". In: *arXiv preprint arXiv:1412.1193* (2014).

[Mar16a] James Martens. *Second-order optimization for neural networks*. University of Toronto (Canada), 2016.

[Mar16b] James Martens. "Second-order optimization for neural networks". PhD thesis. University of Toronto, 2016. URL: `http://www.cs.toronto.edu/~jmartens/docs/thesis%5C_phd%5C_martens.pdf`.

[MBB20] Antoine Maillard, Gérard Ben Arous, and Giulio Biroli. "Landscape Complexity for the Empirical Risk of Generalized Linear Models". In: Proceedings of Machine Learning Research 107 (July 2020). Ed. by Jianfeng Lu and Rachel Ward, pp. 287–327. URL: `http://proceedings.mlr.press/v107/maillard20a.html`.

[McK21] Benjamin McKenna. “Complexity of bipartite spherical spin glasses”. In: *arXiv preprint arXiv:2105.05043* (2021).

[Mec19] Elizabeth S Meckes. *The random matrix theory of the classical compact groups*. Vol. 218. Cambridge University Press, 2019.

[Meh04] Madan Lal Mehta. *Random matrices*. Elsevier, 2004.

[Mez06] Francesco Mezzadri. “How to generate random matrices from the classical compact groups”. In: *arXiv preprint math-ph/0609050* (2006).

[MG15] James Martens and Roger Grosse. “Optimizing neural networks with Kronecker-factored approximate curvature”. In: *International conference on machine learning*. 2015, pp. 2408–2417.

[Mik+13] Tomas Mikolov et al. “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781* (2013).

[MJ01] Larry R Medsker and LC Jain. “Recurrent neural networks”. In: *Design and Applications* 5 (2001), pp. 64–67.

[MM18] Charles H Martin and Michael W Mahoney. “Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning”. In: *arXiv preprint arXiv:1810.01075* (2018).

[MO14] Mehdi Mirza and Simon Osindero. “Conditional Generative Adversarial Nets”. In: *arXiv preprint arXiv:1411.1784* (2014).

[MPV87] Marc Mézard, Giorgio Parisi, and Miguel Virasoro. *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*. Vol. 9. World Scientific Publishing Company, 1987.

[MS06] Gérard Meurant and Zdeněk Strakoš. “The Lanczos and conjugate gradient algorithms in finite precision arithmetic”. In: *Acta Numerica* 15 (2006), pp. 471–542.

[MS12] James Martens and Ilya Sutskever. “Training deep and recurrent networks with Hessian-free optimization”. In: *Neural networks: Tricks of the trade*. Springer, 2012, pp. 479–535.

[Mur91] Fionn Murtagh. “Multilayer perceptrons for classification and regression”. In: *Neurocomputing* 2.5-6 (1991), pp. 183–197.

[Nes13] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*. Vol. 87. Springer Science & Business Media, 2013.

[Ney+17a] Behnam Neyshabur et al. “Exploring generalization in deep learning”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 5947–5956.

[Ney+17b] Behnam Neyshabur et al. “Geometry of optimization and implicit regularization in deep learning”. In: *arXiv preprint arXiv:1705.03071* (2017).

[Ney17] Behnam Neyshabur. “Implicit regularization in deep learning”. In: *arXiv preprint arXiv:1709.01953* (2017).

[Nis01] Hidetoshi Nishimori. *Statistical physics of spin glasses and information processing: an introduction*. 111. Clarendon Press, 2001.

[Noc16] André Nock. “Characteristic Polynomials of Random Matrices and Quantum Chaotic Scattering”. PhD thesis. Queen Mary University of London, 2016.

[Noc17] Andre Nock. “Characteristic Polynomials of Random Matrices and Quantum Chaotic Scattering”. PhD thesis. Queen Mary University of London, 2017.

[NTS14] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. “In search of the real inductive bias: On the role of implicit regularization in deep learning”. In: *arXiv preprint arXiv:1412.6614* (2014).

[Pap18] Vardan Papyan. “The full spectrum of deepnet hessians at scale: Dynamics with sgd training and sample size”. In: *arXiv preprint arXiv:1811.07062* (2018).

[Pap19] Vardan Papyan. “Measurements of three-level hierarchical structure in the outliers in the spectrum of deepnet hessians”. In: *arXiv preprint arXiv:1901.08244* (2019).

[Pap20] Vardan Papyan. “Traces of class/cross-class structure pervade deep learning spectra”. In: *Journal of Machine Learning Research* 21.252 (2020), pp. 1–64.

[Pas+17] Adam Paszke et al. “Automatic differentiation in pytorch”. In: 2017.

[Pas20] Leonid Pastur. “On random matrices arising in deep neural networks. gaussian case”. In: *arXiv preprint arXiv:2001.06188* (2020).

[Pas22] L. Pastur. “Eigenvalue distribution of large random matrices arising in deep neural networks: Orthogonal case”. In: *Journal of Mathematical Physics* 63.6 (2022), p. 063505. DOI: `10.1063/5.0085204`. eprint: `https://doi.org/10.1063/5.0085204`. URL: `https://doi.org/10.1063/5.0085204`.

[PB17] Jeffrey Pennington and Yasaman Bahri. “Geometry of neural network loss surfaces via random matrix theory”. In: *International Conference on Machine Learning*. JMLR. org. 2017, pp. 2798–2806.

[PB20] Marc Potters and Jean-Philippe Bouchaud. *A First Course in Random Matrix Theory: For Physicists, Engineers and Data Scientists*. Cambridge University Press, 2020.

[Pea94] Barak A Pearlmutter. “Fast exact multiplication by the Hessian”. In: *Neural computation* 6.1 (1994), pp. 147–160.

[Pes18] Michael E Peskin. *An introduction to quantum field theory*. CRC press, 2018.

[PHD20] Vardan Papyan, XY Han, and David L Donoho. “Prevalence of neural collapse during the terminal phase of deep learning training”. In: *Proceedings of the National Academy of Sciences* 117.40 (2020), pp. 24652–24663.

[PJ92] Boris T Polyak and Anatoli B Juditsky. “Acceleration of stochastic approximation by averaging”. In: *SIAM journal on control and optimization* 30.4 (1992), pp. 838–855.

[Pol64] Boris T Polyak. “Gradient methods for solving equations and inequalities”. In: *USSR Computational Mathematics and Mathematical Physics* 4.6 (1964), pp. 17–32.

[PS20] L Pastur and V Slavin. “On random matrices arising in deep neural networks: General iid case”. In: *arXiv preprint arXiv:2011.11439* (2020).

[PS22] Leonid Pastur and Victor Slavin. “On random matrices arising in deep neural networks: General I.I.D. case”. In: *Random Matrices: Theory and Applications* 0.0 (2022), p. 2250046. DOI: 10.1142/S2010326322500460. eprint: https://doi.org/10.1142/S2010326322500460. URL: https://doi.org/10.1142/S2010326322500460.

[PSG18] Jeffrey Pennington, Samuel Schoenholz, and Surya Ganguli. “The emergence of spectral universality in deep networks”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2018, pp. 1924–1932.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher D Manning. “Glove: Global vectors for word representation”. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[PV18] Philipp Petersen and Felix Voigtlaender. “Optimal approximation of piecewise smooth functions using deep ReLU neural networks”. In: *Neural Networks* 108 (2018), pp. 296–330.

[PW17] Jeffrey Pennington and Pratik Worah. “Nonlinear random matrix theory for deep learning”. In: vol. 30. 2017.

[PyT21] PyTorch. *RESNET*. 2021. URL: https://pytorch.org/hub/pytorch%5C_vision%5C_resnet/ (visited on 05/03/2021).

[Rad+18] Alec Radford et al. *Improving language understanding by generative pre-training*. 2018.

[Ric44] Stephen O Rice. “Mathematical analysis of random noise”. In: *The Bell System Technical Journal* 23.3 (1944), pp. 282–332.

[RKK19] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. “On the convergence of Adam and beyond”. In: *arXiv preprint arXiv:1904.09237* (2019).

[RMC15] Alec Radford, Luke Metz, and Soumith Chintala. “Unsupervised representation learning with deep convolutional generative adversarial networks”. In: *arXiv preprint arXiv:1511.06434* (2015).

[Ros+19] Valentina Ros et al. “Complex energy landscapes in spiked-tensor and simple glassy models: Ruggedness, arrangements of local minima, and phase transitions”. In: *Physical Review X* 9.1 (2019), p. 011003.

[RSS11] Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. “Making gradient descent optimal for strongly convex stochastic optimization”. In: *arXiv preprint arXiv:1109.5647* (2011).

[RYH21] Daniel A Roberts, Sho Yaida, and Boris Hanin. “The Principles of Deep Learning Theory”. In: *arXiv preprint arXiv:2106.10165* (2021).

[RYH22] Daniel A. Roberts, Sho Yaida, and Boris Hanin. *The Principles of Deep Learning Theory*. `https://deeplearningtheory.com`. Cambridge University Press, 2022. arXiv: `2106.10165 [cs.LG]`.

[Sag+14] Levent Sagun et al. “Explorations on high dimensional landscapes”. In: *arXiv preprint arXiv:1412.6615* (2014).

[Sag+17] Levent Sagun et al. “Empirical Analysis of the Hessian of Over-Parametrized Neural Networks”. In: *arXiv preprint arXiv:1706.04454* (2017).

[SBL16] Levent Sagun, Léon Bottou, and Yann LeCun. “Eigenvalues of the Hessian in Deep Learning: Singularity and Beyond”. In: *arXiv preprint arXiv:1611.07476* (2016).

[Sch15] Torsten Scholak. *unfoldr*. Accessed 30/10/2020. 2015. URL: `https://github.com/tscholak/unfoldr`.

[Sha+14] Ali Sharif Razavian et al. “CNN features off-the-shelf: an astounding baseline for recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2014, pp. 806–813.

[Shc20] Tatyana Shcherbina. “Characteristic polynomials for random band matrices near the threshold”. In: *Journal of Statistical Physics* 179.4 (2020), pp. 920–944.

[SPS17] Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. “A correspondence between random neural networks and statistical field theory”. In: *arXiv preprint arXiv:1710.06570* (2017).

[SS17] Mariya Shcherbina and Tatyana Shcherbina. “Characteristic polynomials for 1D random band matrices from the localization side”. In: *Communications in Mathematical Physics* 351.3 (2017), pp. 1009–1044.

[Sub17] Eliran Subag. “The complexity of spherical p-spin models - a second moment approach.” In: *Ann. Probab.* 45.5 (2017), pp. 3385–3450.

[SV14] Satish Shirali and Harkrishan L Vasudeva. *An Introduction to Mathematical Analysis*. Alpha Science International, Limited, 2014.

[SW99] Angelika Steger and Nicholas C Wormald. “Generating random regular graphs quickly”. In: *Combinatorics, Probability and Computing* 8.4 (1999), pp. 377–396.

[SWB14] Torsten Scholak, Thomas Wellens, and Andreas Buchleitner. "Spectral backbone of excitation transport in ultracold Rydberg gases". In: *Physical Review A* 90.6 (2014), p. 063415.

[SZ13] Ohad Shamir and Tong Zhang. "Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes". In: *International conference on machine learning*. PMLR. 2013, pp. 71–79.

[SZ14] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[Tao12] Terence Tao. *Topics in random matrix theory*. Vol. 132. American Mathematical Soc., 2012.

[Tel15] Matus Telgarsky. "Representation benefits of deep feedforward networks". In: *arXiv preprint arXiv:1509.08101* (2015).

[Tor20] Magnus Tornstad. *Evaluating the Practicality of Using a Kronecker-Factored Approximate Curvature Matrix in Newton's Method for Optimization in Neural Networks*. 2020.

[TSR22] Matthias Thamm, Max Staats, and Bernd Rosenow. "Random matrix analysis of deep neural network weight matrices". In: *arXiv preprint arXiv:2203.14661* (2022).

[TW94] Craig A Tracy and Harold Widom. "Level-spacing distributions and the Airy kernel". In: *Communications in Mathematical Physics* 159.1 (1994), pp. 151–174.

[Unt19] Jeremie Unterberger. "Global fluctuations for 1D log-gas dynamics. Covariance kernel and support". In: *Electronic Journal of Probability* 24 (2019).

[VDN92] Dan V Voiculescu, Ken J Dykema, and Alexandru Nica. *Free random variables*. 1. American Mathematical Soc., 1992.

[Ver04] Jacobus Verbaarschot. "The supersymmetric method in random matrix theory and applications to QCD". In: *AIP Conference Proceedings* (2004). ISSN: 0094-243X. DOI: 10.1063/1.1853204. URL: http://dx.doi.org/10.1063/1.1853204.

[Ver18] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*. Vol. 47. Cambridge university press, 2018.

[VPF21] Aditya Vardhan Varre, Loucas Pillaud-Vivien, and Nicolas Flammarion. "Last iterate convergence of SGD for Least-Squares in the Interpolation regime." In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 21581–21591.

[Wan+19] Ke Wang et al. "Exact Gaussian processes on a million data points". In: *Advances in Neural Information Processing Systems* 32 (2019), pp. 14648–14659.

[WHS22] Alexander Wei, Wei Hu, and Jacob Steinhardt. "More Than a Toy: Random Matrix Models Predict How Real-World Neural Representations Generalize". In: *arXiv preprint arXiv:2203.06176* (2022).

[Wig67] Eugene P Wigner. “Random matrices in physics”. In: *SIAM review* 9.1 (1967), pp. 1–23.

[Wig93] Eugene P Wigner. “Characteristic vectors of bordered matrices with infinite dimensions I”. In: *The Collected Works of Eugene Paul Wigner*. Springer, 1993, pp. 524–540.

[Wil+17] Ashia C Wilson et al. “The marginal value of adaptive gradient methods in machine learning”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 4148–4158.

[WM08] HA Weidenmuller and GE Mitchell. “Random Matrices and Chaos in Nuclear Physics”. In: *arXiv preprint arXiv:0807.1070* (2008).

[Xie+19] Qizhe Xie et al. *Self-training with Noisy Student improves ImageNet classification*. 2019. arXiv: `1911.04252 [cs.LG]`.

[XRV17] Han Xiao, Kashif Rasul, and Roland Vollgraf. “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms”. In: *arXiv preprint arXiv:1708.07747* (2017).

[Yan+19] Greg Yang et al. “A mean field theory of batch normalization”. In: *arXiv preprint arXiv:1902.08129* (2019).

[Yun+19] Sangdoo Yun et al. “Cutmix: Regularization strategy to train strong classifiers with localizable features”. In: *arXiv preprint arXiv:1905.04899* (2019).

[Zha+16] Chiyuan Zhang et al. “Understanding deep learning requires rethinking generalization”. In: *arXiv preprint arXiv:1611.03530* (2016).

[Zha+18a] Guodong Zhang et al. “Three mechanisms of weight decay regularization”. In: *arXiv preprint arXiv:1810.12281* (2018).

[Zha+18b] Han Zhang et al. “Self-Attention Generative Adversarial Networks”. In: *International Conference on Machine Learning*. 2018, pp. 7354–7363.

[Zha+19] Michael Zhang et al. “Lookahead Optimizer: k steps forward, 1 step back”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 9593–9604.

[Zha+21] Chiyuan Zhang et al. “Understanding deep learning (still) requires rethinking generalization”. In: *Communications of the ACM* 64.3 (2021), pp. 107–115.

[Zhu+17] Jun-Yan Zhu et al. “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2242–2251.

[Zhu+20] Juntang Zhuang et al. “AdaBelief Optimizer: Adapting Stepsizes by the Belief in Observed Gradients”. In: *arXiv preprint arXiv:2010.07468* (2020).

[ZK16] Sergey Zagoruyko and Nikos Komodakis. “Wide residual networks”. In: *arXiv preprint arXiv:1605.07146* (2016).