Artificial Neural Networks as a Probe of Many-Body Localization in Novel Topologies

Cameron Beetar

Thesis Presented for the Degree of

Master of Sciences in Applied Mathematics

in the Laboratory for Quantum Gravity in Strings
of the Department of Mathematics
University of Cape Town
2022

Supervised by Prof. Jeff Murugan; Co-Supervised by
Dr Dario Rosa and Prof. Amanda Weltman

Abstract

We attempt to show that artificial neural networks may be used as a tool for universal probing of many-body localization in quantum graphs. We produce an artificial neural network, training it on the entanglement spectra of the nearest-neighbour Heisenberg spin-1/2 chain in the presence of extremal (definitely ergodic/localizing) disorder values and show that this artificial neural network successfully qualitatively classifies the entanglement spectra at both extremal and intermediate disorder values as being in either the ergodic regime or in the many-body-localizing regime, based on known results. To this network, we then present the entanglement spectra of systems having different topological structures for classification. The entanglement spectra of next-to-nearest-neighbour ($J_1 - J_2$, and, in particular, Majumdar-Ghosh) models, star models, and bicycle wheel models - without any further training of the artificial neural network - are classified. We find that the results of these classifications - in particular how the mobility edge is affected - are in agreement with heuristic expectations. This we use as a proof of concept that neural networks and, more generally, machine learning algorithms, endow physicists with powerful tools for the study of many-body localization and potentially other many-body physics problems.

Acknowledgments

I would like to thank my supervisors, Prof. Jeff Murugan, Dr Dario Rosa, and Prof. Amanda Weltman for their continuous support, regular discussions and interesting anecdotes that were never limited to this particular thesis topic; they were wonderfully energetic and motivated from the outset, and it was a pleasure to work with each of them. I would like to thank my family for being incredibly supportive of my research and goals throughout the sometimes tedious process of producing this work. I wish to thank Hannah Clayton for her seemingly infinite levels of both support and understanding throughout this process.

This work was primarily funded by the South African Research Chairs Initiative (SARChI) of the National Research Foundation in South Africa. Additional funding was provided by the South African College Croll Scholarship.

Thanks must also be given to the University of Cape Town's ICTS High Performance Computing team, as numerous computations were performed using facilities provided by this group, totalling several years worth of total computation time and far exceeding the duration of this thesis' production.

Contents

1 Introduction

Physics problems can be notoriously difficult to solve, and it has become standard practice to make use of the powerful computational machines available to researchers to solve problems that may otherwise take a researcher years (if not centuries or millennia) to compute by hand. These computational devices are, however, often limited to being as efficient and/or precise as the algorithms implemented by the programmer using them, and there are a plethora of algorithms available to the modern researcher [1, 2]. While these algorithms may be efficient when applied correctly, they may require a large amount of inherent information about the problem (and about the dynamics of the problem) to be known; update equations are often approximations of precisely known (but perhaps unsolvable) differential equations. This methodology has served (and will continue to serve) science well – examples such as the Human Genome Project are a testament to this fact – but it has very limited applicability [3].

These numerical methods are applicable to research areas where significant theory is known or where some element of the system is well understood. It is prudent to consider how one may approach a problem where notably fewer details are known about the system being studied, or where the dynamics of the system are not well understood. In such a case, all the information known about the problem may be collected with the hope/expectation that we can learn something from it and perhaps extract some relationship between 'pieces' of information. In such research areas, machine learning has an opportunity to come to the fore, with algorithms that can recognise subtle patterns that a human researcher may not have considered or that may be (practically) intractable – the ideas of 'big data' are (generically) to utilise machine learning to exploit subtle human behaviours so as to make predictions about future behaviour [4]. The extensibility of the machine learning algorithm employed to other (similar) problems is called *transfer learning*.

Phase changes in condensed matter physics are one example where unexpected behaviour was found, in particular, the transition from states that obey the Eigenstate Thermalization Hypothesis (ETH) to those that exhibit Many-Body Localization (MBL) – that is, the transition to states that violate the ETH [5]. While these ETH-violating states are better understood than before, it is still an active area of scientific research[1].

In this thesis, we will show that a simple Artificial Neural Network (ANN) (a single hidden

[1] See, for instance, https://arxiv.org/list/cond-mat.dis-nn/new for the latest research in MBL.

layer multilayer perceptron, to be precise) may be used to classify the entanglement spectra of a Heisenberg spin-1/2 chain with periodic boundary conditions in a random static magnetic field in the $\hat{z}$-direction (at a given disorder – described later – and eigenenergy) as being in either the thermalizing (ETH) or localizing (MBL) regime. Furthermore, we show that the *same* ANN may be used to classify topologically distinct systems from the initial Heisenberg spin-1/2 chain *without* retraining the ANN.

This work is structured to provide the reader with the necessary knowledge to understand the final results. Some knowledge of statistical mechanics is assumed, and no significant knowledge of machine learning is assumed. We will introduce the relevant condensed matter physics topics in §2, discussing the general theory of MBL and introducing the system to be studied and treated as fundamental; the disordered Heisenberg spin-1/2 chain with periodic boundary conditions. In §3 we introduce only the relevant subset of theory on ANN's, in particular, that theory related to the multilayer perceptron implemented for this work. We elaborate on the ANN training method and the training results, comparing these results to other work. In §5 we show the ANN's transferability to other distinct topologies of quantum graph, namely, the $J_1 - J_2$ (Majumdar-Ghosh, next-to-nearest-neighbour) graph, star graph, and bicycle wheel graph are all considered in this context. We conclude in §6 by discussing the results of this work and possible future directions.

While this work does attempt to make all relevant details of theory of both MBL and ANN's known to the reader, there will inevitably be specific questions that this work does not address. In such a case, we urge the reader to make use of the numerous references within each section.

Finally, this work is essentially an expansion of original work by the Author and supervisors of this dissertation, which is currently under review by Physical Review Letters [6].

2 An Overview of MBL

Since one of the primary goals of this thesis is to perform successful classifications of eigenstates of a system as being in either the thermalizing (ETH-obeying) or localizing (ETH-violating) regime, it is essential that we understand what each of these 'labels' means. Indeed, producing/recognising data that is used in the training of an ANN is a prerequisite for performing supervised learning (see §3), and this will be elucidated in the current section of this work. The scope of systems we consider is limited to *isolated* (or *closed*) quantum systems, where to be isolated means that the system is not in thermal contact with an *external/thermal reservoir*, sometimes also called a *heat bath*. In particular, we will initially focus on Heisenberg spin-1/2 chains with periodic boundary conditions in the presence of a random static magnetic field aligned in the $\hat{z}$-direction as our candidate system whose eigenstates we wish to train our artificial neural network to classify.

We are concerned with the long-time behaviour of these closed quantum systems, and the generic goal of this work may be recast as seeking to answer the question: What does the system look like after a 'long' time? Of course, this is a more general question than that which we attempt to answer (there are far too many such systems to answer the question in a uniform way for all of them). For the systems we study, there are two basic outcomes: thermalization (ETH-obeying) and localization (ETH-violating). There does exist what might be called a 'transition' region, where the system changes from one behaviour to the other – but this region, known as the *mobility edge*, is still a source of disagreement within the condensed matter community[2].

2.1 Isolated Quantum Systems

The reader may be concerned that isolated (or 'closed') systems are 'toy' models and may be too unphysical to provide any new insights. However, with technological improvements meaning that modern experimental apparatus, itself, may be so small and purpose-built that *it* may exist in the quantum regime (and not just the system being studied *with* said apparatus), there is cause to consider the whole system (apparatus and system of interest) as a closed quantum system [5]. We will briefly introduce the classes of systems we are interested in, then discuss

[2]For instance, the results in [7] would indicate that the mobility edge is independent of the system size, where the results in [8] are contrary to this.

how expectation values are calculated in isolate systems.

2.1.1 Basis and Hamiltonians

The classification procedure for the ANN will primarily be based upon its ability to classify the entanglement spectra of energy eigenstates (described later) of a nearest-neighbour periodic Heisenberg spin-1/2 chain in the presence of a random magnetic field aligned in the $\hat{z}$-direction (this magnetic field and its randomness is conventionally called the 'disorder', and the half-width W of the uniform distribution from which it is sampled is called the 'disorder strength', or also simply 'disorder' in short.). At every site in the chain there is a 2-dimensional state space consisting of the spin-up and spin-down state and all (complex) linear combinations thereof. Spin-1/2 systems of this kind are spanned by a basis of local eigenstates of $\hat{\sigma}^3_\alpha$ operators (where α is an index for the sites) – other choices of basis are possible but this choice of basis is typical in the literature [9]. For the particular case of a nearest-neighbour Heisenberg spin-1/2 chain (or simply the 'Heisenberg spin-1/2 chain/system' for this work), only adjacent sites are coupled (when referring to the 'Heisenberg spin-1/2 model' we shall always mean the periodic nearest-neighbour version thereof). Given an N-site chain, knowing that each site has a 2-dimensional state space, there are 2^N simultaneous eigenstates of the system as a whole. The space of possible operators that act on the N-site state space is spanned by the tensor product of local operators that may act on the states, and we consider these in the conventional spin-1/2 SU(2) representation, that is, where there are four linearly independent operators associated to each site: $\hat{\mathbb{I}}$ (identity) and $\hat{\sigma}^i_\alpha, i \in 1,2,3$ (Pauli matrices) where $\alpha \in 1,...,N$. All the Pauli matrices and Hamiltonians we consider as operators and we omit their 'hats' for the remainder of this subsection. Then, if we define $\sigma^0 \equiv \mathbb{I}$ (the '0' in this case is an index, not an exponent) the operator space is spanned by

$$\{\sigma_1^{p_1} \otimes \sigma_2^{p_2} \otimes ... \otimes \sigma_N^{p_N} | (p_1, p_2, ..., p_N) \in P\}, \tag{2.1}$$

where P are the permutations of $\{0,1,2,3\}$ over N sites. A general Hamiltonian will therefore be a linear combination of operators from this local operator product space. The space spanned by such operators is vast and not constrained by any sort of locality requirement. However, the systems we wish to study typically have highly local couplings – the operators that make

up the Hamiltonians should demonstrate this property as well. Indeed, this is the case for an N-site *disordered nearest-neigh Heisenberg spin-1/2 system*, whose Hamiltonian we define by [5]

$$H = \frac{J}{4}\sum_{i=1}^{N} \boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_{i+1} + \frac{1}{2}\sum_{i=1}^{N} h_i \sigma_i^z, \tag{2.2}$$

where the $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \sigma_3)^{\mathrm{T}}$ and σ_i is the i^{th} Pauli matrix, the J is a uniform coupling coefficient that we will conventionally take to be unity $J = 1$, the h_i moderate the local magnetic field at each site and are randomly sampled from a uniform distribution $h_i \in [-W, W], W \in \mathbb{R}^+$, and $\boldsymbol{\sigma}_{N+1} = \boldsymbol{\sigma}_1$. In general, we will discuss three other systems to attempt to test the generalizability of the ANN after training it on systems having Hamiltonians like that in (2.2). The first of these is the *disordered Majumdar–Ghosh model*, with Hamiltonian [10]

$$H = \frac{J}{4}\sum_{i=1}^{N} \boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_{i+1} + \frac{J}{8}\sum_{i=1}^{N-2} \boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_{i+2} + \frac{1}{2}\sum_{i=1}^{N} h_i \sigma_i^z, \tag{2.3}$$

which includes a next-to-nearest-neighbour coupling that is precisely half the strength of the nearest-neighbour coupling. Additionally, we consider a *disordered bicycle wheel model*, with Hamiltonian

$$H = \frac{J}{4}\boldsymbol{\sigma}_{N-1} \cdot \boldsymbol{\sigma}_1 + \frac{J}{4}\sum_{i=1}^{N-2} \boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_{i+1} + \frac{J}{8}\sum_{i=1}^{N} \boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_N + \frac{1}{2}\sum_{i=1}^{N} h_i \sigma_i^z, \tag{2.4}$$

where the final site is chosen to be connected to all others, and the *disordered star model*, with Hamiltonian

$$H = \frac{J}{4}\sum_{i=1}^{N-1} \boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_N + \frac{1}{2}\sum_{i=1}^{N} h_i \sigma_i^z, \tag{2.5}$$

which is just the bicycle wheel system modulo the nearest-neighbour coupling. These Hamiltonians can be represented graphically as is shown in figure 1. Given that we have now introduced the Hamiltonians we wish to consider (as well as all operators and eigenstates) it is natural to discuss expectation values and, in particular, how they are calculated for isolated systems.

2.1.2 Expectation Values

Consider a (Schrödinger picture) time-independent localized (short-range-interacting only) Hamiltonian, $\hat{H}$, such that this Hamiltonian operator drives the time evolution of the states in the usual way (similarly for the Heisenberg picture) by defining the usual unitary time evolution

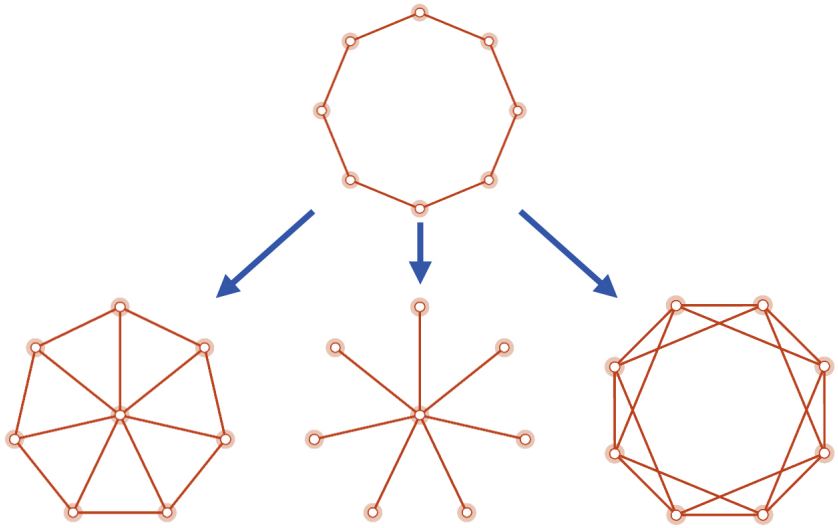


Figure 1: Graphical representation of the Hamiltonians for an 8-site system and their couplings. Top: Periodic Heisenberg spin-1/2 system. Bottom (L-R): Bicycle wheel system, star system, and Majumdar-Ghosh (next-to-nearest-neighbour) system. Image taken from work done by the author [6].

operator

$$\hat{U}(t) \equiv e^{-i\hat{H}t} \Longrightarrow \begin{cases} |\psi(t)\rangle & = \hat{U}(t)\,|\psi(t=0)\rangle\,, \text{ (Schrödinger picture),} \\ \hat{O}(t) & = \hat{U}^{-1}(t)\hat{O}(t=0)\hat{U}(t), \text{ (Heisenberg picture),} \end{cases} \tag{2.6}$$

for some Schrödinger picture state, $|\psi(t)\rangle$, or some Heisenberg picture operator, $\hat{O}$, and where we have implicitly used $\hbar = 1$ (from now on, we also omit the 'hats' on operators unless there is ambiguity). As we are not primarily concerned with the ground state, but with arbitrary excited states (or, later, the whole spectrum), we will make use of *density matrices*; a matrix derived from the density operator[3] [9]

$$\rho \equiv \sum_i p_i\,|i\rangle\,\langle i|\,, \tag{2.7}$$

where the $|i\rangle$ label the various states that a fraction, p_i, of the ensemble of particles are found in (and these are not necessarily orthogonal). Then, in an appropriate orthonormal basis $\{B_j\}, j \in \{1, ..., D\}$ for a D-dimensional space, the density matrix is

$$\rho_{jk} = \langle B_j|\,\rho\,|B_k\rangle\,. \tag{2.8}$$

[3]See Appendix §A for an introduction to the density operator and density matrix.

We consider the density operator to be in the Heisenberg picture and let all other operators remain in the Schrödinger picture, as this allows us, first, to describe the dynamics of the density operator taking the time derivative of the time-evolved operator from (2.6)

$$\frac{d\rho(t)}{dt} = i[H, \rho(t)], \tag{2.9}$$

and second, we may consider time-dependent averages of observables via[4] [5]

$$[O]_t = \mathrm{Tr}(O\rho(t)). \tag{2.10}$$

In this subsection we have taken a succinct look into a small subclass of systems that we wish to study and their expectation values. Thermalization and/or localization, as concepts that are more generally applicable to isolated quantum systems than those whose Hamiltonians, operators and states we described above, are more generically defined, and we discuss these ideas in the coming sections.

2.2 Thermalization

Thermalization refers to the process of reaching *thermal equilibrium*: a state in which the entire system may be described by only the parameters of extensive conserved quantities[5]. However, if one considers the difference between the thermal state, characterised by only a few parameters, and the initial state, that may be completely specified, it is evident that information has been lost; the system has lost its ability to recapture the information of the initial state. The system has gone through a process called *decoherence*[6] [13]. Decoherence is used to explain the counter-intuitive fact that, despite unitary time evolution preserving all information about the system, the initial conditions are 'forgotten' by the thermal state. In isolated systems, if we isolate a (relatively small) subsystem, then decoherence means that the remainder of the system acts as a heat bath for the subsystem. This is important: despite the total isolation of the model it may still thermalize – the system may 'forget' its initial conditions.

[4] See again Appendix §A.

[5] Recall, extensive quantities change in proportion to the size of a system; intensive quantities are independent of the system size [11].

[6] Since its conception, whether or not decoherence solves the measurement problem has been heavily debated, but this is outside the scope of research discussed here (*cf.* [12] for a brief introduction to the history of this debate).

When performing the separation of a system into an 'isolated environment' (the part of the isolated system that will act as a reservoir) and the small subsystem, we implicitly focus on the subsystem and wish to ask questions about the expectation values of the subsystem *only*. The density operator (2.7) and matrix (2.8) must be modified to account for this new focus on the subsystem. Supposing that we label the subsystem as system 'A' and the remainder of the system as 'B', we know that we have enforced any state, $|\Psi\rangle$, of the system to be in a product space of the Hilbert space of the environment, $\mathcal{H}_B$, and of the subsystem, $\mathcal{H}_A$ (*i.e.* $|\Psi\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$). It is understood that there is no way to associate a definite pure state to the subsystem[7]. We may nonetheless define a *reduced density operator*, ρ_A, of a system in a state, $|\Psi\rangle$ (note, the density operator for this state is $|\Psi\rangle\langle\Psi|$) whose matrix elements may be found in the usual way (see (2.8)) as [5, 14, 15]

$$\begin{aligned}\rho_A(t) &= \sum_{|j\rangle\in B} \left(\mathbb{I}_A \otimes |j\rangle\right)^\dagger |\Psi\rangle\langle\Psi| \left(\mathbb{I}_A \otimes |j\rangle\right) \\ &= \sum_{|j\rangle\in B} \left(\mathbb{I}_A \otimes \langle j|\right) |\Psi\rangle\langle\Psi| \left(\mathbb{I}_A \otimes |j\rangle\right) \\ &= \mathrm{Tr}_B |\Psi\rangle\langle\Psi| \\ &= \mathrm{Tr}_B(\rho),\end{aligned} \tag{2.11}$$

where $\mathbb{I}_A$ is the identity operator for the subsystem, A, and the sum over $|j\rangle \in B$ is over an orthonormal basis of the environment, B. It is the reduced density operator that we may use as a 'check' to see if a system undergoes thermalization in a more precise (mathematical) manner. Consider the set of initial states of the whole system that would thermalize to a temperature, T, *if they were to thermalize*. This (whole) system will have a Boltzmannian density ('probability') operator when in thermal equilibrium defined in the usual way[8]

$$\rho_{\mathrm{eq}}(T) = \frac{e^{-H/k_B T}}{\mathcal{Z}}, \tag{2.12}$$

where k_B is the Boltzmann constant, $\mathcal{Z}$ is the partition function and $H \equiv \hat{H}$ is the Hamiltonian of the system. We can, again, trace out the degrees of freedom of B to analyse the subsystem's density operator $\rho_{A,\mathrm{eq}}(T) = \mathrm{Tr}(\rho_{eq}(T))$. Now, if in the thermodynamic limit (that is, in the

[7] See Appendix §B for a canonical example of why this is the case.
[8] See appendix §C for a brief explanation of this statement.

limit where the size of the environment $N_B \to \infty$ *and* the simultaneous long-time limit $t \to \infty$) we have $\rho_A(t) \to \rho_{A,\mathrm{eq}(t)}$, then the system can be said to thermalize [5, 16]. The simultaneous limits are essential, as if $N_B \to \infty$ with t held finite then the system will never thermalize due to limits on the rate of diffusion; similarly, holding N_B fixed and taking $t \to \infty$ will result in quasiperiodic dynamics [5]. To be explicit,

$$\lim_{\substack{N\to\infty \\ t\to\infty}} \rho_A(t) = \rho_{A,\mathrm{eq}}(T) \Longrightarrow \text{System thermalizes.} \tag{2.13}$$

An interesting question to ask is: 'Under what conditions does the system thermalize?' This sort of question is essentially seeking out the *limits* of thermalization by seeking out where it *fails*. Indeed, Huse makes the observation [5]

> "Thermalization is of particular interest for initial states that are well out of equilibrium. These are atypical states because at equilibrium the system is at typical states. Thus, when we say a system thermalizes, for this to be an interesting statement it must certainly apply to some atypical states."

The thermalization of these *atypical* states is what might lead one to conclude that *all* initial states thermalize for a specific temperature. This is a natural point to consider the situation where all the initial states *do* thermalize – the setting for the Eigenstate Thermalization Hypothesis.

2.2.1 The Eigenstate Thermalization Hypothesis

The Eigenstate Thermalization Hypothesis (ETH) is a profound statement about what Rigol *et al.* refer to as a type of *thermodynamic universality* (*i.e.* a dynamic-independent property), independently proposed by Deutsch and Srednicki, and can be described as follows [16–18]: Suppose we have a system at a given temperature that *does* thermalize for any initial state – then if the initial state is an energy eigenstate the density operator is a constant in time[9], $\rho(t) = \rho(0)$, and, since we assumed the system thermalizes, this means that *every* energy eigenstate *is* thermal. Importantly, this also implies the reduced density operator is constant/thermal[10]. An equivalent statement of the ETH may be made in terms of observables. In the context of an

[9] Since $|s(t)\rangle = U(t)\,|s\rangle = \exp(-iHt)\,|s\rangle = \exp\left(-iE_s t\right)|s\rangle$ for an energy eigenstate $|s\rangle$ with eigenvalues E_s, whose exponential will cancel with the contribution from the time-evolved adjoint state $\langle s(t)|$.

[10] Since, for an energy eigenstate initial condition $|\Psi(0)\rangle = |s\rangle$ the reduced density operator evolves according to $\rho_A = \mathrm{Tr}_B\,|s(t)\rangle\,\langle s(t)| = \mathrm{Tr}_B\,|s\rangle\,\langle s|$.

observable, $\hat{O}$, the ETH means that the expectation value of this observable – when calculated for a small subsystem of an isolated quantum system that thermalizes for any initial state – satisfies the equality below (supposing that the initial state of the system is an energy eigenstate) [18]

$$\lim_{t\to\infty} \langle s(t)|\, \hat{O}\, |s(t)\rangle = \lim_{t\to\infty} [\hat{O}]_t = \lim_{t\to\infty} \mathrm{Tr}\Big(O\rho(t)\Big) = \langle O\rangle_{\mathrm{th}}(E_s), \tag{2.14}$$

for some energy eigenket $|s\rangle$ with eigenvalue E_s, and where $\langle\cdot\rangle_{\mathrm{th}}(E_s)$ is the thermal expectation value associated with the energy E_s.

The most relevant result of systems obeying the ETH is the consequence is has for the *entanglement entropy* between the subsystem (A) and the environment (B) (whose composition '$A \oplus B$' we again emphasise is a composite isolated system) which is defined by

$$S_{AB} = -k_B \mathrm{Tr}_A\Big(\rho_A \log(\rho_A)\Big). \tag{2.15}$$

In the ETH context, the entanglement entropy S_{AB} and (equilibrium) thermal entropy of the subsystem, S_A, are equivalent. This is a useful equivalence for another means of 'detecting' thermalization (not directly utilized in this work), since the thermal entropy demonstrates volume-law scaling when in a thermal eigenstate [5, 19]. There are many noteworthy consequences of the ETH, and verification of the ETH for any particular system is a difficult numerical problem. In fact, constructing initial states that do *not* thermalize in the ETH context is subtle and is not discussed here. The ETH is most powerful in that it allows for calculations of the properties of the system at thermal equilibrium given only an energy eigenstate. However, in this work we are primarily concerned with where this hypothesis *fails* to hold – this is the domain of localized systems.

2.3 Localization

In Anderson's seminal 1958 paper, he showed that lattices with randomness, introduced by including random variations in the energy at each lattice site, could demonstrate an "absence of diffusion" [20]. Recalling briefly the Hamiltonians we investigate, (2.2)-(2.5), we immediately observe one direct analogy to the situation investigated by Anderson – the random static local variable h_i. Disorder appearing in this manner (present for all times and unchanging) are called *quenched*. We briefly discuss *Anderson localization* before introducing *many-body localization*.

2.3.1 Anderson's "Nontransport" Theorem

Anderson localization refers to the localization of the wavefunction as in a lattice with quenched disorder – this localization occurs in the absence of interactions and so is an example of *single-particle localization*. In this section we will not re-derive the result, but rather provide some succinct explanations of what Anderson found in his analysis.

Theorem 1. *The "Nontransport" Theorem [20]. Suppose we have a lattice in 3 spatial dimensions, isolated from any environment (i.e. not in contact with a thermal reservoir). The lattice may be either random or regular in its construction. Suppose at each lattice site there is some entity (e.g. a spin or an electron) which we call 'spin'. Suppose also that if a spin is at some lattice site α, for instance, then it has an associated energy E_α which is sampled randomly from some probability distribution with characteristic width, W. Finally, suppose all lattice interactions are short range only, with forces $F < 1/r^3$ as $r \to \infty$. Then, for lattice site densities below some critical density, quantum transport does not occur.*

In presenting a statement of theorem 1, the reader should take note of the many assumptions that may compromise its applicability – the lattice is isolated, it is sufficiently sparse and has short range forces. It may also be noted that the Hamiltonians (2.2)-(2.5) do satisfy these conditions.

An interesting consequence of Anderson localization is in the *Anderson transitions* – the transition of a system from a (delocalized) metallic phase to a (localized) insulating phase [21]. In the localized phase, eigenstates/eigenfunctions are exponentially localized for each site, i, as [5, 21]

$$|\Psi_i^2(\boldsymbol{x})| \sim e^{-|\boldsymbol{x}-\boldsymbol{x}_i|/\xi}, \tag{2.16}$$

where ξ is the *localization length*, which typically depends on the strength of the disorder strength, W, and $\boldsymbol{x}_i$ is the physical location of the lattice site i. While Anderson localization is interaction-free localization (that is, it encourages the study of the localization properties for a single particle that may 'hop' between sites of the lattice and is free from interactions with other particles) which is referred to as *single-particle localization*, we may also study various classes of multi-particle systems (with Hamiltonians containing interaction terms) where localization is then known as *many-body localization*.

2.3.2 From Single-Particle Localization to Many-Body Localization

To study single-particle localization we need two things: a single particle state and a Hamiltonian whose dynamics conserve particle number and a term in the Hamiltonian that contains some stochastic local potential. A conventional model for numerical simulation that has these features is the *Anderson tight-binding model* [21]

$$H = t\sum_{\langle ij\rangle} c_i^\dagger c_j + \sum_i u_i c_i^\dagger c_i, \tag{2.17}$$

where $\langle ij\rangle$ specifies the sum is over nearest-neighbour sites (for what is not necessarily a 1-dimensional system so we cannot use the $i, i+1$ summation as in the previous Hamiltonians), $c_i^\dagger$, c_i are the creation and annihilation operators for a particle at the lattice site i, respectively, and u_i is a local potential randomly sampled from a uniform distribution $[-W, W]$. One can see how this Hamiltonian will evolve the system: a particle at site j can hop to site i through the action of the first term, and each site has its own local potential[11]. In $d \leq 2$ dimensions, this system has exponentially localized eigenstates falling off as in (2.16), and this can be extended to $d > 2$ with sufficiently strong disorder (*i.e.*, by increasing the width of the distribution from which u_i is sampled; that is, choosing a larger W).

A convenient link to extend the understanding found in single-particle localization to many-body localization is made by considering what is already a many-body Hamiltonian and making use of an initial condition that behaves something like that of a single particle. To that end, one may consider the disordered Heisenberg spin-1/2 chain of (2.2), with the initial condition of the spin at all lattice sites being spin-up but for one (which is spin-down) or *vice versa*. If, at any site, there are localized states with non-zero probabilities, then it is possible that the spin-down may still be there in the $t \to \infty$ limit – this constitutes some 'memory' of the initial conditions and a breakdown of decoherence – the system may never thermalize [5]. It should be noted that the extension of Anderson localization to interacting systems has been shown to occur in perturbative calculations – and many-body eigenstates may thus become localized with sufficiently strong disorder [7, 23, 24]. While this single antiparallel disordered Heisenberg model is, perhaps, expedient, it is meant to serve only as a link (for the reader)

[11] In the context of Anderson transitions, Wegner showed that the relationship between localization length (ξ) and conductivity (σ) at the transition region is given by $\xi \propto (E_c - E)^{-\nu}$, $\sigma \propto (E - E_c)^s$, with $s = \nu(d-2)$ for a d-dimensional space and E_c the critical energy at which the transition occurs [21, 22].

between single-particle and many-body localization, the latter of which we discuss below.

Many-body localization generalizes single-particle localization by arising from systems that typically include both more particles and interactions between these particles. The models we study for this thesis are 'very' finite spin-models[12]. This is due to the extremely computationally expensive calculation associated with exact diagonalization.

Let us now consider again the disordered Heisenberg spin-1/2 chain (2.2). Focusing now on this Hamiltonian, one sees that, because the local disorder (local magnetic fields) h_i are sampled from $[-W, W]$, the parameter W (the characteristic width of the distribution; the disorder strength) is the scale to which one can compare the relative strength of the interaction coupling, J. Thus, we consider the system's behaviour being characterised by some relative-strength parameter, $|J/W|$.

- $|J/W| = 0$: The Hamiltonian has nearest-neighbour interactions which, if 'turned off' by setting $J = 0$, will yield a completely decomposable space whose eigenstates are product states of local eigenstates. This system is clearly localized as the spins are unable to hop to any other lattice site.

- $|J/W| \ll 1$: This would constitute a Hamiltonian perturbed by interaction terms, and one may construct perturbative many-body eigenstates. These systems typically localize [5, 23, 24].

- $|J/W| \sim 1$: Perturbation theory breaks down and may indicate that: (i) perturbation theory is simply an insufficient tool for the purposes of studying these systems, and/or (ii) there is a transition region called the *many-body mobility edge* where the system is neither classifiable as thermalizing nor localizing; the outcome depending on which eigenstate is chosen as the initial condition[13] [24].

Critically, the appearance of the Many-Body Localizing (MBL) behaviour represents an altogether different sort of behaviour than that characterised by quantum statistical mechanics and the ETH. Moreover, the phase transition from ETH-MBL, that *may* be characterised by a mobility edge with some thermalizing and some localizing states, is highly contentious as

[12] We study systems consisting of no more than 16 spins. Luitz *et al.* were able to consider systems of up to 22 spins [7].

[13] Basko *et al.* studied perturbation theory in the temperature parameter of weakly-coupled electron systems with quenched disorder to study the transition from an insulating to a metallic system, but this is analogous to the transition of a spin system from a many-body localizing system to a thermal system [24].

the ETH is a statement about *all* eigenstates of a system thermalizing under fixed parameters. We do not address the arguments for/against a mobility edge in this work as we consider only insufficiently large system sizes (and the mobility edge's behaviour is contentious in the context of a thermodynamic limit, far outside the models in this work); we assume that the appearance and placement of a mobility edge is finite in our 'very' finite models, which is an uncontested point. In particular, Grover discusses the existence of a third intermediate region of a "a non-ergodic delocalized phase" [25].

We have now introduced some (mostly qualitative) means of understanding the thermalizing (ETH/ergodic) and localizing (MBL/non-ergodic) behaviour in isolated quantum systems. The hope was to introduce the reader, in a very generic way, to the sort of behaviour seen in these classes of systems. Detailed knowledge of *all* thermalizing/localizing systems and their individual nuances is not necessary for understanding the goal of this work. However, we have not yet described what characteristic it is that we will use to identify a system as being in either of these two phases.

2.4 The Entanglement Spectrum and Classification of States

The previous sections have given a generic overview of various isolated systems, and were intended to equip the reader with some intuition for ETH and MBL systems. Now, we make the difference between these systems more precise by introducing the *entanglement spectrum*; the 'object' we will use in our classification scheme.

The entanglement spectrum *is* a spectrum in the conventional sense as it corresponds to the eigenvalues of the *entanglement Hamiltonian*, defined in terms of the reduced density matrix as follows [15, 26]

$$\rho_A \equiv e^{-H_e} \Rightarrow H_e = -\ln(\rho_A), \tag{2.18}$$

where A labels the subsystem of interest within an isolated system. Solving for the entanglement Hamiltonian's eigenvalues, $\lambda_i, i \in 1, ..., \dim(H_e)$ produces the entanglement spectrum. A natural question to ask at this point is: '*What does this have to do with classifying a system as being ETH or MBL?*' The answer comes in four parts, as there are four well-established ways in which entanglement may generally be used for classifying states as ETH or MBL.

2.4.1 The Schmidt Gap

The *Schmidt gap* is the difference between the two largest eigenvalues of the entanglement Hamiltonian for a given eigenenergy and disorder [27, 28]. Chiara *et al.* showed in numerical simulations that in spin-1/2 and spin-1 chains the Schmidt gap is a good predictor of phase transitions from ETH-MBL near critical disorder (that is, near the region of transition) [29]. In particular, in the ETH region, the Schmidt gap 'closes', and is zero in the thermodynamic limit [29]. On the other hand, in the MBL phase the Schmidt gap approaches a value of 1. Thus the Schmidt gap provides one probe for classification of this quantum phase transition[14].

2.4.2 Entanglement Entropy Scaling

The entanglement entropy of the MBL regime demonstrates area-law scaling [30]. This is in contrast to the extensive scaling (*i.e.* volume-law scaling) associated with the ETH regime [31, 32]. The analysis of this scaling requires measuring the entanglement entropy dependence on the subsystem size N_A. As previously mentioned, the systems considered in this work are small – consisting of no more than 16 spins – due to the computational complexity associated with the numerical diagonalization of these systems. Thus, while a perfectly reasonable procedure for detecting (non-)ergodic states would be to compute the entanglement entropy (2.15) for various choices of the subsystem size and then analyse the scaling dependence, this will not be performed in our analysis.

2.4.3 Entanglement Entropy Standard Deviation

The standard deviation of the entanglement entropy provides another means of ETH/MBL phase detection [19, 33]. This is computed in the following way [15]: (i) One calculates the entanglement entropies for eigenenergies in some range $\lambda_{\text{Energy}} \in [E, E + \Delta E]$ for a system at a given disorder, W; (ii) One then computes the standard deviation of these entanglement entropies, $\sigma_{S_{AB}}$; (iii) If the system is near the phase transition region, then $\sigma_{S_{AB}}$ diverges, whereas, near the transition region $\sigma_{S_{AB}} \ll 1$ [19]. This method of phase detection will also not play a direct role in the classification scheme we employ for the ANN.

[14] It is difficult to emphasize the unexpectedness of the power of the Schmidt gap as a classification tool; two eigenvalues of the entanglement Hamiltonian of a *subsystem* of an isolated system allowing for classification of behaviour of the whole system.

2.4.4 Entanglement Spectrum Level Spacings

The entanglement spectrum has eigenvalue separation characteristics that allow for a means of predicting the regime a system is in [34]. These methods make use of random matrix theory (with particular emphasis on *Gaussian Unitary/Orthogonal Ensembles*) [34, 35]. In the ETH phase the entanglement spectrum level density distribution follows a *Marchenko-Pastur* density distribution [34]. The MBL phase deviates strongly from this distribution; Yang *et al.* showed that the average entanglement spectrum of highly-excited eigenstates obeys a Marchenko-Pastur (MP) distribution, and suggested that the fraction of states that are 'universal' (in the sense that they follow the MP distribution) may act as a suitable order parameter, that is, it appropriately vanishes in the fully localizing (typically strongly-disordered) regime [34].

The four relationships, stated above, between the entanglement entropy/spectrum and the ETH/MBL phase of the system is sufficient motivation for the use of the entanglement spectrum as an input for a classifier ANN. Indeed, all the ANN's implemented for this work take the largest $n \in \mathbb{Z}^+$ entanglement eigenvalues for a system at a given disorder and for a particular eigenenergy. It is important to emphasize that the ANN is not initialized with any 'precursory knowledge' of the identifiable features of the entanglement spectra in either regime (or near the transition); it is randomly initialized and uses known algorithms to optimize the output. Before we proceed with classification of entanglement spectra, we must introduce the requisite knowledge to understand what it is the ANN is doing, in order to understand what the output of the ANN means. This will be the goal of §3.

3 The Artificial Neural Network

Machine learning is a vast field with many sub-fields and algorithmic constructions that are produced to solve certain categories of problems. Due to the great diversity of both the algorithms and the problems we attempt to solve with machine learning, we shall focus only on the subset of machine learning models related to the contents of this thesis; Artificial Neural Networks (ANN's) and, in particular, on the Multi-Layer Perceptron (MLP) network implemented for this work.

3.1 Description of Perceptrons

The fundamental ideas that underpin ANN's are not recent developments. In 1943 McCulloch and Pitts showed that, due to the binary "all-or-nothing" characteristic associated with biological nervous activity, biological neural networks function in a similar manner to that of digital computers and the brain could, therefore, be thought of as a computational device [36].

3.1.1 Information Processing Levels

If we choose to understand the brain as an information processing device (as the brain is the original motivator for the study of ANN's), then we can make use of Marr's three *levels of information processing* in an attempt to develop an information processing device [37]. These levels are [37]:

- *Computational theory*: The highest level of abstraction. This level seeks to define the goals of the computations and the overall logic which may be used to perform the computation.
- *Representation and Algorithm*: The intermediate level of abstraction. This level elucidates the way in which the input(s) and output(s) are represented, and should make explicit how the inputs are transformed into the outputs.
- *Hardware implementation*: The lowest level; minimal abstraction. This level should detail how the algorithm may be physically realised on hardware.

While it should be self-evident that there does exist some dependency between the levels, it should be noted that these dependencies are not necessarily 'bijective' – there may exist multiple algorithms that can be implemented on a given hardware system to perform the same task (for

example, there exist numerous sorting algorithms that can be implemented using the same hardware). Indeed, some phenomena may not necessarily be tractable at more than two levels [37]. In the case of the problems *we* wish to solve, the computational theory is based on the issue of *classification*, the representation we choose is to specify that the inputs and outputs are vectors (the algorithm to transform between them will be discussed later), and the hardware implementation is done on a digital computer.

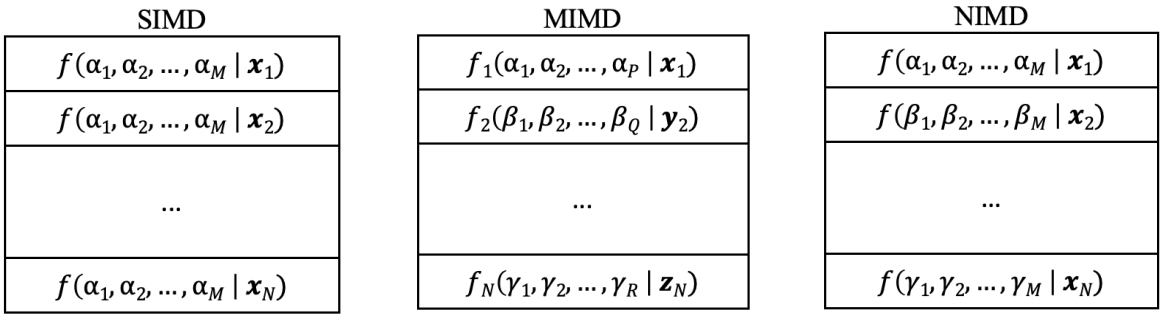
3.1.2 Parallel Processing and Neural Networks

There are two standard models for parallel computations: Single Instruction, Multiple Data (SIMD) models, and Multiple Instruction, Multiple Data (MIMD) models [4]. In SIMD, identical programs are run across multiple processors, but on different data values. In MIMD, different programs may run on different processors and may act on different data values. SIMD-based programs are easier to implement, but lack the generality of MIMD-based programs, and each paradigm has complexities that must be addressed to run effectively (*e.g.* synchronisation).

The paradigm adopted in neural networks is (subtly) distinct from both SIMD and MIMD, and is referred to as the Neural Instruction, Multiple Data (NIMD) model. [4]. This model has characteristics of both SIMD and MIMD: a single function (or 'instruction') of some parameter(s) is implemented across all processors as in SIMD, however, each processor is assumed to have some local memory whereby it may store *particular values* for the parameters of the function to be used – enabling a MIMD-like computation. In such a set-up, the analogy between biology is made concrete by identifying the processors with *biological neurons* and the local parameters with *synaptic weights* – the overall structure is analogous to a *biological neural network* [4]. The three aforementioned parallel computation paradigms are abstractly represented in figure 2.

3.1.3 A Single Perceptron

The neuron structure proposed by McCulloch and Pitts (or *McCulloch-Pitts Neurons*) may have been the historical first-step towards the complex ANN's we have today, but they are simplistic; inputs are binary (0 or 1) and outputs are also binary [36]. The generalisation of this has already been hinted at in 3.1.2 – the notion of modified *synaptic weights*. The biological motivation for this modified weights model was proposed by Hebb in 1949, and was (in conjunction with the



(a) (b) (c)

Figure 2: Abstract visualisation of different parallel computation paradigms assuming there are N processors. Each processor is a 'box'. The instructions are represented as functions, f, with the index of functions labelling distinct functions/instructions. Parameters are indexed Greek symbols and the data on which the functions act are the indexed vectors $\boldsymbol{x}_i$, although the data need not be vectors in general. (a) SIMD: identical instructions, different data but data of the same type and dimension. (b) MIMD: differing instructions, differing data type and/or dimension. (c) NIMD: same form of instructions but with unique parameters, data of same type and dimension.

McCulloch-Pitts neuron) the motivation for Rosenblatt's *perceptron* [38–40]. In this perceptron model, each input, $x_i \in \mathbb{R}$, $i = 1, ..., d$, has an associated *connection/synaptic weight* $w_i \in \mathbb{R}$, $i = 1, ..., d$. These weights were accompanied by a *bias/intercept* value, w_0 and an additional *bias unit*, $x_0 = 1$ such that the transformation of the perceptron can be written as a Euclidean inner-product

$$\boldsymbol{x} \cdot \boldsymbol{w} = y, \tag{3.1}$$

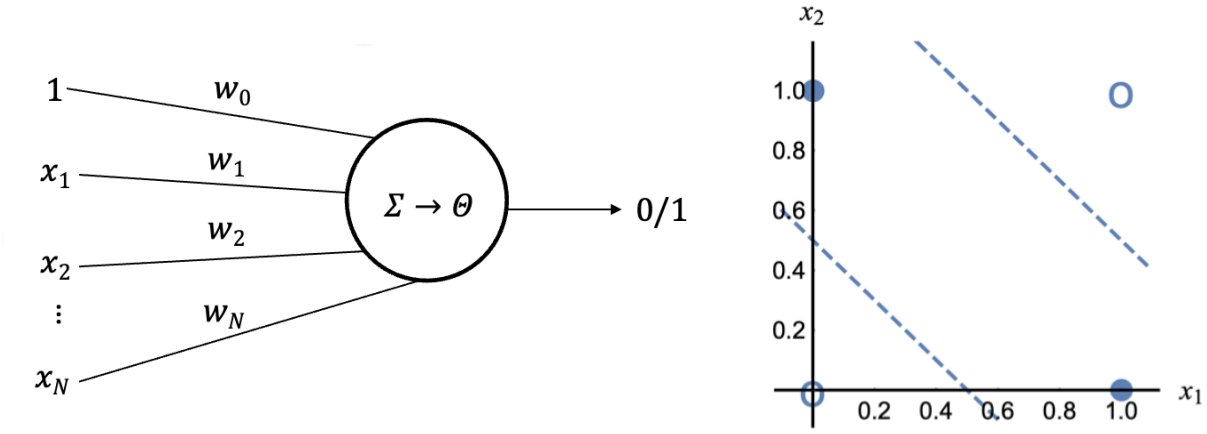
where $\boldsymbol{x} = (1, x_1, ..., x_d)$ is the *input vector*, $\boldsymbol{w} = (w_0, w_1, ..., w_d)$ is the *weight vector*, and y is the *output* of the perceptron. The goal is then to learn the weights, $\boldsymbol{w}$ associated to the input, $\boldsymbol{x}$. In 1-dimension ($d = 1$), (3.1) is nothing more than the equation of a line. For $d \geq 2$ (3.1) defines a (hyper)plane and hence one may solve for the weights using a multivariate linear fit. The process of *classification* of data may then be done by noting the hyperplane partitions the space into values above and below it; in other words, the space is partitioned into two classes, C_1 and C_2. If binary classification is all that is required, a linear discriminant function can be used to classify the data[15], by using the Heaviside function, or to be consistent with ANN

[15]It must be noted that there is the implicit assumption when using a linear discriminant function that the classes C_1 and C_2 are linearly separable. See chapter 10 of [4] as well as [41].

literature, a *threshold function*, Θ,

$$\Theta(y) = \begin{cases} 1 & \text{if } y > 0 \implies \boldsymbol{x} \to C_1, \\ 0 & \text{if } y \leq 0 \implies \boldsymbol{x} \to C_2, \end{cases} \tag{3.2}$$

where $y = \boldsymbol{x} \cdot \boldsymbol{w}$, C_1 and C_2 are the classes that the input, $\boldsymbol{x}$, is mapped to by the perceptron, and the particular classification map we have chosen is arbitrary; C_1 and C_2 could be interchanged if appropriate.



(a) (b)

Figure 3: (a) Visualisation of a perceptron for a binary classification scheme. To each component of the input is an associated weight with which the component is multiplied. The results are summed at the node, followed by an application of the threshold function which maps to either 0 or 1, representing each class. (b) The failure of the perceptron to separate the classes of an XOR function defined by binary variables $x_i \in \{1, 0\}$, $i \in \{0, 1\}$. The dashed lines are examples of the type of partition the perceptron can perform for two variables.

Despite its usefulness, this perceptron model was shown to be limited in its capabilities by Minsky and Papert, who showed that perceptrons could only solve linearly separable functions [41]. The canonical demonstration of this limitation is in trying to produce a perceptron classifier for the XOR units of two binary inputs – no one line may be constructed that separates the various outputs into two classes, as shown in figure 3b.

In the perceptron model above, the threshold function is what is known as an *activation function*. The earliest perceptron model was limited to using the threshold function as its activation function, although many other activation functions are now commonplace: Logistic, Tanh (hyperbolic tangent), Rectified Linear Unit (ReLU), and leaky ReLU to name a few [42].

3.1.4 Multiple Single-Layer Perceptrons

When there are more than 2 potential classes data may fall into, then a more sophisticated classification scheme is adopted. Suppose there are $2 < Q$ classes to which an input may be mapped, then there are Q perceptrons, and to each perceptron is an associated weight vector, $\boldsymbol{w}_j$, $j \in \{1, ..., Q\}$.

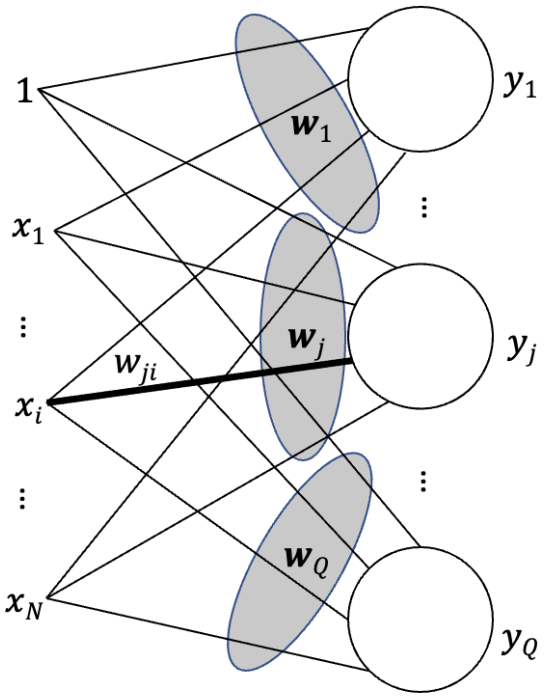


Figure 4: Q perceptrons for classification into Q classes. The x_i, $i \in \{0, ..., N\}$ with $x_0 = 1$ constitute the inputs. For each perceptron there is a weight vector whose limits are demonstrated by shaded ellipses (however some overlap was unavoidable). The result after the weighted sum at each perceptron is y_j, $j \in \{1, ..., Q\}$. Notice we have not specified an activation function. The thicker connection weight labelled w_{ji} is suggestive of the matrix methods to be implemented later, noting that one may label each connection weight with the first index for the perceptron label, the second for the input vector component.

In figure 4, we do not explicitly state what is done after the weighted sums are calculated at each perceptron; rather we state that if posterior probability is desired then the *softmax* function (discussed later) should be used [4, 43]. The previous statement requires some recourse to Bayesian decision theory. In the multiple single-layer perceptron case, one can make use of the suggestive double index on w_{ji} to write

$$y_j = \boldsymbol{w}_j \cdot \boldsymbol{x} = \sum_{i=0}^{N} w_{ji} x_i, \tag{3.3}$$

which is nothing more than an indexed version of the vector equation $\boldsymbol{y} = W\boldsymbol{x}$ where W is the weight matrix defined by w_{ji}. One could connect the output $\boldsymbol{y}$ to additional perceptrons in a similar manner to that shown in figure 4, however, since both transformations are linear (assuming we have not implemented a nonlinear activation function in computing $\boldsymbol{y}$) this two-layer network would be equivalent to a single-layer network.

At this stage, we will refrain from discussing the softmax function and cross-entropy, as these notions are better suited to a discussion on the training of the network, and the aim of §3 is to introduce, in particular, a general multilayer perceptron structure rather than the exact ANN used for this research - this will follow in later sections within this chapter.

3.1.5 Interlude: Bayesian Classification

For the cases we study in this thesis, we ask an ANN to 'recognise' the phase (or *class*) as being either ETH or MBL of some data (the entanglement spectra). Thus we want to know the phase, C, conditioned on the observable data, $\boldsymbol{x}$. Then a useful measure for classification is the conditional probability $P(C|\boldsymbol{x})$, which can be read as the probability of being in phase C, $P(C)$, given that the event $\boldsymbol{x}$ has occurred. $P(C)$ is the *prior probability* of being in phase C (the "prior" refers to being prior to the observation of $\boldsymbol{x}$), which are generally assumed to be equal [43]. A simple classification scheme that makes use of this could be, for example,

$$\text{Phase} = \begin{cases} C = 1, & \text{if } P(C = 1|\boldsymbol{x}) > 0.5 \\ C = 0, & \text{otherwise,} \end{cases} \tag{3.4}$$

where $C = 1$ corresponds to the ETH phase and $C = 0$ corresponds to the MBL phase.

For this classification to be implemented, one must be able to calculate $P(C|\boldsymbol{x})$, which may be expanded using *Bayes' Rule*

$$P(C|\boldsymbol{x}) = \frac{P(C)p(\boldsymbol{x}|C)}{p(x)}, \tag{3.5}$$

where $P(C)$ is the prior probability of being in phase C, $p(\boldsymbol{x}|C)$ is the *class likelihood* (the conditional probability that class C has an observation $\boldsymbol{x}$ associated with it), and $p(\boldsymbol{x})$ is the *evidence* (the probability that observation $\boldsymbol{x}$ is made) [4]. $P(C|\boldsymbol{x})$ is then called the *posterior probability*, as it represents the conditional probability of being in class C having made observation $\boldsymbol{x}$.

3.1.6 Multilayer Perceptrons

As discussed, the single-layer perceptron models are limited to solving classification problems that can be solved with a linear discriminant - this means that the single-layer perceptron is limited to solving linear functions. However, the single-layer perceptron models have a natural

generalisation that greatly improves their classification generality.

Recall that in §3.1.4 we noted that one could add additional perceptrons to the existing single layer of perceptrons, taking the output $\boldsymbol{y}$ of that layer as input for the next. This is an example of a *feedforward* network, where output from a layer is 'fed forward' to the next. However, we noted at the same time that these 'stacked' linear transformations would be reducible to a single-layer model. This limitation can be immediately overcome by the inclusion of an activation function, and henceforth a *layer* will consist of a linear transformation and a nonlinear activation function. This form of ANN is called a *multilayer perceptron* and has some remarkable provable properties, the most relevant of which come from a class of theorems called *Universal Approximation Theorems* (UAT's).

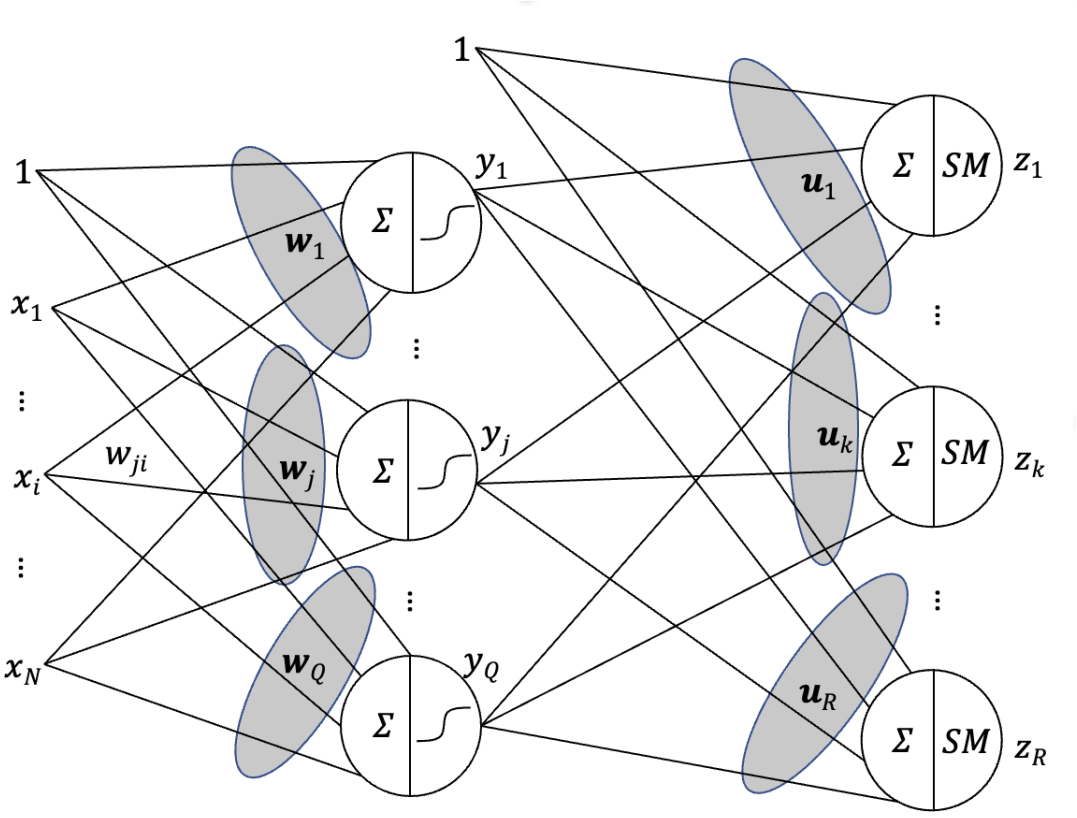


Figure 5: A single hidden layer multilayer perceptron. At each node we have included a Σ to represent the sum at the node. In the y nodes we have included a sigmoid-like symbol to represent the nonlinear activation function. The SM in the z nodes represents the softmax function. The weights are indicated by the weight vectors, $\boldsymbol{w}_j, j \in \{1, ..., Q\}$ and $\boldsymbol{u}_k, k \in \{1, ..., R\}$. Note also the inclusion of the bias unit $x_0 = y_0 = 1$ at each layer.

In 1989, Hornik, Stinchcombe, and White proved that a network with only a single hidden layer was capable of approximating (to arbitrary accuracy) any function (linear or nonlinear) of the input [44]. Multilayer feedforward neural networks do, given this result, form a class of

universal approximators. It should be noted that the result of Hornik, Stinchcombe, and White assumes a hidden layer that can be extended to 'arbitrary width' – that is, perceptrons can (theoretically) be added *ad infinitum*, and that this is obviously not realisable practically on hardware. Most of the time, however, one does not need infinite-level precision, and a finite-width hidden layer is sufficient for the purpose of the network. This result accounts for the choice made in this work to use a multilayer perceptron as the model of choice for the ANN to be used in investigating the ETH/MBL phase transition.

Taking figure 5 to be the archetype for this work, we can write out the full action of the network mathematically as

$$f_{\text{net}}(W, U, \boldsymbol{w}_0, \boldsymbol{u}_0 | \boldsymbol{x}) = \text{Softmax}[U \cdot \text{Softsign}(W \cdot \boldsymbol{x} + \boldsymbol{w}_0) + \boldsymbol{u}_0], \tag{3.6}$$

where f_{net} is the action of the MLP as a function based on the parameters W, U, $\boldsymbol{w}_0$ and $\boldsymbol{u}_0$, W is the $N \times Q$ dimensional (Real) matrix of the weights $\boldsymbol{w}_j$, $\boldsymbol{w}_0$ is the Q-dimensional bias vector (now separated from the weight matrix), Softsign(α) is an element-wise *softsign* activation function (discussed later), U is the $Q \times R$ dimensional (Real) matrix of the weights in the second layer $\boldsymbol{u}_k$, again separated from the second layer's R-dimensional bias vector $\boldsymbol{u}_0$, and Softmax($\boldsymbol{\alpha}$) is the softmax function (discussed later). We explicitly include 'dot' ($\cdot$) notation above to distinguish between linear transformations (having the '$\cdot$' present) and the nonlinear functions. The technical jargon for any layer that is not the final layer (recall, a layer is a linear transformation followed by a nonlinear function) is a *hidden layer*. Hence, the ANN we study is an MLP with a single hidden layer. The goal while training the MLP is then to optimise the free parameters – the weights and biases – such that the network accurately classifies an input vector as belonging to either the thermalizing or localizing regime. The optimisation process is performed during the training of the network.

As a final note, there is no reason one should feel limited to have a single hidden layer MLP – one can extend the number of layers to arbitrary *depth* while maintaining finite width such that other UTA theorems, specifically those that address arbitrarily deep ANN's, may apply [45].

3.2 Training an MLP

At the heart of training MLP's is *backpropagation*. The backpropagation algorithm, in simple terms, is a method of updating the weights and biases within an ANN. However, before discussing this algorithm for the case of MLP's, it is prudent to introduce the concepts of *gradient descent* and *logistic discrimination*.

3.2.1 (Stochastic) Gradient Descent

In using logistic discrimination (and in discriminant-based approaches in general) the parameters are optimized to minimize the cost function of training data. That is, given a cost function $\mathcal{C}$ with parameters $\boldsymbol{w}$, and the data set $\mathcal{X}$, we wish to use an MLP that has weights

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \mathcal{C}(\boldsymbol{w}|\mathcal{X}), \tag{3.7}$$

that is, $\hat{\boldsymbol{w}}$ are the weights in the ANN that will minimise the cost function, $\mathcal{C}$ for the data set $\mathcal{X}$, which may be written

$$\mathcal{C} \equiv \mathcal{C}(\boldsymbol{w}|\mathcal{X}) = \sum_{\boldsymbol{x}\in\mathcal{X}} e(\boldsymbol{x}, \boldsymbol{w}), \tag{3.8}$$

where e is some measure of the error of a data point (the cross-entropy for logistic regression). In general, there need not be an analytical solution to (3.7) and an iterative numerical approach becomes necessary. *Gradient descent* is one such method for minimizing $\mathcal{C}$: we (randomly) initialise the parameters we assume that $\mathcal{C}(\boldsymbol{w})$ is a differentiable function of the vector of parameters $\boldsymbol{w}$ and we then compute the *gradient vector*,

$$\nabla_{\boldsymbol{w}}\mathcal{C} = \left(\frac{\partial\mathcal{C}}{\partial w_1}, \frac{\partial\mathcal{C}}{\partial w_2}, ..., \frac{\partial\mathcal{C}}{\partial w_P}\right), \tag{3.9}$$

after whose calculation one calculates the adjustment to be made to each parameter

$$\Delta w_i = \eta\frac{\partial\mathcal{C}}{\partial w_i} \tag{3.10}$$

for each component $i \in \{1, ..., P\}$, and applied the change using the update equation

$$w_i^{\text{new}} = w_i - \Delta w_i, \tag{3.11}$$

where the update is negative as we wish to minimise the cost function. It should be noted that $\mathcal{C}$ incorporates the entire data set. The parameter η in (3.10) is a *hyperparameter* called the *learning rate*. A hyperparameter is a parameter that is not directly related to the model that is being fit, but relates to how that fit is performed. They are chosen before training of the model starts. In the case of the learning rate, it specifies how large of an update should be used – large learning parameters converge quickly to a minimal region, but if too large they may never converge to an optimal solution, whereas too small of a learning rate may converge to slowly to be practical [42]. One can see from (3.10) that when all the entries in the gradient vector are zero then the update to all components will be zero and the loss will be (locally) minimised – the cost function is not limited to forming a 'convenient' individual global minimum and, in general, the hypersurface of the cost function can be complicated and the local minimum that an algorithm converges to may be critically dependent on initial conditions[16].

Gradient descent, as it is described above, is the most basic algorithm for convergence. It has numerous shortcomings (such as strong dependence on initial conditions, sensitivity to choice of learning rate *etc*) and other methods may be used instead [42].

A common variant of gradient descent is Stochastic Gradient Descent (SGD) [46–48]. SGD makes use of *minibatches* – subsets of a full data set on which to calculate the gradient – to incorporate stochasticisty. For a choice of minibatch size, M, in a full dataset size N, there are $\sim N/M$ minibatches, and an iteration over all the minibatches is called an *epoch* [42]. In the case of SGD, the update amount for a given minibatch is

$$\Delta w_{i,B_k} = \eta \frac{\partial \mathcal{C}_{B_k}}{\partial w_i}. \tag{3.12}$$

where $\mathcal{C}_{B_k}$ is shorthand for the minibatch cost function

$$\mathcal{C}_{B_k} \equiv \mathcal{C}_{B_k}(\boldsymbol{x}, \boldsymbol{w}) = \sum_{\boldsymbol{x} \in B_k} e(\boldsymbol{x}, \boldsymbol{w}), \tag{3.13}$$

where B_k is one of the $k \in \{1, ..., N/M\}$ minibatches. The update equation retains a similar form

$$w_i^{\text{new}} = w_i - \Delta w_{i,B_k}, \tag{3.14}$$

[16] See, for example, the exercises in §4 of [42] for an exercise on – and illustration of – local convergence based on initial condition selection.

such that the update at a given iteration is not due to the entire data set, but rather due to the gradient as a result of the data in a given minibatch.

SGD is favoured over gradient descent primarily for two reasons:

- **Stochasticity**: the introduction of stochasticity into the update algorithm is a feature that reduces the chance of the algorithm converging to a local minimum. Furthermore, this stochasticity may prevent *overfitting* of the data.
- **Speed**: since the gradient is not being calculated over the entire data set for each update sequence, SGD implemented on hardware is less computationally taxing per update sequence.

While the SGD method is a very simple update method, it is effective in most instances [4]. There are, of course, numerous other methods that have been developed by the machine learning community but that are not used here. Examples of these include SGD with momentum, second order methods (ADAM, RMSProp), however, while these methods may reduce the error on the data on which they are trained, that sometimes comes at the cost of lack of generalizability [42, 49–51]. It is worth noting that these updating rules may be used as they are (up to a potential additional index for the weights of multiple single-layer perceptrons) for updating the weights in a single-layer perceptron model. A slightly more sophisticated update scheme must be employed for MLP's, and this is precisely the problem *backpropagation* (discussed later) aims to solve. However, it is necessary to introduce that which we wish to minimise in a coherent way; we must discuss logistic regression.

3.2.2 Logistic Regression

Logistic regression (or logistic discrimination) is a method of learning discrete outcomes from input [4, 42]. In the case of this thesis, we wish to classify states (based on their entanglement spectra) as being in either the ETH or MBL phase, and this makes logistic regression the ideal tool for the classification problem we wish to solve.

The logistic function

$$\sigma(\alpha) = \frac{1}{1 + e^{-\alpha}} \tag{3.15}$$

is an example of a function that, given an input, α, can return a probability of being in a given class [42]. Notice that for two classes, C_1 and C_2 being such that $y = 1 \Rightarrow C_1$ and $y = 0 \Rightarrow C_2$

the posterior probabilities can be expressed as

$$\begin{aligned} P(C_1|\boldsymbol{x}, \boldsymbol{w}) &= \frac{1}{1 + e^{-\boldsymbol{x}\cdot\boldsymbol{w}}}, \\ P(C_2|\boldsymbol{x}, \boldsymbol{w}) &= 1 - P(C_1|\boldsymbol{x}, \boldsymbol{w}), \end{aligned} \tag{3.16}$$

for an observation $\boldsymbol{x}$ and weights $\boldsymbol{w}$. We shall now see how the *cross-entropy* cost function naturally arises when using logistic discrimination.

Consider using Maximum Likelihood Estimation (MLE) to choose parameters, $\boldsymbol{w}$, that maximise the probability of observing N given data pairs $(\boldsymbol{x}_i, y_i)$, $i \in \{1, ..., N\}$ and $y_i = 1$ or 0. The probability of making all N observations given weights $\boldsymbol{w}$ is given by the product

$$P(\{(\boldsymbol{x_1}, y_1), ..., (\boldsymbol{x_N}, y_N)\}|\boldsymbol{w}) = \prod_{i=1}^{N} \sigma(\boldsymbol{x_i} \cdot \boldsymbol{w})^{y_i} [1 - \sigma(\boldsymbol{x}_i \cdot \boldsymbol{w})]^{1-y_i}, \tag{3.17}$$

after which we take the logarithm to compute the log-likelihood

$$\mathrm{LL}(\mathcal{X}, \boldsymbol{w}) = \sum_{i=1}^{N} y_i \ln \Big[\sigma(\boldsymbol{x_i} \cdot \boldsymbol{w})\Big] + (1 - y_i) \ln \Big[1 - \sigma(\boldsymbol{x_i} \cdot \boldsymbol{w})\Big], \tag{3.18}$$

where we have used $\mathcal{X}$ above to represent the set of all observed input-output pairs. In a process that anti-parallels that of finding the value of the parameters that minimizes the error (see (3.7)), we are now concerned with the values of the parameters that *maximise* the probability of making the given observations, hence we wish to find the optimal parameters

$$\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} \Big[\mathrm{LL}(\mathcal{X}, \boldsymbol{w})\Big]. \tag{3.19}$$

The cost function used for linear regression is the negative log-likelihood since the process of finding a likelihood function to minimise is equivalent to having a cost function to minimise (this is nothing but the statement that (3.7) $\equiv$ (3.19)) which, in this case, leads to the cross-entropy [4, 42]

$$\begin{aligned} \mathcal{C}(\mathcal{X}, \boldsymbol{w}) &= -\mathrm{LL}(\mathcal{X}, \boldsymbol{w}) \\ &= -\sum_{i=1}^{N} y_i \ln \Big[\sigma(\boldsymbol{x_i} \cdot \boldsymbol{w})\Big] + (1 - y_i) \ln \Big[1 - \sigma(\boldsymbol{x_i} \cdot \boldsymbol{w})\Big]. \end{aligned} \tag{3.20}$$

As this is precisely the cost function we wish to minimize it is easy to write down the equation

to be solved

$$\nabla_{\boldsymbol{w}}\mathcal{C} \stackrel{!}{=} 0, \tag{3.21}$$

for $\mathcal{C} \equiv \mathcal{C}(\mathcal{X}, \boldsymbol{w})$, however, performing closer inspection of this equation (by explicitly writing out the derivative) gives the transcendental equation

$$\begin{aligned}\nabla_{\boldsymbol{w}}\mathcal{C} &= -\sum_{i=1}^{N}\Big\{\nabla_{\boldsymbol{w}}\Big[y_i \ln\big(\sigma(\boldsymbol{x}_i\cdot\boldsymbol{w})\big) + (1-y_i)\ln\big(1-\sigma(\boldsymbol{x}_i\cdot\boldsymbol{w})\big)\Big]\Big\} \\ &= -\sum_{i=1}^{N}\Big\{y_i\big[1-\sigma(\boldsymbol{x}_i\cdot\boldsymbol{w})\big]\boldsymbol{x}_i + (1-y_i)\big[-\sigma(\boldsymbol{x}_i\cdot\boldsymbol{w})\big]\boldsymbol{x}_i\Big\} \\ &= \sum_{i=1}^{N}\Big[\sigma(\boldsymbol{x}_i\cdot\boldsymbol{w}) - y_i\Big]\boldsymbol{x}_i,\end{aligned} \tag{3.22}$$

where in the first line we have immediately applied the distributivity of the derivative over the sum and we used that $\nabla_\alpha\sigma(\alpha) = \sigma(\alpha)(1-\sigma(\alpha))$ in going from the first line to the second. The problem with the transcendental equation in (3.22) is that it has no closed-form solution, and hence we resort to the methods described in §3.2.2 to minimize the cost function [42].

While the presence of the cross-entropy as a cost function is implicit within a logistic activation function, we have already stated that we do not use this function, but rather that we use the SoftSign (SS) activation function

$$\mathrm{SS}(\alpha) = \frac{\alpha}{1+|\alpha|}, \tag{3.23}$$

whose derivative is polynomial

$$\frac{\partial[\mathrm{SS}(\alpha)]}{\partial\alpha} = \frac{1}{(1+|\alpha|)^2}, \quad \alpha\in\mathbb{R}, \tag{3.24}$$

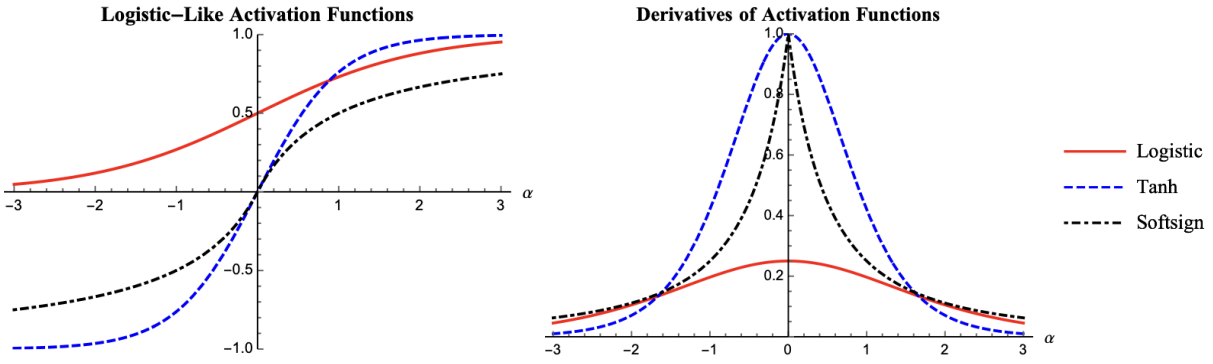
which should be contrasted with Tanh

$$\tanh(\alpha) = \frac{e^{\alpha}-e^{-\alpha}}{e^{\alpha}+e^{-\alpha}}, \tag{3.25}$$

whose derivative is exponential

$$\frac{\partial[\tanh(\alpha)]}{\partial\alpha} = 1-\tanh^2(\alpha). \tag{3.26}$$

The Logistic, Tanh, and Softsign activation functions are all examples of sigmoid activation



(a) (b)

Figure 6: Three similar activation functions and their derivatives: Logistic (red, solid line), Tanh (blue, dashed), and Softsign (black, dot-dashed). (a) The activation functions. We use α as a generic parameter. (b) The derivatives of the activation functions. Notice that, near zero, the Softsign function has a greater derivative than the Logistic function, but not as steep as the Tanh function.

functions. While the derivation of the cross-entropy cost function came about naturally when using the logistic function this is not necessarily the 'best' choice for training a network because [52]:

- Symmetric sigmoid functions (like Tanh and Softsign) often converge faster.
- Tanh and Softsign are more likely to produce values close to zero (notice that the logistic function will always have a positive mean output).

There is a simple relationship between the Tanh and Logistic functions

$$\tanh(\alpha) = 2\sigma(2\alpha) - 1 \tag{3.27}$$

where σ is the logistic function as in (3.15). While a similar relation does not exist between Softsign and the other functions – Tanh converges exponentially and Softsign converges polynomially. Of course, since the Tanh and Softsign both range from (-1,1) they cannot be interpreted as probabilities. This is alright, so long as there is a data normalizer, such as the *Softmax function* [4, 43]; suppose we have K classes and we wish to extract the posterior probability for an observation to be in a particular class, C_i, $i \in \{1, ..., K\}$, then we can find this (by treating the

classes uniformly) by using the Softmax function

$$y_i = P(C_i|\boldsymbol{x}, \boldsymbol{w}, f) = \frac{\exp\left[f(\boldsymbol{x} \cdot \boldsymbol{w}_i)\right]}{\sum_{j=1}^{K} \exp\left[f(\boldsymbol{x} \cdot \boldsymbol{w}_j)\right]}, \tag{3.28}$$

where f corresponds to the choice of nonlinear activation function, $\boldsymbol{x}$ is the observed data and $\boldsymbol{w}_i$ are the current weights of the i^{th} class.

In choosing an activation function we performed empirical comparisons of the results of the MLP's trained with different activation functions (discussed later).

3.2.3 Backpropagation

The case of a single-layer perceptron model has simple update rules for its weights as discussed in §3.2.1. When there are multiple layers (as in an MLP), however, we need a method for updating the weights in the hidden layers. Supposing we have a 2-layer network (that is we have a single hidden layer), then we may consider the output from the hidden layer as the input to the final layer and may use the SGD update method without modifications. For the weights in the hidden layer, we must use the chain rule. Referring to figure 5, we can calculate the updates for the weights U_{kj}, $k \in \{1, ..., R\}$, $j \in \{1, ..., Q\}$ using SGD. The updates associated with the weights W_{ji}, $i \in \{1, ..., N\}$ are then calculated using the chain rule

$$\Delta W_{ji} = \eta \sum_k \left(\frac{\partial \mathcal{C}}{\partial z_k} \frac{\partial z_k}{\partial y_j} \frac{\partial y_j}{\partial W_{ji}} \right), \tag{3.29}$$

where $\mathcal{C}$ is the cost function and η is the learning rate. We do not indicate whether minibatches are being used the above equation as the equation is identical, it is only what data the cost function is calculated on that is changed. As an intuitive explanation of equation 3.29, the error is 'propagated backwards' from the final layer back to the hidden layer(s). The update equation for the hidden layer may then be simply written as

$$W_{ji}^{\text{new}} = W_{ji} - \Delta W_{ji}. \tag{3.30}$$

We can now explicitly write out the update equations. For the second layer we have the Softmax function written as

$$z_k = \frac{\exp(v_k)}{\sum_j \exp(v_j)} \tag{3.31}$$

where v_k is the value of the input after the linear transformation U is applied to y_j, so $v_k = U_{kj}y_j$, and the cross-entropy loss function

$$\mathcal{C} = -\sum_k \hat{z}_k \ln(z_k) \tag{3.32}$$

where $\hat{z}_k$ is the target value for the k^{th} class. The derivative of z_k with respect to v_a is

$$\begin{aligned}
\frac{\partial z_k}{\partial v_a} &= \frac{\partial}{\partial v_a}\left(\frac{\exp(v_k)}{\sum_j \exp(v_j)}\right) \\
&= \exp(v_k)\frac{\partial v_k}{\partial v_a}\Big[\sum_j \exp(v_j)\Big]^{-1} - \exp(v_k)\Big[\sum_i \exp(v_i)\Big]^{-2}\exp(v_a) \\
&= \delta_{ka}z_k - z_k z_a,
\end{aligned} \tag{3.33}$$

where we know $\partial v_k/\partial v_a = \delta_{ka}$ is the Kronecker-delta symbol. Using this result we compute

$$\begin{aligned}
\frac{\partial \mathcal{C}}{\partial v_a} &= -\sum_k \left[\frac{\partial}{\partial v_a}\Big(\hat{z}_k \ln(z_k)\Big)\right] \\
&= -\sum_k \left[\hat{z}_k \frac{1}{z_k}\Big(\delta_{ka}z_k - z_k z_a\Big)\right] \\
&= -\hat{z}_a + \hat{z}_a z_a - \sum_{k\neq a}\Big(-\hat{z}_k z_a\Big) \\
&= z_a\Big(\sum_k \hat{z}_k\Big) - \hat{z}_a \\
&= z_a - \hat{z}_a,
\end{aligned} \tag{3.34}$$

where we have used that the target vector, $\hat{\boldsymbol{z}}$ is a *one-hot* vector when used in training; it will have precisely one entry that is non-zero and that entry will take the value 1. Recall that we chose $v_k = U_{kj}y_j$. The result of taking the derivative over U_{kj} is nothing but a trivial application of the chain rule, and does nothing more than add an additional factor of y_j in both derivatives above, giving the update value for the outer matrix U as

$$\Delta U_{kj} = \eta\frac{\partial \mathcal{C}}{\partial U_{kj}} = \eta(z_k - \hat{z_k})y_j \tag{3.35}$$

with the update equation explicitly being

$$U_{kj}^{\text{new}} = U_{kj} + \eta(\hat{z_k} - z_k)y_j, \tag{3.36}$$

where we have multiplied through the bracket with the leading minus sign. We now wish to calculate the updates in the hidden layer. We know that we must use backpropagation and solve for $\frac{\partial \mathcal{C}}{\partial z_k}\frac{\partial z_k}{\partial y_j}\frac{\partial y_j}{\partial W_{ji}}$. The first of these is straightforward

$$\frac{\partial \mathcal{C}}{\partial z_k} = \frac{\partial}{\partial z_k}\Big[-\sum_a \hat{z}_a \ln(z_a)\Big] = -\sum_a \frac{\hat{z}_a}{z_a}\frac{\partial z_a}{\partial z_k} = -\sum_a \frac{\hat{z}_a}{z_a}\delta_{ak} = -\frac{\hat{z}_k}{z_k}. \tag{3.37}$$

The remaining two derivatives are slightly more tedious, but we can use the chain rule applied to (3.31) and(3.33) to easily find

$$\frac{\partial z_k}{\partial y_j} = \frac{\partial z_k}{\partial v_a}\frac{\partial v_a}{\partial y_j} = (\delta_{ka}z_k - z_k z_a)U_{aj}, \tag{3.38}$$

and the final derivative (making use of (3.24))

$$\frac{\partial y_j}{\partial W_{ji}} = \frac{\partial y_j}{\partial \alpha_l}\frac{\partial \alpha_l}{\partial W_{ji}} = \frac{\delta_{lj}x_i}{(1+|\alpha_l|)^2} = \frac{x_i}{\Big(1+|\sum_b W_{jb}x_b|\Big)^2}, \tag{3.39}$$

so that we may finally explicitly write the update rule for the hidden parameters

$$\Delta W_{ji} = \eta\sum_k (z_k - \hat{z}_k)U_{kj}\frac{x_i}{\Big(1+|\sum_b W_{jb}x_b|\Big)^2}, \tag{3.40}$$

with the update equation thus being

$$W_{ji}^{\text{new}} = W_{ji} - \eta\sum_k (\hat{z}_k - z_k)U_{kj}\frac{x_i}{\Big(1+|\sum_b W_{jb}x_b|\Big)^2}. \tag{3.41}$$

What we have neglected to include in the update equations is the sum over minibatches of data. These sums have no effect on the form of the equation, and would just require an additional index to be summed over in the cost function.

3.2.4 Improving Training: Techniques

The backpropagation procedure forms the basis of the learning mechanism, however there are additional well-known techniques that are used to improve various aspects of an ANN training procedure [4, 42, 52]. In this work, we have implemented: heuristic rules for weight initialisation, weight decay, dropout regularisation, confidence optimisation, and cross-validation. We will

discuss these briefly for completeness, however it is recommended that the interested reader consult the references herein for more thorough discussions.

Heuristic Weight Initialisation: As was noted in §3.2.1, the initial conditions can have a significant effect on how an ANN converges, and to which minimum. When using a sigmoid activation function (such as Softsign), the weights should be initialised such that the sigmoid function is near its maximal gradient – that is, near zero [52]. This is especially important for sigmoid functions as, if they are acting on values far away from zero, then they will *saturate*; they will be in a region with too small a gradient for practical learning speeds where they may not easily be able to escape. Recall that backpropagation makes use of a 'chain' of derivatives and, considering figure 6b it is clear how learning may be stunted from the outset for values incorrectly initialized. Given the choice of the Softsign of activation function, we implement the following heuristics whenever weights are initialised [52]: weights are initialised by random sampling from a Gaussian distribution with a mean of 0 and a standard deviation of 0.1. With this choice of distribution as the one from which random sampling is performed, we can

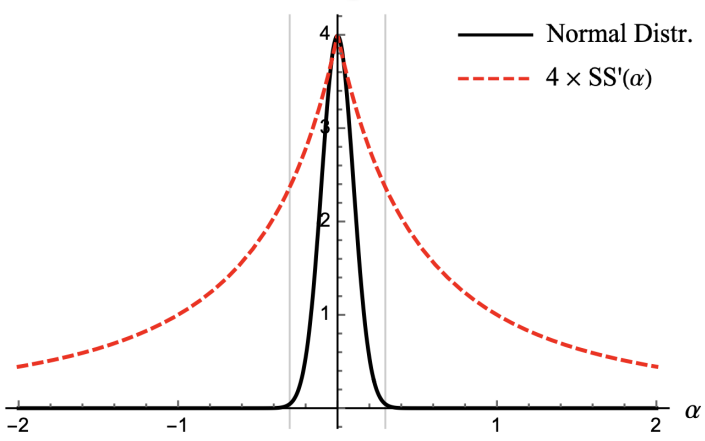


Figure 7: The distribution from which the initial parameters are sampled and the (scaled-by-four) gradient of the Softsign function in that region. The symmetric vertical lines are placed 3 standard deviations from the mean ($\sigma = \pm 0.3$) and represent the region in which 99.7% of the initial parameters are sampled.

be confident that most of the data will be sampled in a region in which the derivative of the Softsign function is in the range $[\sim 0.59, 1]$ (up to 3 standard deviations) and parameters should not be highly saturated after initialisation.

Weight Decay: An important part of any model-fitting process is the *complexity* of the model (how many free parameters there are that may be optimised) and the complexity of the system we are trying to successfully fit and subsequently classify. It is a well-known issue that fitting a model that is too complex to simple data can result in *overfitting*, where the general features of data are lost to a very specific fit to a single data set – a fit may be free from any difference for a given data set but may *generalise* poorly [4, 42, 53]. Since generalisation is

fundamental in being able to make predictions about 'unseen' data, it is necessary to balance complexity with generalizability [52]. It would not be unreasonable to train multiple MLP's of varying degrees of complexity and test its classification skill on a data set that is known to the experimenter but unseen by the network, sometimes referred to as a *test set*. However, it is often more efficient to use *structural adaptation* of the network size: to dynamically alter the network size during training. This may be done *constructively* (increasing complexity by increasing the number of free parameters) or *destructively* (removing unnecessary free parameters) [4]. *Weight decay* is an example of destructive structural adaptation; parameters (weights) that have, at some point during training, become non-zero but that do not contribute further to the minimisation of the cost function should decay to zero – superfluous parameters should be removed by the training process. This is included as an additional term in the parameter update equations [53]

$$\Delta W_{ji}^{\mathrm{WD}} = \eta \frac{\partial \mathcal{C}}{\partial W_{ji}} + \beta W_{ij} \tag{3.42}$$

(recall the update value is subtracted from the existing value as we have defined it in (3.30)) where β is an additional hyperparameter used to control the rate of weight decay – which is equivalent to performing gradient descent on the cost function and including the square of the l_2-norm in the cost function [4, 15, 53]

$$\mathcal{C}^{\mathrm{WD}} = \mathcal{C} + \beta |W|_{l_2}^2. \tag{3.43}$$

In this way, we can focus on providing a sufficiently large network to classify the data and let unnecessary free parameters decay so that they the model fitted by the network is not too complex.

Dropout Regularisation: *Dropout regularisation* is another method for preventing overfitting. In dropout regularisation, only some of the parameters are trained in a given training iteration. This is done by completely removing neurons and their connections for a given training run [54]. This reduces the co-adaption and co-dependence of parameters and significantly improves generalisation [15, 54]. For the training implemented in this work, we did not select a particular number of neurons to be randomly removed from a particular training iteration. Instead, we chose that each neuron should be deactivated with a probability of 50% for each run such that, after many training iterations, the average number of trained neurons per train-

ing iteration converges to half the total number of neurons. This method has been shown to further reduce the classification error on a test data set when combined with the weight decay implemented in this work (l_2-norm weight decay). [15, 54]

Confidence Optimisation: The methods in §3 discussed thus far have centred primarily around supervised learning techniques where for each training data input vector there is an expected/known output that we wish our MLP can learn to predict. In making a prediction about the class of some data, we wish insofar as is reasonable that the classification is maximally precise – we want the posterior probability of being in a particular class C_i due to observation $\boldsymbol{x}$, $P(C_i|\boldsymbol{x})$, to be as close to 1 or 0 as possible. Idealistically, an output posterior probability vector would be a one-hot vector, like the target vector. Practically, this rarely happens due to the limited numerical accuracy of computers or the learning algorithm itself. We should still attempt to implement a maximally confident network. We have thus far, had a general discussion about the network we implement without making many direct references to the classification problem we are solving – we will now briefly mention a property of the classification problem we are attempting to solve to explain how confidence optimization is performed.

For the data we classify there are 3 data regions: low-disorder and thermalizing, high-disorder and localizing, and intermediate disorder with unknown behaviour. For the supervised learning portion of the learning algorithm, we specify the data and the targets at low and high disorder values. The intermediate data phase is never trained in a supervised manner. However, only implementing this form of learning will lead to a network which is very decisive in the extremal disorder regions and 'agnostic' in the intermediate regions. To avoid this, we implement a trivial unsupervised learning algorithm for the intermediate disorder region that runs parallel to the supervised learning algorithm and that will contribute to a shared cost function. We want to penalise uncertainty in classification of the intermediate disorder data, and a simple method for doing this is by calculating the *Shannon entropy* of its output [55, 56]

$$f_{\text{net}}(\boldsymbol{x}_{\text{int}}) \ln \Big[f_{\text{net}}(\boldsymbol{x}_{\text{int}}) \Big], \tag{3.44}$$

where f_{net} is the action of the MLP that is acting on the intermediate data point $\boldsymbol{x}_{\text{int}}$, and attempting to minimise the negative of its value. The negative Shannon entropy cost function that we use to minimise the classification uncertainty is shown in figure 8. This is the final

addendum to the cost function we use in training. Additionally, we know that we will be using a posterior probability classification scheme distinguishing between two classes only (ETH/MBL), and, therefore, we will have a two-dimensional one-hot vector as our target vectors used in supervised learning of the extremal disorder regions, where the output of the MLP will be a 2-dimensional vector of the posterior probabilities of being in a particular class. Minibatches are used in training, so a single batch iteration of the cost function can be written as

$$\begin{aligned} \mathcal{C}_{\text{full},k}(f, f_{\text{net}}) = &- \sum_{\boldsymbol{x}\in B_{\text{ext},k}} \sum_{i=1}^{2} \left(f_i(\boldsymbol{x}) \ln \left[f_{\text{net},i}(\boldsymbol{x}) \right] \right) \\ &- \gamma \sum_{\boldsymbol{x}\in B_{\text{int},k}} \sum_{i=1}^{2} \left(f_{\text{net},i}(\boldsymbol{x}) \ln \left[f_{\text{net},i}(\boldsymbol{x}) \right] \right) + \beta |W|_{l_2}^2, \end{aligned} \tag{3.45}$$

where we have added a hyperparameter, γ, to moderate the relative importance of the confidence penalty, the sets $B_{\text{ext(int)},k}$ represent the k^{th} minibatch of the extremal(intermediate) data, $f_{\text{net},i}$ is the i^{th} component of the output of the network, f_i is the i^{th} component of the target vector, and we have included the weight decay term from (3.43).

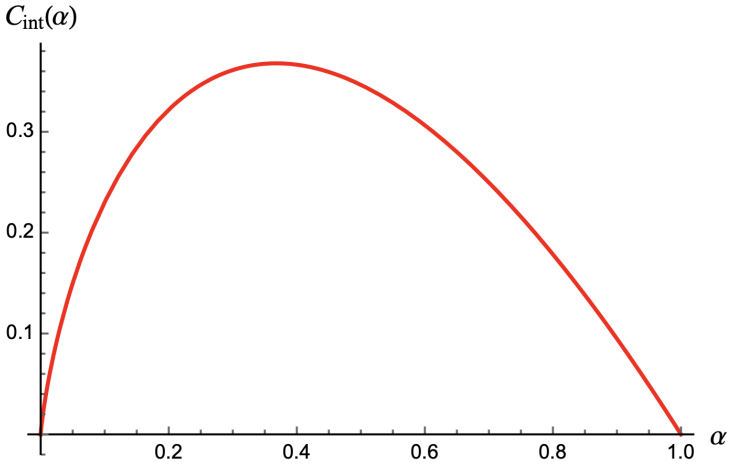


Figure 8: The confidence penalty associated with the negative Shannon entropy. Notice that minimizing this cost function on intermediate data, where the classification uncertainty may be higher, will 'push' the network away from agnostic classifications.

Cross-Validation: Another method of training that aims to reduce overfitting (or increase generalizability) is called *cross-validation* [57, 58]. In cross-validation, the data is divided into three sets: a *training set*, a *validation set*, and a *test set* [4]. The training set is used as previously discussed to fit parameters to minimize the cost function on the data. The validation set is used to test the generalizability of the fitted model – a better choice of model is one that minimises the error when calculated on a validation set as well as the training data (assuming that the validation and training sets are sufficiently large). The test set is then used if we wish to report the expected error of the model – we can no longer use the validation set as, by including it in

the training process, it essentially becomes training data – so the test set plays the same role as the validation set, but is reserved for testing the error on the post-training MLP.

There are a plethora of appendages and methods of optimization that may be used with an MLP to enhance its ability to perform the task of classification. The field of artificial neural networks is a rapidly developing one, and, because of this rate of development, it is probable that a 'better' classification scheme/algorithm exists or is in development. Nonetheless, the features of MLP's and the adaptations we give to the MLP we use are an effective[17] classification tool. That being said, there is often a simple way to improve the MLP's output – by giving it more input data or by including some intentional data groupings (recall §2.4.3). Such inclusions can make designing the ANN more difficult and make the network structure notably more complex. This is not our goal – instead, we seek to use what *is* understood about the ETH/MBL phase transition to sensibly select what should be meaningfully classifiable input data (the entanglement spectrum) and then to use the ANN as a tool to probe into systems where results are limited to conjecture or not known at all. This way, we may test the strength of what is conjectured and/or provide some intuition for researchers looking to understand problems that are potentially intractable so that they might be able to (eventually) justify an *ansatz* that might take the physics/research community forward. The emphasis is on letting ANN's act as a tool to probe beyond what is known that it may eventually be (provably) understood.

[17] 'Effective' the sense that it is able to effectively classify the system, regardless of subtle underlying intricacies; it detects *something* in the entanglement spectra that allows for effective classification.

4 Results

Having introduced some of the theory describing MBL and now having introduced the features of the MLP[18] that we use to optimise our classification, we are poised to present the results. This discussion will follow a chronological order: we start with the results of training before describing the results of giving the MLP 'novel' topologies to classify.

4.1 Implementation, Training, and Training Results

In §3.2 we discussed how one might train an MLP. We now present the results of the training algorithms implemented for this work. We benchmark the ANN against the work of Schindler *et al.*, Luitz *et al.* and Šuntajs *et al.* [7, 15]. The reader should note a strong similarity with the work of Schindler *et al.* – this is intentional as their research was a strong motivator for this work. However, where Schindler *et al.* opted for the ReLU activation function, we opt for the Softsign activation function [15]. The results of this section and §5 are the results underlying original work by the author and supervisors currently under review by Physical Review Letters and available on the arXiv [6].

4.1.1 Implementation of the MLP

Before discussing how the network was trained, we present the network as it was prior to and during the training process. This should serve as an abstract model to keep in mind when considering the network hereafter, and should also serve to solidify the many concepts introduced in §3. The MLP may be abstractly modelled as in figure 9. We now discuss the many aspects of the network. This discussion is aimed at identifying what each element of the network does, as opposed to describing *how* it realises that task, as those issues were discussed in §3. The following is with reference to figure 9 and is for the case of $N = 16$ for definiteness[19]:

- **A**: Unsupervised input data; an ordered length-35 Entanglement Spectrum (ES).
- **B**: Supervised input data (target **Q**); an ordered length-35 ES.
- **C**: Input stacking layer; stacks input data into a length-70 vector.

[18] At this stage and hereafter, we use 'the MLP/ANN/neural network/network/...etc' interchangeably.

[19] We use the terminology 'layer' very loosely here – the natural alternative of 'node' also has meaning in the ANN context. The 'layers' described here should not be confused with the formal layers described in §3.

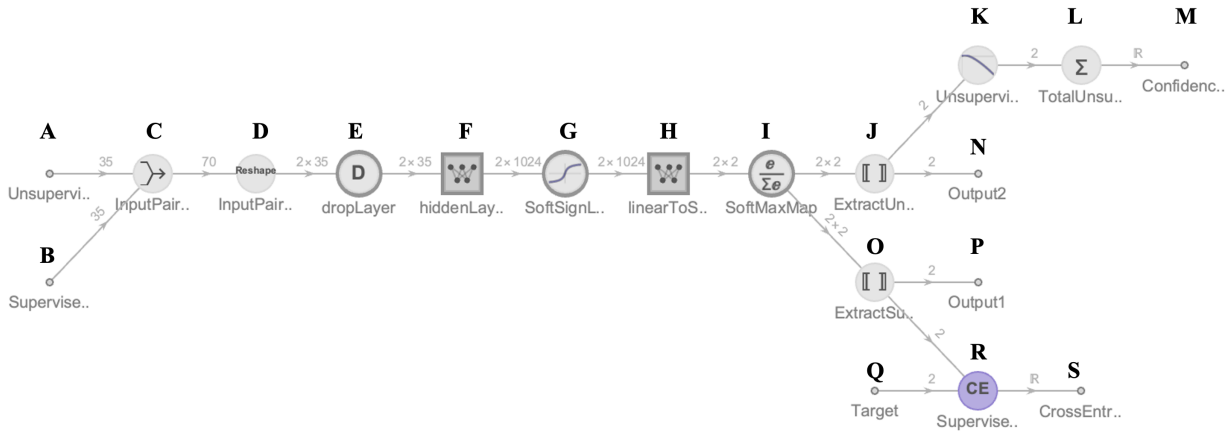


Figure 9: The training network for $N = 16$ entanglement spectra. Every element of the ANN is labelled for ease of discussion. Graphic made by the author; edited from [6].

- **D**: Reshape layer; converts the length 70 vector into a 2×35 matrix where each row is treated independently by the network hereafter.
- **E**: Dropout regularization layer; performs dropout regularization on output of **D** such that a random number of neurons are 'turned off' (*i.e.* set to zero). The number dropped is normally distributed and centred at 50%.
- **F**: Hidden layer; an (*entanglement spectrum length*)$\times$1024 matrix.
- **G**: Softsign (or alternative) activation function.
- **H**: Final linear layer; a 1024×2 matrix.
- **I**: Softmax layer; turns output into posterior probabilities.
- **J**: Extraction layer 1; extracts unsupervised result.
- **K**: Unsupervised loss function; calculates the confidence penalty of each component of the output of the unsupervised data. See figure 8 and (3.44).
- **L**: Sum layer; sums confidence penalties.
- **M**: Output; Confidence penalty loss function result.
- **N**: Output; Unsupervised classification result – a 2-component posterior probability with Prob(ETH) in component 1 and Prob(MBL) in component 2 with Prob(MBL) = [1 - Prob(ETH)].

- **O**: Extraction layer 2; extracts supervised data result.
- **P**: Output; Supervised data classification result (similar to **N**).
- **Q**: Target data; used to compute cross-entropy loss on the network's predictions. A 2-component output vector that is either $(1, 0)$, that is 'ETH-definite' OR $(0, 1)$, that is 'MBL-definite'.
- **R**: Cross-entropy loss layer; calculates the cross-entropy loss described by (3.20).
- **S**: Output; cross-entropy loss function result.

Once we are satisfied with the network's output, we can reduce this abstract model significantly, retaining only those elements needed to perform a classification. This model 9 is trained using

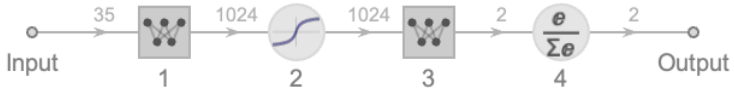


Figure 10: The trained MLP: extracted from the full MLP used in training. To be precise, the extracted elements here correspond to: $1 \leftrightarrow$ **F**; $2 \leftrightarrow$ **G**; $3 \leftrightarrow$ **H**; $4 \leftrightarrow$ **J** (all referring to figure 9). Graphic made by the author; also in [6].

the methods described in §3.2. Before describing the results of the trained network, it is necessary to describe what the training data *is* – this is the subject of §4.1.2.

4.1.2 Training Data

We train the MLP on the entanglement spectra of the disordered Heisenberg spin-1/2 chain with periodic boundary conditions at a given disorder and having a given eigenenergy. We restrict ourselves to the subspace of configurations of a spin-chain having no net spin ($S^z_{\text{tot}} = 0$). Prior to this restriction, for an N-site system there are 2^N possible states (since spin can only be $\pm 1/2$ at each site). When restricting ourselves to the $S^z_{\text{tot}} = 0$ subspace the number of possible states decreases to $\binom{N}{N/2}$. Additionally, we restrict the subsystem we consider to *also* have net 0 spin. For example, for an 8-site system (letting $0 \leftrightarrow -1/2$ and $1 \leftrightarrow 1/2$ for readability) $\{1, 1, \underline{1, 1, 0, 0}, 0, 0\}$ and take the underlined part as our subsystem. Thus there are $\binom{N/2}{N/4}$ possible states in the subsystem we consider. We now explain how the data generation algorithm works.

The data generation algorithm is based on a function that takes 3 parameters: N (total number of sites in the Hamiltonian), a positive integer divisible by 4; W (disorder parameter –

see comments under (2.2)), a non-negative real number; and $\{h_i\}$, a normalized random magnetic field (an array of numbers of length N that is sampled from $[-1, 1]$). The only peculiarity at this point is the separation of the h_i and W. We perform this separation because it means that, prior to running the data generation function, we can generate several lists of N numbers sampled from a uniform distribution between $[-1, 1]$ and can simply use W to scale the disorder[20], which means that producing a phase diagram for a *specific* set of disorder configurations (at a later stage) will be easier. A list of all combinations of $N/2$ 0's and $N/2$ 1's is created, and a list of possible transformation rules is instantiated based on the coupling. For example, suppose we have an 8-site Heisenberg system with the one state labelled as $\{1, 1, 1, 1, 0, 0, 0, 0\}$; then this state can evolve to the state $\{0, 1, 1, 1, 0, 0, 0, 1\}$ or $\{1, 1, 1, 0, 1, 0, 0, 0\}$ with equal probabilities (where again we use $0 \leftrightarrow -1/2, 1 \leftrightarrow 1/2$ for ease of reading). The random magnetic field is then scaled to appropriate disorder and added to the Hamiltonian. This Hamiltonian's eigenvalues and eigenvectors are computed. We then 'cut the chain' in half, choosing half of a connected length of the chain to *be* the subsystem, provided that this partition itself has no net spin, as mentioned in the previous paragraph – translation invariance[21] ensures that any choice of subsystem is acceptable. The remainder of the system is traced over to produce the reduced eigenvectors, and these reduced eigenvectors are used to produce the reduced density matrix, ρ_A. The eigenvalues of the reduced density matrix are computed and, finally, we take the negative of their natural logarithm to generate the entanglement spectrum. Due to some eigenvalues of the reduced density matrix being very near zero, there can be floating point errors that cause these values to be very slightly negative. As these reduced density eigenvalues are so small as to contribute nothing physical to the system, they are ignored – this is implemented generically by only considering the largest 50% of a given entanglement spectrum. Additional modifications are made to the data for memory optimization and to label the entanglement spectrum with its associated eigenenergy and disorder. Thus, for an 8-site system, when partitioning the system into two equal length systems, A and B, tracing out B and finding the entanglement spectrum at a given eigenenergy and disorder (and labelling the spectrum with these values), we get a

[20] This is equivalent to *defining* the h_i as *always* being randomly sampled from a uniform distribution $[-1, 1]$ and then letting the disorder parameter, W, be a constant that multiplies into the last of the sums of (2.2)-(2.5).

[21] Obviously this is not *true* invariance due to the local h_i, but because these data are randomly sampled the inability to know *a priori* which particular section of the chain constitutes the subsystem is sufficient for any net-spin-0 selection to be valid.

list of labelled entanglement spectra[22] that will generically look something like

$$\{W, \lambda_{\text{energy}}, \lambda_1, \lambda_2\lambda_3\} \overset{eg.}{=} \{0.25, -3.71, 6.15, 2.31, 0.41\} \tag{4.1}$$

where the λ_i are the ordered (largest to smallest) upper 50% of the entanglement eigenvalues for a particular random static field configuration at disorder $W = 0.25$ (and where some rounding has been performed on this example data for readability). A trained MLP, in this example, would only be given $\{6.15, 2.31, 0.41\}$ as input to be classified. The output data will be a 2-component vector (v_1, v_2) with constraint $v_1 + v_2 = 1$ that describes with *what* probability the network classifies the state as being in the ETH phase (v_1) or the MBL phase (v_2). During training we need a way to specify what the output of the network *should* be – to do this we make two critical assumptions.

Critical Assumptions

For the disordered Heisenberg spin-1/2 chain:

- Data generated from disorder values of $W \leq 0.25$ in the central 80% of eigenenergies obey the ETH. These data have target values of $(1.0, 0.0)$.
- Data generated from disorder $W \geq 12.0$ are ETH-violating and fully MBL. These data have target values $(0.0, 1.0)$.

Notice that no assumptions are made for the intermediate disorder $0.25 < W < 12.0$.

For an *untrained* MLP, the input data is given with its appropriate target value. That is, the input to the MLP appears as

$$\{6.15, 2.31, 0.41\} \overset{\text{target}}{\longrightarrow} \{1.0, 0.0\}, \tag{4.2}$$

where this is an example of an ETH-obeying state (*i.e.* a state sampled from $W = 0.25$). Recall now that we have also implemented a confidence penalty scheme – we wish the network to avoid 'agnosticism' in its classification. To implement this, we randomly generate entanglement spectra sampled from the intermediate disorder region $0.25 < W < 12.0$ such that every time a supervised input is given with its appropriate target, so too is an unsupervised input with *no* target value. Thus, for any single unbatched input, the MLP will take in a composed object of

[22] Labels are **not** given to the ANN, but are included so that plotting of data is possible.

these elements. An example of the full (unbatched) input to the network for an 8-site system is thus

$$\begin{aligned} \text{Supervised: } \{6.15, 2.31, 0.41\} &\xrightarrow{\text{target}} \{1.0, 0.0\}; \\ \text{Unsupervised: } \{5.28, 3.90, 1.84\}&. \end{aligned} \tag{4.3}$$

The final restriction we place on the training data is that it must come from the central 80% of the energy eigenvalues, as the system behaviour is known to deviate strongly at extremal eigenenergies [15].

We have now presented the form of the network used in training and the training data. All that remains is to describe how this network performs under various training procedures.

4.1.3 Activation Functions

As this work is strongly motivated by work done by Schindler *et al.* [15] and Luitz *et al.* [7], these works will serve as a measure for the quality of our network[23].

We will use a 12-site system to compare the various activation functions. To perform this comparison, a network is trained in precisely the same way for each activation function and we compare the ETH/MBL phase diagrams that they produce when presented with the same disorder realisations.

Training was performed over 500 rounds using stochastic gradient descent with batch sizes of 100. An initial learning rate of $\lambda_l = 0.0001$ was chosen, as this was the largest value over which none of the activation functions tested encountered unexpected divergences. After training, each network was tasked with classifying (unseen) states generated from disorders of $W = 0.25$ and $W = 12.0$.

Critical Definition:

We define a state to be classified by the network as in the ETH(MBL) phase if the network is over 90% confident in this classification.

The latter (MBL) case was not at all enlightening; after 18480 classifications – that is, 20

[23] [7] will serve as a numerical goal we wish to approach in some way; [15] serves as a proof of concept of the network and these results should be replicable; both provide qualitatively similar results that we consider desirable for the network to reproduce. The results of Šuntajs *et al.* from [8] also serve as an approximate numerical goal.

full classifications of the 924 eigenenergies' entanglement spectra or, phrased differently, as a classification of all entanglement spectra of 20 distinct disorder realisations at $W = 12$ – all the networks classified the states as MBL for *every* state[24]. The variance of these systems was also negligible ($< 10^{-4}$) and so we are forced to abandon the thought of using the MBL states for comparison of activation functions. The ETH systems were marginally more interesting for study when considering full spectra[25].

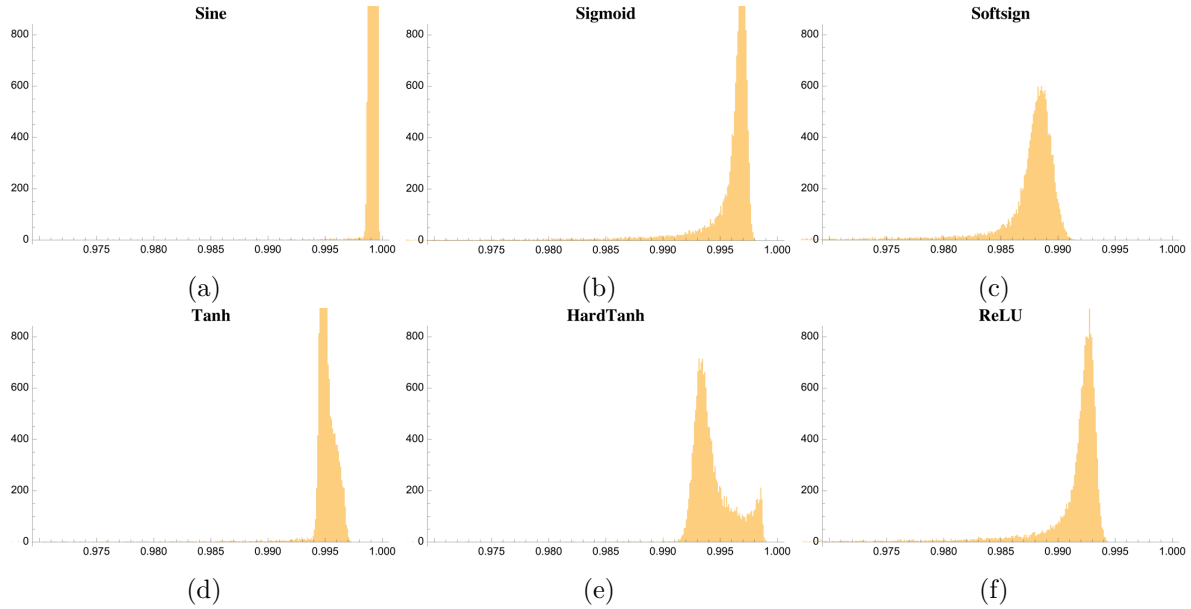


Figure 11: Activation function classification comparison for ETH data ($W = 0.25$). Each graphic is a histogram over the region $[0.97, 1]$, where the width of an interval is 0.0001. The x-axis is the confidence of classification as ETH and the y-axis is a count of the number of classifications made within a given confidence interval width.

The first observation one makes when considering figure 11 is that these classification confidences are all around very high values.

The first observation that one can make from the data in 1 is that all of the activation functions successfully perform the classification task. It is interesting to note that while the sine function is periodic and is therefore often taken to be an inappropriate choice for an activation function, it does appear to be a viable option for this classification task. Nonetheless, to avoid any potential unexpected issues, we do not use it[26].

[24] That is, 18480/18480 states were classified (by our definition) as being in the MBL phase. This procedure was repeated and checked for errors several times.

[25] As with the MBL case, however, when confining the test to the central 80% of eigenenergies and their associated entanglement spectra the network classified 100% of the data as ETH, regardless of the activation function used.

[26] Justifying the use of the sine function is another research topic in and of itself, and it is unclear (at the time of writing) what issues may have arisen had we chosen this as our activation function.

Table 1: Classification results for 20 disorder realisations in the ETH regime. As per our assumptions, we should expect that almost all the entanglement spectra generated in this regime are classified as ETH. The second column is what percent of all given states were classified as ETH by our critical assumption rule (confidence $\geq 90\%$). The remainder of the data are the statistics after classifying 18480 entanglement spectra taken from $W = 0.25$.

Activation Function	% ETH	Mean Confidence	Median	Variance
Sine	98.4	98.7	99.9	1.0
Sigmoid	95.8	97.4	99.7	1.4
Softsign	96.4	97.0	98.8	1.0
Tanh	98.0	98.3	99.5	0.8
HardTanh	97.6	97.7	99.4	1.4
ReLU	97.9	98.3	99.2	0.4

The sigmoid (11b), softsign (11c) and ReLU (11f) all produce qualitatively similar results. Hardtanh (11e) behaves in an unexpected way, with its extended right tail leading to a second peak nearer to 1. Tanh (11d) also has an extended base which is not easily understood.

While these results are inconclusive, we elect to use the softsign activation function. The reason we choose this function compared to some of the other, arguably better activation functions, is because it was the only activation function that was robust against divergences during training. This meant we could reliably train the network with more input data and over more training rounds without the training run being stopped early due to a divergence. The other feature it has that makes it a desirable choice is a more *natural* classification peak[27]; we should expect that the network classifies data that is dominated by some form of behaviour that we would not expect to spontaneously appear but that there should be 'symptoms' of the behaviour before the behaviour is realised, and this is true even more so in the case of an effective averaging procedure. Thus, due in part to its demonstrated naturalness and, more importantly, due to the robustness it demonstrated during training, we choose the softsign activation function for the network.

As a final comment on the activation functions shown in this section, these activation functions are being used to train networks with other optimization features such as weight decay and confidence optimization. The activation functions' behaviour may not be (and in fact, was observed *not* to be) the same without these additions. However, in designing a network a researcher attempts to simultaneously optimize all features to meet the goal of the network, so

[27] *Naturalness* is not an uncommon concept in physics – discussions of naturalness of the sizes of coupling parameters in the standard model are known to most physicists.

we do not consider the behaviour of these functions in a network *without* the aforementioned optimizations.

4.1.4 Effect of Weight Decay and Confidence Optimisation

We briefly demonstrate the effects of including weight decay and a confidence penalty in the network. We do this by training networks on the same 12-site training data over 500 rounds using stochastic gradient descent with all parameters identically chosen but for the confidence penalty 'weight' (γ) and the weight decay 'strength' (β) (refer to (3.45)). The confidence penalty weight gives the relative strength of the confidence penalty loss as compared to the cross-entropy loss; if $\gamma = 0$ the confidence penalty is turned off and does not contribute to the total loss function. The weight decay parameter, β tells the network how quickly individual neuron weights in the hidden layer should decay during training and is essentially a penalty for overfitting; $\beta = 0$ turns this penalty off while non-zero β enforces L_2-norm weight decay.

Using our critical assumptions, we binarize classifications of the entanglement spectra $\boldsymbol{x}_{ES}$ such that

$$p_{\mathrm{ETH}}(\boldsymbol{x}_{ES}) \to \begin{cases} 1, & p_{\mathrm{ETH}}(\boldsymbol{x}_{ES}) \geq 0.9; \\ 0, & \text{otherwise.} \end{cases} \tag{4.4}$$

In order to see a 'smooth' transition region in terms of a temperature/heat map, we average over the results of classification of 5 distinct disorder realisations at each disorder step. Thus, the results displayed in figure 12 are averages taken over 5 realisations when increasing the disorder of each system from $W = 0.25$ up to $W = 4$ in steps of $\Delta W = 0.0625$.

Figure 12 very clearly demonstrates the effect that including weight decay and confidence penalties may have on the final classification scheme derived from the trained network. There are two primary effects that can be observed by varying the hyperparameters γ and β. First, implementing weight decay 'pushes back' the mobility edge; we see the edge somewhere near $W = 4$ in 12a and this edge shifts to near $W = 3$ for $\beta = 0.5$ in 12b. Finally, it is tentatively pushed back to $\sim W = 2.5$ in 12c, however there is a large region of uncertainty in classification that extends beyond $W = 4$. The confidence penalty hyperparameter γ has the effect of reducing this classification agnosticism near the transition region and, again, 'pushing back' the mobility edge by an amount that is small compared to the effect of weight decay. We see this by noting that the region of uncertainty in 12c that stretches above $W = 4$ is narrowed significantly in

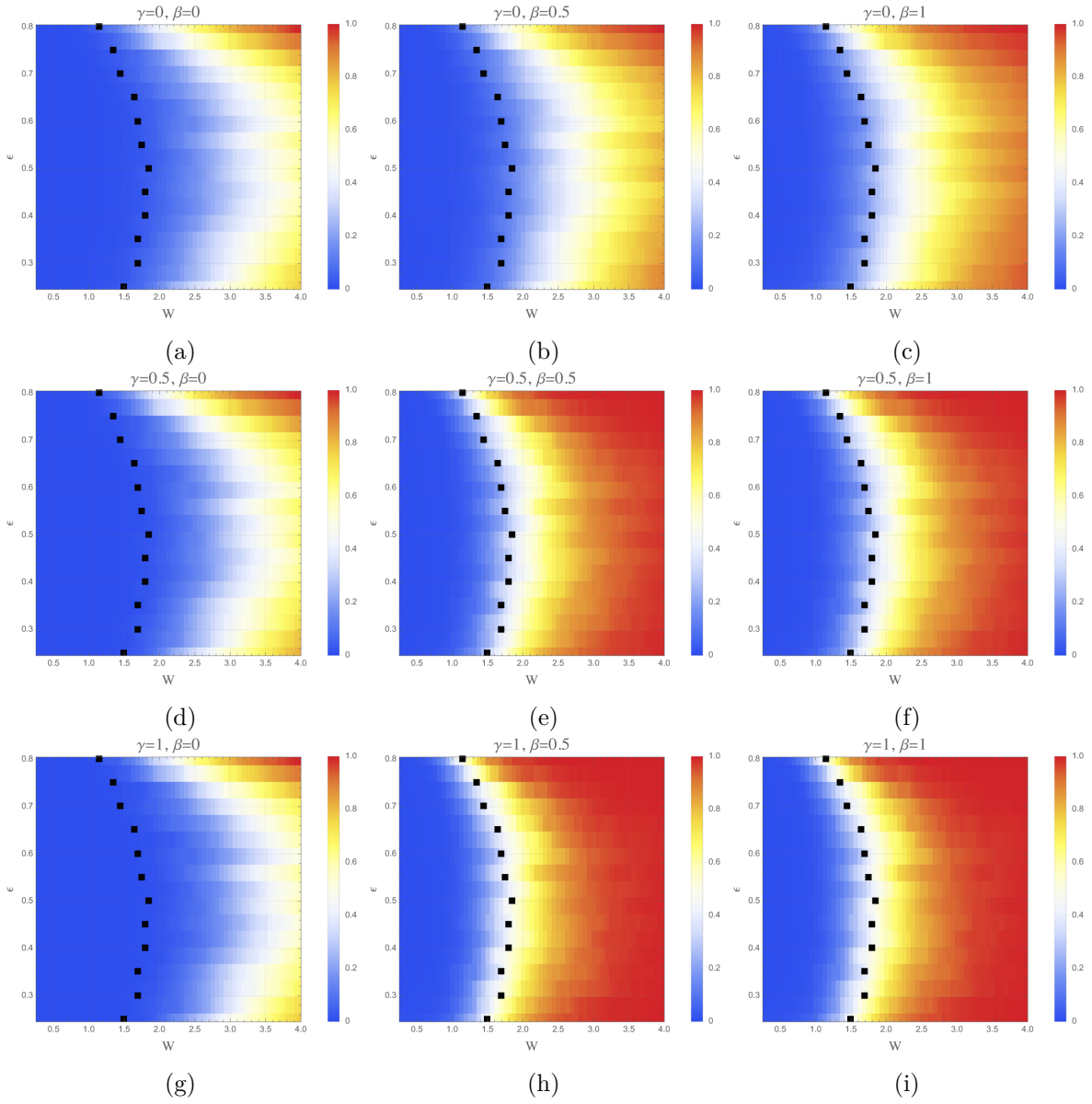


Figure 12: A graphical confidence optimization - weight decay 'matrix' for $N = 12$ site classification. A given row of images corresponds to having the same confidence penalty weighting (γ), whereas a given column has a fixed weight decay parameter (β). The black squares are data values for which the behaviour of entanglement entropy changes from volume-law to area-law proportionality in numerical simulations done by Luitz *et al.*, and represent numerical values the network should approach in a meaningful classification scheme [7]. The x-axis is for the disorder strength, and the y-axis is for a normalised and zero-centred energy; $\epsilon = (E - E_{\text{min}})/(E_{\text{max}} - E_{\text{min}})$, for energy eigenvalue E having maximum energy eigenvalue E_{max} and minimum energy eigenvalue E_{min} at a given disorder. These ϵ values are binned with bin-'widths' (heights) of 0.035. The heat map shows the network's confidence in an entanglement spectrum at a given disorder and ϵ value being classified as MBL (to be explicit, for the data shown: blue $\leftrightarrow$ ETH, red $\leftrightarrow$ MBL).

12f and is narrowed and pushed back further still in 12i.

The overall effect of including weight decay is thus to strongly push back the mobility edge to lower disorders, where the inclusion of a confidence penalty pushes back the mobility edge more

(but not as much as weight decay) and reduces the classification agnosticism region, that would extend far from the disorder of the mobility edge, otherwise. Increasing γ and β beyond values of ~ 1 produces no observable benefit and can sometimes cause divergences during training. However, we see that with values of $\gamma = \beta = 1$ we are able to approximate the ETH/MBL transition region as found by Luitz *et al.* reasonably well[28].

Before continuing, it is also useful to introduce another way to represent the classification data the neural network produces in a way that highlights the transition region. We can do this by focusing on entanglement spectra that the network classifies as neither being ETH nor MBL. Recalling our critical assumption, this is tantamount to highlighting states with an ETH confidence between $(0.1, 0.9)$. Producing a graphic in this way, one can make observations about the *extent* of the region of uncertainty in classification (in other words, one can observe the width of the transition region, where the network classifies states with minimal confidence, more easily). One can use a similar averaging procedure as used to generate 12 to smooth-out the results (as well as to justify the results more generally). In practice this is implemented by enforcing a binary scheme; for a given ETH classification probability $p_{\text{ETH}}(\boldsymbol{x}_{ES})$ of the entanglement spectrum $\boldsymbol{x}_{ES}$

$$p_{\text{ETH}}(\boldsymbol{x}_{ES}) \to \begin{cases} 0, & p_{\text{ETH}}(\boldsymbol{x}_{ES}) \leq 0.1 \quad \text{OR} \quad p_{\text{ETH}}(\boldsymbol{x}_{ES}) \geq 0.9; \\ 1, & \text{otherwise.} \end{cases} \tag{4.5}$$

The approximate smoothness of the the images will come from averaging over realisations. Indeed, when considering figure 13, we can see very clearly how including a confidence penalty 'tightens' the width around the transition regions and contributes to minimizing the region of uncertain classification. Figure 13 thus re-confirms our initial assessment that the confidence penalty reduces the mobility edge's width and contributes (but less so than weight decay) to a pushed-back mobility edge.

4.2 Initial Results

At this stage, we have compared activation functions and we have compared the effects of implementing a confidence penalty and weight decay during training. We did this by training many networks to classify 12-site entanglement spectra over 500 training rounds. Now, however,

[28] It is also interesting to note that in [15] Schindler *et al.* made similar observations for their hyperparameters.

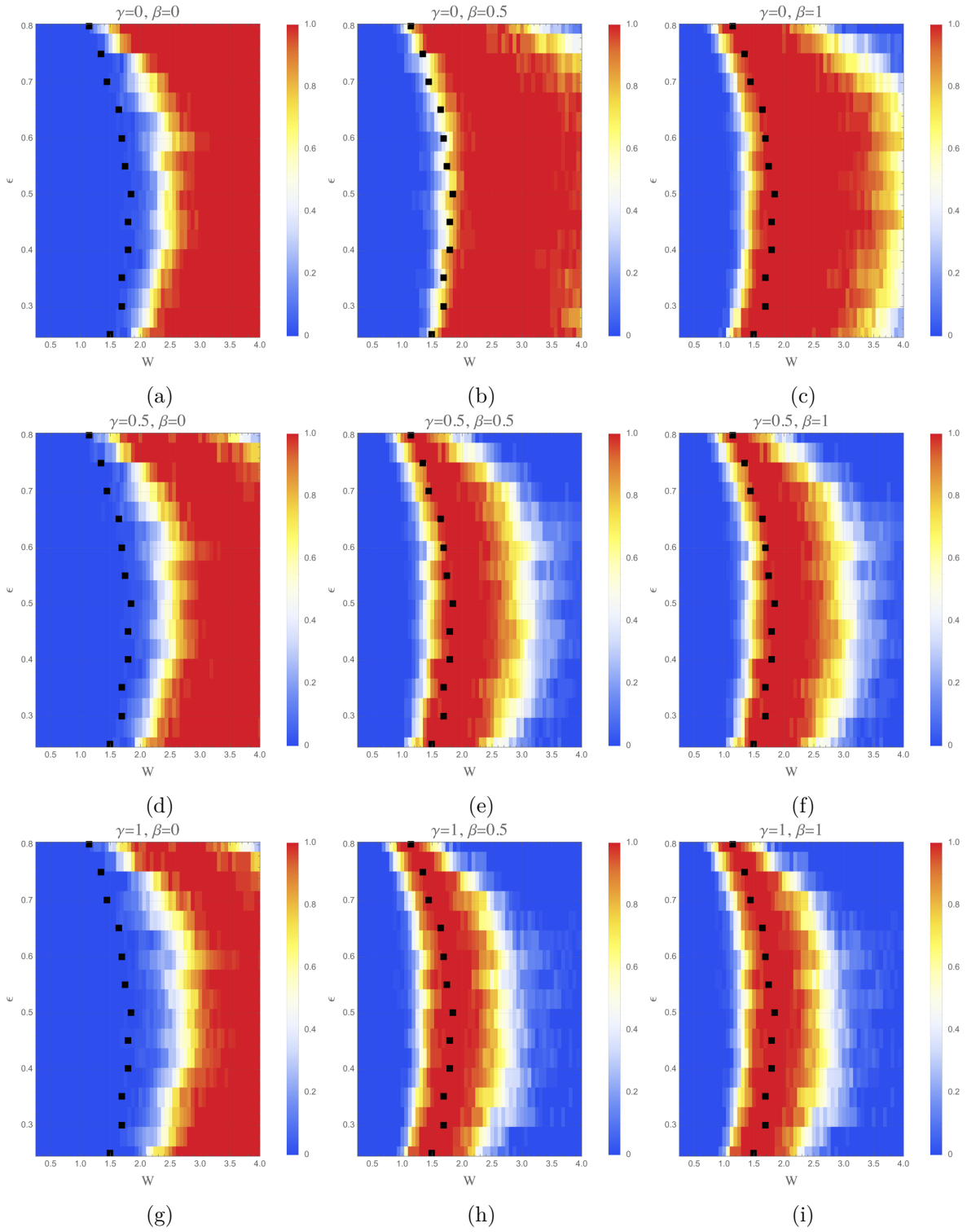


Figure 13: A graphical confidence optimization - weight decay transition region 'matrix' for $N = 12$ site classification. Using this method of displaying data, it is more easily observed that the effect of confidence optimisation is to reduce the uncertainty classification region of the network.

we have chosen an activation function and the hyperparameter values we will use. Therefore, we are now in a position to perform an unconstrained training – we train two networks, one for an

N=12 site systems and another for N=16 site systems. In this subsection we will consider only the disordered Heisenberg spin-1/2 chain as our system with $N = 12, 16$ sites. For this training procedure we produce 100 realizations of data; 50 produced with disorder values $W_{\text{ETH}} = 0.25$ and another 50 produced with disorder values $W_{\text{MBL}} = 12$. These realizations, according to our critical assumptions, all obey the ETH (for $W = 0.25$) or are all many-body localizing (for $W = 12$). This data is thus labelled with a target value of $(1, 0)$ for ETH entanglement spectra and $(0, 1)$ for MBL entanglement spectra. We take the central 80% of energy eigenvalues associated to a given disorder value, as the behaviour at the extremal eigenenergy values may differ. For $N = 12$, this means we consider 721 of the 924 entanglement spectra at a given disorder value for a single disorder realization. Of these, we randomly sample 18400 data from each disorder value, randomly arrange all 36800 entanglement spectra, then partition this set into two sets of equal size – the first set of the partition is taken as the test data that will be used at the start of a training round; the second set of the partition is chosen to be a validation set to help prevent overfitting during training. For $N = 16$, we consider 10297 of the 12870 entanglement spectra at a given disorder value for a single disorder realization. We do as we did in the $N = 12$ case, but this time the random sample contains 500000 total entanglement spectra (250000 from each regime). This data constitutes our supervised data.

We also need unsupervised data. This data comes from 50 disorder realizations for W randomly sampled from $(0.25, 12)$. These data are not labelled as the classification assumptions we made do not apply to them. We randomly sample 36800 (500000) entanglement spectra from these *intermediate* $N = 12$ ($N = 16$) entanglement spectra.

As was justified previously, we choose the confidence penalty and weight decay hyperparameters to be $\gamma = 1$, $\beta = 1$. In order to end the training, we require that the loss during a training round to be unimproved on average over a period of 50 training rounds. We use stochastic gradient descent as our training optimization procedure.

After training, unlike the previous cases where we used only 5 unique disorder realisations to average over in producing an image, we now take 48 unique disorder realisations to be averaged over. This is to reduce the effects of anomalous sampling causing a shift in the mobility edge. The process of classifying all these states and compiling an image has a very large memory

requirement[29]

Figure 14 allows for a direct comparison in four ways.

- **System Size - Uncertainty Region Relation:** When moving from the $N = 12$ to $N = 16$ system, we obtain a better resolution of the transition region in spite of the fact that the disorder step sizes for the $N = 12$ site system were $125\times$ smaller than that of the $N = 16$ site system. This behaviour is expected to continue with larger system sizes, but could not be tested with sufficient resolution for the next system size up ($N = 20$) due to computational constraints.

- **Neural Network Comparison:** As this work is strongly motivated by work done in [15], it is natural to compare the results produced by the network used for that work. We find that our network pushes the transition region back more strongly when comparing 14b and14d to data presented in figure 1 of [15] (the work from which the black circles are sampled). While this is a potential source of concern, it is mitigated by the fact that, qualitatively, the classification results are similar – a slightly down-shifted peak for the transition region and greater width of uncertainty near the central ϵ values of the transition region (not shown in figures) is the behaviour seen for similar graphics in [15].

- **Numerical Data Comparison:** Numerical simulations performed by Luitz *et al.* in [7] demonstrate what appears to be a strong correlation with the classification produced by the network. We can not make this statement precise due to a lack of access to the data produced in [7], however, we see a stronger agreement with these numerical results as compared to the classifications from the network in [15], so we may be confident in the network's classification efficacy.

- **Ergodicity Indicator Comparison:** In [8], Šuntajs *et al.* found that their *ergodicity indicator*, g, demonstrated a sudden drop when plotted as a function of system size over the Berezinskii-Kosterlitz-Thouless (BKT) correlation length, N/ξ_{BKT}. While this work does not explicitly explore this indicator, we make use of the observations in [8] to note that there is large observed overlap between this region and the transition region at central eigenenergies. This indicator and, more specifically, its application in [8], is also useful

[29] Even with the code written to minimize the required memory it still exceeded 128GB of Random Access Memory. This seems an apt point to, again, thank the University of Cape Town's High Performance Computing Facility for use of their clusters.

as it is one of the very few simulations that produces measurable transition regions for a $J_1 - J_2$ model; a generalisation of the Majumdar-Ghosh model.

Before proceeding with the classification on systems that are topologically distinct from the Heisenberg spin-1/2 chain, we introduce another way of graphically representing the classifications of the network.

We see, again, in figure 15 that the results of Schindler *et al.* demonstrate a transition region at higher disorders than we find using our network. We also see the same reduction in width with an increase in system size using this presentation. A key qualitative difference when using this eigenvalue-binned presentation is that the transition region appears more plateaued than in the ϵ presentation.

We have seen that our network produces results that appear to be qualitatively consistent with results found in previous work and they appear to reproduce numerical results – we refrain from making this statement definite due to a lack of access to the numerical values of [7]. With these results in mind, we proceed to attempt an analysis of systems that the network has had no training with whatsoever, and analyse the results of classification in these unseen topologies.

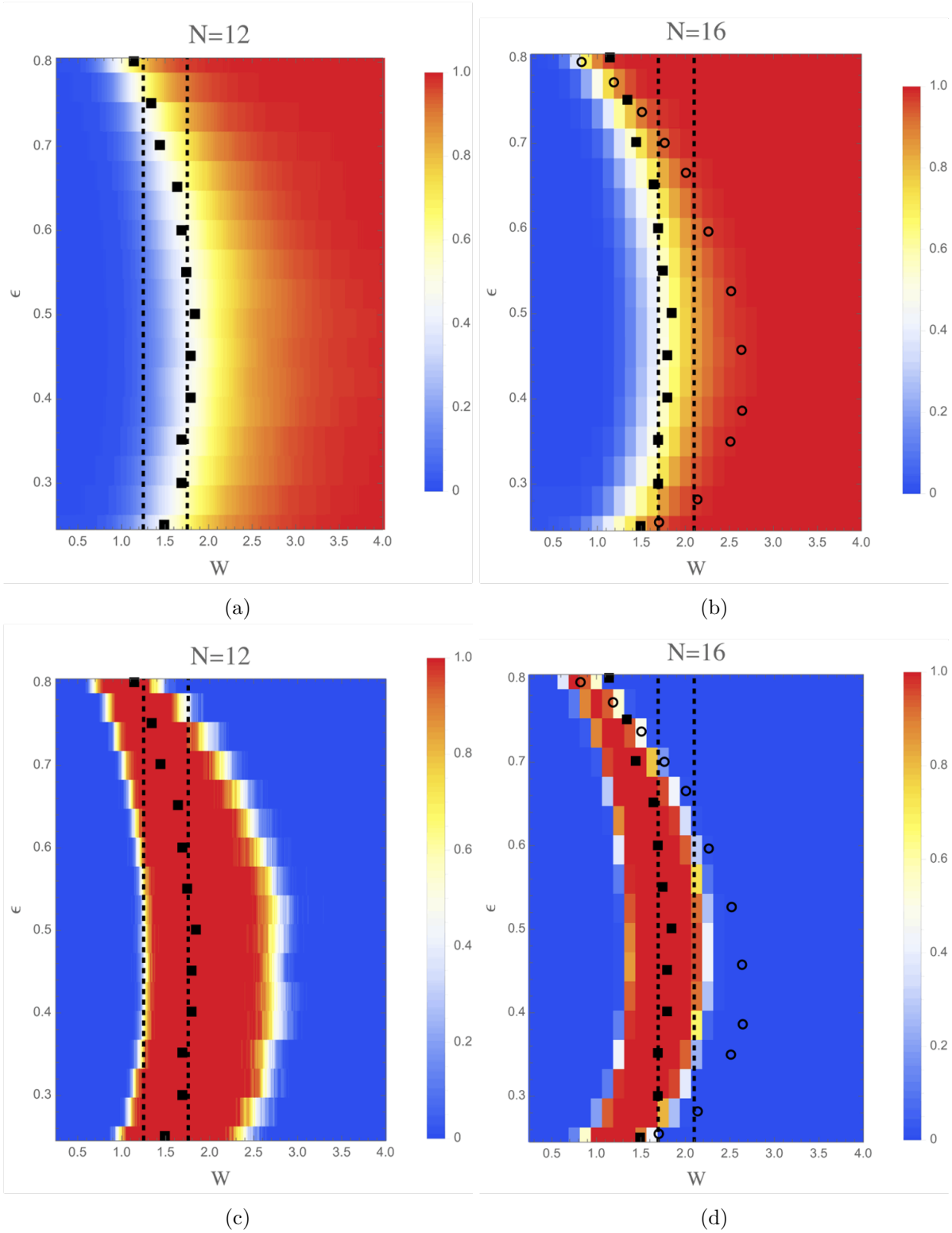


Figure 14: Results of classifications averaged over 48 disorder realisations. The $N = 12$ case (a,c) use disorder steps of $\Delta W = 0.01$, while the $N = 16$ case uses $\Delta W = 0.125$. The heat maps (a) and (b) are the averaged confidence of being in the MBL regime at a given disorder and normalized-shifted energy ϵ, as in figure 12. The heat maps in (c) and (d) are intended to highlight the region in which the network is least confident (classification confidence $< 90\%$ for either regime). The solid square black markers are results sampled from Luitz *et al.* in [7]. The open black circle markers in (b) and (d) are approximate samplings from data produced by Schindler *et al.* in [15]. The vertical dashed lines enclose the region in which Šuntajs *et al.* found that their 'ergodicity indicator' sharply declined [8].

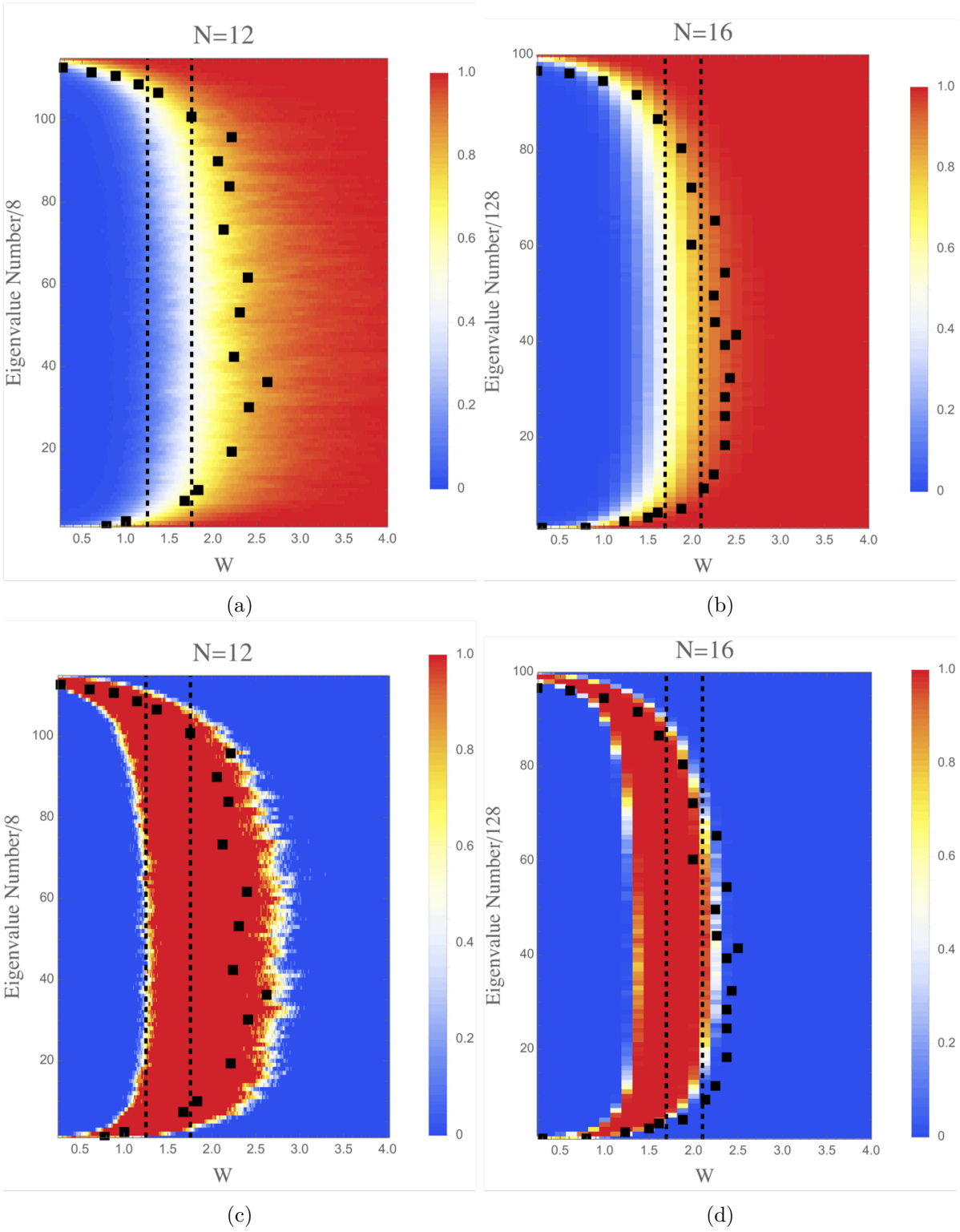


Figure 15: Results of classifications averaged over 48 disorder realisations in an eigenvalue-binned presentation. These graphics are taken from the same data as in figure 14, however, instead of using the ϵ parameter, we choose to bin the results of the ordered eigenenergies in bins consisting of 8 consecutive eigenenergies for the $N = 12$ system, and in bins of 128 consecutive eigenenergies for the $N = 16$ system. The solid square black markers are results sampled from Schindler *et al.* in [15]. The vertical dashed lines are the same as those in figure 14 [8].

5 Altering the Topology

We have shown that the network can reliably classify data for a disordered Heisenberg spin-1/2 chain. We now consider three different systems: the Majumdar-Ghosh model, the bicycle wheel model, and the star model (see figure 1 and equations (2.3)-(2.5)). In using the network to classify the entanglement spectra of these models, we move into a regime of classification for which the network has received no direct training. In these systems we use only eigenvalue number binning. We choose this *over* the ϵ-binning introduced and used in figures 12-14 because of a potential binning issue; in order to compute the average of an ϵ bin one needs at least one data point to lie within that bin-width. However, in these *new* topologies, we do not necessarily have eigenvalues sufficiently evenly spaced for this binning procedure to be free of an empty bin. Furthermore, trying to increase the ϵ bin width inevitably leads to a loss of resolution of the transition region, which does not appear to spontaneously change its behaviour for all eigenenergies at a given disorder (that is, we do not see 'strong' ETH). Fortunately, regardless of the eigenenergy spacing, we are always able to know *how many* eigenenergies there are. This makes the eigenvalue number binning procedure a more robust tool in probing these novel topologies.

5.1 The Majumdar-Ghosh Model Results

The simplest generalisation of a disordered Heisenberg spin-1/2 chain is to extend the coupling from strictly nearest-neighbour to nearest-neighbour *and* next-to-nearest-neighbour coupling. The Majumdar-Ghosh model has coupling between next-to-nearest-neighbours that is precisely half the value of the nearest-neighbour coupling, and we will add the disorder at each spin site just as we did in adding disorder to the Heisenberg spin-1/2 chain. We would like the various systems to be directly comparable, with their only difference being in the additional coupling. Thus we use the same unique 48 disorder realisations to produce averaged-over phase diagrams as were used in generating figure 15. The Majumdar-Ghosh model is a special, exactly solvable case of the $J_1 - J_2$ model; the $J_1 - J_2$ model is more general in that the coupling to next-to-nearest-neighbours is not precisely 1/2 the value of the nearest-neighbour coupling, but is still a constant value. One key observation to make at this point is the relative locations of the transition region 'peaks' and their respective system sizes in the context of the data from [8].

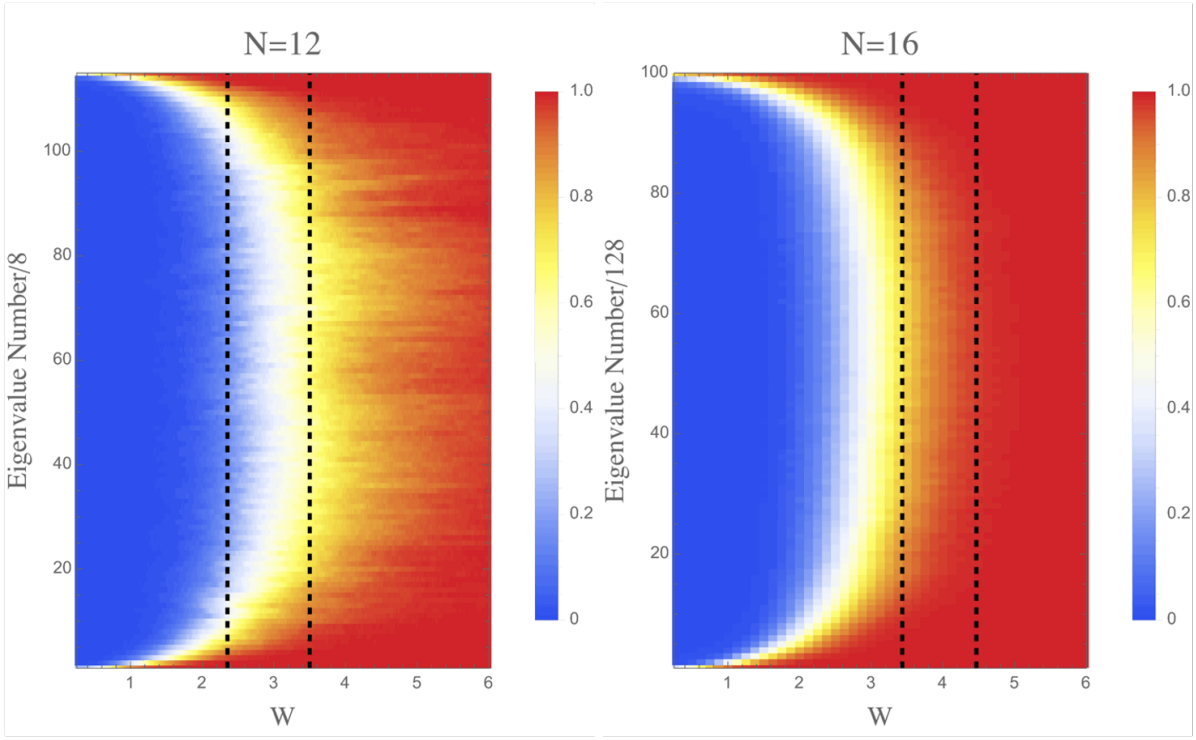


Figure 16: Results for classification of entanglement spectra from the Majumdar-Ghosh model over 48 unique disorder realisations of the $N = 12$ and $N = 16$ size systems, respectively. Notice the disorder width has increased to show values $W \in [0.25, 6]$ to make the edge clear for both cases. The vertical dashed lines are, again, results sampled from [8].

To do this, we contrast the placement of the vertical dashed lines in figure 15 and figure 16. Qualitatively, we see very similar results; in 15 the $N = 12$ transition region has the predictions of [8] being almost aligned with the low-disorder transition region at central eigenenergies, which persists into the central disorder values of the transition region. Similarly, for the $N = 16$ cases we see similar idiosyncrasies; the region enclosed by the dashed line predictions of [8] encloses the upper central region of the disorder. There is a noticeable shift in the relative locations of the dashed lines in 16 when compared to 15. This is easily understood; in producing the results in [8], Šuntajs *et al.* made the choice of coupling to be $|J_1| = |J_2| = 1$, where the Majumdar-Ghosh model uses $|J_1| = 2|J_2| = 1$ (which applies to the data we use). Increasing the coupling strength to the next-to-nearest-neighbours is effectively increasing the strength of coupling of the subsystem we consider (half of the total system with net spin of 0) to the heat bath (the other half of the system that is acting as a thermal reservoir). An increase in the strength of coupling to the thermal reservoir will necessarily increase the range of ergodicity, as stronger disorders will be necessary to cause localization. Thus, we should be encouraged to see this small relative shift in the vertical dashed lines when going from figure 15 to figure 16.

These results are extremely encouraging; with no training on this system, the network appears to classify states successfully and with an apparently good agreement with results that make use of an ergodicity indicator this network has never directly encountered.

5.2 Prediction: The Bicycle Wheel Model Results

The network, having demonstrated some extensibility to the modified problem of the Majumdar-Ghosh model, is now tasked with classifying a model for which we do not have any numerical results to work with in any form – there is no image with which we may hope to qualitatively compare the results and so we are now in the realm of prediction and conjecture. In spite of an overall lack of numerical/graphical data for this system (and the star model, too), we have shown that the network produces results that share a strong resemblance to the numerical results of [7], that are qualitatively similar to the results of [15], and that demonstrate predictable agreement (with an expected shift) to results in [8].

We thus proceed to let the network classify two additional systems and claim that these predictions will be found when future numerical work is done to compute the transition region. These numerical computations are outside what is computable given the resources at hand when

producing this work[30] The prediction in figure 17 aligns with heuristic expectations; the system has nearly double the coupling density – by choosing a central site and connecting it to all other sites we have increased the effective coupling to the part of the system we take to be the thermal reservoir (the half of the system that was traced over). This increased coupling would lead to an increase in the resilience of the system to withstand the influence of including disorder, so that the system only localizes at higher disorder values, which is precisely the behaviour demonstrated in 17.

The qualitative behaviour of the extremal eigenenergy regions also agrees with the behaviour found in the disordered Heisenberg spin-1/2 system; long 'legs' of classification uncertainty extend deep into the extremal eigenenergy/minimal disorder region. These features are encouraging, and serve to partly justify our claim that the network's classification of this system is qualitatively accurate, and is likely numerically accurate as well.

5.3 Prediction: The Star Model Results

As in §5.2, we now claim the network predicts the transition from the ETH regime to MBL regime as a function of the disorder for the star model. Prior to computation, it is useful to consider what we might expect; the star model is not as connected as the bicycle wheel model, but the separation between any two spin sites is maximally 2 lattice steps. Thus the coupling to the heat bath should certainly be stronger than that of the disordered Heisenberg spin-1/2 system initially considered, but should be less strongly coupled to the heat bath as compared to the bicycle wheel model, whose coupling is a superposition of the star and nearest-neighbour systems. We should reasonably expect that the network places the central transition region of the star system somewhere between the Heisenberg model and the bicycle wheel model.

We see precisely what we might expect; the mobility edges of the star model in figure 18 are placed, as heuristically expected, between that of the disordered Heisenberg spin-1/2 model in figure 15 and the disordered bicycle wheel model in figure 17. We do, however, see one definite numerical artefact in the star model, and another *potential* numerical artefact.

In figures 18c and 18d, we see small 'bubbles' of uncertainty at very low disorders. This is a numerical artefact attributed to the ordering of eigenenergies – an unexpected eigenenergy

[30]The fact that numerical values have not been computed elsewhere for these relatively simple systems may be indicative of the difficulty of computing applicable numerical values (*e.g.* entanglement entropy) for these systems.

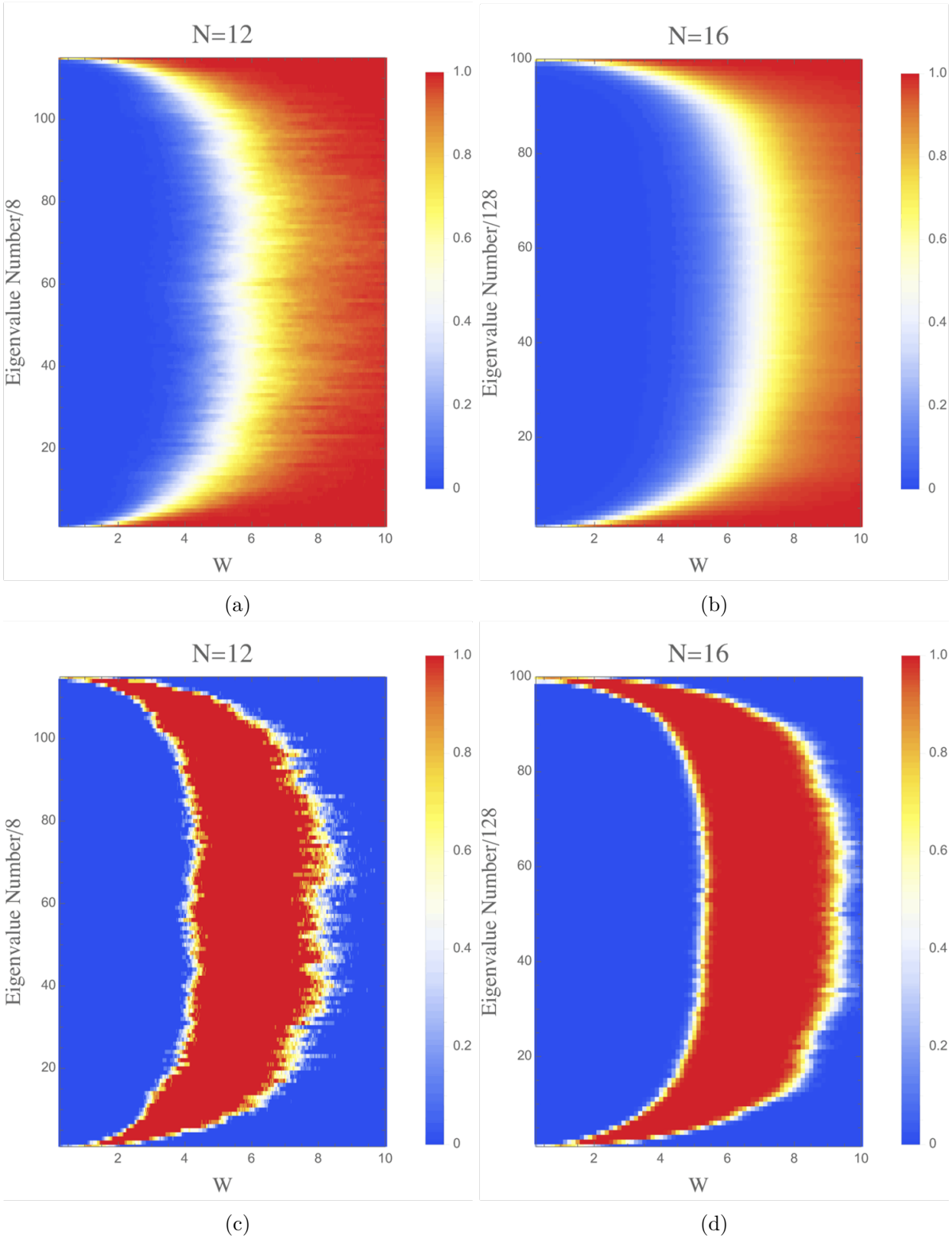
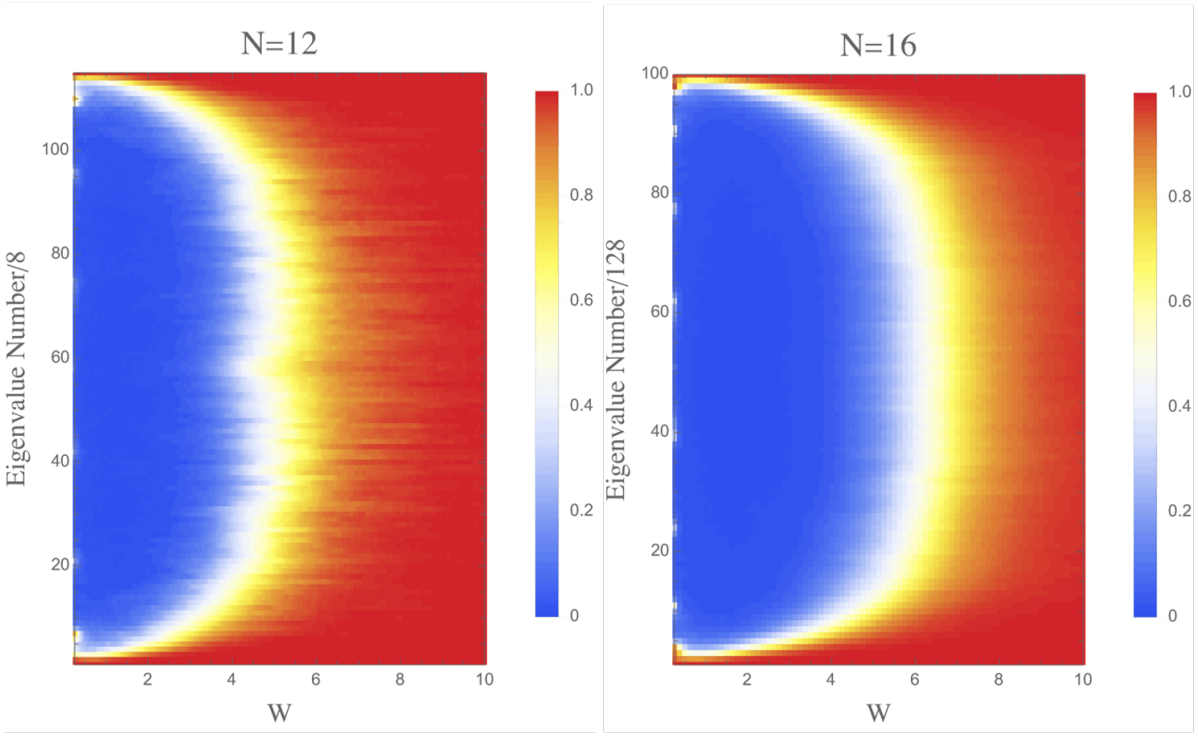


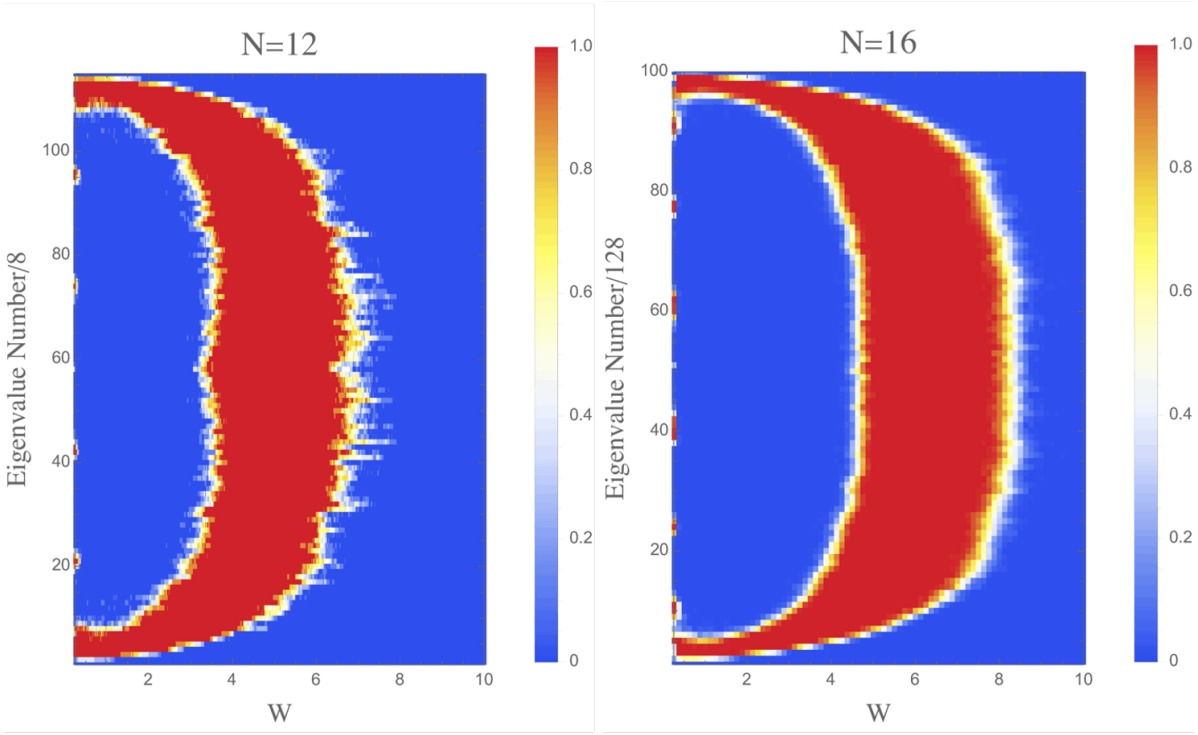
Figure 17: Network predictions of ETH/MBL phase transition as a function of disorder averaged over 48 disorder realizations. Notice the disorder axis now extends as $W \in [0.25, 10]$ in order to capture the full uncertainty region.

degeneracy was found to occur at perturbative values of disorder $W \in [0, 0.3]$ that vanishes for non-trivial disorder.



(a)

(b)



(c)

(d)

Figure 18: Network predictions of the ETH/MBL transition for the star model averaged over 48 disorder realizations. Note that the disorder axis runs over $W \in [0.25, 10]$.

The potential numerical artefact appears most prominently in figures 18a and 18c, but can also be partially seen in 17a and 17c. We are referring to the 'divot' at the central eigenenergy part of the transition region for the $N = 12$ systems. It is unclear whether there is a physical

explanation of this indentation in the transition region, or whether it is a numerical artefact that is a result of considering a very small system. However, this artefact does not detract meaningfully away from the strength of the network as a general predictor of the disorder-eigenenergy 'location' of the transition region.

6 Conclusions and Recommendations

The primary goal of this work has been to demonstrate that artificial neural networks trained on data from a relatively simple system, the disordered Heisenberg spin-1/2 model, could be used to probe more complex models. To do this, we outlined some of the principle physics ideas in §2, built-up the ideas necessary to understand the Artificial Neural Network (ANN) we made use of in §3 before reporting the results of the ANN's classification of the entanglement spectra of our principle system of consideration: the disordered Heisenberg spin-1/2 model. Finally, the extensibility of the ANN's classification capabilities were conjectured, demonstrated and discussed in §5.

6.1 Review

In §2, we introduced the basis and Hamiltonian(s) that we made use of in this work. This discussion was something of a preamble for our discussion of thermalization and what thermalization *is*, which culminated in the Eigenstate Thermalization Hypothesis (ETH). With our description of thermalization in mind, we proceeded to discuss the central focus of this work: systems that fail to thermalize – systems that *localize*. Localization was described in terms of single particle localization and this discussion was immediately broadened to include Many-Body Localization (MBL). The discourse on localization – paired with the goal of this work – necessitated a description of how such localizing states are identifiable. It is at this point where entanglement spectra came to the fore, and its use as a meaningful predictor of the ETH and MBL regimes was justified.

The following section, §3, served to articulate the intricacies of the network in a 'ground-up' approach; if it is pedagogical it is because building the ANN was an exercise in self-learning. The articulation of the components of the ANN followed a chronological/relevance-restricted description, beginning with the original 1943 notions of perceptrons and spanning up to the 1989 multilayer feedforward networks as universal approximators – a key justification for the use of an ANN in this work. This section concluded with a discussion of the training techniques/methodologies and a mathematical outline of their mechanism for 'learning'.

The physical theory outlined and the mathematical intricacies described, the conceptual construction required to understand this work was complete. Section §4 was aimed at explaining

the realization of the ANN that was used in this work (based on the discussion in §3), before providing evidence for the ANN as a tool that could successfully at perform its classification task. We showed that the results produced by our ANN tended to match those of numerical simulations done by Luitz *et al.* in [7] and found the network predicted the transition region to be in approximately the same region as found in work done by Šuntajs *et al.* in [8] – which appeared to be numerically more consistent than results found in a similar ANN classification scheme used by Schindler *et al.* in [15]. These results provided grounds to accept the ANN was indeed performing the task of classification of ETH/MBL entanglement spectra successfully.

The final task presented to the network was a test of its generalizability; the subject of §5. The most basic generalization was to the Majumdar-Ghosh model – a generalization by including next-to-nearest-neighbour coupling that was half the strength of the nearest-neighbour coupling. Numerical results for verification of the ANN's results are not common in the literature. Indeed, there was only one measure of consistency; results for the $J_1 - J_2$ model studied by Šuntajs *et al.* in [8]. However, these results were consistent with the results of classification performed by the ANN. The generalizability was tested further by testing the ANN's classifications of two systems whose structure was much more highly connected than that of the Heisenberg model, namely, the bicycle wheel model and the star model. We discussed how these systems could be expected to behave, and found that the ANN matched this predicted behaviour in its classifications.

6.2 Conclusions and Predictions

The results of this work can be discussed in 3 categories: basic goal, extensibility goals, and predictions.

First, we consider the ANN performing the task it was trained to perform. *Did the ANN learn how to classify entanglement spectra of a disordered spin-1/2 system?* Yes, the ANN performed very well when tasked with the classification of entanglement spectra for systems of the type it was trained on. This assertion is based on three comparisons to existing results. It is unfortunate that *how* well the ANN performed cannot be given a numerical value due to a lack of access to simulated data value results. Nonetheless, the results presented in figure 14 are profoundly striking – the ANN was trained with supervision at two disorder values *only*, $W = 0.25$ for ETH data and $W = 12$ for MBL data – nothing of the intermediate structure was specified during training, only that the ANN should be penalized for a lack of

confidence in the intermediate disorder region, $W \in (0.25, 12)$). That the ANN learned results comparable to values produced in precise numerical simulations is, to use the scientifically imprecise term, 'somewhat astonishing'[31]. The ANN is fed neither eigenenergy data nor disorder data in training; that it identifies them are evidence of it having 'learned' something *about* the structure of the entanglement spectra that encodes which of these entanglement spectra are attributed to ergodic states, and which are attributed to localizing states.

The extensibility of the ANN is the next area of interest. We wished to show that the ANN, trained on a system that *does* have numerical/qualitative results for comparison, could be used as a probe of similar – but nonetheless distinct – systems. As problems within condensed matter physics often require that a 'thermodynamic limit' be taken, the limitations and lack of computability is strongly felt even in systems as small as 22 bodies of, and even for models as simple as the disordered Heisenberg spin-1/2 model [7]. A tool that could reliably probe models that are, for practical purposes, otherwise incomputable, would be an invaluable tool. We found that the ANN produces results that share an agreement with other numerical predictions (up to an expected shift) for the Majumdar-Ghosh model [8]. We took this is a positive indicator that the ANN was successfully classifying these entanglement spectra. This result is somewhere between a prediction and a result – the dashed vertical lines of figure 16 are insufficient grounds to confidently say that the ANN performs classification of the Majumdar-Ghosh model. However, this model, which should have behaviour dominated by its Heisenberg terms but should simultaneously stay ergodic for higher disorders than did the Heisenberg model, *did* produce the expected behaviour according to the ANN's classification. We took this to be a reasonable indicator of the effectiveness of the ANN in the Majumdar-Ghosh case. Thus, we claim that this result should be replicable in numerical experiments.

The next two systems were distinct from the Heisenberg and Majumdar-Ghosh models as they are more strongly coupled to the rest of the system, which acts as a heat bath for the subsystem we consider, and they have increased effective coupling to this heat bath. These models have no readily available numerical results with which we could compare, and so these results, we conjecture, are predictions for the ETH/MBL phase transition in the bicycle wheel and star models. The bicycle wheel model introduced the major modification of a central spin site connected to all others, superposed with a Heisenberg model on the other sites. This model,

[31] We are relegated to making qualitative statements by the computational resources we have access to – which were abundant but still insufficient.

being the most connected of all the models we considered, was expected to have ergodicity that persisted to much higher disorders in the central eigenenergy region than that of the Heisenberg and Majumdar-Ghosh models. Indeed, this is what was found, with a transition region extending beyond disorder values of $W = 8.0$ in figure 17, while still maintaining the characteristic long arched 'legs of uncertainty' down to the extremal eigenenergies at low disorder. We claim this result accurately predicts the shape of the ETH/MBL phase diagram and that numerical simulations should find a similar diagram.

The final model we considered, the star model, has *no* features of the Heisenberg model – the bicycle wheel model had a Heisenberg model superposed with a star. This model represents a fundamental change in the classification requirements of the ANN; to classify it requires 'knowledge' of something universal about the ETH and/or MBL states in these 'very' finite quantum graphs. Heuristic estimates for where the transition region for the star system should be were discussed (between the Heisenberg and bicycle wheel models) and the results of classification were in agreement with this expectation. Given a lack of numerical data, this result is limited to being a prediction at the time of writing. We claimed that these results would also be found when precise numerical experiments could be performed.

Taken altogether, we have four results:

- The trained ANN can classify entanglement spectra as ETH/MBL for the disordered Heisenberg spin-1/2 model very well;
- The trained ANN appears to generalize to the Majumdar-Ghosh model's entanglement spectra for ETH/MBL classification;
- We conjecture that the trained ANN can classify the bicycle wheel model's entanglement spectra, and the classification will be of approximately the form shown in figure 17;
- We conjecture that the ANN can classify the star model's entanglement spectra, and that the classification will be of approximately the form shown in figure 18.

Claiming the ability of the ANN to generalize may seem speculative, but Schindler *et al.* showed that an ANN could learn the entanglement spectra 'shape' [15]. We suspect that our ANN has learnt something about the entanglement spectra and their relation to random matrix theory, described by Yang *et al.* in [34]. A proof of this suspicion is a notoriously subtle task and beyond the scope of this work.

Artificial neural networks can be incredibly powerful tools – understanding *what* it is they 'learn' is tantamount to understanding a problem at what is often a much deeper level than is perhaps already understood by the scientific community as a whole. In some ways this can feel like 'cheating science' – the ability to use an ANN to provide *an* answer when one does not yet know the answer is not a conventional approach to research, and flies in the face of evidence-based science at first sight. However, this is not what this work seeks to promote. Instead, in much the same way that experimental physicists produce data that theoretical/mathematical physicists may use for guidance of their research, so too can artificial neural networks, especially in situations where numerical simulations can be time-consuming or, perhaps, incomputable.

6.3 Future Directions

This thesis has two areas in which future work is both possible and an exciting prospect.

The first is the numerical verification of the results produced in this work. Verification that the ANN's predictions for the Majumdar-Ghosh, bicycle wheel, and star models is a necessity to fully understand the extensibility of the ANN.

The second area for consideration is understanding what the ANN has 'learned'. We strongly suspect that elements of random matrix theory are being learnt by the ANN; proving this and making this result precise is an ambitious but highly-rewarding goal.

There are other potential areas worth exploring, the most obvious of which is enhancing the network, perhaps including additional layers and/or converting it into a deep neural network, which may allow for a better understanding of what it is that the ANN has learned.

As is often the case when training an ANN, we were strongly limited by the data we had access to. We have already stated that it would be ideal to verify the predictions of the ANN, but to take this further, it would be interesting to break down quantum systems into something akin to *simplices* used in topology. Training on these 'quantum graph simplices' and then building theoretical systems out of them (whose properties we can attempt to predict) would definitely be a satisfying and rewarding direction to pursue.

References

[1] W Press et al. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. 3rd ed. Cambridge University Press, 2007. ISBN: 0521880688. URL: http://www.amazon.com/Numerical-Recipes-3rd-Scientific-Computing/dp/0521880688/ref=sr_1_1?ie=UTF8&s=books&qid=1280322496&sr=8-1.

[2] R Burden and J Faires. *Numerical Analysis*. Ninth. Cengage Learning. Brooks/Cole, 2011.

[3] International Human Genome Sequencing Consortium. "Finishing the euchromatic sequence of the human genome". In: *Nature* 431 (2004), pp. 931–945. DOI: https://doi.org/10.1038/nature03001.

[4] E Alpaydin. *Introduction to Machine Learning*. 3rd ed. Adaptive Computation and Machine Learning. Cambridge, MA: MIT Press, 2014. ISBN: 978-0-262-02818-9. URL: https://mitpress.mit.edu/books/introduction-machine-learning-third-edition.

[5] R Nandkishore and D Huse. "Many-Body Localization and Thermalization in Quantum Statistical Mechanics". In: *Annual Review of Condensed Matter Physics* 6.1 (2015), pp. 15–38. DOI: 10.1146/annurev-conmatphys-031214-014726. URL: https://doi.org/10.1146/annurev-conmatphys-031214-014726.

[6] C Beetar, J Murugan, and D Rosa. *Neural Networks as Universal Probes of Many-Body Localization in Quantum Graphs*. 2021. arXiv: 2108.05737 [cond-mat.dis-nn].

[7] D Luitz, N Laflorencie, and F Alet. "Many-body localization edge in the random-field Heisenberg chain". In: *Phys. Rev. B* 91 (8 2015), p. 081103. DOI: 10.1103/PhysRevB.91.081103. URL: https://link.aps.org/doi/10.1103/PhysRevB.91.081103.

[8] J Šuntajs et al. "Quantum chaos challenges many-body localization". In: *Phys. Rev. E* 102 (6 2020), p. 062144. DOI: 10.1103/PhysRevE.102.062144. URL: https://link.aps.org/doi/10.1103/PhysRevE.102.062144.

[9] J Sakurai and J Napolitano. *Modern Quantum Mechanics*. 2nd. Pearson New International Edition, 2014.

[10] C Majumdar and D Ghosh. “On Next-Nearest-Neighbor Interaction in Linear Chain. I”. In: *Journal of Mathematical Physics* 10.8 (1969). DOI: http://dx.doi.org/10.1063/1.1664978.

[11] D Schroeder. *An Introduction to Thermal Physics*. Addison-Wesley, 1999.

[12] M Schlosshauer. “Decoherence, the measurement problem, and interpretations of quantum mechanics”. In: *Reviews of Modern Physics* 76.4 (2005), 1267–1305. ISSN: 1539-0756. DOI: 10.1103/revmodphys.76.1267. URL: http://dx.doi.org/10.1103/RevModPhys.76.1267.

[13] H Zeh. “On the interpretation of measurement in quantum theory”. In: *Foundations of Physics* 1.1 (1970), pp. 69–76.

[14] P Dirac. “Note on Exchange Phenomena in the Thomas Atom”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 26.3 (1930), 376–385. DOI: 10.1017/S0305004100016108.

[15] F Schindler, N Regnault, and T Neupert. “Probing many-body localization with neural networks”. In: *Phys. Rev. B* 95 (24 2017), p. 245134. DOI: 10.1103/PhysRevB.95.245134. URL: https://link.aps.org/doi/10.1103/PhysRevB.95.245134.

[16] M Srednicki. “Chaos and quantum thermalization”. In: *Physical Review E* 50.2 (1994), 888–901. ISSN: 1095-3787. DOI: 10.1103/physreve.50.888. URL: http://dx.doi.org/10.1103/PhysRevE.50.888.

[17] J Deutsch. “Quantum statistical mechanics in a closed system”. In: *Phys. Rev. A* 43 (4 1991), pp. 2046–2049. DOI: 10.1103/PhysRevA.43.2046. URL: https://link.aps.org/doi/10.1103/PhysRevA.43.2046.

[18] M Rigol, V Dunjko, and M Olshanii. “Thermalization and its Mechanism for Generic Isolated Quantum Systems”. In: *Nature* 452 (7189 2008), pp. 854–858. DOI: 10.1038/nature06838. URL: https://doi.org/10.1038/nature06838.

[19] J Kjäll, J Bardarson, and F Pollmann. “Many-Body Localization in a Disordered Quantum Ising Chain”. In: *Phys. Rev. Lett.* 113 (10 2014), p. 107204. DOI: 10.1103/PhysRevLett.113.107204. URL: https://link.aps.org/doi/10.1103/PhysRevLett.113.107204.

[20] P Anderson. “Absence of Diffusion in Certain Random Lattices”. In: *Phys. Rev.* 109 (5 1958), pp. 1492–1505. DOI: 10.1103/PhysRev.109.1492. URL: https://link.aps.org/doi/10.1103/PhysRev.109.1492.

[21] F Evers and A Mirlin. “Anderson transitions”. In: *Reviews of Modern Physics* 80.4 (2008), 1355–1417. ISSN: 1539-0756. DOI: 10.1103/revmodphys.80.1355. URL: http://dx.doi.org/10.1103/RevModPhys.80.1355.

[22] F Wegner. “Electrons in Disordered Systems. Scaling near the Mobility Edge”. In: *Zeitschrift für Physik B Condensed Matter* 25.4 (1976), pp. 327–337. ISSN: 1539-0756. DOI: 10.1007/BF01315248. URL: https://doi.org/10.1007/BF01315248.

[23] I Gornyi, A Mirlin, and D Polyakov. “Interacting Electrons in Disordered Wires: Anderson Localization and Low-T Transport”. In: *Phys. Rev. Lett.* 95 (20 2005), p. 206603. DOI: 10.1103/PhysRevLett.95.206603. URL: https://link.aps.org/doi/10.1103/PhysRevLett.95.206603.

[24] D Basko, I Aleiner, and B Altshuler. “Metal–insulator transition in a weakly interacting many-electron system with localized single-particle states”. In: *Annals of Physics* 321.5 (2006), 1126–1205. ISSN: 0003-4916. DOI: 10.1016/j.aop.2005.11.014. URL: http://dx.doi.org/10.1016/j.aop.2005.11.014.

[25] T Grover. *Certain General Constraints on the Many-Body Localization Transition*. 2014. arXiv: 1405.1471 [cond-mat.dis-nn].

[26] H Li and F Haldane. “Entanglement Spectrum as a Generalization of Entanglement Entropy: Identification of Topological Order in Non-Abelian Fractional Quantum Hall Effect States”. In: *Phys. Rev. Lett.* 101 (1 2008), p. 010504. DOI: 10.1103/PhysRevLett.101.010504. URL: https://link.aps.org/doi/10.1103/PhysRevLett.101.010504.

[27] G De Chiara et al. “Entanglement Spectrum, Critical Exponents, and Order Parameters in Quantum Spin Chains”. In: *Physical Review Letters* 109.23 (2012). ISSN: 1079-7114. DOI: 10.1103/physrevlett.109.237208. URL: http://dx.doi.org/10.1103/PhysRevLett.109.237208.

[28] A Turner, F Pollmann, and E Berg. “Topological phases of one-dimensional fermions: An entanglement point of view”. In: *Physical Review B* 83.7 (2011). ISSN: 1550-235X. DOI: 10.1103/physrevb.83.075102. URL: http://dx.doi.org/10.1103/PhysRevB.83.075102.

[29] G Torlai, K McAlpine, and G De Chiara. “Schmidt gap in random spin chains”. In: *Physical Review B* 98.8 (2018). ISSN: 2469-9969. DOI: 10.1103/physrevb.98.085153. URL: http://dx.doi.org/10.1103/PhysRevB.98.085153.

[30] J Eisert, M Cramer, and M Plenio. “Colloquium: Area laws for the entanglement entropy”. In: *Rev. Mod. Phys.* 82 (1 2010), pp. 277–306. DOI: 10.1103/RevModPhys.82.277. URL: https://link.aps.org/doi/10.1103/RevModPhys.82.277.

[31] L Vidmar et al. “Volume Law and Quantum Criticality in the Entanglement Entropy of Excited Eigenstates of the Quantum Ising Model”. In: *Phys. Rev. Lett.* 121 (22 2018), p. 220602. DOI: 10.1103/PhysRevLett.121.220602. URL: https://link.aps.org/doi/10.1103/PhysRevLett.121.220602.

[32] L Vidmar et al. “Entanglement Entropy of Eigenstates of Quadratic Fermionic Hamiltonians”. In: *Phys. Rev. Lett.* 119 (2 2017), p. 020601. DOI: 10.1103/PhysRevLett.119.020601. URL: https://link.aps.org/doi/10.1103/PhysRevLett.119.020601.

[33] V Khemani et al. “Critical Properties of the Many-Body Localization Transition”. In: *Phys. Rev. X* 7 (2 2017), p. 021013. DOI: 10.1103/PhysRevX.7.021013. URL: https://link.aps.org/doi/10.1103/PhysRevX.7.021013.

[34] Zhi-Cheng Yang et al. “Two-Component Structure in the Entanglement Spectrum of Highly Excited States”. In: *Physical Review Letters* 115.26 (2015). ISSN: 1079-7114. DOI: 10.1103/physrevlett.115.267206. URL: http://dx.doi.org/10.1103/PhysRevLett.115.267206.

[35] Y Atas et al. “Distribution of the Ratio of Consecutive Level Spacings in Random Matrix Ensembles”. In: *Phys. Rev. Lett.* 110 (8 2013), p. 084101. DOI: 10.1103/PhysRevLett.110.084101. URL: https://link.aps.org/doi/10.1103/PhysRevLett.110.084101.

[36] W Mcculloch and W Pitts. “A Logical Calculus of Ideas Immanent in Nervous Activity”. In: *Bulletin of Mathematical Biophysics* 5 (1943), pp. 115–133. DOI: 10.1007/BF02478259. URL: https://doi.org/10.1007/BF02478259.

[37] D Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Cambridge, MA: MIT Press, 2010. ISBN: 978-0262514620. URL: https://mitpress.mit.edu/books/vision.

[38] D Hebb. *The organization of behavior: a neuropsychological theory*. New York: Wiley, June 1949. ISBN: 0-8058-4300-0. URL: https://www.amazon.com/Organization-Behavior-Neuropsychological-Theory/dp/041565453X.

[39] F Rosenblatt. *The perceptron - A perceiving and recognizing automaton*. Tech. rep. 85-460-1. Buffalo, New York: Cornell Aeronautical Laboratory, 1957.

[40] F Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. New York: Spartan Books, 1962. URL: https://www.amazon.com/Organization-Behavior-Neuropsychological-Theory/dp/041565453X.

[41] M Minsky and S Papert. *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA, USA: MIT Press, 1969. ISBN: 9780262130431. URL: https://mitpress.mit.edu/books/perceptrons.

[42] P Mehta et al. “A high-bias, low-variance introduction to Machine Learning for physicists”. In: *Physics Reports* 810 (2019), pp. 1–124. DOI: 10.1016/j.physrep.2019.03.001. URL: https://doi.org/10.1016/j.physrep.2019.03.001.

[43] J Bridle. “Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition”. In: *Neurocomputing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990, pp. 227–236. ISBN: 978-3-642-76153-9. DOI: 10.1007/978-3-642-76153-9_28. URL: https://doi.org/10.1007/978-3-642-76153-9_28.

[44] K Hornik, M Stinchcombe, and H White. “Multilayer feedforward networks are universal approximators”. In: *Neural Networks* 2.5 (1989), pp. 359–366. ISSN: 0893-6080. DOI:

https://doi.org/10.1016/0893-6080(89)90020-8. URL: https://www.sciencedirect.com/science/article/pii/0893608089900208.

[45] Z Lu et al. "The Expressive Power of Neural Networks: A View from the Width". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/32cbf687880eb1674a07bf717761dd3a-Paper.pdf.

[46] H Robbins and S Monro. "A Stochastic Approximation Method". In: *The Annals of Mathematical Statistics* 22.3 (1951), pp. 400 –407. DOI: 10.1214/aoms/1177729586. URL: https://doi.org/10.1214/aoms/1177729586.

[47] J Kiefer and J Wolfowitz. "Stochastic Estimation of the Maximum of a Regression Function". In: *The Annals of Mathematical Statistics* 23.3 (1952), pp. 462 –466. DOI: 10.1214/aoms/1177729392. URL: https://doi.org/10.1214/aoms/1177729392.

[48] L Bottou. "Stochastic Gradient Descent Tricks". In: *Neural Networks: Tricks of the Trade: Second Edition*. Ed. by G Montavon, G Orr, and K Müller. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 421–436. ISBN: 978-3-642-35289-8. DOI: 10.1007/978-3-642-35289-8_25. URL: https://doi.org/10.1007/978-3-642-35289-8_25.

[49] D Kingma and J Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015. URL: http://arxiv.org/abs/1412.6980.

[50] T Tieleman, G Hinton, et al. "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude". In: *COURSERA: Neural networks for machine learning* 4.2 (2012), pp. 26–31.

[51] A Wilson et al. *The Marginal Value of Adaptive Gradient Methods in Machine Learning*. 2018. arXiv: 1705.08292.

[52] Y LeCun et al. "Efficient BackProp". In: *Neural Networks: Tricks of the Trade: Second Edition*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 9–48. ISBN: 978-3-642-35289-8. DOI: 10.1007/978-3-642-35289-8_3. URL: https://doi.org/10.1007/978-3-642-35289-8_3.

[53] A Krogh and J Hertz. “A Simple Weight Decay Can Improve Generalization”. In: *Advances in Neural Information Processing Systems 4*. San Francisco, CA: Morgan Kaufmann, 1992, pp. 950–957. URL: https://proceedings.neurips.cc/paper/1991/file/8eefcfdf5990e441f0fb6f3fad709e21-Paper.pdf.

[54] N Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[55] C Shannon. “A mathematical theory of communication”. In: *The Bell System Technical Journal* 27.3 (1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[56] C Shannon. “A mathematical theory of communication”. In: *The Bell System Technical Journal* 27.4 (1948), pp. 623–656. DOI: 10.1002/j.1538-7305.1948.tb00917.x.

[57] M Stone. “Cross-Validatory Choice and Assessment of Statistical Predictions”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 36.2 (1974), pp. 111–133. DOI: https://doi.org/10.1111/j.2517-6161.1974.tb00994.x. URL: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1974.tb00994.x.

[58] M Stone. “An Asymptotic Equivalence of Choice of Model by Cross-Validation and Akaike’s Criterion”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 44–47. DOI: https://doi.org/10.1111/j.2517-6161.1977.tb01603.x. URL: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1977.tb01603.x.

[59] N Chandra and R Ghosh. *Quantum Entanglement in Electron Optics: Generation, Characterization, and Applications*. Springer, 2013.

[60] J Von Neumann. *Mathematical Foundations of Quantum Mechanics*. Princeton university press, 2018.

A The Density Operator and Density Matrix

The density operator is a subtle object, so we introduce it with some (perhaps excessive) detail. First, recall that one of the postulates of quantum mechanics is that a system's wavefunction contains complete information about the system, however, wavefunctions describe only pure states [59]. We wish to consider mixed states (linear combinations of pure states) to represent imperfect knowledge of the system, which can be done using John von Neumann's density operator formalism [60]. Given an ensemble of systems, a mixed ensemble is one having some fraction of systems p_1 in state $|1\rangle$, p_2 in state $|2\rangle$ (generically, p_i in state $|i\rangle$) such that the fractions make up the total population ($\sum_i p_i = 1$) and the states $|i\rangle$ are not necessarily orthogonal and are not necessarily less numerous than the dimension of the vector space of $|i\rangle$ (that is, $i \not\leq D$ for a D-dimensional vector space).

Given the above set-up, we may now ask what the mean value of some observable, O would be over many repeated measurements. This is precisely the *ensemble average* of the observable given by [9]

$$\begin{aligned}
[O] &\equiv \sum_i p_i \langle i| O |i\rangle \\
&= \sum_i \sum_\lambda p_i \langle i|\lambda\rangle \langle\lambda| O |i\rangle \\
&= \sum_i \sum_\lambda p_i \lambda |\langle\lambda|i\rangle|^2,
\end{aligned} \tag{A.1}$$

where we have inserted an instance of the identity $\sum_\lambda |\lambda\rangle \langle\lambda|$ in the second line with $O|\lambda\rangle = \lambda|\lambda\rangle$ (*i.e.* $|\lambda\rangle$ is an eigenket of O). It is worth noting the presence of the dual probabilistic elements in the sum; the weights p_i and the quantum probability $|\langle\lambda|i\rangle|^2$.

It is now prudent to choose a 'good' orthonormal basis $\{B_j\}, j \in 1, ..., D$ that spans the vector space and consists of $N = D$ basis vectors. Then applying the same idea as in the previous calculation (twice), we find

$$\begin{aligned}
[O] &= \sum_i \sum_j \sum_k p_i \langle i|B_j\rangle \langle B_j| O |B_k\rangle \langle B_k|i\rangle \\
&= \sum_j \sum_k \Big(\sum_i p_i \langle i|B_j\rangle \langle B_k|i\rangle \Big) \langle B_j| O |B_k\rangle \\
&= \sum_j \sum_k \Big(\sum_i p_i \langle B_k|i\rangle \langle i|B_j\rangle \Big) \langle B_j| O |B_k\rangle ,
\end{aligned} \tag{A.2}$$

which naturally leads one to define the *density operator* [9, 59, 60]:

$$\rho \equiv \sum_i p_i \left|i\right\rangle \left\langle i\right| . \tag{A.3}$$

With this definition, we can easily define the *density matrix* with entries ρ_{jk} as [9]

$$\rho_{jk} \equiv \left\langle B_j\right| \rho \left|B_k\right\rangle . \tag{A.4}$$

Now, for a few concluding observations for the density operator, notice that the last line of (A.2) can be rewritten as

$$\begin{aligned} [O] &= \sum_j \sum_k \rho_{kj} O_{jk} \\ &= \mathrm{Tr}(\rho \cdot O), \end{aligned} \tag{A.5}$$

and that

$$\begin{aligned} \mathrm{Tr}(\rho) &= \sum_i \sum_j p_i \left\langle B_j|i\right\rangle \left\langle i|B_j\right\rangle \\ &= \sum_i p_i \left\langle i|i\right\rangle \\ &= 1. \end{aligned} \tag{A.6}$$

Finally, the density operator for a *pure* ensemble (an ensemble consisting of the entire population in one state) is a projection operator since it is clearly idempotent for some $p_i = \delta^{ik}$ for an ensemble purely described by the state $\left|k\right\rangle$ for a particular k.
This concludes the short introduction to the density operator and density matrix. Additional discussion beyond this can be found in most modern textbooks on quantum mechanics, but this appendix entry follows the style and convention of [9].

B Composite Systems and Pure States

We claim that, for a composite system consisting of, for instance, two subsystems, one cannot (in general) associate a pure state to either subsystem. The argument in support of this claim is well-known to most undergraduates who have taken a quantum mechanics course, but we include a canonical example for completeness (and, perhaps, to aid my parents if they ever try to read this thesis).

Consider a composite system consisting of the tensor product space of two spin-1/2 particles, each having a 2-dimensional state space consisting of eigenstates that we label as $|\pm 1\rangle_A$ and $|\pm 1\rangle_B$; the +'s for spin-up and -'s for spin-down. Then a state of the composite system is an element of the tensor product Hilbert space, $|\Psi\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$. In this product space consider the state

$$|\Psi\rangle = \frac{1}{\sqrt{2}}\Big(|+1\rangle_A \otimes |-1\rangle_B + |-1\rangle_A \otimes |+1\rangle_B \Big). \tag{B.1}$$

This state is properly normalised, but neither subsystem A nor subsystem B are in a definite pure state. One can see this for the A subsystem by tracing out the B subsystem and considering the overlap of the $|+1\rangle_A$ state with the reduced state $\mathrm{Tr}_B\,|\Psi\rangle$

$$_A\langle +1| \Big(\mathrm{Tr}_B\,|\Psi\rangle \Big) = {}_A\langle +1| \frac{1}{\sqrt{2}} \Big(|+1\rangle_A + |-1\rangle_A \Big) = \frac{1}{\sqrt{2}}, \tag{B.2}$$

which is a calculation similar to that used to find reduced density matrices. Of course, for our example we specifically made use of an entangled state of the whole system to demonstrate our point – and there *are* unentangled (or *separable*) states that will demonstrate no issues whatsoever if one wanted the subsystems to be in pure states (consider $|\Psi\rangle = |+1\rangle_A \otimes |-1\rangle_B$, for example).

The example above is trivial, but easily generalizable and quite obvious in its emphasis: for a general state of a composite system there is not an implied pure state for its subsystems.

C The Partition Function/Boltzmann Factor

In truth, this derivation of the Boltzmann distribution is almost certainly superfluous to any physicist/student who has taken a statistical mechanics course. However, it is quite simple, and is one of my favourite derivations in all of statistical physics. I follow a line of reasoning that is similar to (and adopt the same notation as) that of [11].

The setting is simple: we have an isolated system consisting of a reservoir and small subsystem, in thermal equilibrium (in this setting, we are presupposing that thermal equilibrium occurs when entropy is maximised). We suppose that the reservoir is sufficiently large that it may be considered to be at a fixed temperature, T, while the subsystem is small and is subject to variations in temperature (and energy).

Consider two microstates of the subsystem s_1 and s_2 that have corresponding energies $E(s_1)$ and $E(s_2)$ as well as corresponding probabilities $P(s_1)$ and $P(s_2)$, respectively. Instead of trying to find out what the probabilities of these states are *directly*, we decide to try to find a simple formula for the *ratio* of the probabilities. To do this, we make use of the fundamental assumption of statistical mechanics [11]:

> "For an isolated system, all accessible microstates are equally probable."

The isolated system in this setting is the composite system consisting of the reservoir and the subsystem. We then consider what the multiplicity of the reservoir is when the subsystem is in state s_1 ($\Omega_R(s_1)$) versus when the subsystem is in state s_2, ($\Omega_R(s_2)$). Suppose, without loss of generality, that $E(s_1) < E(s_2)$. In each case we have *specified* the microstate of the subsystem. Then we can 'count' the number of ways that s_1 may be achieved in the subsystem by counting the number of possible microstates of the reservoir *given* that the subsystem is in state s_1 (*i.e.* $\omega_R(s_1)$). If we do precisely the same thing but for s_2, we arrive at an expression for the relative probabilities of the *subsystem* microstates

$$\begin{aligned}\frac{P(s_2)}{P(s_1)} &= \frac{\Omega_R(s_2)}{\Omega_R(s_1)} \\ &= \frac{\exp(S_R(s_2)/k_B)}{\exp(S_R(s_1)/k_B)} \\ &= \exp((S_R(s_2) - S_R(s_1))/k_B),\end{aligned} \tag{C.1}$$

where we have made use of the multiplicity definition of entropy, $S = k_B \ln(\Omega)$, with

k_B=Boltzmann's constant.

The expression for the ratio of probabilities of the subsystem states is now described in terms of the change in entropy of the reservoir. Suppose now that the subsystem is sufficiently small as compared to the reservoir that this change in entropy is small so that we can make use of the thermodynamic identity [11]

$$dS_R = \frac{1}{T}\Big(dU_R + pdV_R - \mu dN_R\Big), \tag{C.2}$$

where U is the internal energy, p is the pressure, V is the volume, μ is the chemical potential, and N is the number of 'particles'. The latter two terms we can consider to be vanishing because we are interested in holding the subsystem structure as fixed (*i.e.* we do not wish to exchange particles so $dN_R = 0$) and because the change in pdV is much smaller than dU (the change in volume for atomic particles would be $\sim (10^{-10}\text{m})^3$ which requires pressures much higher than are relevant in this work to produce any significant change at the level of dU). Then

$$S_R(s_2) - S_R(s_1) \approx \frac{1}{T}\Big(U_R(s_2) - U_R(s_1)\Big) = -\frac{1}{T}\Big(E(s_2) - E(s_1)\Big), \tag{C.3}$$

where the change of sign in the last equality is simply an energy conservation statement since a change in energy of $+\Delta E$ in the reservoir corresponds to a change of $-\Delta E$ in the energy of the subsystem. Substituting this expression back into (C.1) we find

$$\begin{aligned} \frac{P(s_2)}{P(s_1)} &= \exp\Big[\Big(-E(s_2) + E_(s_1)\Big)/k_B T\Big] \\ \implies \frac{P(s_1)}{e^{-E(s_1)/k_B T}} &= \frac{P(s_2)}{e^{-E(s_2)/k_B T}} \equiv \frac{1}{\mathcal{Z}}, \end{aligned} \tag{C.4}$$

where $\mathcal{Z}$ is the partition function, and $\exp(-E/k_B T)$ is called the *Boltzmann factor*. The expression for the probability of the subsystem being in a particular microstate, s, is then given by

$$P(s) = \frac{e^{-E_s/k_B T}}{\mathcal{Z}}. \tag{C.5}$$

This describes a probability distribution known as a *Boltzmann distribution*. We are therefore led to a natural definition of a Boltzmann density/probability operator for a subsystem in

thermal equilibrium with a reservoir

$$\hat{\rho}_{\mathrm{eq}}(T) = \frac{e^{-H/k_B T}}{\mathcal{Z}}, \tag{C.6}$$

which is precisely the object introduced in (2.12) of the main text.