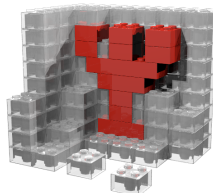


UNIVERSITY OF KENT

School of Physical Sciences
Physics of Quantum Materials group



A thesis submitted for the Doctor of Philosophy degree

Artificial Intelligence Approaches to Quantum Materials: Quantifying Entanglement

AUTHOR:
TYMOTEUSZ TULA

SUPERVISORS:
DR JORGE QUINTANILLA, DR GUNNAR MÖLLER

CANTERBURY, MARCH, 2025

*"And when we meet on a cloud
I'll be laughing out loud
I'll be laughing with everyone I see
Can't believe how strange it is to be anything at all"*

Jeffery Nye Mangum, In the Aeroplane Over the Sea

Declaration of originality

I hereby declare that the work presented in this dissertation, entitled "**Artificial Intelligence Approaches to Quantum Materials: Quantifying Entanglement**", is a result of my own research, under the guidance of my supervisors: Dr Gunnar Möller and Dr Jorge Quintanilla. This thesis is a partial requirement for the award of **Doctor of Philosophy in Physics** and has not been submitted for a purpose of qualification of any other degree at different university/institution. The material covered in Chapter 5 is a summary of published work by T. Tula, G. Möller, J. Quintanilla, S. R. Giblin, A. D. Hillier, E. E. McCabe, S. Ramos, D. S. Barker and S. Gibson [1, 2]. The material covered in Chapter 6 is based on a joint preprint by T. Tula, J. Quintanilla, G. Möller [3]. Unless explicitly stated, all of the data in figures are my own. For the purpose of proofreading this dissertation, I have used recommendations from Writefull's AI model [4].

List of publications

- **2021** - T. Tula, G. Möller, J. Quintanilla, S. R. Giblin, A. D. Hillier, E. E. McCabe, S. Ramos, D. S. Barker, and S. Gibson, “Machine learning approach to muon spectroscopy analysis”, *Journal of Physics: Condensed Matter*, vol. 33, p. 194002, 5.
- **2022** - T. Tula, G. Möller, J. Quintanilla, S. R. Giblin, A. D. Hillier, E. E. McCabe, S. Ramos, D. S. Barker, and S. Gibson, “Joint machine learning analysis of muon spectroscopy data from different materials”, *Journal of Physics: Conference Series*, vol. 2164, p. 012018, 3.
- **2022** - Z. Shan, P. K. Biswas, S. K. Ghosh, T. Tula, A. D. Hillier, D. Adroja, S. Cottrell, G.-H. Cao, Y. Liu, X. Xu, Y. Song, H. Yuan, and M. Smidman, “Muon spin relaxation study of the layered kagome superconductor CsV_3Sb_5 ”, *Phys. Rev. Research*, vol. 4, p. 033145.
- **2024** - T. Tula, G. Möller, J. Quintanilla, “Fitness landscape for quantum state tomography from neutron scattering”, *arXiv preprint*, arXiv:2412.10502

Abstract

Artificial intelligence has gained much interest over the last years — broadly, but also in the field of physics specifically. Here we present a numerical study of machine learning (ML) methods applied to data analysis for strongly correlated quantum systems. The main goal of this work is to establish a framework for entanglement learning: an application of neural networks (NNs) — a supervised ML technique — to detect the entanglement of many-body magnetic systems at finite temperatures. As a preliminary result, we applied an unsupervised ML technique called principal component analysis (PCA) to detect superconducting phase transitions in exotic magnetic materials from real-life muon spectroscopy data. We find that the PCA can identify phase transition points correctly, even when the data vary in a subtle manner, which proves to be difficult for more traditional methods. Moreover, the joint PCA of data from materials with vastly different magnetic properties and different kinds of phase transitions enhances rather than diminishes the performance of the method. In order to demonstrate that the entanglement can in principle be learned, we first examine to which extent the two-point correlators constrain entanglement measures. To this end, we conduct a study of the fitness landscape of neutron-scattering functions obtained from Heisenberg-like Hamiltonians. We discover that the fitness landscape shows a linear correlation as a function of distances between quantum states at zero temperature, and a sub-linear correlation for finite temperatures. This makes the scattering functions — or closely related two-point correlators and structure factors — perfect candidates as input for entanglement learning. We find that entanglement entropy can be correctly predicted by NNs from previously mentioned observables, even when trained on a fraction of the full data set, or when trained on data from a different system than the one for which the prediction is made. Specifically, when trained on observables from an anisotropic transverse-field XY model, in order to obtain an accurate prediction, we only require 3% (6%) of the data to train the network if using the dynamic two-point correlators (structure factors) for learning.

Acknowledgments

From the moment I started my PhD degree I had troubles envisioning actually finishing it. Now that I am so close to submission, I am not at all surprised that so many PhD candidates struggle or even quit before making it to the end. I can count myself as one who struggled. It is only because of the overwhelming support and understanding that I received from my closed ones, that I made it through. In the end, I am not regretting doing my doctorate degree. In addition to learning about new and exciting computational techniques, the PhD experience taught me many things, and I am feeling like a different person from the one that started as a PhD student, better equipped to face the hardships ahead.

Among the many people who I would like to thank for their support, first and foremost are my main supervisors: Dr Jorge Quintanilla and Dr Gunnar Möller. Throughout my PhD, they provided me with insightful advice in my research and moral support; the latter especially near the end of my writing, when I needed it most. Having two main supervisors meant that I was often offered two different perspectives for any scientific topic, which although sometimes confusing, in retrospective helped me a lot to understand it. Jorge's enthusiasm was infectious, and in our conversations, he often helped me see the bigger picture behind what I was doing. I could always count on Gunnar's excellent advice, whenever I needed assist in my calculations, or in more personal matters. I am very glad that I could study under their guidance.

From other people at University of Kent, I would especially like to thank Dr Stuart Gibson who was my additional supervisor and an advisor in the field of machine learning. Conversations with him, early in my PhD studies, have helped me to understand techniques central to my research. Also, I thank other members (or previous members) of the Physics of Quantum Materials group, with special acknowledgment to Dr Sylvia Ramos, Dr Emma McCabe and Dr Sudeep Ghosh, for their contributions to the works we published together. I want to thank other PhD students that I have worked alongside and who helped me especially in guiding through earlier stages of PhD studies: Leon Schoonderwoerd, Henry Sheehy, John Pooley, Tyler Hewitt and Johnathan Martin.

Among researchers from other institutions that I want to acknowledge are: Dr Sean Giblin from Cardiff University and Dr Adrian Hillier from ISIS Neutron and Muon Source facility — for their collaboration in work that is one of the main results of this thesis — and Dr Stefanos Kourtis from Université de Sherbrooke for his support throughout the difficult time of my PhD thesis writing period.

Lastly, I want to thank my friends and family. I could not complete my PhD without the moral and financial support from my sister Anastazja, and my parents Bogusław and Beata. I thank Iwona Maxwell for providing me with the best writing retreat for the past six months, in the form of a room at her house in Berlin. My special thanks go to my friends who supported me along the way: Dąbrowka Biegańska, Krystian Kulas, Tobiasz Szafraniak, and Maciek Stawicki.

This work was financially supported by EPSRC through a DTA studentship under Grant No. EP/R513246/1 and by the School of Physical Sciences, University of Kent.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Chapter content	2
2	Quantum Entanglement	4
2.1	History	5
2.1.1	Einstein-Podolski-Rosen paradox	5
2.1.2	Hidden variables theory	6
2.1.3	Bell's inequality	6
2.1.4	Experimental confirmation of violation of Bell's inequality	8
2.2	Quantifying entanglement	8
2.2.1	Von Neumann entropy and Renyi entropies	8
2.2.2	1-tangle	9
2.2.3	Concurrence and 2-tangle	10
2.2.4	Quantum fisher information	10
2.3	Entanglement transition	11
3	Tensor networks and calculation of correlation functions	13
3.1	Tensor networks and matrix product states	13
3.1.1	Tensors and graphical notation	14
3.1.2	Matrix product states	16
3.1.3	AKLT model	17
3.1.4	Schmidt decomposition	18
3.1.5	Canonical form	19
3.2	Entanglement and Area Law states	22
3.3	Purification MPS	22
3.3.1	Matrix product operators	23
3.3.2	Thermal density matrix in QM	23
3.4	Calculating observables	24
3.4.1	Time-evolving block-decimation	24
3.4.2	MPO evolution	25
3.4.3	Optimising calculations	26
4	Machine learning methods	30
4.1	Machine learning - background	30
4.2	Artificial neural networks	32
4.2.1	Linear threshold units and perceptron	33
4.2.2	Multi-layer Perceptron and Backpropagation	35
4.3	Restricted Boltzmann Machine	37

4.4	Methods used in the research	39
4.4.1	Principal component analysis and Autoencoders	39
4.4.2	Convolutional neural networks	42
5	Machine learning application to muon spectroscopy	46
5.1	Muon spectroscopy experiment	46
5.2	PCA for asymmetry functions	50
5.2.1	Experimental error in PCA	50
5.2.2	Results - simulated data	52
5.2.3	Results - experimental data	53
5.3	Extensions	57
5.3.1	Cumulative principal component score	57
5.3.2	Principal component score - calculating error	58
6	Towards quantum state tomography from neutron-scattering functions	60
6.1	Background	60
6.2	Generalisations	62
6.2.1	Systems at finite temperature	62
6.2.2	Systems with linear spin terms	64
6.3	General framework	65
6.3.1	Neutron scattering functions	65
6.3.2	Fitness landscape of scattering functions	66
6.4	State tomography	67
6.5	Finite temperature	71
6.6	Discussion and Conclusions	72
7	Entanglement learning	74
7.1	General framework	74
7.1.1	Toy models	74
7.1.2	Observables	76
7.1.3	Entanglement of purification	80
7.1.4	Cost function	81
7.1.5	Neural network architecture	82
7.2	Results	82
7.2.1	Learning entanglement landscape within given toy model	83
7.2.2	Learning entanglement landscape between two models	89
7.2.3	Learning from dynamical observables	94
7.3	Class activation maps	95
7.4	Conclusions	100
8	Conclusions	101
8.1	Future work	102

List of Figures

2.1	An example of the EPR paradox: a) An electron and a positron annihilate each other creating two gamma particles (high energy photons). Assuming that the initial spins of electron and positron were opposed, in order to preserve momentum and angular momentum of the system after the annihilation process, two photons have to travel in opposing directions, and their spin have to cancel each other; b) The wave function describing two receding photons is $ \psi_{12}\rangle = \frac{1}{\sqrt{2}} (\uparrow_1\downarrow_2\rangle + \downarrow_1\uparrow_2\rangle)$. The photons are entangled. Polarisation of the first photon is going to be measured by Alice (A); c) A measurement by Alice collapses the wave function of combined system of two photons, making them entangled no more. A measurement by Bob (B) will now show opposite polarisation to the one measured by Alice. The first measurement seems to affect the second photon instantaneously, violating the principle of local causality.	5
2.2	CHSH setup. Two particles, each set with values (X, Y) of properties P_X and P_Y , are sent to Alice (A) and Bob (B), where they can only measure one of the properties with 50/50 probability of which measurement is chosen. Carol (C) can prepare the particles in any way she wants, meaning that the kind of correlations that occur in quantum mechanics (for example EPR paradox) can be simulated.	7
2.3	Sketch of the ET for the anisotropic XY Heisenberg model in the thermodynamic limit. The AFM long-range interactions (AFM LRO) prevail in the presence of external magnetic field up to the $h = h_c = 1$, where the quantum fluctuations are strong enough to destroy AF order. At $h = h_f = \sqrt{1 - \gamma^2}$, the wave function becomes fully factorised and entanglement measures such as entanglement entropy, t-entanglement or concurrence will go to 0 at that point. At critical point, the nature of the entanglement is changed as well.	12
3.1	Diagrammatical representation of tensors and basic operations/manipulations of tensors: a) A single n -rank tensor is drawn as a box (or a node) with n legs representing indices $a_1, a_2, \dots$; b-c) The exchange of the legs is the permutation of indices of a tensor; d) If two tensors are connected with a single (or more) legs, this indicates a contraction of that index between them, which can be changed to represent a single tensor; e) Joining legs together creates a composite index.	15
3.2	Matrix product state representation for a 1-D chain of N spins. The N -rank tensor on the left is a full wave function, while tensors on the R.H.S each correspond to a local contribution to the information about the state. . .	16

3.3	The representation of AKLT ground state. Each $S=1$ site σ_i can be divided into two spin- $\frac{1}{2}$ partons: a_i and b_i , which connect through singlet interactions with their neighbours from the other sites.	17
3.4	Bipartition of a 1-D chain of spins into subsystems A and B.	18
3.5	Canonical forms and calculation of local observables: a) The matrices that form left canonical form are left-orthonormal, which means that if there is a contraction with their hermitian conjugate (usually drawn at the bottom of a diagram), and there are no virtual legs on the left, the contraction results in an identity. Identity can be represented as a line in a diagram [5]; b) The same holds true for right canonical form, with the only difference that there should be no virtual legs on the right; c) Contraction of local observables in a mixed canonical form. Calculation of some local observable at position l can be done by using left canonical form to the left and including l and right canonical form to the right of l . Contractions are done one by one - from the left, by using a) property, one can first turn $A^{[1]}$ into a line on a diagram, and then merge that line with the one that connects $A^{[2]}$ and its conjugate. No additional legs on the left means that a) relationship can again be applied and so on. The same procedure for the matrices B means that eventually only the diagram on R.H.S remains.	20
3.6	Purification MPS representing density matrix. Trace over auxiliary space is represented by contractions $q_1, q_2 \dots q_N$	24
3.7	TEBD algorithm for a system with two-site nearest neighbour interactions: a) In a single iterative step, the initial state ψ_0 is connected to odd and even parts of the Hamiltonian (see equations (3.43) and (3.44)); b) The process of multiplying a single two-site evolution tensor to the MPS. The physical σ and virtual a legs are connected in order to reshape 4-tensor into matrix. This new matrix contains two left canonical matrices from the MPS, evolution tensor and one of the singular value matrix. After performing SVD the updated $\tilde{A}^{[j]}$ is composed of left orthonormal matrix $\tilde{U}$ from the decomposition, while - in order to preserve left canonical form [6] - right $\tilde{A}^{[j+1]}$ has to include new singular value matrix, $\tilde{V}^\dagger$ and the inverse of singular values $S^{[j+1]}$	25
3.8	Comparison of entanglement growth for a typical imaginary time evolution, with and without the application of disentangler. Both the maximal peak and entanglement entropy at lower temperatures are smaller with application of disentangler.	27
3.9	Evolution of the local spin operator in Heisenberg picture. The initial block in the middle consist of identity matrices and the local operator at some position j . The evolution is applied both to the bra (down) and ket (up) indices, and all sites have both physical and auxiliary legs. The unitary disentangler block applies only to the auxiliary indices, while the evolution U applies to the physical indices.	28
4.1	MNIST set of handwritten digits, used in one of the first papers on NNs and the categorization tasks. From then, it is commonly used as a testing bed for new ML algorithms, and as a common set for learning ML.	31

4.2	Examples of LTU (a) and perceptron with multiple LTUs and a bias input (b). A single LTU combines the weighted input $\sum_i w_i x_i$ and runs it through some activation function h . The perceptron contains several LTUs, which share the same inputs but not necessarily the strength of connections $\mathbf{w}$	33
4.3	Example of a two-layer perceptron. Initial input is passed to a intermediate layer in the same fashion as for perceptron. Then the output of the hidden layer is passed to the second one.	35
4.4	Restricted Boltzmann Machine structure. Physical spins in the interacting system are connected not directly, but through the hidden layer of h_i units. Weights $\{a_i\}$, $\{b_j\}$ and $\{w_{(i,j)}\}$ are trained to simulate ground state of a system defined through the Hamiltonian, by using the variational principle.	38
4.5	PCA of a data in 2-D space x_1, x_2 . Because of the linear correlation in the data, PCA finds the direction which holds the most covariance v_1 - first PC. The second PC v_2 is orthogonal to the first.	40
4.6	Scree plots. The Horizontal axis represents order of PCs, and the vertical axis the principal values connected to those components. a) Typical case with exponential decay. The few first PCs hold almost all of the covariance b) An example of bad performance, which usually suggests that the data are not linearly correlated.	41
4.7	Convolutional layers for 1-D input. The connections between neurons from consecutive layers is defined by the size of the filter. In the above example, l_1 layer takes input from three neighbouring neurons in the previous layer, while l_2 takes it from seven neurons from layer l_1 . The weights are shared between all units within layers. The boundary conditions can be solved by decreasing size of neurons in a layer (l_1) or by introducing zero input in previous layer (l_2 , greyed neurons at border).	43
4.8	Convolutional layer for 2-D input with multiple maps and multiple filters. The two maps in layer l_1 are treated as an input for layer l_2 . The three filters of l_2 detect diagonal, vertical and horizontal lines in the maps from l_1 respectively (greyed out parts of 3x3 sections above l_2). The input is taken from all of the maps in the previous layer.	44
4.9	General structure for 1-D CNN, with multiple channels. Following the input, there are multiple convolutional layers, which are decreasing in size, but include more maps, representing different filters. At the beginning those detect simple shapes such as lines, but the patterns get more complicated with deeper networks. Pooling layers are used to translate those patterns into simple scores, for which a fully connected network can analyse it further.	45
5.1	Different muon states and the distribution of positron decay. a) Thermalized muons can be implanted in three different states: positively charged muon, muon coupled with two electrons and neutral muonium atom. b) After the decay into positron and two neutrinos, the momentum of the positron is positively correlated with muon's spin at the time of the decay.	47

5.2	Muon SR setup in backward-forward (BF) geometry. a) The stream of muons enters the sample through a hole in backward (B) set of detectors. Particles thermalize and are implanted somewhere within the sample (S). b) Larmor precession due to the combined local and external magnetic fields. c) Muons with a half-time of $2.2\mu s$ eventually decay into positrons, which are detected either by B or forward (F) set.	48
5.3	Asymmetry function from a ZF μSR of $LaNiGa_2$ at $T = 0.5K$ and the Kubo-Toyabe function fit to the data.	49
5.4	Graphical representation of PCA performed on μSR data. a) The j -th asymmetry function in a set of data can be represented as a N dimensional vector. b) High-dimensional vector space of all asymmetry spectra. If the functions are linearly correlated, they will lie near smaller sub-space (represented by reddish plane). c) The PCA finds new orthogonal basis $(\pi_1, \pi_2) \rightarrow (\pi'_1, \pi'_2)$, which holds most covariance in least amount of basis vector in the data.	51
5.5	PCA performed on simulated Kubo-Toyabe functions. Each column represents data with different amount of error, with third column ($R = 0.0025$) corresponding to error comparative to the experimental data. First row shows the 1st vs 2nd most important PC scores, with colormap corresponding to change in temperature. On Second and third row, the individual changes in 1st and 2nd PC scores can be seen as a function of temperature. The fourth column shows changes in PC vectors for different error. The faded teal plot on second, third and fourth column indicates the 1st PC vector without the error.	53
5.6	An illustration of how PC analysis can be used to reduce the dimensionality of a muon data set. The set consists of a sizeable number of experimentally-obtained muon asymmetry functions $A(t)$. The black curves in panels (a)–(c) present three particular examples. Each curve has 110 time stamps and therefore constitutes a point in a 110-dimensional space. PCA yields a small number of PCs which, through linear combination, can accurately describe any curve in the data set. In our case, we find the two most important PCs shown in panel (e). The reconstruction of the original data using the PCs and the average (d) (see Section 4.4.1) can be obtained by the formula “reconstruction = average + 1st PC score $\times$ 1st PC + 2nd PC score $\times$ 2nd PC + (higher PCs that were omitted)”. From that we can interpret the PCs as the most common deviations from the average curve. The reconstructions are shown, for our three examples, by the red curves in panels (a)–(c). Using only the first two PCs gives an accurate reconstruction in this case and therefore enables us to represent each curve by a single point on a two-dimensional plane (f). For this example we used 25 $A(t)$ curves for the material $BaFe_2Se_2O$ obtained at 25 different temperatures.	54
5.7	Results of PCA performed separately for all sets of data. Each column represents experimental data coming from a given magnetic material. First row shows first and second PC curves. In Second/third row the 1st/2nd PC score dependency on temperature is plotted, and the last row shows the scree plots for each set of data.	55

5.8	Results of joint PCA of all of the magnetic materials under study. First/second row shows the dependence of 1st/2nd PC scores on temperature correspondingly. Red vertical lines signify values of phase transitions taken from [7, 8].	55
5.9	The change of PC vectors with addition of new batches of data. Each column represents different PC curve, with the last being the scree plot of PCA. In the top row, only asymmetry spectrums of LaNiGa_2 data are provided, and the data from different materials are gradually added in following rows (as indicated on left of the plots). The blue vertical line in scree plots indicate the amount of PCs needed to hold 80% of covariance.	56
5.10	a) The temperature dependence of first four PC scores for asymmetry functions of magnetic materials. Red vertical lines indicate phase transitions [7, 8]. b) CPCS of first four PCs for corresponding materials.	58
5.11	PCA of muon spectroscopy of CsV_3Sb_5 . PCA was performed jointly with other magnetic materials mentioned in this chapter. a) The cPCS for CsV_3Sb_5 . b-d) The three most important PC scores with error estimations. Figure taken from [9].	59
6.1	Quantum state tomography and Hamiltonian learning. The way to obtain observables from a Hamiltonian is straightforward. Ground state (or thermal density matrix for $T > 0$) is obtained by a suitable method, such as numerical ED. Then the expectation value of an observable $\langle \hat{O} \rangle$ is calculated from $\langle \psi \hat{O} \psi \rangle$ (or $\text{Tr} [\rho \hat{O}]$). The reverse function generally does not exist. The process of deducing a state from experimental observables is called quantum state tomography (blue arrows). We call obtaining initial Hamiltonian from observables Hamiltonian learning (red arrow).	61
6.2	Geometries of magnetic systems used in simulating scattering functions: a) Magnetic molecule with a ring geometry. To preserve the symmetry of local coordinates for all spin sites, the transformation from local $\{x', y'\}$ to global $\{x, y\}$ coordinates has to be performed (according to the equation (6.18)), with the ϕ angle given by the deviation from global coordinates; b) Linear molecule with long range interactions. The target Hamiltonian describing that system in our research had non-decaying interactions for every pair of spin sites, technically making it a non-dimensional system. We still refer to it as a linear geometry, due to the fact that every local coordinate system is the same as the global one, which affects the calculated scattering functions.	66

- 6.3 Fitness landscape for quantum state tomography using the ground-state, diffuse magnetic neutron-scattering function. The target state ψ is the ground state of a spin-1/2 ring tetramer with nearest-neighbour, AFM Heisenberg interactions. This corresponds to the zero-field Hamiltonian (6.1) with all coupling constants equal to zero except $J_{j,j+1}^{\alpha,\alpha} = 1$ for all $j = 1, 2, 3, 4$ and $\alpha = x, y, z$. (a) Cost function ΔS [Eq. (6.21)] as a function of the distance $\text{dist}(\psi, \psi')$ between the target ground state and a trial state vector ψ' [Eq. (6.23)]. We used three sets of 1000 random state vectors each with amplitudes from drawn the probability distribution centred on ψ described in the main text. Each individual state vector is shown by a filled circle. Each set had a different value of the parameter $\sigma_{\max}$, which gives the upper limit to the standard deviation sampled from uniform distribution (see explanation in section 6.4) (b) The target diffuse magnetic scattering function $S(\mathbf{q})$ obtained by ED. (c-e) Three scattering functions obtained from the three random state vectors highlighted with stars on panel (a), with c corresponding to the leftmost (least deviation) and e to the rightmost star (most deviation). 68
- 6.4 Dependence of the fitness landscapes on the number of spins L in the cluster. For each L , 1000 target states were sampled with normal-distributed perturbations around the target wave function (see main text). Target states were chosen as ground states of: (a) A spin-1/2, nearest-neighbour anti-ferromagnetic Heisenberg model in a ring geometry, i.e., the same model as in Figure 6.3, but with varying L . (b) A random long-range spin-1/2 Hamiltonian in a linear open chain geometry, given by Equation (6.14) with random exchange constants $J_{i,j}^{\alpha,\beta}$ and a random global magnetic field with components $h_i^\alpha = h^\alpha$. For each L , Hamiltonian parameters $J_{i,j}^{\alpha,\beta}$ and h^α were drawn from a normal distribution with zero average and standard deviation equal to J (the nearest-neighbour coupling of the Heisenberg Hamiltonian used in panel (a)). Dashed lines in (a,b) show the linear relation in Equation (6.24). 70
- 6.5 Dependence on L (see legend) of the fitness landscape for finite-temperature quantum tomography of a random Hamiltonian. The Hamiltonian was obtained in the same way as in Figure 6.4 (b). The target density matrix ρ is the thermal equilibrium density matrix of that Hamiltonian at temperature $T = J$. 1000 trial states (random density matrices) were tested against this target state. The distance between random density matrices and the target is measured on the horizontal axis and defined by equation (6.25). For $T = 0$ this would have coincided with the distance $\text{dist}(\psi, \psi')$ between ground states. The dashed line represents equation (6.24), for comparison. For the calculation of the scattering functions the atoms were assumed to be on a straight line. The inset shows similar output with different random targets at each size. 71
- 7.1 Framework for testing entanglement learning. Simulated thermal density matrix for a model Hamiltonian $\hat{H}$ can be obtained using TNs (1). From density matrix one can calculate both observables (2) and entanglement entropy (3), which can then be fed to ML algorithm in order to reproduce the entanglement (4). 75

7.2	Entanglement entropy as a function of magnetic field, for the lowest achieved temperature $T = 0.1$, for XY model with anisotropy $\gamma = 0.25$. Red vertical line indicates the factorisation field. Clearly, there is no discontinuity at $h = h_f$	77
7.3	Examples of static two-point correlation functions $\langle S^x S^x \rangle$, $\langle S^y S^y \rangle$ and $\langle S^z S^z \rangle$ from the XY model (7.1) with anisotropy $\gamma = 0.25$, as a function of distance j . The figures are drawn for two different values of β (left column $\beta = 3$, right column $\beta = 10$), and three different values of magnetic field: long-range AFM (top), at factorisation field (middle), after factorisation and critical fields (bottom).	78
7.4	Close up of structure factors $\langle S^x S^x \rangle$, $\langle S^y S^y \rangle$ and $\langle S^z S^z \rangle$ from the XY model (7.1) with anisotropy $\gamma = 0.25$, as a function of frequency domain index $k = \frac{qL}{2\pi}$ (outlier zero modes for $\langle S^z S^z \rangle$ are not shown to focus on the features of the rest of structure factors). The figures are drawn for two different values of β (left column $\beta = 3$, right column $\beta = 10$), and three different values of magnetic field: long-range AFM (top), at factorisation field (middle), after factorisation and critical fields (bottom).	79
7.5	Entanglement of purification for 1D purification MPS. a) After joining physical indices σ with the auxiliary q , the von Neumann entropy of any bisection can be calculated as with regular MPS according to formula depicted, where Σ_L and Q_L are physical and auxiliary Hilbert spaces of the left section accordingly. Density matrix $\rho_{\Sigma_L Q_L}$ is given by tracing out the right section. b) Due to the fact that the thermal density matrix is invariant to any unitary matrix U applied to auxiliary indices, but the von Neumann entropy for the purification MPS will change as a function of U , a well defined measure is the minimum of von Neumann entropy for all different unitary operations U , which defines entanglement of purification.	81
7.6	The general scheme for entanglement learning, presented using the example of the XY model with anisotropy $\gamma = 0.25$. A portion of the entire data is chosen as the training set. Each data point consists of spatial correlation functions (or structure factors) and the entanglement entropy obtained from the same state as those observables (red arrow pointing to the specific value of entanglement entropy). For clarity, only $\langle S^x S^x \rangle$, $\langle S^y S^y \rangle$ and $\langle S^z S^z \rangle$ are displayed in the left panel. The colours indicate the same observables as in Figures 7.3 and 7.4. Entanglement entropy was plotted for a single value of $h = 1.1$ and entire range of β in right panels. Using CNN with the correlators as an input and the entanglement entropy as an output, the estimation of entanglement entropy (depicted by the violet points in the right bottom plot, the greyed out points are the actual entanglement entropy, as also depicted on the right top plot) can be calculated after training.	83
7.7	Different entanglement learning strategies for training on two-point correlation functions. Scans of external magnetic fields h at chosen inverse temperature β . The data is shown for XY model with $\gamma = 0.25$ (top 6 plots) and XY model with $\gamma = 0.5$. (bottom 6 plots). Red horizontal lines indicate data that was used as a training set. The true entanglement entropy is displayed in the top left plot for each model. The contour lines show the values of prediction for each training strategy, while the number at the top is the measure of fitness defined in (7.7).	85

7.8	Different entanglement learning strategies for training on two-point correlation functions. Scans of inverse temperature β at chosen external magnetic fields h . The data is shown for XY model with $\gamma = 0.25$ (top 6 plots) and XY model with $\gamma = 0.5$. (bottom 6 plots). Red horizontal lines indicate data that was used as a training set. The true entanglement entropy is displayed in the top left plot for each model. The contour lines show the values of prediction for each training strategy, while the number at the top is the measure of fitness defined in (7.7).	86
7.9	Different entanglement learning strategies for training on structure factors. Scans of external magnetic fields h at chosen inverse temperature β . The data is shown for XY model with $\gamma = 0.25$ (top 6 plots) and XY model with $\gamma = 0.5$. (bottom 6 plots). Red horizontal lines indicate data that was used as a training set. The true entanglement entropy is displayed in the top left plot for each model. The contour lines show the values of prediction for each training strategy, while the number at the top is the measure of fitness defined in (7.7).	87
7.10	Different entanglement learning strategies for training on structure factors. Scans of inverse temperature β at chosen external magnetic fields h . The data is shown for XY model with $\gamma = 0.25$ (top 6 plots) and XY model with $\gamma = 0.5$. (bottom 6 plots). Red horizontal lines indicate data that was used as a training set. The true entanglement entropy is displayed in the top left plot for each model. The contour lines show the values of prediction for each training strategy, while the number at the top is the measure of fitness defined in (7.7).	88
7.11	Entanglement learning between two different XY Hamiltonians, trained on two-point correlators. Left column shows the true and predicted entanglement entropy at the bisection in the middle of $L = 30$ spin chain for a XY model with anisotropy $\gamma = 0.25$. Right column shows the same for XY model with anisotropy $\gamma = 0.5$. The plots are functions of external magnetic field with a range of β values. Top two figures show the true entanglement entropy. Two middle figures show predictions when trained on two-point correlators from just the $\gamma = 0.25$ model. The bottom two figures show predictions when trained on two-point correlators from just the $\gamma = 0.5$ model. Red border and axes indicates which of the prediction is done on the model that served as the training set for the NN.	90
7.12	Entanglement learning between two different XY Hamiltonians, trained on structure factors. Left column shows the true and predicted entanglement entropy at the bisection in the middle of $L = 30$ spin chain for a XY model with anisotropy $\gamma = 0.25$. Right column shows the same for XY model with anisotropy $\gamma = 0.5$. The plots are functions of external magnetic field with a range of β values. Top two figures show the true entanglement entropy. Two middle figures show predictions when trained on structure factors from just the $\gamma = 0.25$ model. The bottom two figures show predictions when trained on structure factors from just the $\gamma = 0.5$ model. Red border and axes indicates which of the prediction is done on the model that served as the training set for the NN.	91

7.13	Entanglement learning between XY and XYZ Hamiltonians, trained on two-point correlators. Left column shows the true and predicted entanglement entropy at the bisection in the middle of $L = 30$ spin chain for a XY model with anisotropy $\gamma = 0.25$. Right column shows the same for XYZ model with anisotropy $\gamma = 0.5$ and $\Delta = 0.25$. The plots are functions of external magnetic field with a range of β values. Top two figures show the true entanglement entropy. Two middle figures show predictions when trained on two-point correlators from just the $(\gamma, \Delta) = (0.25, 0)$ model. The bottom two figures show predictions when trained on two-point correlators from just the $(\gamma, \Delta) = (0.5, 0.25)$ model. Red border and axes indicates which of the prediction is done on the model that served as the training set for the NN.	92
7.14	Entanglement learning between XY and XYZ Hamiltonians, trained on structure factors. Left column shows the true and predicted entanglement entropy at the bisection in the middle of $L = 30$ spin chain for a XY model with anisotropy $\gamma = 0.25$. Right column shows the same for XYZ model with anisotropy $\gamma = 0.5$ and $\Delta = 0.25$. The plots are functions of external magnetic field with a range of β values. Top two figures show the true entanglement entropy. Two middle figures show predictions when trained on structure factors from just the $(\gamma, \Delta) = (0.25, 0)$ model. The bottom two figures show predictions when trained on structure factors from just the $(\gamma, \Delta) = (0.5, 0.25)$ model. Red border and axes indicates which of the prediction is done on the model that served as the training set for the NN.	93
7.15	Entanglement learning for the XY Hamiltonian with anisotropy $\gamma = 0.25$, trained on dynamic two-point correlators. The ratio N_t/N_{all} is the amount of data in the training set as compared to the total number of data points. $D(\text{pred})$ is calculated according to Equation (7.7) for all data points. The error bars show the variance between different training sets in multiple runs. The training sets were chosen at random.	94
7.16	Entanglement learning for the XY Hamiltonian with anisotropy $\gamma = 0.25$, trained on dynamic structure factors. The ratio N_t/N_{all} is the amount of data in the training set as compared to the total number of data points. $D(\text{pred})$ is calculated according to Equation (7.7) for all data points. The error bars show the variance between different training sets in multiple runs. The training sets were chosen at random.	95
7.17	Example of 2-D scattering functions for an hexagonal molecular magnet with the XY Hamiltonian (7.1), with anisotropy parameter $\gamma = 0.4$. Left column show scattering functions before h_f , and right columns are the scattering functions after h_f	97
7.18	CAMs for identifying the phase below ET. Left column shows the CAMs from a scattering functions of a anisotropic XY model (7.1) with anisotropy parameter $\gamma = 0.4$ with external magnetic fields smaller and close to h_f . Right column shows CAMs for the same model h_f	98
7.19	CAMs for identifying the phase after ET. Left column shows the CAMs from a scattering functions of a anisotropic XY model (7.1) with anisotropy parameter $\gamma = 0.4$ with external magnetic fields smaller and close to h_f . Right column shows CAMs for the same model after h_f	99

List of Tables

5.1	Error margins for first three PCs for CsV ₃ Sb ₅ data.	59
-----	--	----

Abbreviations

ML	M achine L earning
AI	A rtificial I ntelligence
PC	P rincipal C omponent
PCA	P rincipal C omponent A nalysis
NN	N eural N etwork
CNN	C onvolutional N eural N etwork
DNN	D eep N eural N etwork
ED	E xact D iagonalization
TN	T ensor N etwork
EPR	E instein- P odolsky- R osen
CHSH	C lauser- H orne- S himony- H olt
QFI	Q uantum F isher I nformation
QPT	Q uantum P hase T ransition
AFM	A nti F erro M agnetic
LRO	L ong- R ange O rders
ET	E ntanglement T ransition
MPS	M atrix P roduct S tate
MPO	M atrix P roduct O perator
AKLT	A ffleck- K ennedy- L ieb- T asaki
TEBD	T ime- E volving B lock- D ecimation
SVD	S ingular V alue D ecomposition
RBM	R estricted B oltzmann M achine
LTU	L inear T hreshold U nit
MLP	M ulti- L ayer P erceptron
ZF μSR	Z ero- F ield μ -on S pectroscopy
TRSB	T ime- R eversal S ymmetry B reaking
cPCS	cumulative P rincipal C omponent S core
DFT	D ensity F unctional T heory
CAM	C lass A ctivation M ap

Chapter I

Introduction

This dissertation is a summary of results on machine learning (ML) applications in condensed matter physics, with the focus on the study of quantum entanglement in many-body interacting systems. The research presented in this thesis is mostly based on numerical studies. The data used to train ML algorithms was obtained through simulation of small (using exact diagonalization (ED)) or larger (using tensor networks (TNs)) models. There are, however, some original results that used data from real-life experiments, and the research was a collaboration with an experimental team. The overall goal of this thesis is to find whether it is possible to use ML approaches to discover the relationship between quantum entanglement of a many-body state and observables that can be obtained from experiment. If such a connection can be found, it would motivate further research into experimental entanglement measurements, for which so far there are no universal protocols.

The research undertaken as part of this dissertation is divided into three main topics, broadly connected to answer how well ML can be used to determine important qualities of magnetic materials, namely phase transition and entanglement, with a focus on the latter. These topics are, in the order that they are presented in the dissertation:

- Using ML to identify time-reversal symmetry breaking (TRSB) superconducting phase transitions from muon spectroscopy data.
- Finding how well can the experimentally available observables determine the ground state and finite-temperature state of magnetic materials.
- Using the same observables to determine a measure of entanglement in certain magnetic materials with ML techniques.

I.1 Motivation

The major hindrance in numerical studies of quantum many-body physics has been historically, and still is, the exponential growth of the dimensionality of Hilbert space with the number of degrees of freedom. The size of system that can be directly solved using ED is restricted by memory requirements on classical computers, which today in practice limits the study of interacting lattice models to around 24 sites/particles¹. Apart from a few exceptions, this is not enough to correctly predict the behaviour of electrons or quasi-particles in crystals which are made of $N \approx 10^{23}$ atoms. On the other hand, this large number can be helpful

¹This estimation is for the full diagonalisation (not just low-lying eigenvalues) of systems with local Hilbert spaces of dimensionality $d = 2$. For $d > 2$, the restriction is even greater.

in analytical studies. When combined with spatial symmetry, it can often be used to solve ordered condensed matter systems by treating the lattice as infinite $N \rightarrow \infty$, which allows for looking at a broader picture, rather than focusing on the microscopic details. Counter-intuitively, it seems like models that can be predicted out exactly in quantum physics are either very small or infinitely large, and the hardest problems (at least numerically) are between those two extremes.

Not surprisingly, much of the research in the field of condensed matter has been focused on finding ways to approximate the results without having to directly solve Hamiltonians. Some of these methods are based on a restriction of initial Hilbert space to a smaller subspace, hoping that the desired state which describes the system lies within or close to it. Others, like Variational Quantum Monte Carlo [10], use stochastic approach, which gives approximate results by design but does not require to keep track of all possible configurations of the model. Recently, there has been much interest in employing ML techniques in studying quantum states [11] of many-body systems. Calculation frameworks such as neural networks (NNs) are in principle "universal functions", meaning that they can approximate any mapping with precision that can be easily adjusted. This is very advantageous for quantum simulation, since usually the goal is to find connections between, for example, Hamiltonians and their ground states ($H : \rightarrow \psi_0$) or initial states and their time-evolved counterparts ($\psi(t = 0) : \rightarrow \psi(t > 0)$). If a well-behaved function between some set of observables $\{\langle \hat{O} \rangle\}$ and entanglement of a quantum system exist, the NN should be able to approximate it.

The possibility of predicting entanglement properties of a system with experimentally available observables without a need to know about microscopic physics is highly desirable. Currently, there is no universal entanglement measure that can be directly obtained from measurements of condensed systems. To underline the importance of research on quantum entanglement, an introduction to this topic is provided in Chapter 2.

1.2 Chapter content

Apart from the introduction and conclusions, this thesis comprises six chapters organised so that all of the results obtained during my PhD studies are presented in a logical order. The first three chapters show the motivation behind the research (the history behind, and why it is important to know entanglement in many-body physics) and explanation of the computational methods. Following that, the final three chapters contain all the results obtained during my studies, with the topic of each chapter corresponding to the ones listed at the beginning of the introduction. The next paragraphs in this section are short summaries of each chapter.

Chapter 2 is an introduction to the topic of quantum entanglement. This serves as a more elaborate explanation of the motivation behind the project. The chapter starts by explaining arguments raised against Copenhagen interpretation of Quantum Mechanics in the early days of the theory. The famous Einstein-Podolsky-Rosen (EPR) paradox and Bell's theorem are presented, from which the concept of quantum entanglement is first derived. Later in the chapter, the idea of many-body entanglement is explained. It will be shown that a full insight into microscopic information is needed in general to get a whole picture of quantum entanglement. This proves to be a problem in experimental research, and that is why approximate measures of entanglement are often introduced. Some of these measures are presented in more detail in this chapter.

In Chapter 3, the methods of obtaining simulated training data for entanglement learn-

ing are explained. For small systems, ED can be used to obtain observable quantities, which simulate experimental data. We decided to use TNs to obtain observables for larger 1-D systems. TN have the additional advantage of naturally providing detailed information about entanglement in many-body systems, which then can be used for supervised ML. This chapter introduces the basic concepts behind TN like matrix product states (MPSs), graphical notation, and canonical representation. In addition, more advanced methods are explained in the context of obtaining finite-temperature dynamical observables.

Chapter 4 focuses on introducing concepts of ML and its application to physics in recent years. This part of the thesis explores why ML methods can be successfully applied to the physical sciences and what types of algorithms are particularly useful in this field. The techniques used for my final results are described in more detail. Among these are principal component analysis (PCA), NNs and convolutional neural networks (CNNs).

In Chapter 5, the first original results are presented — the application of ML to muon spectroscopy data. In collaboration with an experimental team from the ISIS Neutron and Muon Source, we found that certain unsupervised ML techniques can be used to enhance the analysis of magnetic materials. We show that some exotic phase transitions, for which the standard regression methods are not always viable, can be identified instead by using techniques such as PCA.

The following two chapters contain the main results of this dissertation. Chapter 6 shows evidence that there exists a mapping from two-point magnetic correlators to the quantum state of the material. Inspired by density functional theory (DFT), we show that for certain magnetic systems, there exists a one-to-one correspondence between parameters of a Hamiltonian and its finite-temperature correlation functions. We further propose that experimentally available observables (like scattering functions) can be used for quantum state tomography: an inference of system state from other sets of data related to the sample studied. In the last part of the chapter, the results of a stability analysis of scattering functions are shown. The possibility of quantum state tomography in magnetic systems suggests that the entanglement learning should also be possible, since entanglement is also a function of the state.

Chapter 7 shows the results of applying CNN algorithm to two-point magnetic correlator data obtained by TNs for testing at finite temperature and on larger systems than ED allows for. The main aim is to study whether it is possible for CNN to learn about entanglement from observables evaluated at different temperatures or external magnetic fields. We train CNN to predict either the entanglement entropy at each bisection of the 1-D spin chain, or the entanglement entropy averaged over all bisections, which is just a single number.

The last chapter summarises the main results of the thesis and proposes possible directions of further research.

Chapter 2

Quantum Entanglement

This chapter serves as a summary of concepts surrounding quantum entanglement, its connection to quantum information, and the quantities that measure it. Intuitively, quantum entanglement is the property between two or more parts of an interacting system, which holds that system inseparable i.e. the full wave function of non-separable system cannot be described by a product of wave functions of subsystems:

$$|\psi\rangle_{AB} \neq |\psi\rangle_A \otimes |\psi\rangle_B. \quad (2.1)$$

If above condition is satisfied for every possible wave functions $|\psi\rangle_A, |\psi\rangle_B$, we say that subsystems A and B are in mixed states, and describing the pure state $|\psi\rangle_{AB}$ by the states from subsystems A and B in general requires using the full basis from both

$$|\psi\rangle_{AB} = \sum_{i,j} c_{ij} |\psi_i\rangle_A \otimes |\psi_j\rangle_B. \quad (2.2)$$

Such construction means that acting on one part of the system (either by collapsing or unitary evolving the wave function), inevitably affects the other, since the coefficients that are changing c_{ij} influence both subsystems. Therefore, A and B become inseparable or entangled with each other.

Although the concept of quantum entanglement is widely accepted as a good description of reality and is experimentally confirmed [12, 13, 14, 15], unlike some other quantities describing quantum phenomena, there is no single agreed definition of a measure of entanglement [16]. Heuristically, such a measure should be proportional to the amount of information about the state of the subsystem A , which can be extracted from the mixed state of subsystem B . Looking at the description of the full state (2.2), it is obvious that the entanglement measure will be some function of coefficients c_{ij} , but it should be basis independent.

There are many entanglement measures currently used in the field of many-body physics [17]. In strongly correlated systems, the entanglement plays a major role in understanding some of the most important phenomena exhibited. Most of them, in general, require microscopic knowledge of the wave function.

To understand the importance of quantum entanglement, a brief history of the development of the concept will be presented along with the well-known controversy. Later in the chapter, some commonly used measures of entanglement will be shown and, using those, an entanglement transition (ET) will be defined for an example of a transverse field anisotropic XY model.

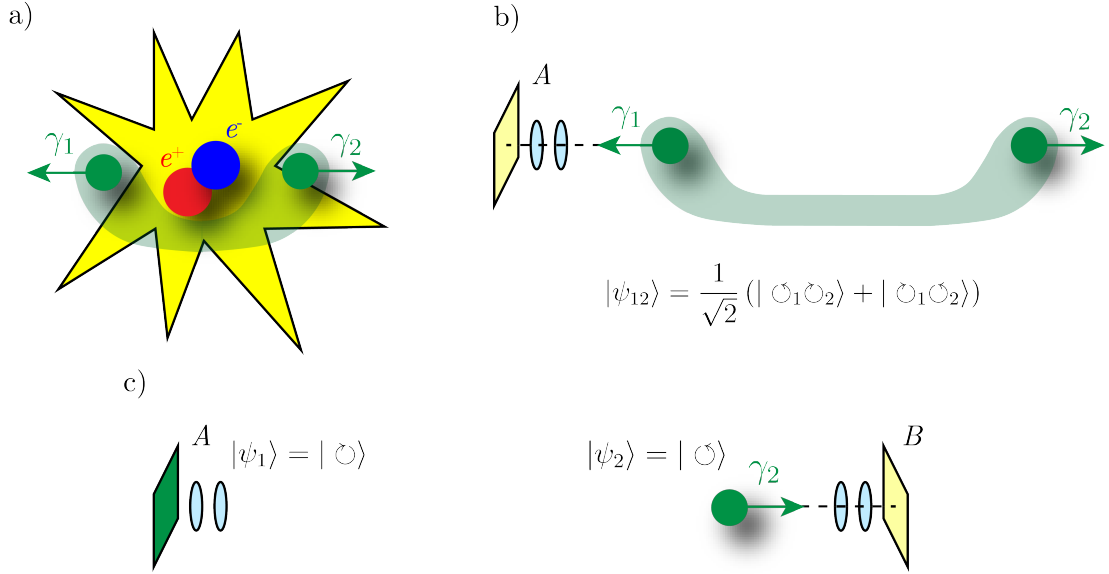


Figure 2.1: An example of the EPR paradox: a) An electron and a positron annihilate each other creating two gamma particles (high energy photons). Assuming that the initial spins of electron and positron were opposed, in order to preserve momentum and angular momentum of the system after the annihilation process, two photons have to travel in opposing directions, and their spin have to cancel each other; b) The wave function describing two receding photons is $|\psi_{12}\rangle = \frac{1}{\sqrt{2}} (|\oplus_1 \oplus_2\rangle + |\ominus_1 \ominus_2\rangle)$. The photons are entangled. Polarisation of the first photon is going to be measured by Alice (A); c) A measurement by Alice collapses the wave function of combined system of two photons, making them entangled no more. A measurement by Bob (B) will now show opposite polarisation to the one measured by Alice. The first measurement seems to affect the second photon instantaneously, violating the principle of local causality.

2.1 History

Quantum entanglement was an early part of quantum theory of physics and led to large disagreement between Einstein and proponents of Copenhagen interpretation, which eventually was resolved by experimental confirmation of violation of Bell's inequality.

2.1.1 Einstein-Podolski-Rosen paradox

The phrase "spooky action at a distance", coined by Alfred Einstein [18], refers to non-locality of interacting quantum systems. Indeed, quantum entanglement can be considered a very unintuitive concept at first. This is most readily shown in famous thought experiment, where an electron and its antiparticle — positron — undergo annihilation. From this event two photons are created, which will have to travel in opposite directions (consequence of conservation of momentum) and will have opposite spin directions (consequence of conservation of angular momentum). Because of the superposition, they both can be in a state with equal probabilities of $+1$ and -1 spin, but after measurement is performed on one of them, the combined wave function will collapse, essentially changing the state of the other photon (see Figure 2.1). Symbolically, we can write the full state of both photons as

$$|\psi\rangle = \frac{1}{\sqrt{2}} (|\oplus\rangle_1 \otimes |\oplus\rangle_2 + |\ominus\rangle_1 \otimes |\ominus\rangle_2). \quad (2.3)$$

Measuring the wave function of the first photon, i.e. acting with the bra vector $\langle\phi|_1$ collapses both the first and second photon wave functions, essentially changing the state of the second photon non-locally.

This thought experiment is known as the Einstein-Podolski-Rosen (EPR) paradox, after the authors of the article published in 1935 [19], which was aimed at criticising quantum mechanics as incomplete or outright wrong picture of reality. The paradoxical nature of this experiment comes from the axiom of locality, which was crucial in Einstein's theory of relativity. According to it, no interaction can travel faster than the speed of light, and it seems like the measurement on one photon causes instant change in the state of the other.

2.1.2 Hidden variables theory

The alternative explanation provided by Einstein was the concept of "hidden variables" [20]. It assumes that the interactions between particles are local and that the evolution of systems is deterministic in nature. Quantum mechanics only appear to be random, but it is due to the fact that there are intrinsic variables that cannot be measured in experiment, which nonetheless affect the result of the measurement. These hidden variables are deterministic and can be influenced by other particles, meaning that both the causality and locality can be preserved in physics. "God does not play dice" is a direct reference to this rejection of Copenhagen interpretation of quantum mechanics.

This dispute is seemingly about the philosophy or interpretation of quantum physics. It only affects the way in which we think about physical reality, and does not have an impact on the results in the laboratory. However, in 1964 John Stewart Bell showed that in general, local hidden variables theory cannot reproduce the results of quantum mechanics exactly, meaning that there are experimental ways to find out if there are any hidden variables affecting the physics. At the time, it was known that there were no discrepancies between quantum theory and experiment, and the later measurements in 1972 [13] confirmed that by reconstructing Bell's test, originally only a thought experiment. Most physicists now accept the "weirdness" of quantum entanglement and the non-locality, although the non-local version of hidden variables theory has not been disproved by Bell's theorem.

2.1.3 Bell's inequality

The assumption behind hidden variables is that the two entangled variables cannot "talk" to each other if they are separated. One of the possible setup testing this claim is known as Clauser-Horne-Shimony-Holt (CHSH) inequality (after the authors of article introducing the concept [12]), where there are two possible sets of experiments for two particles measured independently and apart from each other.

Let us assume that the person at point C prepares two particles, each of which has two general properties P_X, P_Y , that can take on a value of either $+1$ or -1 . From C, a particle is sent to position A (see Figure 2.2). We can denote the value of its properties as X_A and Y_A . The particle A has those values objectively, whether they are measured or not. At position A, only one of the properties of the particle is detected. Whether it is P_X or P_Y is chosen at random with a 50/50 chance (for example, by tossing a coin by a person at A). The same procedure is used for another particle that is sent from C to position B. We denote the values of that particle as X_B and Y_B , the same random process is performed for a measurement and the detection of the value follows. Crucially, everything should be timed so that the detection of particles at A and B happens simultaneously, such that the event of detection at one place does not affect the other.

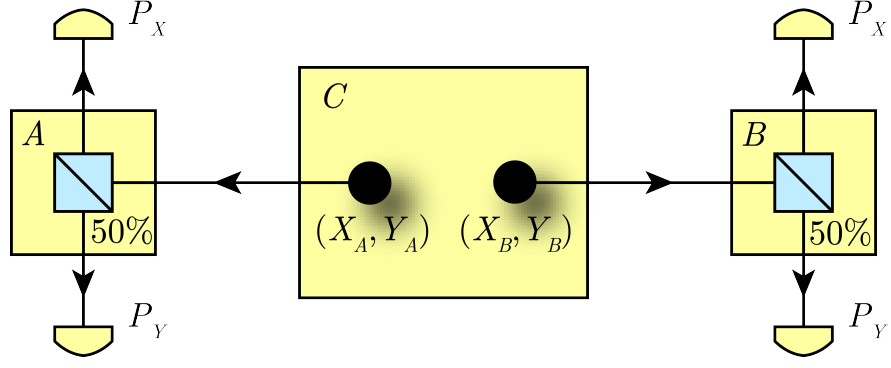


Figure 2.2: CHSH setup. Two particles, each set with values (X, Y) of properties P_X and P_Y , are sent to Alice (A) and Bob (B), where they can only measure one of the properties with 50/50 probability of which measurement is chosen. Carol (C) can prepare the particles in any way she wants, meaning that the kind of correlations that occur in quantum mechanics (for example EPR paradox) can be simulated.

With that experimental setup, we can calculate expectation values of some quantities, which should confirm the assumption of locality. Let us take a closer look at quantity

$$X_A X_B + Y_A X_B + Y_A Y_B - X_A Y_B = (X_A + Y_A) X_B + (Y_A - X_A) Y_B. \quad (2.4)$$

Since it is true that X_A and Y_A can be either $+1$ or -1 , only one of the terms at R.H.S of 2.4 is nonzero. That term will have a value of either $+2$ or -2 . Therefore no matter how the particles are prepared in C and which measurements are chosen at positions A and B , the following statement is true

$$X_A X_B + Y_A X_B + Y_A Y_B - X_A Y_B = \pm 2. \quad (2.5)$$

It follows that the expectation value of that repeated experiment will be

$$\begin{aligned} E(X_A X_B + Y_A X_B + Y_A Y_B - X_A Y_B) = \\ E(X_A X_B) + E(Y_A X_B) + E(Y_A Y_B) - E(X_A Y_B) \leq 2. \end{aligned} \quad (2.6)$$

The equation above is one of the examples of Bell's inequality. Notice that, unlike with the EPR paradox, there is no assumption on the correlation between properties P_X and P_Y of two particles. A person who sends both particles from C can prepare them however they like. Nor do we specify how a local measurement can alter the properties of the particle. What is important is that particles A and B are measured at the same time (and therefore, the measurement of one cannot affect the other causally) and the values of their properties are objectively real, even when not measured. However, a specific prediction from quantum mechanics can violate inequality (2.6).

Let us assume that the properties of particles in this example are the projections of a spin direction on the proper axis. If the two particles are prepared in a mixed Bell state [12, 21]:

$$\psi = \frac{1}{\sqrt{2}} (|\uparrow\rangle_A \otimes |\downarrow\rangle_B - |\downarrow\rangle_A \otimes |\uparrow\rangle_B), \quad (2.7)$$

and the measurements are corresponding to

$$\begin{aligned} X_A &= \sigma_z^A, & Y_A &= \sigma_x^A \\ X_B &= -\frac{1}{\sqrt{2}} (\sigma_z^B + \sigma_x^B) & Y_B &= \frac{1}{\sqrt{2}} (\sigma_z^B - \sigma_x^B), \end{aligned} \quad (2.8)$$

where the σ_α^Γ are Pauli matrices. One can calculate that in such example, the expectation values are

$$\begin{aligned} E(X_A X_B) &= E(Y_A X_B) = E(Y_A Y_B) = \frac{1}{\sqrt{2}} \\ E(X_A Y_B) &= -\frac{1}{\sqrt{2}}, \end{aligned} \tag{2.9}$$

and after substituting those numbers to (2.6), the L.H.S is equal to $2\sqrt{2}$ and the inequality is not satisfied.

2.1.4 Experimental confirmation of violation of Bell's inequality

The example of a CHSH inequality clearly shows that there is a discrepancy between the results from hidden variables and quantum theories. Which one is correct? That can be answered by setting up a real-world experiment in a manner similar to the CHSH setup and equations (2.7) and (2.8). Usually these type of experiments are referred to as Bell tests [22]. The first Bell tests were performed by John Clauser and Stuart Freedman in 1972 [13], for which John Clauser was awarded a Nobel Prize in 2022 [23]. This experiment and the many that came after it confirmed that nature violates Bell's inequalities, and thus the local version of hidden variables theory is not correct. For now, the best description of microscopic physics is given by quantum theory.

2.2 Quantifying entanglement

The philosophical considerations of the quantum entanglement are still debated to this day. More practical applications of our understanding of quantum entanglement have also been proposed, which mainly began with the developments in the field of quantum information, such as quantum computing [24], quantum teleportation [25], and quantum cryptography [26]. In these fields of research, entanglement is often viewed as a resource, allowing for a speed up in calculations in quantum computers and being an essential tool in quantum communication protocols [27, 28]. Over time, this also influenced research in many-body physics, which, among others, led to an emerging field of tensor network (TN) methods [29], where entanglement plays a major role as an indicator of the viability of the method. The connection between TNs and the entanglement is explained in more detail in Chapter 3. In this section, I will present few entanglement measures commonly used in quantum many-body physics starting from ones that are defined between single sites and moving towards ones that are defined for particular bisections of large systems.

2.2.1 Von Neumann entropy and Renyi entropies

The most common measure of the entanglement is the von Neumann entropy defined for the partition of any quantum system. Supposing that the subsystem of a larger system is in a mixed state (that is, the entanglement between it and the remaining part is not zero), it can be described by the density matrix ρ . The von Neumann entropy can be written then as [17]

$$S(\rho) = -\text{Tr} [\rho \ln \rho], \tag{2.10}$$

which mirrors the definitions of entropy from statistical physics and information theory. Since the density matrix can be diagonalised and written as

$$\rho = \sum_i \lambda_i |\phi_i\rangle\langle\phi_i|, \quad (2.11)$$

it follows that the von Neumann entropy becomes

$$S(\rho) = - \sum_i (\lambda_i) \ln \lambda_i. \quad (2.12)$$

Therefore, the λ_i values define the entanglement entropy of a system. Importantly, if the state is pure, diagonalizing ρ should return just one nonzero eigenvalue (which will be equal to one since ρ should be normalised) and one can check that the von Neumann entropy goes to zero in such a case.

Mimicking the definition from thermal density matrix¹, the eigenvalues can be instead written as logarithms²: $\xi_i = -\ln \lambda_i$, and called entanglement energies, or entanglement spectrum as a set which points to corresponding Schmidt eigenstates. The full information about the entanglement between a subsystem and the remaining part lies in the entanglement spectrum³.

The generalisation of von Neumann entropy are the Rényi entropies, defined with the parameter α as

$$H_\alpha(\rho) = \frac{1}{1-\alpha} \ln \sum_i \lambda_i^\alpha. \quad (2.13)$$

In the limit of $\alpha \rightarrow 1$, the Rényi entropy $H_1(\rho)$ will be equal to the von Neumann entropy. Usually, the application of Rényi entropies is due to the different values obtained from the entanglement spectrum [31]. With larger α , the contribution from states that have the largest eigenvalues will be greater then for the Rényi entropy of smaller α .

While both von Neumann and Rényi entropies are good measures of entanglement, because they require microscopic knowledge of the material, they are not accessible experimentally. This is why there exist a need for other metrics which can be defined in terms of observables that can be measured. Usually, the definition of such metrics will be in their relationship to entanglement entropies.

2.2.2 1-tangle

To classify an average amount of entanglement between single site spin and the rest of the system, we can use the determinant of the reduced density matrix [32]

$$\tau_1(j) = 4 \det \rho_j^{(1)}, \quad (2.14)$$

where j is referencing j -th spin site, $\rho_j^{(1)} = \text{Tr}_{B_j} [\rho]$, and B_j is all of the sites in the system excluding j . In the thermodynamic limit, if the system has spatial symmetry, the determinant will not depend on the position j and can be written simply as τ_1 . This measure is called 1-tangle [17, 32, 33] and while it is not obvious that it directly relates to entanglement, it was

¹ $\rho(T) = \sum_i e^{\frac{E_i}{T}} |\phi_i\rangle\langle\phi_i|$

²Because the density matrix is semi-positive definite, its eigenvalues are positive and therefore the logarithm is real.

³Interestingly, the entanglement spectrum can be connected to topological properties of a physical edge state via Li-Haldane conjecture [30].

proven that there exist a direct connection between it and the von Neumann entropy. The most important feature of 1-tangle is that it goes to zero when the system is factorised i.e. when all spins act as if they are separate from each other⁴.

For magnetic Hamiltonians, the 1-tangle has the advantage of being easy to calculate, since the determinant (2.14) can be shown to be a function of the magnetisation:

$$\tau_1 = 1 - 4 \sum_{\alpha=x,y,z} (M^\alpha)^2, \quad \text{where } M^\alpha = \langle S^\alpha \rangle. \quad (2.15)$$

It is important to note that, although 1-tangle has a particularly easy form in spin systems, it contains pairwise and higher multipartite entanglement between spins. However, the information about their ratios cannot be read just from this measure, and so other metrics are often needed to get the full picture of the entanglement.

2.2.3 Concurrence and 2-tangle

Similarly to the 1-tangle, one can think of a quantity that would contain information about the pairwise entanglement in the system. The concurrence [32] serves as such a quantity. For magnetic systems it can be defined as a function of the separation between two spins r as⁵ $C_r = 2\max\{0, C'_r, C''_r\}$ where

$$C'_r = |\langle S_i^y S_{i+r}^y \rangle + \langle S_i^z S_{i+r}^z \rangle| - \sqrt{\left(\frac{1}{4} + \langle S_i^x S_{i+r}^x \rangle\right)^2 - (m^x)^2} \quad (2.16)$$

$$C''_r = \langle S_i^x S_{i+r}^x \rangle + |\langle S_i^y S_{i+r}^y \rangle - \langle S_i^z S_{i+r}^z \rangle| - \frac{1}{4}.$$

The sum of the squares of concurrences is called the 2-tangle $\tau_2 = \sum_{r \neq 0} C_r^2$, and quantifies the total pairwise site-site entanglement in the system.

Looking at equation (2.16), it is again not obvious that it relates to entanglement at all. The relationship comes from the notion of “entanglement of formation” [34, 33], which is the amount of entanglement needed to copy the current state. The authors of [34] show how such a measure can be extracted from the eigenvalues of the Hermitian conjugate matrix $R_{jk} = \sqrt{\rho_{jk}^{(2)}}((\sigma_y)_j \otimes (\sigma_y)_k)(\rho_{jk}^{(2)})^*((\sigma_y)_j \otimes (\sigma_y)_k)\sqrt{\rho_{jk}^{(2)}}$, where $\rho_{jk}^{(2)}$ is the reduced density matrix obtained by tracing out all sites except j -th and k -th. The concurrence is defined through those eigenvalues. Importantly, the 2-tangle will go to zero when there is no entanglement in the system, just as the 1-tangle. Another useful property is the monogamy inequality [35], which states that $\tau_1 \geq \tau_2$. This means that the 2-tangle stores just the information about the pairwise entanglement, and the residual quantity $\tau_1 - \tau_2$ is the amount of multipartite entanglement. Since both τ_1 and τ_2 are measurable in spin systems through definitions (2.14) and (2.16), a signature of multipartite entanglement is accessible in the experiment.

2.2.4 Quantum fisher information

While combined 1-tangle and 2-tangle measures can provide information about multipartite entanglement indirectly, there are other quantities that are more directly connected to it.

⁴Which means that they are not entangled with any other spin in the system.

⁵The full definition can be found for example in [17]

One of such measures is quantum fisher information (QFI), which is defined through a sum of the expectation values of local operators [36, 37]. In the case of magnetic systems, those local observables are spin operators, and it was shown [38, 39] that QFI can be expressed with two-point correlation functions as

$$f_Q(k, T) = \frac{2}{N^d \pi} \int_{-\infty}^{\infty} d\omega \tanh^2\left(\frac{\omega}{2T}\right) S(\omega, k), \quad (2.17)$$

where $S(\omega, k)$ is the structure factor:

$$S(\omega, k) = FT[S(t, j)] = FT \left[\sum_{\alpha \in \{x, y, z\}} \langle S_{N/2+j}^{\alpha}(t) S_{N/2}^{\alpha}(0) \rangle \right]. \quad (2.18)$$

How does it relate to the entanglement? The QFI was shown [38, 39] to obey a simple inequality, namely:

$$f_Q > m, \quad (2.19)$$

where m is the divisor of the system, meaning that we can divide it into $m + 1$ subsystems that are completely disentangled. More specifically the full wave function ψ with a divisor m can be expressed as:

$$|\psi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle \otimes \dots \otimes |\phi_m\rangle \otimes |\phi_{m+1}\rangle. \quad (2.20)$$

Inequality (2.19) gives an upper bound on the amount of factorisation of the wave function through the observables that in principle can be obtained from the experiment. Lower QFI signifies more entanglement in the system, since the state cannot be written in a more factorised way. Unfortunately, the QFI does not tell anything about the lower bound on m . Nevertheless, it still provides a useful insight where potentially there could be a large factorisation in the phase diagram, especially since it is a function of temperature.

2.3 Entanglement transition

Entanglement provides additional insights into the properties of many-body interacting systems. The usual investigation into interesting Hamiltonians often focuses on identifying and characterising phase transitions and critical points on a phase diagram. In statistical physics, phase transitions are connected to the interplay between thermal fluctuations and some ordering mechanism coming from the interactions between particles, spins, etc. Quantum mechanics additionally gives rise to so-called quantum fluctuations, which can come from, for example, the uncertainty rule regarding different orientations of the spin. This is why the quantum phase transition (QPT) can occur even at zero temperature. Moreover, at finite temperature, the sharp transition at a single critical point broadens and becomes another phase. It turns out [40, 41, 42] that in some systems, there can be another type of transition, where the wave function of the system becomes at one point fully factorised i.e. all of the local particles are in pure states. This is sometimes called an “entanglement transition”, due to the fact that the particles at the factorisation point are completely disentangled, and the nature of the entanglement between two phases is usually very different.

To illustrate the difference between QPT and ETs, let us consider an example of an anisotropic XY spin chain model with a perpendicular field. The Hamiltonian of such a system is as follows

$$H_{XY} = \sum_j (1 + \gamma) S_j^x S_{j+1}^x + (1 - \gamma) S_j^y S_{j+1}^y + h S_j^z. \quad (2.21)$$

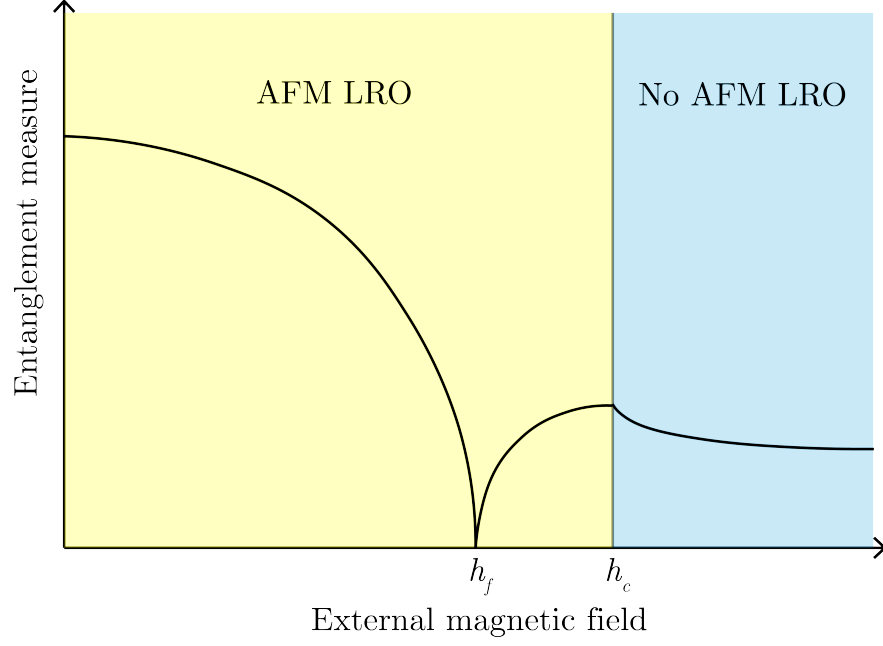


Figure 2.3: Sketch of the ET for the anisotropic XY Heisenberg model in the thermodynamic limit. The AFM long-range interactions (AFM LRO) prevail in the presence of external magnetic field up to the $h = h_c = 1$, where the quantum fluctuations are strong enough to destroy AF order. At $h = h_f = \sqrt{1 - \gamma^2}$, the wave function becomes fully factorised and entanglement measures such as entanglement entropy, 1-tangle or concurrence will go to 0 at that point. At critical point, the nature of the entanglement is changed as well.

In the thermodynamic limit, the XY model has a QPT at the critical point $h_c = 1$, where the quantum fluctuations caused by the transverse field are strong enough to get from the ordered antiferromagnetic (AFM) phase to disordered phase. This is irrespective of the anisotropy parameter γ . However, at $h_f = \sqrt{1 - \gamma^2}$, the ground state is factorised, and with no entanglement, the spins behave as a classical system. This is surprising, in part because the Hamiltonian is not classical at h_f , and also the factorisation point and the critical points are separate (if the anisotropy parameter is larger than zero), meaning that the ground state fundamentally changes within the ordered phase. This means that entanglement plays a large role in understanding the fundamental aspects of many-body physics.

The entanglement measures mentioned before in this chapter can be used to identify ETs. Figure 2.3 shows an example sketch of such an entanglement measure for an anisotropic XY model, with the factorisation and critical fields marked. For an isotropic model ($\gamma = 0$), those two critical fields would be identical, meaning that the moment in which the wave function becomes fully factorised is the same as the system changing from an ordered to disordered phase. Introduction of anisotropy separates those two points and the factorisation is totally within the ordered phase, meaning that those two behaviours are not intrinsically connected.

The models with ETs can provide a good test for a machine learning algorithm that learns to predict the entanglement. This will be explored in more detail in Chapter 7 of this thesis.

Chapter 3

Tensor networks and calculation of correlation functions

Introduced in the research by Stephen White [43], matrix product states (MPSs) are a way of representing interacting many-body quantum states while storing information about quantum entanglement between any bipartition of that system. They proved to be a very effective tool for finding ground states and calculating observables for strongly correlated systems, especially in 1-D cases. They also have a useful graphical notation, which often allows for calculation to be simplified by manipulation of diagrams, rather than through equations. More generally, the techniques that came from studying MPS are now called tensor networks (TNs) [44] and are widely used in computational condensed matter physics and other connected fields of study. Depending on the particular system, they can outperform many similar techniques such as variational Monte Carlo, neural networks, or quantum computing. As an example, it was shown that recent result of Quantum Supremacy [45] was outperformed by a classical computer using TNs [46].

To achieve the research goals of this thesis, we have opted to use TNs in order to generate data from larger systems than exact diagonalization would allow for. Apart for calculating observables such as static/dynamic correlators at finite temperature, TNs provide a natural way to obtain information about the entanglement for any bipartition of the system. Both of those facts served as a justification for choosing this numerical method.

This chapter is a first part of explaining the numerical methods used in the research of this thesis (the second being the machine learning techniques in the following chapter). An introduction to MPS and TN graphical notation will be shown, with specifics on how to calculate the local observables in an efficient fashion. Following that, the concept behind matrix product operators (MPOs) will be presented along with the representation of a density matrix operator in MPO form, which will lead to a discussion on how to calculate finite temperature and time-dependent density matrices effectively, using methods such as disentanglers.

3.1 Tensor networks and matrix product states

The beginnings of modern TN methods can be traced back to Wilson's solution of the Kondo problem by the renormalization group technique [47]. In the later article by S. R. White [48], he formulated the general method of representing 1-D quantum systems through the reduced density matrix of two parts of a system (left and right). Building upon that work, the computational physicist community has created a framework for techniques

aimed at calculating large systems effectively. The following section serves as an introduction to TN methods, which closely follows some of the more comprehensive texts on the subject (for example, [44], [5], [6]).

3.1.1 Tensors and graphical notation

In the context of Hilbert spaces¹, tensors can be described as generalizations of vectors and matrices to n -dimensional arrays of numbers, where n is called the rank of the tensor. Scalars, vectors, and matrices are 0-rank, 1-rank, and 2-rank tensors, respectively. The standard notation for a general n -rank tensor is

$$T_{a_1 a_2 a_3 \dots a_n}, \quad (3.1)$$

where each index a_k runs through $a_k = 1, 2, 3, \dots, d_k$. This notation is both used as a representation of the tensor (meaning the entire array) and a single element of the tensor, depending on the context. Elements are usually real or complex numbers. The algebraic operations on tensors consist of:

- Addition of two tensors S and R . If both tensors have the same dimensions then the elements of resulting tensor are the sum of elements of S and R :

$$T_{a_1 a_2 a_3 \dots a_n} = S_{a_1 a_2 a_3 \dots a_n} + R_{a_1 a_2 a_3 \dots a_n} \quad \text{for all } a_k, \quad k = 1, 2, 3, \dots, n \quad (3.2)$$

- Inner product or contraction of two tensors

$$T_{a_1 \dots a_s a_{s+2} \dots a_n b_1 \dots b_r b_{r+2} \dots b_m} = \sum_c S_{a_1 \dots a_s c a_{s+2} \dots a_n} R_{b_1 \dots b_r c b_{r+2} \dots b_m}. \quad (3.3)$$

T in this case is a $(n + m - 2)$ -rank tensor, because the summation over the c index in both S and R has contracted this dimension. c does not have to be in the same position in S and R . The only condition is that those dimensions have to be equal in size (while other dimensions $a_1, a_2, \dots$ and $b_1, b_2, \dots$ generally do not). Matrix and matrix-vector multiplication are examples of a tensor contraction.

- Outer product or tensor product, defined as

$$T_{a_1 a_2 \dots a_n b_1 b_2 \dots b_m} = S_{a_1 a_2 \dots a_n} R_{b_1 b_2 \dots b_m}. \quad (3.4)$$

T in this case is a $(n + m)$ -rank tensor with each element being a multiplication of elements from S and R .

The graphical notation of tensors is presented in Figure 3.1. Usually, the blocks are representations of tensors as objects, with the lines coming from the blocks representing indices. The index is "free" if it is not connected to the index of another tensor. The connection of two indices makes a contraction between two tensors² along those dimensions. The efficient representation of contractions is actually the main advantage of this notation. Outer products are rarely introduced in those diagrams (usually by placing two tensors near each other). Element-wise additions can be expressed by writing the sum within the single box. Nevertheless, it proves to be a useful tool when creating representations of quantum states.

¹Tensors can be more generally defined for more complex manifolds than flat spaces [49].

²The contraction can be also between two indices from the same tensor

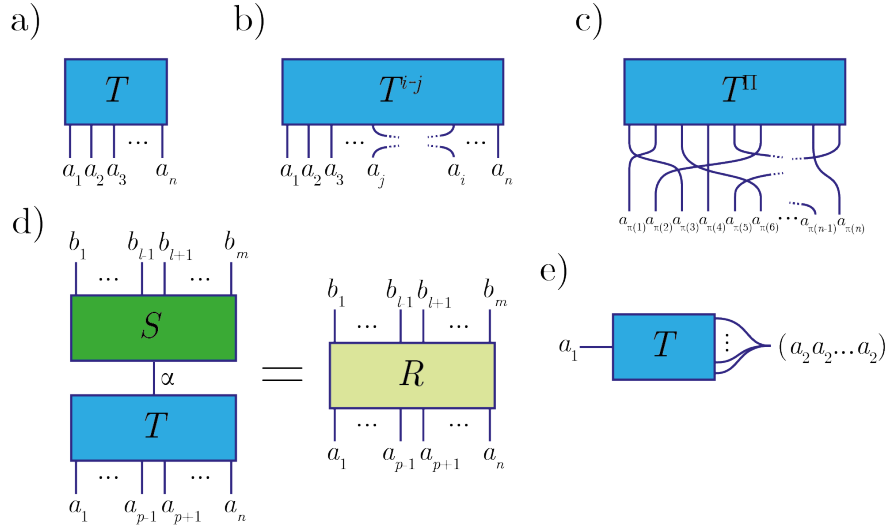


Figure 3.1: Diagrammatical representation of tensors and basic operations/manipulations of tensors: a) A single n -rank tensor is drawn as a box (or a node) with n legs representing indices $a_1, a_2, \dots$; b-c) The exchange of the legs is the permutation of indices of a tensor; d) If two tensors are connected with a single (or more) legs, this indicates a contraction of that index between them, which can be changed to represent a single tensor; e) Joining legs together creates a composite index.

Multiple tensors connected by different contraction form a type of graph in this notation (with tensor as nodes and indices forming connections); hence the name tensor networks.

Apart the algebraic operations, there are additional manipulations that can be performed on TNs. Among the most important are

- Transposition of two indices

$$T_{a_1 a_2 \dots a_i \dots a_j \dots a_n}^{(i \leftrightarrow j)} = T_{a_1 a_2 \dots a_j \dots a_i \dots a_n}, \quad (3.5)$$

which in graphical notation means switching two lines with each other.

- Permutation of many indices

$$T_{a_1 a_2 a_3 \dots a_n}^\Pi = T_{a_{\pi(1)} a_{\pi(2)} a_{\pi(3)} \dots a_{\pi(n)}}. \quad (3.6)$$

Generalisation of transposition to multiple indices at the same time.

- Combining multiple indices into one

$$T_{a_1 a_2 a_3 \dots a_n} = T_{a_1, (a_2 a_3 \dots a_n)}. \quad (3.7)$$

In the above example, T changes from the n -rank tensor to a matrix. The reshaping of a composite index can be done in multiple ways. An example of an ordering for two conjoined indices a_1 and a_2 with dimensions d_1, d_2 could be $I = a_1 \times d_2 + a_2$.

- Expansion of a tensor into two with contraction. It can be shown that it is always possible to write any tensor as a inner product of two different tensors

$$T_{a_1 a_2 a_3 \dots a_n} = \sum_{\alpha} S_{a_1 \alpha} R_{\alpha a_2 a_3 \dots a_n}. \quad (3.8)$$

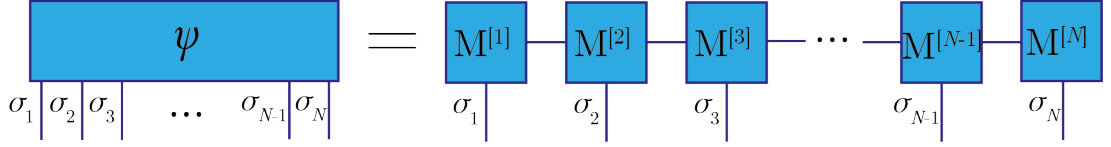


Figure 3.2: Matrix product state representation for a 1-D chain of N spins. The N -rank tensor on the left is a full wave function, while tensors on the R.H.S each correspond to a local contribution to the information about the state.

where in this case the matrix $S_{a_1\alpha}$ is separated from the remaining tensor R , by introducing the virtual index α . In the graphical notation this would be represented by one tensor being replaced by two, with the α index connecting them both (see Figure 3.1 d). This representation of T is not unique, but there are various situations where choosing a particular form of S (which also defines R) is convenient, as will be shown later in the chapter.

3.1.2 Matrix product states

To apply the advantages of TN representation to the quantum physics of many-body systems, it is usually convenient to start with tensor representation of a wave function. Since the traditional language of quantum mechanics is described through linear algebra of matrices and vectors, the generalisation to tensors is quite straightforward. Suppose that the quantum system of interest is a 1-D chain of N spins (or other connected sites with two-level degree of freedom) - a common example in a condensed matter physics problem. A general wave function describing such system is given by

$$|\psi\rangle = \sum_{\sigma_1\sigma_2\sigma_3\ldots\sigma_N} \psi_{\sigma_1\sigma_2\sigma_3\ldots\sigma_N} |\sigma_1\sigma_2\sigma_3\ldots\sigma_N\rangle, \quad (3.9)$$

where $\sigma_i = 1, -1$ is the spin component along z axis of the i -th site in a chain, and $\psi_{\sigma_1\sigma_2\sigma_3\ldots\sigma_N}$ are complex coefficients which hold all the information about the wave function. Instead of thinking of these as a set of numbers, one can treat $\psi_{\sigma_1\sigma_2\sigma_3\ldots\sigma_N}$ as N -rank tensor, which in graphical notation is just a single node with N free legs. Evaluating an amplitude of some given configuration would be equal to connecting those free legs with vectors $|{-1}\rangle_i = (1, 0)$ and $|1\rangle_i = (0, 1)$ according to the given configuration of spins

$$\langle(1)(-1)(-1)(1)(1)\ldots(-1)|\psi\rangle = \psi_{(1)(-1)(-1)(1)(1)\ldots(-1)}. \quad (3.10)$$

The amount of numbers that must be stored in $\psi_{\sigma_1\sigma_2\sigma_3\ldots\sigma_N}$ grows exponentially, and therefore studying larger systems numerically would become prohibitively difficult. However, using a particular form of expansion formula (3.8) (a general way of doing this will be shown in the later section as canonical form) one can transform coefficients in the wave function into a product of matrices

$$\begin{aligned} \psi_{\sigma_1\sigma_2\ldots\sigma_N} &= \sum_{a_1a_2a_3\ldots a_{N-1}} M_{1,a_1}^{[1];\sigma_1} M_{a_1,a_2}^{[2];\sigma_2} M_{a_2,a_3}^{[3];\sigma_3} \ldots M_{a_{N-2},a_{N-1}}^{[N-1];\sigma_{N-1}} M_{a_{N-1},1}^{[N];\sigma_N} \\ &= \mathbf{M}^{[1];\sigma_1} \mathbf{M}^{[2];\sigma_2} \mathbf{M}^{[3];\sigma_3} \ldots \mathbf{M}^{[N-1];\sigma_{N-1}} \mathbf{M}^{[N];\sigma_N}. \end{aligned} \quad (3.11)$$

The graphical representation of that transformation is shown in Figure 3.2. In general, the amount of information that is stored in matrices $\mathbf{M}$ is the same as the full wave function. In practice, it is often possible to reduce the sizes of $\mathbf{M}$ either with or without approximation.

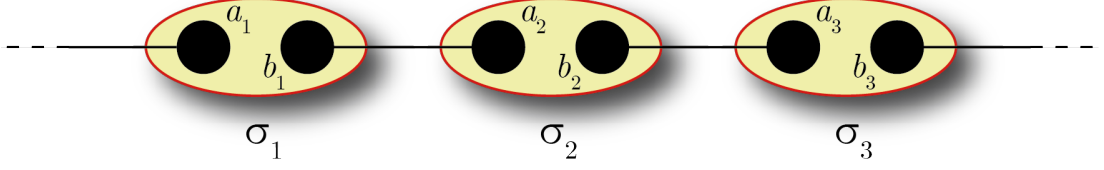


Figure 3.3: The representation of AKLT ground state. Each $S=1$ site σ_i can be divided into two spin- $\frac{1}{2}$ partons: a_i and b_i , which connect through singlet interactions with their neighbours from the other sites.

3.1.3 AKLT model

An example of non-trivial classical representation of a quantum state which is much simpler in an MPS representation is the Affleck-Kennedy-Lieb-Tasaki (AKLT) state [50]. It is a ground state of a spin-1 chain model with a isometric Heisenberg interaction with an additional quadratic term

$$H_{\text{AKLT}} = \sum_i \mathbf{S}_i \cdot \mathbf{S}_{i+1} + \frac{1}{3} (\mathbf{S}_i \cdot \mathbf{S}_{i+1})^2. \quad (3.12)$$

The full ground state of that model is not easy to write. However, by replacing the spin-1 sites $\sigma_i \in \{|-1\rangle, |0\rangle, |1\rangle\}$ by two spin- $\frac{1}{2}$ sites $a_i, b_i \in \{|\uparrow\rangle, |\downarrow\rangle\}$ in a triplet state, i.e.

$$\begin{aligned} |1\rangle &= |\uparrow\uparrow\rangle \\ |0\rangle &= \frac{1}{\sqrt{2}} (|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle) \\ |-1\rangle &= |\downarrow\downarrow\rangle, \end{aligned} \quad (3.13)$$

the ground state can be shown to be a singlet connections between neighbouring spin- $\frac{1}{2}$ sites, coupled with projections to the $S=1$ subspace for each pair (see Figure 3.3). For the simplest non-trivial case of two-dimensions³ of a chain of length L , the state which merely enforces the singlet constraint is

$$|\tilde{\psi}_{\text{AKLT}}\rangle = \sum_{\{\mathbf{a}\}, \{\mathbf{b}\}} S_{b_1, a_2} S_{b_2, a_3} S_{b_3, a_4} \dots S_{b_{N-1}, a_N} S_{b_N, a_1} |\mathbf{a}, \mathbf{b}\rangle, \quad (3.14)$$

where $\mathbf{a} = (a_1, a_2, \dots, a_L)$, $\mathbf{b} = (b_1, b_2, \dots, b_L)$, and $\mathbf{S}$ is the singlet matrix:

$$\mathbf{S} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \quad (3.15)$$

The above state is in the basis of spin- $\frac{1}{2}$ sites. In order to express it in the original spin-1 chain, the mapping given by (3.13), in matrix form

$$\mathbf{M}^{+1} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \mathbf{M}^0 = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \mathbf{M}^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad (3.16)$$

³Meaning that terms $\mathbf{S}_i$ in the Hamiltonian (3.12) have only x and y components.

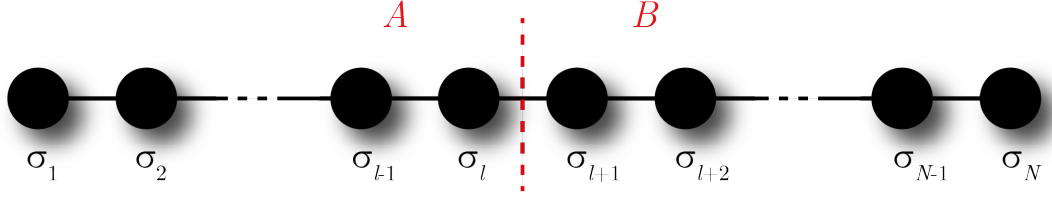


Figure 3.4: Bipartition of a 1-D chain of spins into subsystems A and B.

can be inserted into the pairwise singlet state, in order to construct the ground state of the AKLT model: (3.14)

$$|\psi_{\text{AKLT}}\rangle = \sum_{\{\sigma\}, \{\mathbf{a}\}, \{\mathbf{b}\}} M_{a_1, b_1}^{\sigma_1} S_{b_1, a_2} M_{a_2, b_2}^{\sigma_2} S_{b_2, a_3} \dots M_{a_{N-1}, b_{N-1}}^{\sigma_{N-1}} S_{b_{N-1}, a_N} M_{a_N, b_N}^{\sigma_N} S_{b_N, a_1} |\sigma\rangle. \quad (3.17)$$

By joining together transformation and the singlet matrices for each site

$$A_{a_i, a_{i+1}}^{\sigma_i} = \sum_{b_i} M_{a_i, b_i}^{\sigma_i} S_{b_i, a_{i+1}}, \quad (3.18)$$

the final expression for the coefficients standing in front of a given $|\sigma\rangle$ configuration is the trace of the product of matrices $\mathbf{A}^4$

$$|\psi_{\text{AKLT}}\rangle = \sum_{\{\sigma\}} \text{Tr} [\mathbf{A}^{\sigma_1} \mathbf{A}^{\sigma_2} \mathbf{A}^{\sigma_3} \dots \mathbf{A}^{\sigma_N}] |\sigma\rangle, \quad (3.19)$$

where $\mathbf{A}$ are the same for each site of spins

$$\mathbf{A}^{+1} = \begin{bmatrix} 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 \end{bmatrix} \quad \mathbf{A}^0 = \begin{bmatrix} -\frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \quad \mathbf{A}^{-1} = \begin{bmatrix} 0 & 0 \\ -\frac{1}{\sqrt{2}} & 0 \end{bmatrix}. \quad (3.20)$$

The state $|\psi_{\text{AKLT}}\rangle$ is not normalised yet. It can be shown [5], that in order to normalise it in the thermodynamic limit $N \rightarrow \infty$, the matrices $\mathbf{A}$ have to be put into left-canonical form⁵.

The above transformations show that the AKLT state can be written in the matrix product form, with the entire information about the state encoded in just three 2x2 matrices, irrespective of the size of the system. Of course, this has been possible because of the nature of this particular system, but any state can be expressed as a MPS. A clever way to do this is to use QR or Schmidt decompositions to build a canonical form of an MPS.

3.1.4 Schmidt decomposition

Suppose that there is a state describing a 1-D system of interacting particles (see Figure 3.4) $|\psi\rangle$. Given any bipartition of the chain, which splits the system into parts A and B, there is a way to write down the state as a tensor product between states from left and right subsystems. In general, one can use the basis states from A and B

$$|\psi\rangle = \sum_{\alpha\beta} \psi_{\alpha\beta} |\alpha\rangle_A \otimes |\beta\rangle_B. \quad (3.21)$$

⁴The trace comes from the fact that in the sum (3.17), the last spin- $\frac{1}{2}$ site is the first, i.e. this is a lattice with periodic boundary conditions. For open boundary conditions there is no connection between a_1 and b_N and therefore first and last matrices become vectors.

⁵Meaning scaling them so that $\sum_{\sigma} (\mathbf{A}^{\sigma})^{\dagger} \mathbf{A}^{\sigma} = \mathbf{I}$.

The choice of basis $|\alpha\rangle_A$ and $|\beta\rangle_B$ is arbitrary, and it can be shown [5] that for each state there exist a unique basis such that

$$|\psi\rangle = \sum_n \Lambda_n |n\rangle_A \otimes |n\rangle_B. \quad (3.22)$$

Finding that unique transformation from a generic basis $|\alpha\rangle_A \rightarrow |n\rangle_A$, $|\beta\rangle_B \rightarrow |n\rangle_B$ is called Schmidt decomposition, while Λ_n are Schmidt coefficients. More information on Schmidt decomposition can be found in many introductory texts on TNs [5, 44]. There are two important connections that are worth mentioning here for later concepts:

- Schmidt coefficients are directly linked to the von Neumann entropy between subsystems A and B. For example, writing down the density matrix of subsystem B using (3.22)

$$\begin{aligned} \rho_B &= \text{Tr}_A [|\psi\rangle\langle\psi|] = \sum_{\tilde{n}} \sum_{n,n'} \Lambda_n \Lambda_{n'} (\langle\tilde{n}|_A \langle n|_A \otimes |n\rangle_B) (\langle n'|_A \otimes \langle n'|_B) (|\tilde{n}\rangle_A) \\ &= \sum_{\tilde{n}} \sum_{n,n'} \Lambda_n \Lambda_{n'} (\delta_{n\tilde{n}} |n\rangle_B) (\delta_{n'\tilde{n}} \langle n|_B) = \sum_{n,n'} \Lambda_n \Lambda_{n'} \delta_{nn'} |n\rangle_B \langle n'|_B \\ &= \sum_n (\Lambda_n)^2 |n\rangle_B \langle n|_B, \end{aligned} \quad (3.23)$$

one can write down the entanglement entropy as

$$S = -\text{Tr} [\rho_B \ln(\rho_B)] = -\sum_n (\Lambda_n)^2 \ln((\Lambda_n)^2). \quad (3.24)$$

And of course the same result is obtained by taking trace over system B.

- In terms of calculations, Schmidt decomposition is equal to a singular value decomposition (SVD) of a matrix Ψ which components are the ψ_{ij} coefficients standing in front of the tensor product in (3.21):

$$\Psi = \mathbf{U} \mathbf{\Lambda} \mathbf{V}^\dagger, \quad (3.25)$$

where the $\mathbf{\Lambda}$ is a diagonal matrix with Λ_n coefficients, and $\mathbf{U}$, $\mathbf{V}$ are the transformation basis matrices for $|i\rangle_A \rightarrow |n\rangle_A$ and $|j\rangle_B \rightarrow |n\rangle_B$ respectively. The matrices U and V are semi-unitary, which mean they both obey:

$$\mathbf{U}^\dagger \mathbf{U} = \mathbf{V}^\dagger \mathbf{V} = \mathbf{I}, \quad (3.26)$$

however, the identity relationship does not generally hold for other order of multiplication $\mathbf{U} \mathbf{U}^\dagger = \mathbf{V} \mathbf{V}^\dagger \neq \mathbf{I}$.

3.1.5 Canonical form

The MPS representation of a quantum state is not unique. Given a wave function $\psi_{\sigma_1 \sigma_2 \dots \sigma_N}$, as presented in (3.11), the following transformation of matrices $\mathbf{M}$, leaves the whole wave function invariant:

$$\mathbf{M}^{[i];\sigma_i} \rightarrow \mathbf{M}'^{[i];\sigma_i} = \mathbf{U}^{[i]} \mathbf{M}^{[i];\sigma_i} (\mathbf{U}^{[i+1]})^{-1}. \quad (3.27)$$

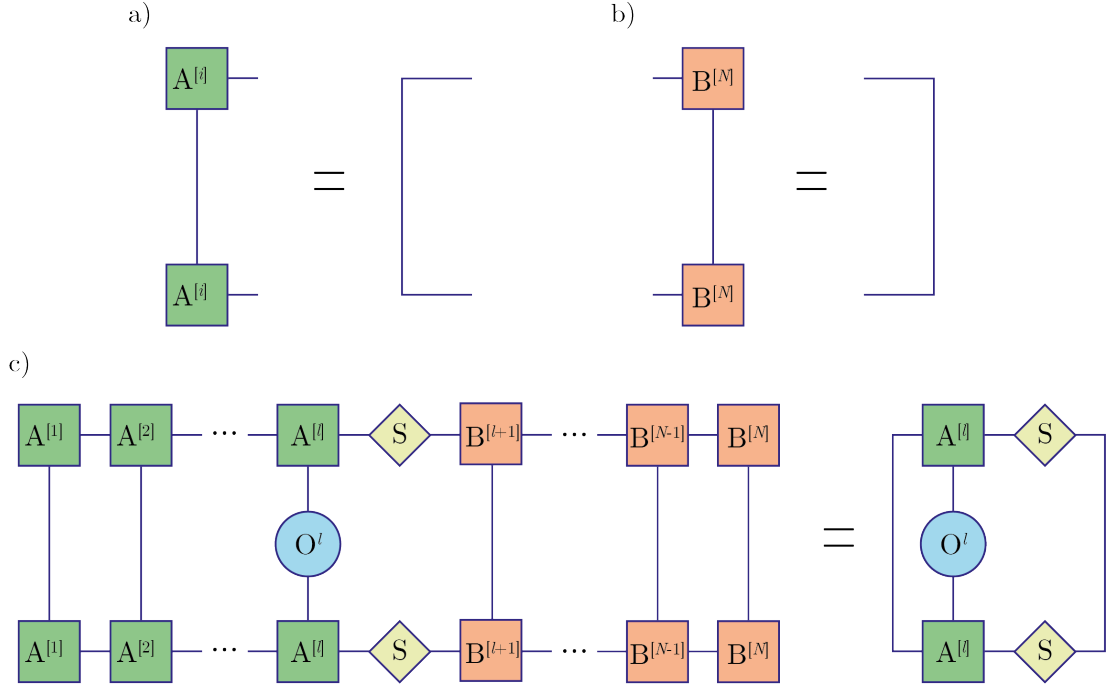


Figure 3.5: Canonical forms and calculation of local observables: a) The matrices that form left canonical form are left-orthonormal, which means that if there is a contraction with their hermitian conjugate (usually drawn at the bottom of a diagram), and there are no virtual legs on the left, the contraction results in an identity. Identity can be represented as a line in a diagram [5]; b) The same holds true for right canonical form, with the only difference that there should be no virtual legs on the right; c) Contraction of local observables in a mixed canonical form. Calculation of some local observable at position l can be done by using left canonical form to the left and including l and right canonical form to the right of l . Contractions are done one by one - from the left, by using a) property, one can first turn $A^{[1]}$ into a line on a diagram, and then merge that line with the one that connects $A^{[2]}$ and its conjugate. No additional legs on the left means that a) relationship can again be applied and so on. The same procedure for the matrices B means that eventually only the diagram on R.H.S remains.

That holds true for every set of matrices $\mathbf{U}^{[1]}, \mathbf{U}^{[2]}, \dots, \mathbf{U}^{[N+1]}$ provided that $\mathbf{U}^{[1]} = \mathbf{U}^{[N+1]} = \mathbf{I}$. Because such gauge transformation is possible, there are certain MPS representations of the state that are particularly helpful in calculating local observables, called canonical forms, which are composed of orthogonal matrices. The process of turning the full state tensor $\psi_{\sigma_1 \sigma_2 \dots \sigma_N}$ into left-canonical MPS form is as follows.

1. N -dimensional tensor $\psi_{\sigma_1 \sigma_2 \dots \sigma_N}$ is turned into a matrix by combining all but one physical indices $\sigma_2, \sigma_3, \dots, \sigma_N$ into a single one

$$\psi_{\sigma_1 \sigma_2 \dots \sigma_N} = \psi_{\sigma_1, (\sigma_2 \dots \sigma_N)} \quad (3.28)$$

2. Using SVD on resulting matrix one gets

$$\psi_{\sigma_1, (\sigma_2 \sigma_3 \dots \sigma_N)} = \sum_{a_1} U_{\sigma_1, a_1} S_{a_1, a_1} (V^\dagger)_{a_1, (\sigma_2 \sigma_3 \dots \sigma_N)} = \sum_{a_1} A_{a_1}^{[1]; \sigma_1} \psi_{a_1 \sigma_2 \sigma_3 \dots \sigma_N}, \quad (3.29)$$

where in the last step the $\mathbf{U}$ matrix was reshaped into set of vectors $A_{a_1}^{[1];\sigma_1}$ and singular values were incorporated into $\mathbf{V}^\dagger$ to form tensor $\psi_{a_1\sigma_2\ldots\sigma_N}$.

3. The same procedure can be performed on the tensor $\psi_{a_1\sigma_2\ldots\sigma_N}$, where we combine the two indices (a_1 and σ_2) on the left and the remainder on the right. This results in

$$\begin{aligned}\psi_{a_1\sigma_2\sigma_3\ldots\sigma_N} &= \psi_{(a_1\sigma_2),(\sigma_3\ldots\sigma_N)} = \sum_{a_2} U_{(a_1\sigma_2),a_2} S_{a_2,a_2} (V^\dagger)_{a_2,(\sigma_3\ldots\sigma_N)} \\ &= \sum_{a_2} A_{a_1,a_2}^{[2],\sigma_2} \psi_{a_2\sigma_3\ldots\sigma_N}.\end{aligned}\quad (3.30)$$

Inserting that into the previous equation for the full state (3.29)

$$\psi_{\sigma_1\sigma_2\ldots\sigma_N} = \sum_{a_1,a_2} A_{a_1}^{[1];\sigma_1} A_{a_1,a_2}^{[2],\sigma_2} \psi_{a_2\sigma_3\ldots\sigma_N}. \quad (3.31)$$

The tensor $\psi_{a_2\sigma_3\ldots\sigma_N}$ has now fewer indices than the full state representation.

4. Continuing decomposition of $\psi_{a_2\sigma_3\ldots\sigma_N}$ and following reduced tensors ψ will finally transform the full tensor into MPS

$$\psi_{\sigma_1\sigma_2\sigma_3\ldots\sigma_N} = \sum_{a_1 a_2 a_3 \ldots a_{N-1}} A_{a_1}^{[1];\sigma_1} A_{a_1,a_2}^{[2];\sigma_2} A_{a_2,a_3}^{[3];\sigma_3} \ldots A_{a_{N-2},a_{N-1}}^{[N-1];\sigma_{N-1}} \Gamma_{a_{N-1}}^{[N];\sigma_N}, \quad (3.32)$$

where the last vector $\Gamma_{a_{N-1}}^{[N];\sigma_N}$ cannot be left orthonormal, since it is the result of multiplying the results of the last SVD: $S_{a_{N-1}a_{N-1}} (V^\dagger)_{a_{N-1},\sigma_N}$.

As mentioned above, the SVD produces semi-unitary matrices $\sum_{\sigma_i} (\mathbf{A}^{[i],\sigma_i})^\dagger \mathbf{A}^{[i],\sigma_i} = \mathbf{I}$. In graphical representation, this means that for each connected pair of tensors, where one is the hermitian conjugate of the other, one can replace it with a single connected leg (see Figure 3.5).

The same process can be done by starting from the right side of tensor $\psi_{\sigma_1\sigma_2\ldots\sigma_N}$, which results in a right-canonical form

$$\psi_{\sigma_1\sigma_2\sigma_3\ldots\sigma_N} = \sum_{a_1 a_2 a_3 \ldots a_N} \tilde{\Gamma}_{a_1}^{[1];\sigma_1} B_{a_1,a_2}^{[2];\sigma_2} B_{a_2,a_3}^{[3];\sigma_3} \ldots B_{a_{N-2},a_{N-1}}^{[N-1];\sigma_{N-1}} B_{a_{N-1}}^{[N];\sigma_N}, \quad (3.33)$$

where matrices $\mathbf{B}$ are right-orthonormal $\sum_{\sigma_i} \mathbf{B}^{[i],\sigma_i} (\mathbf{B}^{[i],\sigma_i})^\dagger = \mathbf{I}$, and again the tensors can be reduced to lines in graphical notation, this time from the right. As before, the first tensor cannot be made right-orthonormal, as it is a product of $U_{\sigma_1,a_1} S_{a_1,a_1}$.

Combining left- and right-canonical forms is particularly useful in calculating observables. Suppose that we would like to know the expectation value of the operator O_l , which acts only on the l -th site of the chain. By taking a full tensor $\psi_{\sigma_1\sigma_2\sigma_3\ldots\sigma_N}$ and performing decompositions from the left and right such that it stops at the l -th site, one gets a mixed-canonical form

$$\psi_{\sigma_1\sigma_2\sigma_3\ldots\sigma_N} = \sum_{a_1 a_2 a_3 \ldots a_N} A_{a_1}^{[1];\sigma_1} A_{a_1,a_2}^{[2];\sigma_2} \ldots A_{a_{l-1},a_l}^{[l];\sigma_l} S_{a_l,a_l} B_{a_l,a_{l+1}}^{[l+1];\sigma_{l+1}} \ldots B_{a_{N-2},a_{N-1}}^{[N-1];\sigma_{N-1}} B_{a_{N-1}}^{[N];\sigma_N}. \quad (3.34)$$

Calculating the expectation value $\langle \psi | O_l | \psi \rangle$ in a graphical notation is equivalent to joining the physical legs of ψ and its hermitian conjugate, with the matrix representing O_l at the site

l as is presented in Figure 3.5. By starting to contract first the foremost left matrices $\mathbf{A}^{[1]}$, one can use the left normality to show that they contract to the identity tensor acting on the left virtual indices of the bra and ket. Following that, the next set of matrices $\mathbf{A}^{[2]}$ can also be changed to identity and so on. Contracting the right-orthogonal matrices in the same way leaves just a very small TN to contract, reducing the computational cost.

3.2 Entanglement and Area Law states

Representation of a wave function in a left or right canonical form suggests a clever method of approximating the quantum state. Using the procedure described in the previous section, we will find that the left canonical form matrices $\mathbf{A}^{[i];\sigma_i}$ have dimensions: $1 \times d, d \times d^2, d^2 \times d^3, \dots, d^2 \times d, d \times 1$ ⁶ (where d is the dimensionality of the local Hilbert space). If we add all these together for each σ_i to get a number of independent coefficients (recalling that some coefficients will be dependent because of left/right-orthonormality), we will get exactly d^N — number of coefficients $\psi_{\sigma_1\sigma_2\sigma_3\dots\sigma_N}$. We could find, however, that the entanglement spectrum for each SVD shows that many states have insignificant contribution to the wave function of the entire system. There are also states like $|\psi_{AKLT}\rangle$, which will be exactly described by just 2×2 dimensional matrices, regardless of system size N . We can define the maximum bond dimension $\chi < d^{N/2}$, which will indicate the number of states with largest Schmidt coefficients that we will keep at each bipartition of the system. Alternatively, we can define maximal truncation error ϵ , such that

$$\left\| |\psi\rangle - \sum_{n=1}^{\chi} \Lambda_n |n\rangle_L \otimes |n\rangle_R \right\| < \epsilon, \quad (3.35)$$

for some χ . The number of parameters is now restricted to $D \simeq Nd\chi^2$, which allows us to obtain large system sizes N , if we can keep the truncation error low. This condition is only satisfied if the entanglement spectrum falls down rapidly enough, i.e. entanglement entropy is low.

It turns out that for most of the states in many-body Hilbert spaces, the entanglement entropy scales with the volume of the system, meaning that even for 1-D lattices, where $S \sim N$, a good description by truncated MPS is not possible. However, there are many important states that obey what is called “Area Law” - scaling with the size of the border between two partitions. For 2-D lattices that still presents a problem, but for 1-D, the entanglement entropy stays constant at each bipartition. Among the Area Law states are ground states of gapped systems, which is a large class of interesting Hamiltonians. Intuitively, the Area Law states can be explained by finite correlation length ξ in the wave function. Adding a site on that lattice which is far away ($i - j > \xi$) from the bipartition should not have any effect on the entanglement spectrum.

3.3 Purification MPS

In previous sections, an efficient TN representation of a quantum wave function was discussed, where the use of a left-/right-canonical form allowed for a simple calculation of observables acting on single sites. For non-local operators, calculations can become cumbersome, but an optimal way of storing information in a TN form is often used and it is called

⁶This is a consequence of the shapes of matrices after performing SVD. For more detail, see Chapter 4 on PCA method.

MPO [51]. A similar structure is also employed when describing thermal quantum state [52], which will be important when calculating finite-temperature observables for the entanglement learning procedure.

3.3.1 Matrix product operators

The generalisation of TNs to quantum operators is straightforward: Each matrix at site i , now need to have two physical indices corresponding to bra and ket part of the operator

$$\begin{aligned}\hat{O} &= \sum_{\sigma_1 \sigma_2 \sigma_3 \dots \sigma_N} \sum_{\sigma'_1 \sigma'_2 \sigma'_3 \dots \sigma'_N} O_{\sigma_1 \sigma_2 \sigma_3 \dots \sigma_N}^{\sigma'_1 \sigma'_2 \sigma'_3 \dots \sigma'_N} |\sigma\rangle \langle \sigma'| \\ &= \sum_{\sigma_1 \sigma_2 \sigma_3 \dots \sigma_N} \sum_{\sigma'_1 \sigma'_2 \sigma'_3 \dots \sigma'_N} \sum_{a_1 a_2 a_3 \dots a_{N-1}} W_{a_1}^{[1]; \sigma_1 \sigma'_1} W_{a_1, a_2}^{[2]; \sigma_2 \sigma'_2} W_{a_2, a_3}^{[3]; \sigma_3 \sigma'_3} \dots W_{a_{N-1}}^{[N]; \sigma_N \sigma'_N} |\sigma\rangle \langle \sigma'|.\end{aligned}\tag{3.36}$$

We call this representation MPOs. Usually in graphical representation, the physical legs σ'_i come from the upper part of the tensors $\mathbf{W}^{[i]}$. By reshaping primed and non-primed indices into one, any previously discussed operation for MPS still holds up for MPO. The canonical form, however, usually has no practical application. SVD is still often used to reduce the bond dimension after operator-operator or operator-wave function multiplications.

3.3.2 Thermal density matrix in QM

The thermal equilibrium state of a quantum system is described by the thermal density matrix, which weighs energy eigenstates by their Boltzmann weight:

$$\rho(T) = e^{-\beta H} = \sum_n e^{-\beta E_n} |n\rangle \langle n| \tag{3.37}$$

where $\beta = \frac{1}{T}$ and $|n\rangle$ are the eigenstates of H . Finding all $|n\rangle$ and representing them in MPS form is usually too difficult. There is another way of representing mixed states in TN, by using so called purification [53, 54]. We can take standard MPS $\mathbf{M}^{[i]}$, which represent a general wave function $|\psi\rangle$ and double the physical space at each site by adding an auxiliary index q_i . Then we represent density matrix as a trace over that auxiliary space

$$\rho = \text{Tr}_Q [|\psi\rangle \langle \psi|]. \tag{3.38}$$

This ensures that ρ can represent mixed states. Graphical representation of taking the trace is pictured in Figure 3.6. This object is similar to MPOs, in that it has two physical indices at each site.

In order to obtain thermal density matrix in TN representation, we initialise ρ at infinite temperature, where for all systems it is proportional to the identity matrix, which is trivial to write as a MPS. If we set $\beta = \tau = it$, we can use any iterative time evolution algorithm to get to finite temperature

$$\rho(T) = e^{-\beta H} \rho(T = \infty). \tag{3.39}$$

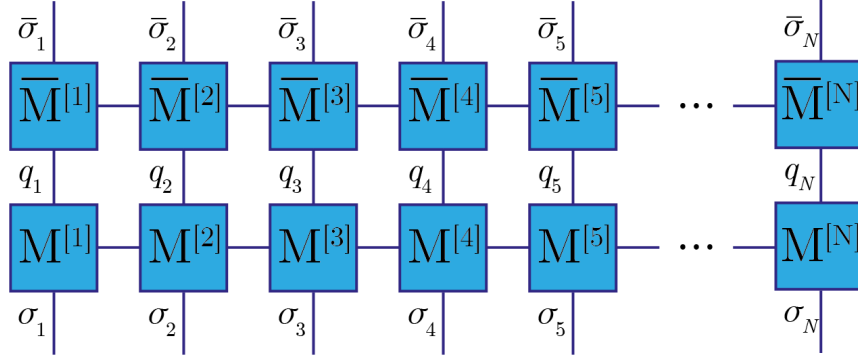


Figure 3.6: Purification MPS representing density matrix. Trace over auxiliary space is represented by contractions $q_1, q_2 \dots q_N$.

3.4 Calculating observables

The general formula for calculating quantum observables at finite temperatures is

$$\langle \hat{O} \rangle = \frac{\text{Tr} [\rho(T) \hat{O}]}{\text{Tr} [\rho(T)]}. \quad (3.40)$$

In a general case, calculations of the above expression for a many-body system larger than a few tens of particles is prohibitively costly. However, if both the thermal density matrix and the observable operator are in purification MPS and MPO forms respectively, the multiplication within the trace can be performed⁷. The main challenge then is to calculate the TNs expressions of $e^{-\beta H}$ and $\hat{O}$. Since some of the observables obtained for the research are time dependent, and the finite temperature density matrix can also be obtained using imaginary time $\beta = \tau = it$, the use of popular TN time-evolution algorithms was an obvious choice.

3.4.1 Time-evolving block-decimation

Time-evolving block-decimation (TEBD) [55] is a method which gives an approximation of the evolution operator, which is easily translated into local tensor terms. It relies on the fact that many interesting systems have Hamiltonians, which can be divided into separate parts, where the terms within each part commute with each other. As an example, the Hamiltonian of 1-D chain with nearest neighbour interactions can be written as the sum of two

$$H = H_{\text{even}} + H_{\text{odd}} = \sum_{i=0} h_{2i} h_{2i+1} + \sum_{i=0} h_{2i+1} h_{2i+2}. \quad (3.41)$$

Using the Baker-Campbell-Hausdorff formula [56] applied to evolution operator of H

$$U(t) = e^{-i(H_{\text{even}} + H_{\text{odd}})t} = e^{-iH_{\text{even}}t} e^{-iH_{\text{odd}}t} + \mathcal{O}(t^2). \quad (3.42)$$

And since all of the terms in H_{even} and H_{odd} commute with each other, both exponentials in 3.42 are products of exponents of local terms

$$\tilde{U}_{\text{even}}(\Delta t) = e^{-i(h_0 h_1) \Delta t} e^{-i(h_2 h_3) \Delta t} \dots e^{-i(h_{N-2} h_{N-1}) \Delta t} \quad (3.43)$$

$$\tilde{U}_{\text{odd}}(\Delta t) = e^{-i(h_1 h_2) \Delta t} e^{-i(h_3 h_4) \Delta t} \dots e^{-i(h_{N-1} h_N) \Delta t}. \quad (3.44)$$

⁷In TN form, trace requires connecting upper and lower physical indices of the purification MPS.

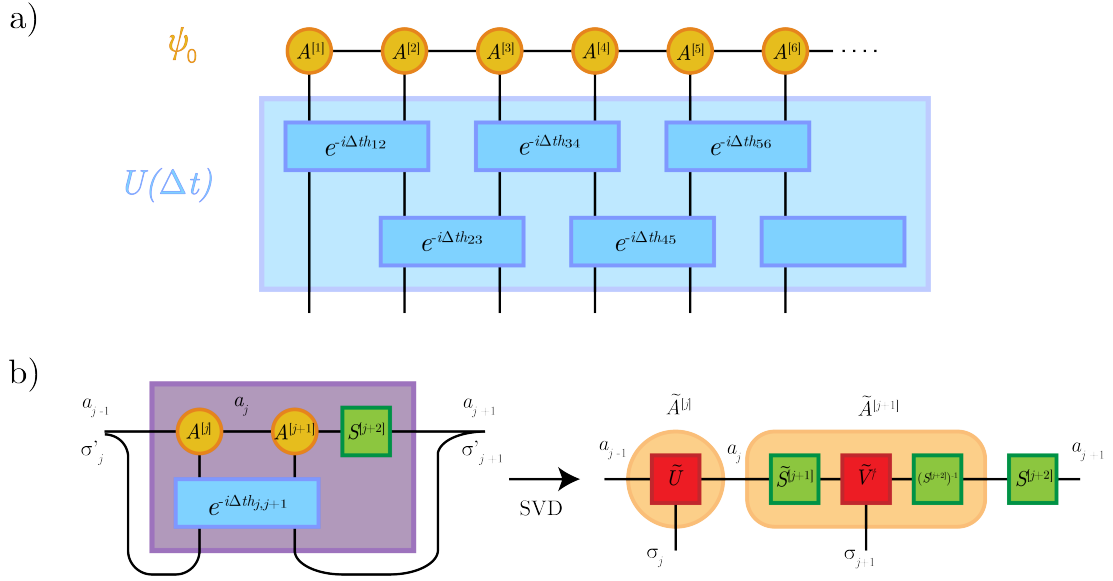


Figure 3.7: TEBD algorithm for a system with two-site nearest neighbour interactions: a) In a single iterative step, the initial state ψ_0 is connected to odd and even parts of the Hamiltonian (see equations (3.43) and (3.44)); b) The process of multiplying a single two-site evolution tensor to the MPS. The physical σ and virtual a legs are connected in order to reshape 4-tensor into matrix. This new matrix contains two left canonical matrices from the MPS, evolution tensor and one of the singular value matrix. After performing SVD the updated $\tilde{A}^{[j]}$ is composed of left orthonormal matrix $\tilde{U}$ from the decomposition, while – in order to preserve left canonical form [6] – right $\tilde{A}^{[j+1]}$ has to include new singular value matrix, $\tilde{V}^\dagger$ and the inverse of singular values $S^{[j+1]}$.

By setting small Δt and applying $\tilde{U}_{\text{even}}$, $\tilde{U}_{\text{odd}}$ consecutively many times, the time evolution of either a quantum state or operators can be performed.

In terms of TNs, each of the local exponents in $\tilde{U}_{\text{even}}$, $\tilde{U}_{\text{odd}}$ is expressed by an independent 4-tensor, which can be contracted with the main MPS/MPO one by one (see Figure 3.7). Each contraction will increase the bond dimension between sites at which the local operator applies. Usually, when that bond reaches a certain maximal value χ_{max} , SVD — followed by truncation — is used to reduce the dimensionality of tensors⁸. The advantage of reducing evolution operator to its approximate block form lies in the fact that it is much easier to contract, and then perform SVD of local terms, then it is for the full tensor representing the entire evolution operator. The error comes from: a) The omission of second and higher-order terms in the expansion (3.42), which is common for all approximate iterative techniques. This error can be controlled by making the time step as small as required; b) Removal of states with lowest entanglement through SVD, after the maximum bond dimension is reached.

3.4.2 MPO evolution

The iterative step of the time evolution using TNs can be done in multiple ways. Rather than dividing the Hamiltonian into non-commuting parts (as in TEBD), one can get a MPO

⁸See Section 3.2

representation directly from the exponential expansion of evolution operator

$$U(t) = 1 + t \sum_j H_j + \mathcal{O}(t^2), \quad (3.45)$$

where H_i represent any local term in the Hamiltonian. Authors of the paper [57] found out that replacement of the Euler step by

$$1 + t \sum_j H_j \rightarrow \prod_i (1 + t H_j) \quad (3.46)$$

has a particularly easy to write MPO representation. Each individual tensor of that MPO corresponding to site i , has a size proportional to the number of interaction terms between that site and its neighbours. For most short-range interaction Hamiltonians that is not a large number and the contraction with the initial state can be done fast. As before, the error in calculations is both due to the linear approximation and the eventual excess of the maximum bond dimension.

3.4.3 Optimising calculations

As discussed above, the iterative methods used to calculate the imaginary and real-time evolution have their limits. At large times (or for low temperatures) the build-up of entanglement connected to moving from the correct solution to a more random one⁹ will eventually produce values of observables far from the true ones. Below are some of the methods for reducing entanglement growth/extending the maximum achievable time for the evolution that we have used in our research.

Disentanglers

Typical entanglement growth for an imaginary time evolution is presented in Figure 3.8. We start with a system at infinite temperature, where the spins are randomly oriented at each site, so that the entanglement entropy at each bipartition will be equal to zero. After some evolution, S will grow due to cooling down (which means that high-energy states will contribute less to ρ), but we expect that for most systems (especially gapped) at low temperatures entanglement will fall down again, because ground state will usually be less entangled than sum of some higher-energy states. In practice, this will not always happen due to errors connected to MPS truncation and approximation to the true evolution.

There is an interesting trick that we can perform in purification MPS to combat the entanglement growth in the system. While performing time evolution, we have only applied $U(\Delta\tau)$ to physical legs, but we can also apply certain operators to auxiliary legs, and as long as they are unitary, they will not change the final result¹⁰. It turns out that doing so can result in the removal of a lot of entanglement from the system [54]. We call those auxiliary space operators disentanglers, and there are many different ways of constructing them, e.g. for real-time evolution it can be the time evolution operator with a change of sign - $U(-t)$.

⁹We usually assume that ground states for ordered physical systems will have a low entanglement.

¹⁰Because we are taking trace over the auxiliary indices and the evolution is performed both on bra and ket parts of the state, disentanglers will cancel out and leave the result unchanged.

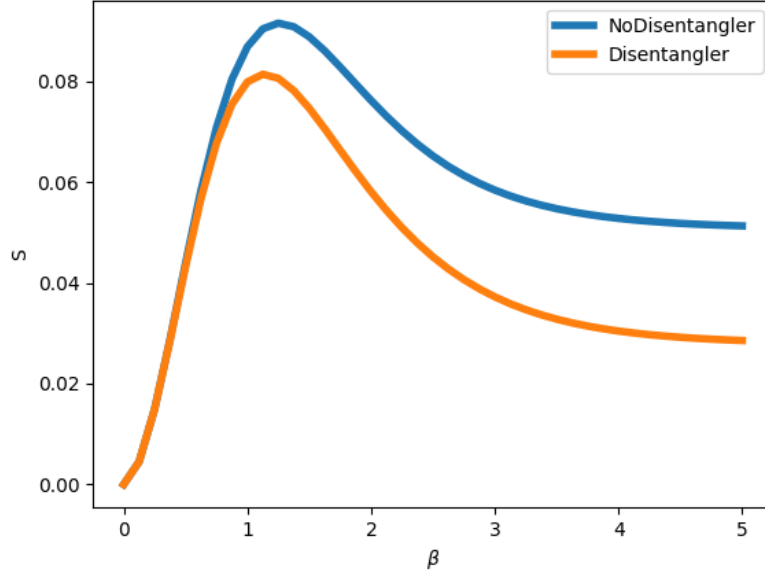


Figure 3.8: Comparison of entanglement growth for a typical imaginary time evolution, with and without the application of disentangler. Both the maximal peak and entanglement entropy at lower temperatures are smaller with application of disentangler.

Using time-translation invariance

One of the observables that we have considered in our study to learn entanglement are the dynamic two-point correlation functions defined as

$$\langle S_{N/2+i}^{\alpha}(t) S_{N/2}^{\alpha}(0) \rangle. \quad (3.47)$$

where S_i^{α} is the $\alpha = x, y, z$ component of the spin operator at the i -th site. Since only one of the operators is time dependent, at first one might try to perform the iterative evolution for MPO representing that local operator. If the Hamiltonian is not time-dependent, the following equality applies

$$\langle S_{N/2+i}^{\alpha}(t) S_{N/2}^{\alpha}(0) \rangle = \langle S_{N/2+i}^{\alpha}(t/2) S_{N/2}^{\alpha}(-t/2) \rangle. \quad (3.48)$$

This allows for simultaneous evolution for both local operators, but the combined computational error for the same time period is smaller for the right-hand side expression of the correlator. For many systems, the entanglement entropy becomes large at $t \approx 10 - 20$ [53]. That result can be doubled using the expression above.

Heisenberg and Schrödinger picture evolutions

Evaluating the two-point correlation functions at finite temperatures means that both the imaginary and the real-time evolutions have to be performed. If we write down purification MPS in the form

$$|\psi\rangle = \sum_{p,q} \psi_{p,q} |p\rangle \otimes |q\rangle, \quad (3.49)$$

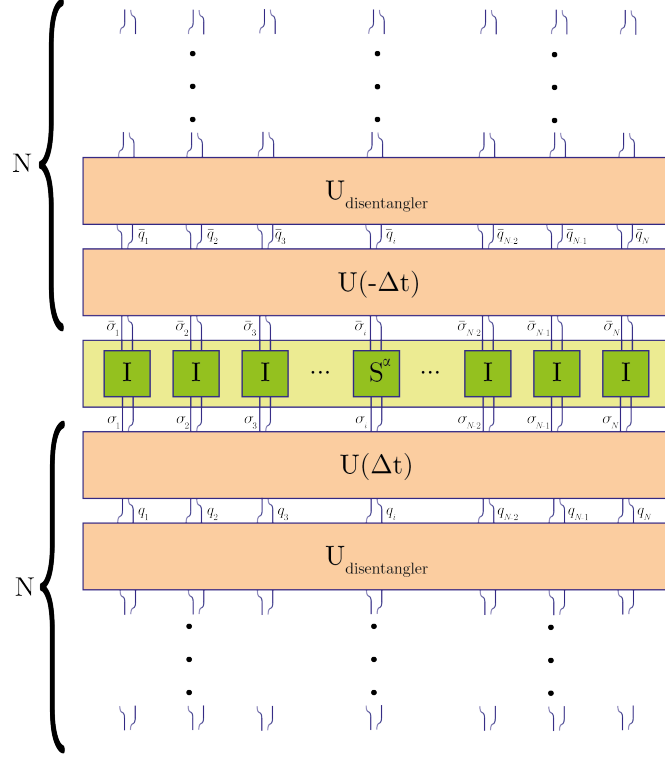


Figure 3.9: Evolution of the local spin operator in Heisenberg picture. The initial block in the middle consist of identity matrices and the local operator at some position j . The evolution is applied both to the bra (down) and ket (up) indices, and all sites have both physical and auxiliary legs. The unitary disentangler block applies only to the auxiliary indices, while the evolution U applies to the physical indices.

where $|p\rangle, |q\rangle$ are the states from physical and auxiliary spaces correspondingly, the correlation functions become

$$\begin{aligned} \langle S_{N/2+j}^\alpha(t/2) S_{N/2}^\alpha(-t/2) \rangle &= \\ &= \langle \psi(T = \infty) | e^{-\beta H} e^{iH \frac{t}{2}} S_{N/2+j}^\alpha(0) e^{-iH \frac{t}{2}} e^{-iH \frac{t}{2}} S_{N/2}^\alpha(0) e^{iH \frac{t}{2}} e^{-\beta H} | \psi(T = \infty) \rangle. \end{aligned} \quad (3.50)$$

The most straightforward approach to calculate this quantity is to build the entire TN, with the evolution operators given by either TEBD or some MPO form. If we were interested in running data for multiple temperatures, this would introduce unnecessary redundancies. Firstly let us start with separating bra and ket parts of the state

$$\begin{aligned} |\psi_1\rangle &= e^{-iH \frac{t}{2}} S_{N/2}^\alpha(0) e^{iH \frac{t}{2}} e^{-\beta H} |\psi(T = \infty)\rangle \\ |\psi_2\rangle &= e^{iH \frac{t}{2}} S_{N/2+j}^\alpha(0) e^{-iH \frac{t}{2}} e^{-\beta H} |\psi(T = \infty)\rangle. \end{aligned} \quad (3.51)$$

The first obvious optimisation is that the state $e^{-\beta H} |\psi(T = \infty)\rangle$ is at the end of both $|\psi_1\rangle$ and $|\psi_2\rangle$, and therefore should be calculated just once. Additionally, to avoid performing real-time evolution each time we want to change temperature of the system, we can calculate evolution of spin operators in a Heisenberg picture, while leaving the imaginary time

evolution in Schrödinger picture. The three TN objects that are to be evaluated then are

$$\begin{aligned}
|\psi_s\rangle &= e^{-\beta H} |\psi(T = \infty)\rangle \\
S_{N/2}^\alpha(t/2) &= e^{-iH\frac{t}{2}} S_{N/2}^\alpha(0) e^{iH\frac{t}{2}} \\
S_{N/2+j}^\alpha(t/2) &= e^{iH\frac{t}{2}} S_{N/2+j}^\alpha(0) e^{-iH\frac{t}{2}}.
\end{aligned} \tag{3.52}$$

If those objects are saved at each imaginary or real-time step in the iterative algorithm, it is easy to multiply them in order to obtain two-point correlators as a function of both β and t . The example of the local time-evolved spin operator in the tensor form is presented in Figure 3.9. The evolution of the initial local tensor is performed for both the bra and the ket indices, with disentanglers applied to both parts of auxiliary indices as well.

Chapter 4

Machine learning methods

The relatively recent techniques that can be described as machine learning (ML) started getting attention from both the scientific world and the general public since around 2006, when an article by Geoffrey Hinton et al. [58] was published, where it was shown how to train a neural network (NN) to recognize handwritten digits (MNIST dataset) with a precision on par with that of a human worker. This started a trend of implementing deep NNs in more and more complicated tasks, such as image recognition [59], text generation [60] and image creation [61]. Currently we are experiencing major breakthroughs in applications of ML, where the AI-generated art can win awards when competing with human artists [62]. The emergence of sophisticated ML methods has stirred up conversations around ethical usage of technology, for example, in the use of image and video evidence in the justice system, or even the meaning of art. By now artificial intelligence (AI) has stopped being a strictly science-fiction or futurologist term and has become one of the most discussed topics for many people.

Although there is currently no danger for physicists to be replaced by AI, ML tools have been widely used in the physical sciences. The most straightforward applications lie with analysing large sets of data, commonly found in physical experiments. Finding patterns and important subsets is one of the easiest tasks for ML. More sophisticated approaches focus on trying to achieve state-of-the-art simulation of many-body systems [11], combining ML, TN, and quantum computational approaches [63] and simulating thermal density matrices of interacting systems [64].

This chapter will focus on presenting the general concept behind ML, showing a NN architecture and a backpropagation algorithm that enabled the current advances in that technology. An example of how those concepts can be applied in modern physics research will be shown with Restricted Boltzmann Machines (RBMs) [11]. The most relevant methods to the results presented later in this text, namely principal component analysis (PCA) and convolutional neural networks (CNNs), will be explained in more detail.

4.1 Machine learning - background

All ML methods aim to extract some information from a given dataset. Of course, this statement fits all other data analysis methods as well. Perhaps the most general description for ML in particular is in their ability to perform better with addition of new data or with the repeating "training sessions" that the algorithm goes through. In more precise language, if we have a function

$$p(\text{data}, \text{algorithm}, \text{hyperparameters}),$$



Figure 4.1: MNIST set of handwritten digits, used in one of the first papers on NNs and the categorization tasks. From then, it is commonly used as a testing bed for new ML algorithms, and as a common set for learning ML.

which measures the performance of a given ML algorithm, then p can increase, not just by changing the algorithm itself, but by adding a new $\text{data}_{\text{new}} = \text{data} + \text{data}_+$. Under this definition, there are many commonly used protocols that could be called ML, such as regression analysis, and in fact they are often defined as such.

Currently, all ML methods are divided into three categories: unsupervised learning, supervised learning, and reinforcement learning. In summary, they can be described as follows.

- Unsupervised learning - is characterised by unlabelled training data, meaning that there is no inherent value ascribed to a given data vector that is used for training. In such type of methods, the task of the algorithm is usually to find the correlation in the training data, with no initial assumption on the data vectors.
- Supervised learning - training data have specific labels, which can represent the different categories for them (for example, when training ML to recognise written numbers, we would like for the computer to know what is the number beforehand). These methods are particularly useful for classification tasks.
- Reinforced learning - differently from the two previous categories, in reinforced learning, we assume that there is an AI agent that operates in some environment, of which it has a partial or full knowledge. We also assume that the agent can make certain decisions about its actions. By attributing rewards and penalties for certain positive or negative actions, we are able to "teach" the agent a desired set of outcomes. These methods work best for training movement in robotics and autonomous cars.

The ML methods can also be a combination of some concept(s) from all of these categories. For example, if we had no knowledge about the labels of the written digits from MNIST set 4.1, then one can get it from applying an unsupervised learning method, which will create 10 different classes of images. Those labels can then be used for a supervised task.

In a usual case of classification task, the whole dataset that is used for ML is divided into training and validation sets¹. As the name implies, training set is used to feed to the NN (or other ML architecture) during the learning process, and the validation set is used to assess the accuracy of the learning (for example, if the ML algorithm is overfitting to the training data). Standard measure of performance for the classification tasks is the accuracy [65] defined as

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{True Positives} + \text{True Negatives} + \text{False Positives} + \text{False Negatives}},$$

where True Positives and True Negatives refer to the number of correct identifications of inputs that are either in the class (True Positives) or not in the class (True Negatives). Similarly, False Positives and False Negatives are the incorrect identifications; either of inputs that are in the class but are identified as not in the class (False Negatives) or inputs that are not in the class but classified as in the class (False Positives). Another, more general metric is validation loss [65], which tells the numerical difference between ML prediction and true value of the label. This is especially useful when the prediction is expected to be a real number. In the process of training ML, these metrics can be tracked for both training and validation set. With some iteration of the learning algorithm, at the beginning the performance should increase for both sets. When a given measure starts to get worse for the validation but still is getting better for the training set, it implies that the overfitting is happening and the best performance for general use was already achieved.

4.2 Artificial neural networks

Although not all ML methods are based on the artificial NN architecture [66, 67, 68], the breakthroughs in these types of algorithms have advanced the field greatly. The deep NNs are now part of every significant AI model [69]. Their main advantages are universality (they can be used in all previously mentioned categories of ML), flexibility (large number of hyperparameters to optimize the best structure for specific task) and the fact that they are able to reproduce any function [70].

Artificial NNs were inspired by the structure of biological nerve systems, which are comprised of neuron cells sending electro-chemical signals through a complex network of connections. When a signal (defined as a pulse in electrical potential between the inside and the outside of the brain cell) of sufficient voltage is received by a neuron it “fires”, subsequently sending the signal to all other neurons connected to the synapses at the end of of its axon. Most of the times a single signal will not cause the neuron to fire, but multiple signals received in a short time period will be able to do so. The currently used architectures of artificial NNs usually ignore temporal aspects of biological NNs, but otherwise they are modelled after their ability to send information to other neurons, and the possible change in the connection strength between them.

The first models of artificial NNs were introduced in 1943 by neurophysiologist Warren McCulloch and the mathematician Walter Pitts [66]. They proposed a simplified model of how the biological NNs may operate using propositional logic. Very simple architectures such as perceptron (Figure 4.2 a) were later introduced and shown to be capable of learning on data using computational processors. Since the perceptron can serve as a simplest example of modern artificial NNs, it is instructional to see how it works in detail.

¹A rule of thumb division for the amount of data in both is around 80% of original data in training set and 20% in the validation. Both sets should be as representative of the original as possible.

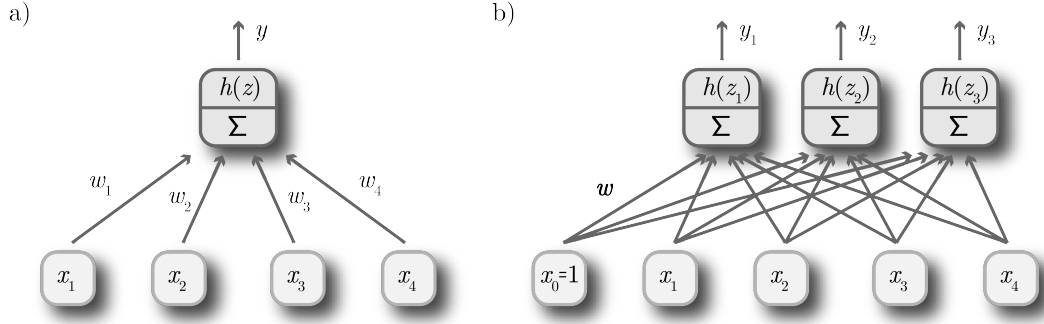


Figure 4.2: Examples of LTU (a) and perceptron with multiple LTUs and a bias input (b). A single LTU combines the weighted input $\sum_i w_i x_i$ and runs it through some activation function h . The perceptron contains several LTUs, which share the same inputs but not necessarily the strength of connections $\mathbf{w}$

4.2.1 Linear threshold units and perceptron

Introduced in 1957 by Frank Rosenblatt [67], a linear threshold unit (LTU) is a model of a single neuron, with multiple inputs $\mathbf{x}$ and a single output y . The inputs are treated as signals from other neurons (they would be an equivalent to the membrane potential) and are summed up with weighted coefficients $\mathbf{w}$. Their weighted sum z is then a variable for the activation function² $h(z)$ which is the output of the LTU. Expressed as an equation, the output is simply

$$y = h(z) = h(\mathbf{w}^T \cdot \mathbf{x}). \quad (4.1)$$

The way that LTU can "learn" is due to the fact that the weights $\mathbf{w}$ can be changed and optimised to approximate a target function $y_{\text{target}}(\mathbf{x})$. Although a single LTU cannot reproduce every possible function, the more complicated neuron structures are able to do so [70]. The changing weights are inspired by the plasticity of biological neurons - their ability to strengthen, or weaken their connections with other neighbouring neurons.

The perceptron is a single layer of LTUs, which means that along with multiple inputs, it contains multiple outputs $\mathbf{y}$ (Figure 4.2), but all of the LTUs that make the perceptron share the same inputs. Again this can be summed up in a single equation that produces the output of the perceptron

$$\mathbf{y} = H(\mathbf{z}) = H(\mathbf{W} \cdot \mathbf{x}). \quad (4.2)$$

The $\mathbf{W}$ now is a matrix with each i -th row corresponding to weights of connections for i -th LTU. Generally, in calculations a constant bias is added alongside the input neurons: $x_0 = 1$. This is done to allow for offset of centre of the $H(\mathbf{z})$ function³

Training

In addition to the structure of the perceptron, the learning rules have to be established to allow for the "training" part of the ML algorithm. The rule chosen by Frank Rosenblatt in his seminal paper on the perceptron is inspired by the idea first proposed by Donald Hebb in his book published in 1949 [71]. He noticed that in biological networks, the connections are strengthened between the neurons that tend to fire at the same time. The concept was

²In the original article, the activation function is a step function, meaning that the output is binary.

³Otherwise no matter how we change the weights, the input $\mathbf{x} = [0, 0, 0, \dots, 0]$ will always yield the same value, since usually $H(\mathbf{z})$ function will return a zero vector for $\mathbf{z} = [0, 0, \dots, 0]$

coined in a catch-phrase: "Cells that fire together, wire together". Rosenblatt proposed the "Hebb's rule" for the perceptron learning which reflected that intuition but also allowed for corrections if the output is different than predicted output⁴. The updates for each single training step following Hebbian learning is as follows

$$w_{i,j}^{\text{new}} = w_{i,j}^{\text{old}} + \mu(y_i - \hat{y}_i)x_j, \quad (4.3)$$

where μ is the learning rate (hyperparameter that can be changed during training), y_i is the output obtained from the perceptron, $\hat{y}_i$ is the target output and x_j is the j -th input. The change is proportional to both the input and the difference between the true solution and the one received from the network. This simple rule is known to work quite well for the perceptron architecture when using binary outputs (which we can assure by choosing the right activation function).

Perceptron can be successfully used to reproduce certain functions, but the input (4.2) is always linear, meaning that some functions cannot be simulated. Indeed, it was shown that for categorization tasks, only the cases in which the different categories are linearly separated can be reproduced [72]. For example one can show explicitly that there is no possible way of reproducing the XOR function using the perceptron. The simplest fix for that, to allow for nonlinearity, is to add additional layers of LTUs.

Activation functions

The functions $h(z)$, $H(\mathbf{z})$ are usually called activation functions in both neuroscience and ML. For perceptrons a common activation functions are heaviside or sign functions

$$\text{heaviside}(z) = \begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{if } z \geq 0 \end{cases}, \quad (4.4)$$

$$\text{sgn}(z) = \begin{cases} -1 & \text{if } z < 0 \\ 0 & \text{if } z = 0 \\ 1 & \text{if } z > 0 \end{cases}. \quad (4.5)$$

These step functions are inspired by biological neurons. Depending on the amount of total stimulation z received from other neurons, the receiving unit is activated or not⁵. It turns out that these functions work well for simple linear classification tasks, but they are not sufficiently general for more complicated work that requires multiple layers of neurons. As is discussed in the next part, the algorithm for training such systems requires the activation functions to have a nonzero derivative at least for some part of the domain. More commonly used functions for that task are

$$\text{Lin}(z) = z, \quad (4.6a)$$

$$\text{ReLU}(z) = \max(0, z), \quad (4.6b)$$

$$\sigma(z) = \frac{1}{1 + \exp(-z)}, \quad (4.6c)$$

$$\tanh(z) = 2\sigma(2z) - 1. \quad (4.6d)$$

⁴This would be an example of supervised learning, when the training set contains both inputs and outputs

⁵In real biological networks this is more complicated because the systems are dynamical. Here, the assumption is that everything is happening at an instant.

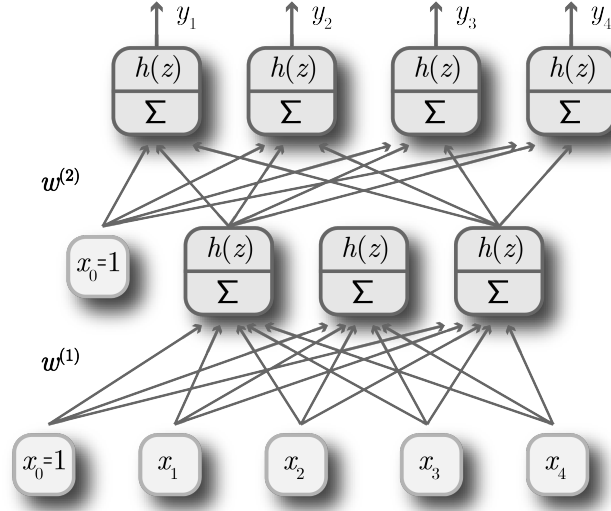


Figure 4.3: Example of a two-layer perceptron. Initial input is passed to a intermediate layer in the same fashion as for perceptron. Then the output of the hidden layer is passed to the second one.

These are all differentiable with finite nonzero derivatives with the exception of ReLU at $z < 0^6$.

4.2.2 Multi-layer Perceptron and Backpropagation

A multi-layer perceptron (MLP), as the name suggest is the expansion of perceptron to multiple layers of LTUs (see Figure 4.3). The first layer takes the input from external data, while each consecutive layer's input is taken from the output of the previous layer. The LTUs in between the initial and the final layers are sometimes called a part of hidden layers since usually their values are just passed through and are not important in and of itself. The MLPs with two or more hidden layers are usually called deep neural networks (DNNs). In mathematical form, the MLPs can be written as nested functions

$$\mathbf{y} = H_{L-1}(\mathbf{W}_{L-1} \cdot H_{L-2}(\mathbf{W}_{L-2} \cdot \dots \mathbf{W}_2 \cdot H_1(\mathbf{W}_1 \cdot \mathbf{x}) \dots)), \quad (4.7)$$

where H_i is the activation function of the i -th layer, and $\mathbf{W}_i$ is weight matrix between i -th and $i - 1$ -th layers. If all the activation functions are linear, then it is still the case that the output will be a linear function of $\mathbf{x}$. However, even slight modification to the linearity of H functions will result in nonlinear behaviour, due to the nested structure of $\mathbf{y}$.

Although MLPs were introduced around the same time as the perceptron, the general application of Hebb's rule was not successful at training MLPs and it wasn't until 1986 when researchers discover a way of successfully training these types of architectures [68].

Training

In principle, to find a minimum of the difference between the actual output of the MLP $\mathbf{y}$ and the target output $\hat{\mathbf{y}}$, one can calculate all derivatives of $\mathbf{y} - \hat{\mathbf{y}}$ with respect to weights and perform the steepest gradient descent accordingly. This strategy will rarely work for NNs,

⁶That can sometimes lead to a problem with "dead" neurons[73] i.e. the single unit can be in a situation where every input from the training set leads to a negative z for that unit.

largely due to the sheer dimensionality of weight space and a smarter approach has to be considered.

The concept behind training MLPs is called “backpropagation”. It was developed long after introduction of single-layered perceptrons (the former had its groundbreaking article in 1986 [68], the latter in 1957), but it marked a drastic change in computational science, since from now on an efficient algorithm for training large-scale NNs existed. Combined with the improvement in computational capabilities of modern CPUs and GPUs, large scale AI systems are now commonly used, and the backpropagation algorithm is still in use in those systems today.

The backpropagation algorithm can be described as performing a gradient descent on weight matrices $\mathbf{W}_i$ of the neuron network layer by layer, starting from the last (the weights connecting the output and one layer before it) and then doing it for the previous set of layers, one by one until reaching the first weight matrix $\mathbf{W}_1$. The first step is to calculate the cost function (any function that calculates how far is the current output from the target one) $C(\mathbf{y})$ by doing a forward pass of the network, while simultaneously keeping track of the output of activation functions at each hidden layer

$$\mathbf{a}_i = H_i(\mathbf{W}_i \cdot \mathbf{a}_{i-1}) \quad (4.8)$$

and the derivatives of activation functions H'_i evaluated at the input

$$\mathbf{z}_i = \mathbf{W}_i \cdot \mathbf{a}_{i-1}. \quad (4.9)$$

The first input is of course the initial layer $\mathbf{x}$.

The first gradient of weights with respect to cost function C can be calculated as (for a derivation see [68])

$$\nabla_{\mathbf{W}_{L-1}} C = (\mathbf{H}_{L-1})'(\mathbf{z}_{L-1}) \cdot \nabla_{\mathbf{y}} C \cdot (\mathbf{a}_{L-2})^T, \quad (4.10)$$

where $(\mathbf{H}_{L-1})'(\mathbf{z}_{L-1})$ means the derivative of $\mathbf{H}_{L-1}$, evaluated at $\mathbf{z}_{L-1}$, and $\nabla_{\mathbf{y}} C$ is the derivative of the cost function with respect to the output (evaluated at the actual output for each given set of training data). The weight matrix can then be updated accordingly

$$\mathbf{W}_{L-1}^{\text{new}} = \mathbf{W}_{L-1}^{\text{old}} + \mu \nabla_{\mathbf{W}_{L-1}} C, \quad (4.11)$$

where again, μ is the learning rate, which should be adjusted to a given set of training data. So far there are no more complicated calculations than simple vector multiplications. Normally in order to evaluate further weight gradients, the several matrix multiplication would have to be performed. For example, the formula for second to last gradients is

$$\nabla_{\mathbf{W}_{L-2}} C = (\mathbf{H}_{L-2})'(\mathbf{z}_{L-2}) \cdot \mathbf{W}_{L-1}^{\text{old}} \cdot t(\mathbf{H}_{L-1})'(\mathbf{z}_{L-1}) \cdot \nabla_{\mathbf{y}} C \cdot (\mathbf{a}_{L-3})^T, \quad (4.12)$$

performing it by simply calculating left and right sides first would mean that two matrices would have to be multiplied⁷. But in (4.12) there is one quantity that has been already calculated in the previous step. The vector $\delta_{L-1} = (\mathbf{H}_{L-1})'(\mathbf{z}_{L-1}) \cdot \nabla_{\mathbf{y}} C$, whose length corresponds to the number of neurons in the layer before $\mathbf{y}$. The multiplication was already performed in (4.10), so it can be saved and put directly into (4.12)

$$\nabla_{\mathbf{W}_{L-2}} C = (\mathbf{H}_{L-2})'(\mathbf{z}_{L-2}) \cdot \mathbf{W}_{L-1}^{\text{old}} \cdot \delta_{L-1} \cdot (\mathbf{a}_{L-3})^T. \quad (4.13)$$

⁷It is worth mentioning that here it would still be fine to multiply from the left to the right, as it would only involve multiplication of matrices by vectors. However, that is not the case for the next gradients.

If every operation on the left of δ_{L-1} is performed first then there is no matrix multiplication in the process. After updating the weights again and saving the new partial derivative for layer $L-2$ $\delta_{L-2} = (\mathbf{H}_{L-2})'(\mathbf{z}_{L-2}) \cdot \mathbf{W}_{L-1}^{\text{old}} \cdot \delta_{L-1}$, one finds the new gradient as

$$\nabla_{\mathbf{W}_{L-3}} C = (\mathbf{H}_{L-3})'(\mathbf{z}_{L-3}) \cdot \mathbf{W}_{L-2}^{\text{old}} \cdot \delta_{L-2} \cdot (\mathbf{a}_{L-4})^T, \quad (4.14)$$

and so the update procedure can be done using the recursive formula for partial derivative vector

$$\delta_i = (\mathbf{H}_i)'(\mathbf{z}_i) \cdot \mathbf{W}_{i+1}^{\text{old}} \cdot \delta_{i+1} \quad (4.15)$$

where only matrix-vector multiplications have to be performed.

In the above presentation of backpropagation, it can be summarized as a steepest gradient descent method used efficiently in a context of highly nested functions, and it has the same trappings as any other optimization based on gradient descent. The two main ones being a) reduction in optimization speed when approaching a local minimum (or encountering a plateau), b) converging to a local minimum, rather than a global one. In the context of NNs, the first problem can be alleviated by introducing a learning rate scheduling, which means changing the learning rate from constant μ to some function $\mu(t)$, which can be pre-determined or based on performance [74]. In addition, the steepest descent step (equation (4.11)) often is replaced by other update methods. For example, the momentum optimization [75] treats $\nabla_{\mathbf{W}} C$ gradient not as a "speed" of descent, but rather as the acceleration, which can handle the previously mentioned plateau problem (intuitively the point in weights space will acquire a large speed at the plateau level).

There are other problems that are specific to the structure of DNNs, but the ones used in the research here were fairly shallow, so there is no need to discussing them in this dissertation.

Hyperparameters

NNs are architectures with many metavariables, which in ML language are usually referred to as hyperparameters. These are, for example, the number of hidden layers, number of units at each layer, activation functions, or the type of the optimizer used in the learning phase. For each individual type of NN, choosing the right hyperparameters might be crucial. There is always a possibility of searching through the space of hyperparameters using cross-validation, but due to the sheer number of possibilities, one can only look at a very small fraction of it. In practice, there are some rules of thumb for given NN structure and application. For example, in later discussed CNNs the number of units within each layer should become smaller with each layer after the input⁸.

4.3 Restricted Boltzmann Machine

One of the first and clearest examples of using recent advances in NNs in physical applications was proposed by Carleo and Troyer in 2016 [11]. They introduced a special architecture of units that model an interacting quantum many-body system. It comprises of two layers (shown in Figure 4.4). The visible one corresponds to the spin configurations $\{\sigma_i\}$ of a modelled quantum system, and the hidden layer $\{h_j\}$ is there to serve as the connections between those spins. Because it is a fully connected network, the interactions can be long-ranged, and

⁸ Although the depth of layer should be growing, check Section 4.4.2.

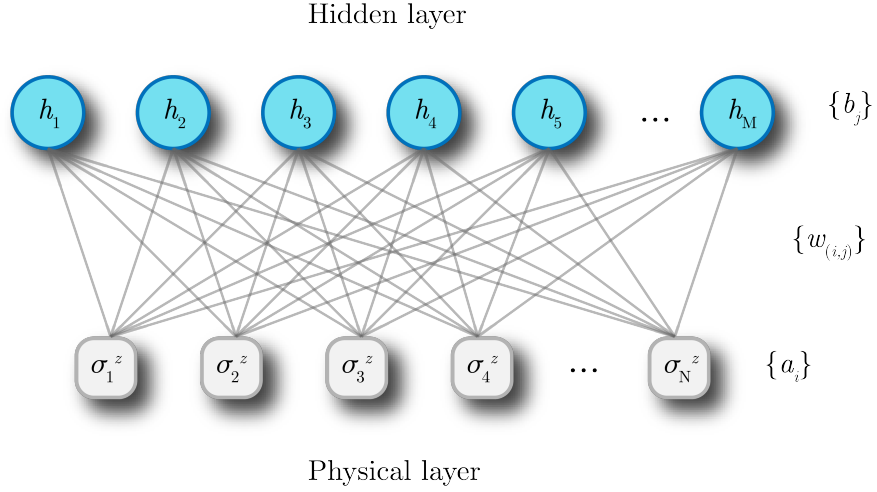


Figure 4.4: Restricted Boltzmann Machine structure. Physical spins in the interacting system are connected not directly, but through the hidden layer of h_i units. Weights $\{a_i\}$, $\{b_j\}$ and $\{w_{(i,j)}\}$ are trained to simulate ground state of a system defined through the Hamiltonian, by using the variational principle.

the lattice is not necessarily 1-D. This type of model has been used in computational science as RBMs [76], named due to the type of ansatz wave function that represent this architecture

$$\psi_{RBM} = \sum_{\{h_j\}} \exp \left[\sum_i a_i \sigma_i + \sum_j b_j h_j + \sum_{i,j} W_{ij} \sigma_i h_j \right], \quad (4.16)$$

where every degree of freedom h_j can only take discrete values $+1, -1$. If the number of neurons in the hidden layer is large, summing over all of the configurations $\{h_j\}$ would be impractical, but because of the simple nature of the ansatz, it can be traced out to give

$$\psi_{RBM} = 2 \exp \left[\sum_i a_i \sigma_i \right] \times \prod_j \cosh \left[b_j + \sum_i W_{ij} \sigma_i \right]. \quad (4.17)$$

Unlike in previous feed-forward NNs, RBM does not have an output layer, so how does the training work and, more importantly, what can be calculated with it? Given some Hamiltonian $\mathcal{H}$ that acts on the Hilbert space of spin sites $\{\sigma_i\}$, the RBM wave function can be used to find the ground state of $\mathcal{H}$, by calculating the energy

$$E = \frac{\langle \psi_{RBM} | \mathcal{H} | \psi_{RBM} \rangle}{\langle \psi_{RBM} | \psi_{RBM} \rangle} \quad (4.18)$$

and performing an update on weights a_i , b_j and W_{ij} , based on a Monte Carlo step and some optimization method⁹ to find the minimum of energy for $\mathcal{H}$ and therefore the ground state, if it is in the class of RBM-like wave functions. The derivatives of (4.18) with respect to the weights are quite simple, and therefore the optimization process can be done quickly.

It was shown that this NN inspired approach to finding ground states of interacting many-body systems can yield very precise results. In the original work, the ground energies of considered models (transverse-field Ising and antiferromagnetic Heisenberg in 1-D

⁹In the original paper, the authors used stochastic gradient descent [77].

and 2-D) were calculated with more precision than with other state-of-the-art methods such as Jastrow and TNs. The RBM is also not restricted to only finding the ground state, as by introducing the time variable to weights $W(t)$, and using the Dirac-Frenkel variational principle [11], the unitary dynamics of a system can also be calculated [11]. Overall, this research promotes the idea that ML methods can be successfully applied to quantum many-body physics calculations, and as such it was partly the inspiration for using similar structures to study phase transitions or information about entanglement in the system in this work.

4.4 Methods used in the research

Before moving on to the results part of this thesis, the particular ML methods that were used here will be explained in more detail. In Chapter 5, we primarily used PCA. For Chapter 7, the main tool for analysis of entanglement were CNNs.

4.4.1 Principal component analysis and Autoencoders

The PCA belongs to the category of unsupervised learning of ML [78]. It is one of the most popular dimensionality reduction tools, although it can also be used for categorization tasks. It can be summarized as a method of finding the hyperplane of the original data space such that all of the data points lie closely to this hyperplane, thus by projecting, only a small amount of information is lost (Figure 4.5). The way of finding this hyperspace is through calculating the covariance matrix, where the eigenstates represent the vectors that hold a large amount of covariance.

The mathematical formulation of the method is as follows. Suppose that there is data with linear correlations (like in Figure 4.5) lying in a d -dimensional space. If the number of data points is N , one can construct a $N \times d$ matrix $\mathbf{X}$

$$\mathbf{X} = \begin{bmatrix} x_1^{[1]} & x_1^{[2]} & x_1^{[3]} & \dots & x_1^{[d]} \\ x_2^{[1]} & x_2^{[2]} & x_2^{[3]} & \dots & x_2^{[d]} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_N^{[1]} & x_N^{[2]} & x_N^{[3]} & \dots & x_N^{[d]} \end{bmatrix}, \quad (4.19)$$

where $x_i^{[j]}$ is the value of j -th dimension of the i -th data point. The centre of the cloud of data usually will be outside of the origin of the coordinates. In order for the method to work, the data have to be shifted to the origin. Otherwise, it is very likely that the direction that holds the most covariance will be the vector from the origin to the centre of the initial data (see Figure 4.5).

$$[\mathbf{Y}]_{i,j} = x_i^{[j]} - \sum_{k=1}^N x_k^{[j]}. \quad (4.20)$$

With the data shifted, the covariance matrix correctly identifies the correlations in the data. If the number of data points exceed the dimensionality of the data (the usual case)¹⁰, the covariance matrix can be written as

¹⁰In other case the covariance matrix should be $\mathbf{S} = \frac{1}{N-1} \mathbf{Y}^T \mathbf{Y}$. The other way to look at this is that the number of PCs should always be the lesser of N and d , since if there are more data points than dimensions, PCA will only rotate the original coordinate system. But if there are fewer data points than d , PCA also finds that the subspace spanned by those points and the dimensionality of that subspace is N .

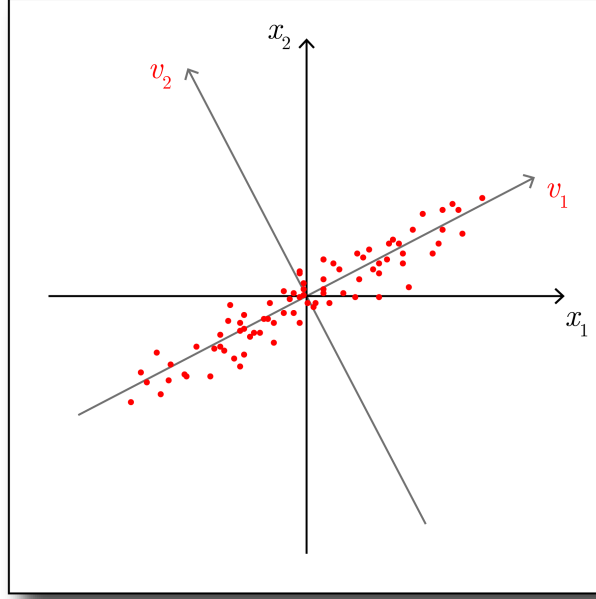


Figure 4.5: PCA of a data in 2-D space x_1, x_2 . Because of the linear correlation in the data, PCA finds the direction which holds the most covariance v_1 - first PC. The second PC v_2 is orthogonal to the first.

$$\mathbf{S} = \frac{1}{N-1} \mathbf{Y} \mathbf{Y}^T. \quad (4.21)$$

The eigenvectors of $\mathbf{S}$ are usually called principal components (PCs) in this analysis, and the corresponding eigenvalues hold information about the covariance (so that the higher the eigenvalue, the larger the covariance). Another way of looking at PCA is that PCs form a new rotated basis of data space, which aligns along the linear correlations. The 1st PC will point towards the largest variation in the data, the 2nd PC will then point to the next largest variation, but in the hyperspace orthogonal to the 1st PC, and so on.

In order to reduce the dimensionality of the data using PCA, one needs to determine how many dimensions are required to properly represent the data. Usually it is done by looking at so-called “scree” plots [79] (named after resemblance to similarly looking rock formations), which have the PC indices on the x axis and the principal values on the y axis. (Examples shown in Figure 4.6). Usually scree plots will have a shape similar to an exponential curve and the cut off for dimensions can be fairly well seen. Sometimes, like in the Figure 4.6 b), the singular values will show slower or no decay, which means that the PCA does not work well with that data. The reasons for this is either a) the data points are not correlated or b) the correlations are not linear.

Once the number of “important” dimensions is established, the data can be represented by projection to this smaller space. The eigenvectors of $\mathbf{S}$ can be written in a matrix as

$$\mathbf{V} = \begin{bmatrix} | & | & | & \dots & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \dots & \mathbf{v}_d \\ | & | & | & \dots & | \end{bmatrix}. \quad (4.22)$$

Suppose that from the scree plot, the number of important dimensions is $p < d$. Then the

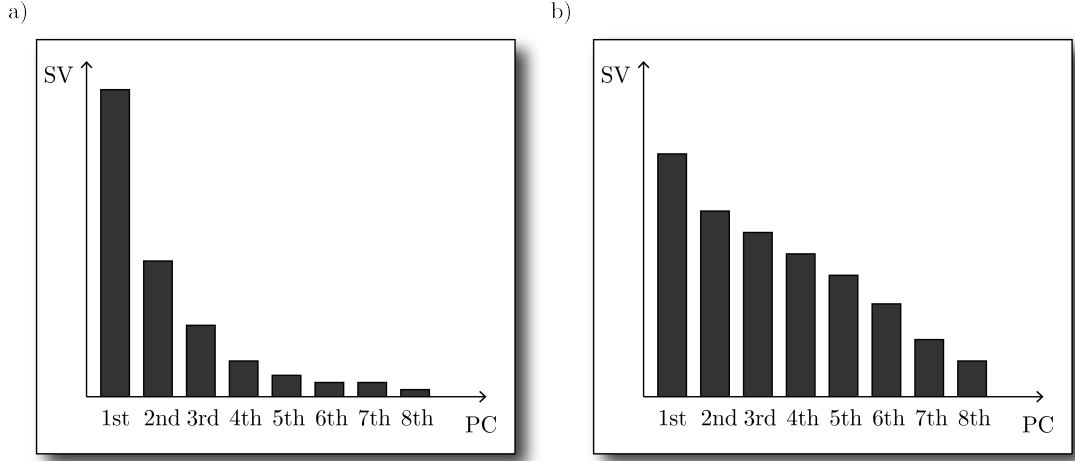


Figure 4.6: Scree plots. The Horizontal axis represents order of PCs, and the vertical axis the principal values connected to those components. a) Typical case with exponential decay. The few first PCs hold almost all of the covariance b) An example of bad performance, which usually suggests that the data are not linearly correlated.

subspace of reduced dimensionality is spanned by

$$\tilde{\mathbf{V}} = \begin{bmatrix} | & | & \dots & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_p \\ | & | & \dots & | \end{bmatrix}. \quad (4.23)$$

The projections of original data points onto the subspace spanned by PCs $\mathbf{V}$ are usually called PC scores

$$\mathbf{C} = \mathbf{Y}\mathbf{V}, \quad (4.24)$$

and a full reconstruction of the data (moving it back to original coordinates) can be performed by

$$\mathbf{R} = \mathbf{C}\mathbf{V}^T. \quad (4.25)$$

However, if the interest is only in the reduced dimensions then one can replace the $\mathbf{V}$ by $\tilde{\mathbf{V}}$ in (4.24), and the reduced scores $\tilde{\mathbf{C}}$ provide partial reconstruction

$$\tilde{\mathbf{R}} = \tilde{\mathbf{C}}\tilde{\mathbf{V}}^T. \quad (4.26)$$

The degree of proximity $\tilde{\mathbf{R}}$ to the full reconstruction depends on the combined covariance that is held by chosen PCs. If the data are highly linearly correlated, this provides an excellent tool for compression, as p can be very small compared to d .

Autoencoders

An interesting generalisation of the PCA method can be found in autoencoders [80]. Essentially these are NNs where the number of units in the input and the output are the same, and the NN tries to reproduce the input i.e. the cost function is the distance from the input. The reduction in dimensionality comes from setting a smaller number of neurons between the input and the output layer, so that a smaller number of parameters (neurons in the hidden layer) can fully reconstruct the initial information. Moreover, this can be used as a generative model, since if the network is trained to completely reconstruct the data, manually tweaking

the values in the hidden layers will result in some change in the output, but it should not be far from other data in the training set. For instance, with an autoencoder trained on the MNIST database, it should be possible to generate other similar looking 28x28 pictures of handwritten numbers. It can be proven that the autoencoder with linear activation functions and mean square cost function performs the same optimization as PCA [81]. Adding more hidden layers and more complicated activation functions can mean that unlike PCA, autoencoder can find nonlinear correlations in the data. The drawback is that it introduces more hyperparameters (like the cost function or number of neurons in hidden layer), while PCA has none.

4.4.2 Convolutional neural networks

As mentioned above, artificial NNs were heavily inspired by models designed to simulate perceptrons and more complicated neurological structures. It turns out that in this framework one of the best understood parts of brain function is the visual system [82]. Experimental research conducted on various mammals [83] showed that the clusters of neurons in the visual cortex are both layered and spatially dependent. The job of the initial "layer" of neurons (the one closest to the eye itself) is to activate whenever some line pattern is registered in a given position of our vision. The orientation of the line pattern is also decided in this layer, which can be described by a ring model of neuron connectivity [84]. Each neuron from the next layer is connected to multiple neurons from the previous one, such that it gathers information from multiple neighbouring angles of our vision. Combined with the orientation resolution of the previous layer, neurons start to activate when recognizing simple shapes, such as triangles and crosses. The layer after that will combine those shapes to build even more complicated patterns. The visual cortex can be described as preprocessor of the initial image that helps animals with quick pattern recognition. It is not surprising then that the architecture of the first computational NN that was designed for image processing was inspired by structure of neurons in the visual cortex.

The first attempt to create an artificial model on such a basis was the neocognitron [85], which was then followed by more work, with an important breakthrough in 1998 [86], when the LeNet-5 architecture was introduced to recognize handwritten numbers. Today, such structures are referred to as CNNs or networks with convolutional layers (because it is common for architectures to have multiple types of neuron layers). Unlike the fully connected NNs, the number of weight parameters in convolutional layers is much lower, which helps with dealing with high-resolution pictures¹¹, or other data with potentially large number of parameters in the input layer. The underlying assumption is that the neighbouring neurons have to be correlated. This is true for many potential observables in physics, and that is why CNNs were used in this research.

Convolutional layer

The part of CNNs that allows for a larger number of neuron units and the locality of analysis is the structure of the convolutional layer. Unlike the fully connected case, the weights between layers are no longer represented by matrices $\{w_{i,j}^{(l)}\}$ (with listings of all connections between neurons from one layer to the other), but by a single vector $\{w_i^{(l)}\}$, because the

¹¹If we consider the input of a current standard picture format (1920x1080), the fully connected network would require $1920 \times 1080 \times (\text{Number of neurons in first hidden layer})$ size matrices to hold the weights between input and the first layer, which would severely limit the number of hidden layer neurons.

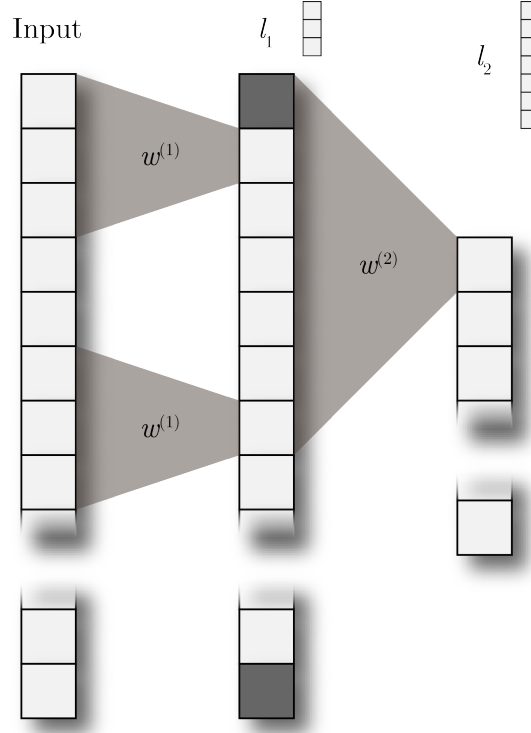


Figure 4.7: Convolutional layers for 1-D input. The connections between neurons from consecutive layers is defined by the size of the filter. In the above example, l_1 layer takes input from three neighbouring neurons in the previous layer, while l_2 takes it from seven neurons from layer l_1 . The weights are shared between all units within layers. The boundary conditions can be solved by decreasing size of neurons in a layer (l_1) or by introducing zero input in previous layer (l_2 , greyed neurons at border).

weights are now shared between all neurons from the forward layer. Of course, if the connections were still exactly the same, all of the neurons from the forward layer would receive the same input; however, the difference is that the weight vector references the relative position of the neurons from the previous layer, and should only count some local region of neurons. Therefore, the input will depend on the position of neurons from the forward layer. For example, in the 1-D CNN structure, if there are two layers with the same number of neurons, the equation for the output z_i of each neuron from the second convolutional layer k would be

$$z_i^{(k)} = \sum_{p=0}^{2l} w_p^{(k)} a_{i+p-l}^{(k-1)}, \quad (4.27)$$

where in this case there are $2l + 1$ connected neurons from previous layer and the index p is running through those neurons. In the standard case where CNNs are used, namely analyzing images, the inputs are 2-D from the start, so the sum in (4.27) would run over two indices. In addition, neurons at the borders of the convolutional layer would take input from neurons outside the range of the previous layer. Those activation inputs can be set to zero or, if it makes sense for the input data, periodic boundary conditions can be introduced. Alternatively, the convolutional layer can be set to have a smaller number of neurons, so it fits to the size of the weight vector $\mathbf{w}^{(k)}$.

Much of what the convolutional layer can detect is directly coupled to the shape of con-

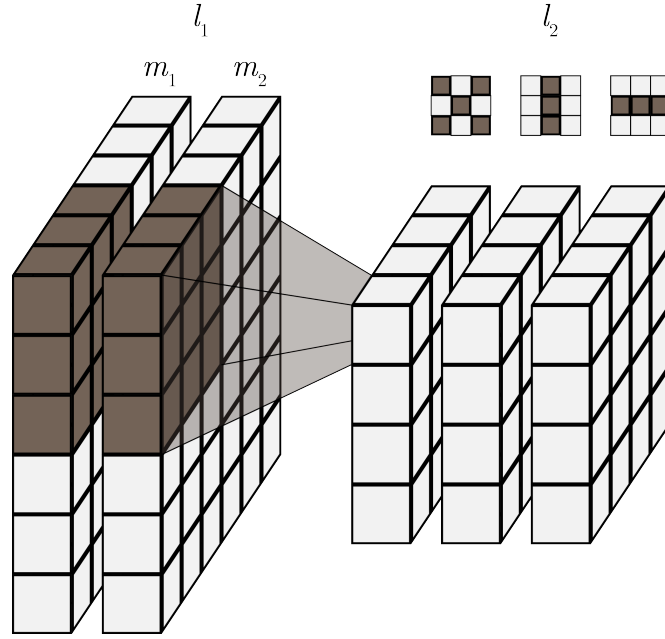


Figure 4.8: Convolutional layer for 2-D input with multiple maps and multiple filters. The two maps in layer l_1 are treated as an input for layer l_2 . The three filters of l_2 detect diagonal, vertical and horizontal lines in the maps from l_1 respectively (greyed out parts of 3×3 sections above l_2). The input is taken from all of the maps in the previous layer.

nections represented by the weight vector. In the case of (4.27), neurons summarize the input from a few adjacent units centered around the output, but, in general, the connections can be more complicated, especially in two dimensions. For example if the NN is trained to detect diagonal 3 pixel lines, the weight vector should have nonzero entries only on diagonal connections (see Figure 4.8). For detection of horizontal or vertical lines, the weights would also have to be shaped accordingly. In each convolutional layer, there can be multiple such weight patterns, usually called "filters", so that the same NN can detect multiple "feature maps" at once. If there are two convolutional layers with multiple feature maps, the output of the next layer is calculated in a following manner

$$z_i^{(k,m)} = \sum_{p=0}^l \sum_{n=1}^N w_p^{(k,m,n)} a_{(x,y)_p}^{(k-1,n)}, \quad (4.28)$$

where m points to a given feature map from the following layer k , and the n index runs through all maps from the previous layer. The index p runs through all the neurons from the previous layer, just as the one defined in (4.27).

Overall structure

Although the main feature of CNNs lies in the structure of weight connections between layers, some insight is gained through understanding of most usual architectures of such networks because they do not consist only of convolutional layers. The general architecture of CNNs is made up of:

- Input layer with possible multiple channels. For example, if a network analyzes color pictures, the input layer might consist of three channels with RGB values for each

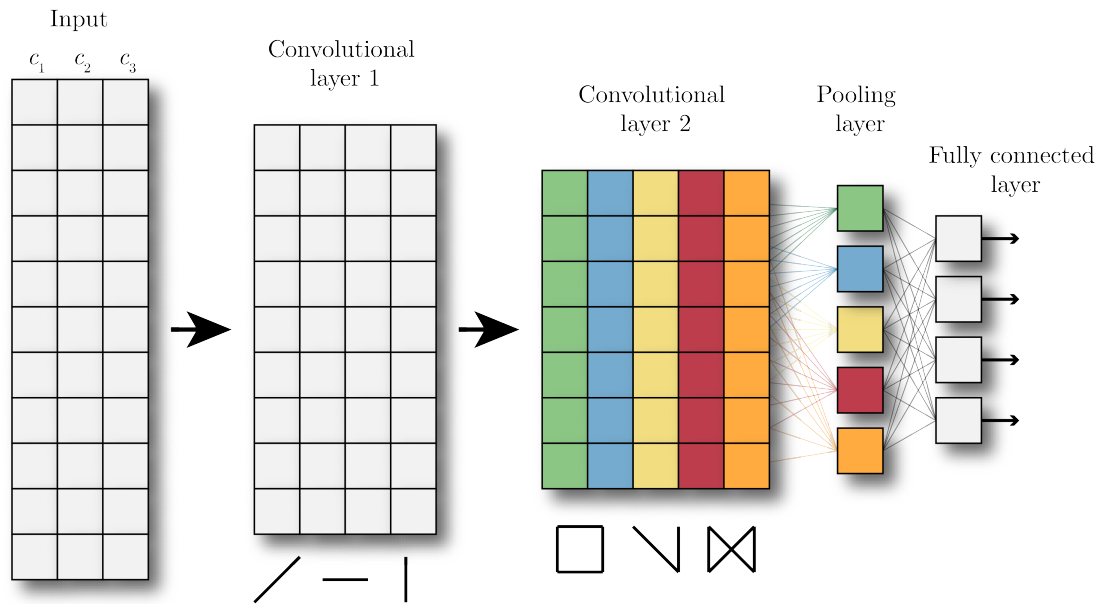


Figure 4.9: General structure for 1-D CNN, with multiple channels. Following the input, there are multiple convolutional layers, which are decreasing in size, but include more maps, representing different filters. At the beginning those detect simple shapes such as lines, but the patterns get more complicated with deeper networks. Pooling layers are used to translate those patterns into simple scores, for which a fully connected network can analyse it further.

pixel. Those channels work in the same way as multiple feature maps in the convolutional layer.

- Set of convolutional layers, each represented by the weight tensor $w_p^{(k,m,n)}$. The weights indicate both the strength of the connections between neurons from two layers and the filter shape, i.e. which of the connections have nonzero value. Filters in the first layers are usually set to detect small features, such as horizontal and vertical lines. Those are then connected by filters in consecutive layers into more complicated shapes, similar to the operation of biological neurons in the visual cortex. Usually the size of layers shrinks in their initial x and y pixel dimensions¹² but increases in the number of filters to account for a larger number of possible more complicated patterns that the network can detect.
- The pooling layers, which combine or reduce the units from convolutional layers. They act by either finding the maximum value or a sum of few neighbouring neurons, or even an entire map. Usually, they are introduced at the end of the convolutional layer to transform the architecture for a prediction task. Because they are expressed as simple max, average, or sum functions, those layers do not take part in the learning procedure.
- Fully connected layers, which take the input from the pooling layers, perform some additional nesting and end in the output layer.

An example of such structure can be seen in Figure (4.9). Those types of networks seem to be especially good for categorisation tasks, where the distinction between classes is dependent on some patterns in the data. Additionally, because the weights do not depend on the positions of the neurons, the activation will occur independently of the position of the pattern.

¹²Because of the larger size of filters.

Chapter 5

Machine learning application to muon spectroscopy

This chapter serves as a quick aside from the main topic of the thesis - the learning of entanglement through machine learning (ML). Nevertheless, it was an important part of my early Doctoral training, as I was able to learn and apply principal component analysis (PCA) to a set of real-life data. The research I conducted eventually resulted in the publication of two articles as a main author and a third as a co-author. The following sections are a summary of those results.

In the first part, a short introduction to muon spectroscopy will be provided, along with an explanation of why there are some situations where utilizing ML might be preferable to standard analysis. Our approach was to use an unsupervised ML method — PCA (for detailed explanation see Chapter 4), which will be presented here in the context of the muon spectroscopy data, along with more details regarding handling the error. In the last part, our published findings and the extensions on the PCA method will be shown.

5.1 Muon spectroscopy experiment

Muons are exotic fundamental particles that were first discovered in 1936 by Carl David Anderson and Seth Neddermeyer, during the study of positron creation by cosmic rays [87]. They are naturally created in higher parts of the atmosphere of the Earth, where the high-energy cosmic rays interact with the nuclei of atoms [88, 89], creating short-lived pions, which decay into muons with an average life time of 26 ns. The muons themselves are also unstable and will eventually turn into electrons or positrons, depending on the initial charge of the pion. The standard model classifies muons as one of the generations of leptons [90], along with electron/positron and tau particles. With a weight of ≈ 207 of m_e , they are sometimes considered heavy electrons, since their other properties are similar.

Most famously, the detection of naturally created muon particles in the laboratories on Earth is often cited as early evidence for Einstein’s special theory of relativity [91]. Since muons are created where cosmic radiation has high intensity, even though their average speeds reach $v_\mu \approx 0.991c$, the short life-time $t_\mu = 2.2\mu s$ would prevent them from reaching Earth. However, due to the time dilation, particles “live” longer (or the distance from space to the Earth is shortened from the perspective of muons) and are captured by detectors on the surface. The experiment that measured that the momentum-dilation relationship is exactly as predicted by Einstein was conducted by Bruno Rossi and David Hall from the University of Chicago in 1941 [92].

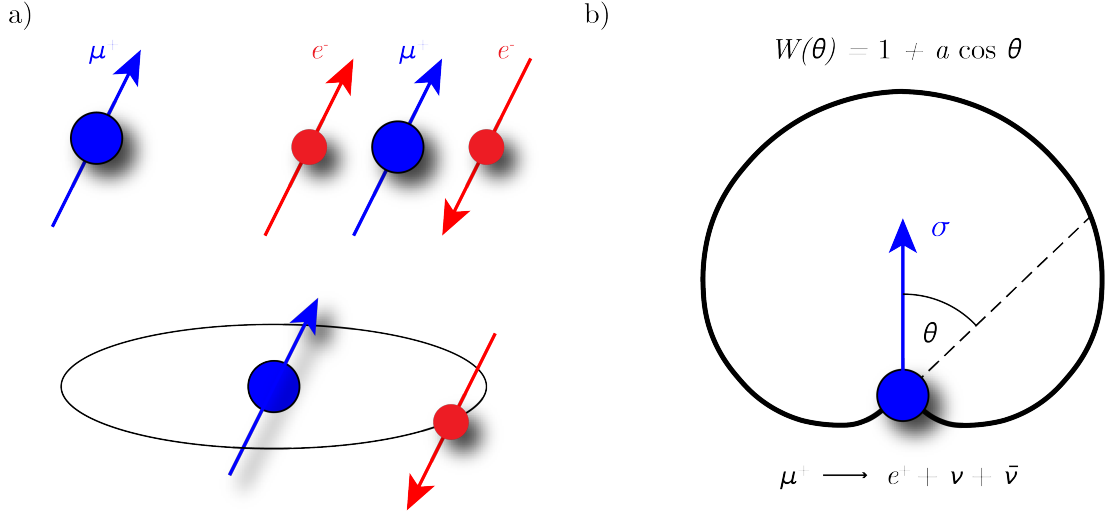


Figure 5.1: Different muon states and the distribution of positron decay. a) Thermalized muons can be implanted in three different states: positively charged muon, muon coupled with two electrons and neutral muonium atom. b) After the decay into positron and two neutrinos, the momentum of the positron is positively correlated with muon's spin at the time of the decay.

The idea of using muon streams as an analysis tool in low-energy physics was introduced after the first successful creation of pions in proton accelerators, at the end of the 1950s [89]. The two-body decay of pions results in the creation of polarized positive muons, which have just the right energy to enter the sample and thermalize, then implant themselves in the bulk. The thermalization of muons involves a few consecutive processes: The initial kinetic energy (ranging in 4-100 MeV) is mostly lost due to ionisation of atoms and scattering with electrons due to Coulomb interactions. The next process depends on the electronic state of the probed material. If muons do not couple to any electrons in the sample, or they do couple to paired electrons in a singlet state, they form a positive or negative diamagnetic state (this state is not charge neutral). However, if muons can couple to single unpaired electrons, they will form a neutral muonium, which mimics the hydrogen atom. This is also referred to as a paramagnetic muon state. At this point in both cases, muons have lost enough kinetic energy that they thermalize completely in picoseconds. The whole process between muons entering the sample and stopping in the bulk takes approximately 10^{-9} s, which is much less than the life-time of the particle.

Inside the material, due to the local magnetic field, muonium or diamagnetic muon will undergo Larmor precession - the spin of the muon will start to rotate around the direction of the field. Considering that: 1) Initial muon beam was fully polarised and 2) The momentum of the positron that is created in the decay is dependent on the spin direction of the muon, the distribution of positron velocity will depend on the total magnetic field, although the correspondence between momentum and spin direction is not one-to-one (see Figure 5.1 b). Quantum field theory gives the probability of positron momentum as a function of an angle as [89]

$$W(\theta) = 1 + a \cos \theta, \quad (5.1)$$

where θ is an angle between spin direction and the positron momentum after the decay, and a is a parameter that depends on the energy of the positron. Given these facts, it is possible to obtain information about the intrinsic magnetic field distribution in the sample by looking

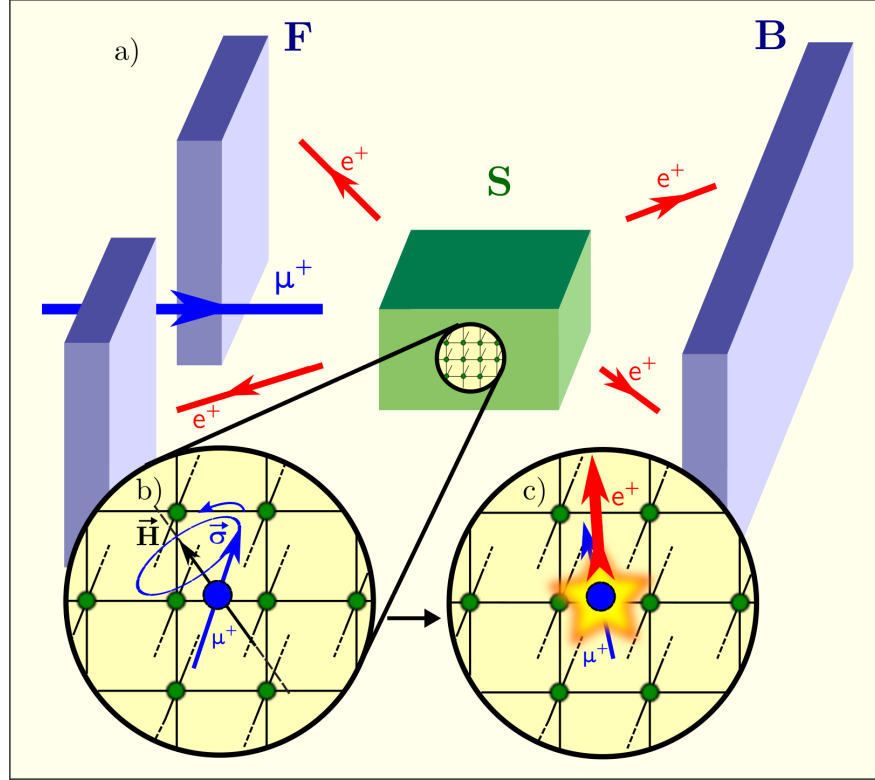


Figure 5.2: Muon SR setup in backward-forward (BF) geometry. a) The stream of muons enters the sample through a hole in backward (B) set of detectors. Particles thermalize and are implanted somewhere within the sample (S). b) Larmor precession due to the combined local and external magnetic fields. c) Muons with a half-time of $2.2\mu\text{s}$ eventually decay into positrons, which are detected either by B or forward (F) set.

at the angular distribution of the positron emission, and especially how it changes in time.

The type of experimental output from muon spectroscopy (often abbreviated as μ SR) depends both on the geometry of the detector setup and whether the external magnetic field is applied. One of the examples of the setup is presented in Figure 5.2. Detectors count positrons that are directed forward (F) or backward (B). The important quantity for this setup is usually the asymmetry spectrum defined as¹

$$A(t) = \frac{N_B(t) - \alpha N_F(t)}{N_B(t) + \alpha N_F(t)}, \quad (5.2)$$

where $N_{F/B}(t)$ are the number of counting events for F and B at time t , and α is the ratio between efficiencies of F and B positron detectors (usually in the range of 0.2 to 0.27). The shape of the asymmetry function will depend on the combined internal and external magnetic fields “felt” by muons.

An interesting version of this experiment is the zero-field muon spectroscopy (ZF μ SR), where the muons only probe internal fields. The usual approach for analysing the data is to perform a regression analysis with a good guess for a fitting function. Using some sensible approximations, one can derive an analytical function for ZF μ SR with the following derivation [88]: Starting from the expectation value of the muon spin direction in a magnetic field

¹This definition works for pulsed muon sources, where the background detection is negligible. For more general definition see [89].

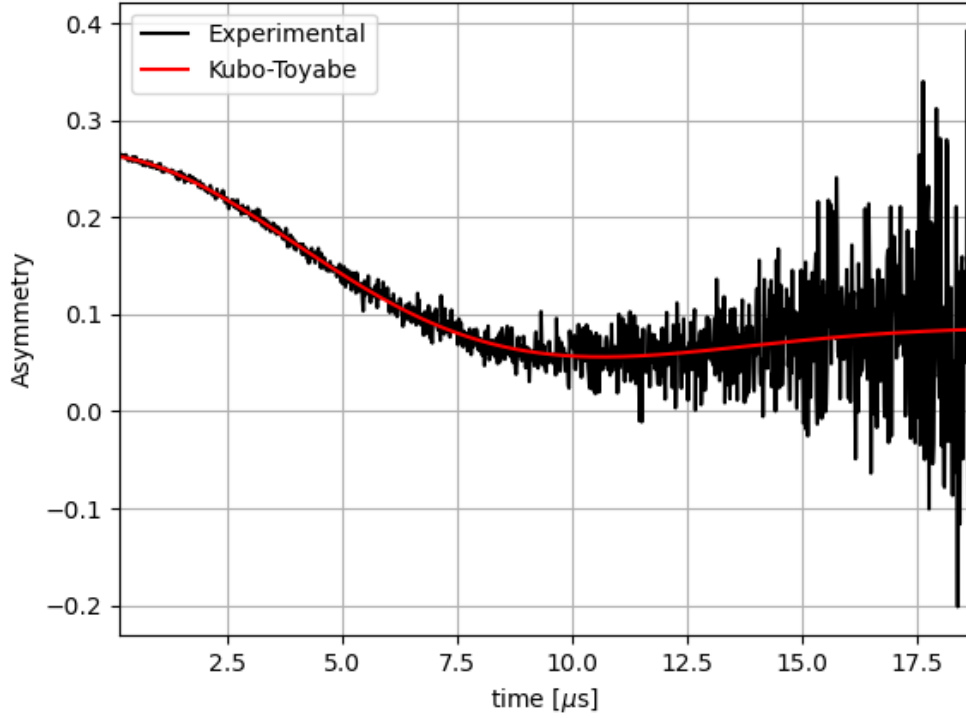


Figure 5.3: Asymmetry function from a ZF μSR of LaNiGa_2 at $T = 0.5K$ and the Kubo-Toyabe function fit to the data.

$\mathbf{H}$, which is given by

$$\langle \sigma_\alpha(t) \rangle = \cos^2 \theta + \sin^2 \theta \cos(\gamma_\mu H t), \quad (5.3)$$

where θ is an angle between muon spin and magnetic field, γ_μ corresponds to the gyromagnetic ratio of the muon (equal to $135.5 \text{ } 2\pi \text{ MHz T}^{-1}$), and $\alpha = x, y, z$ runs through all components of the spin. Positron detection is a stochastic process, and the distribution of magnetic field $p(\mathbf{H})$ has to be taken into consideration when adding all the contributions from different muons to the asymmetry function. Integrating over all possible fields, we get directional polarisation function

$$P_\alpha(t) = \int d^3H p(\mathbf{H}) \langle \sigma_\alpha(t) \rangle. \quad (5.4)$$

If we assume that the magnetic field does not depend on the spin component α , and has a Gaussian distribution with zero mean and a standard deviation Δ , the integration will produce Kubo-Toyabe function [88, 93, 94, 95]

$$G_{KT}(t) = \frac{1}{3} + \frac{2}{3} e^{-\frac{1}{2} \Delta^2 t^2} (1 - \Delta^2 t^2) \quad (5.5)$$

The fitting function to asymmetry spectrum will have additional factors, related to the background detection and the fact that muons will decay in time. The standard assumption of exponential decay will give us the fitting function of the form

$$A_{\text{phen.}}(t) = A_0 \left(\frac{1}{3} + \frac{2}{3} e^{-\frac{1}{2} \Delta^2 t^2} (1 - \Delta^2 t^2) \right) e^{-\lambda t} + A_{\text{bckg}}, \quad (5.6)$$

where A_0 refers to the initial asymmetry, λ is the muons decay rate and A_{bckg} is the amount of background asymmetry.

The Kubo-Toyabe function works in many cases of ZF μ SR experiment. However, the assumption of an isotropic Gaussian distribution of the field inside the sample does not have to be satisfied by all interesting magnetic materials. Specifically, the assumption of Gaussian distribution with zero mean excludes any long-range magnetic ordered states, meaning that for example any unconventional superconductors, which exhibit magnetic order, cannot be successfully fitted to Kubo-Toyabe. One such example is the recently studied time-reversal symmetry breaking (TRSB) superconductors, LaNiGa_2 and $\text{Cu}_x\text{LaNi}_{1-x}\text{C}_2$. The changes in the asymmetry functions of those materials at different temperatures are very subtle and the regression analysis seems to not fully capture the phase transitions. On the other hand, there are interesting materials for which the assumption is satisfied like quantum spin liquids. Here we aim to develop a method that could potentially work for all those materials, since it does not depend on the assumption like the mean zero Gaussian distribution. The next section explains how the PCA can be used for the analysis of muon spectroscopy data.

5.2 PCA for asymmetry functions

As explained in Chapter 4, the PCA is an unsupervised ML method, which means that it searches for correlation in the data without any prior bias. This can be beneficial for analysing μ SR data, since there are no assumptions stated on the material probed (such as the distribution of the field, etc.). The only requirement for the method is that there are enough data points, so that the covariance between them is larger than the measurement error. At this point, there are also no hyperparameters, although one will be provided later in this section.

To perform a singular value decomposition (SVD), a set of measurements has to be written in the form of a matrix. Each measurement can be thought of as a vector (see Figure 5.4), provided that all the asymmetry functions have the same number of time bins. These vectors can be then written as columns of matrix $\mathbf{A}$

$$\mathbf{A} = \begin{bmatrix} A_1(t_1^1) & A_2(t_1^2) & \dots & A_M(t_1^M) \\ A_1(t_2^1) & A_2(t_2^2) & \dots & A_M(t_2^M) \\ \vdots & \vdots & \ddots & \vdots \\ A_1(t_N^1) & A_2(t_N^2) & \dots & A_M(t_N^M) \end{bmatrix} \quad (5.7)$$

If all of the times in each row $t_0^i, t_1^i, \dots$ represent the same times in the measurement, then the PCA is a viable method.

5.2.1 Experimental error in PCA

Looking at the typical asymmetry spectrum from a pulsed muon source, one can notice that the deviation from the mean seems to grow over time. This is a consequence of the decreasing number of muons in the sample due to positron decay. In regression analysis, this can be taken into account by adding a weight to each point (equal to the inverse of error) or by re-binning, so that each new time interval holds approximately equal amount of standard deviation [96]. To improve our ML results we have tried a similar procedure applied to PCA.

We can write down the error of asymmetry spectrum in a matrix form, similar to the data

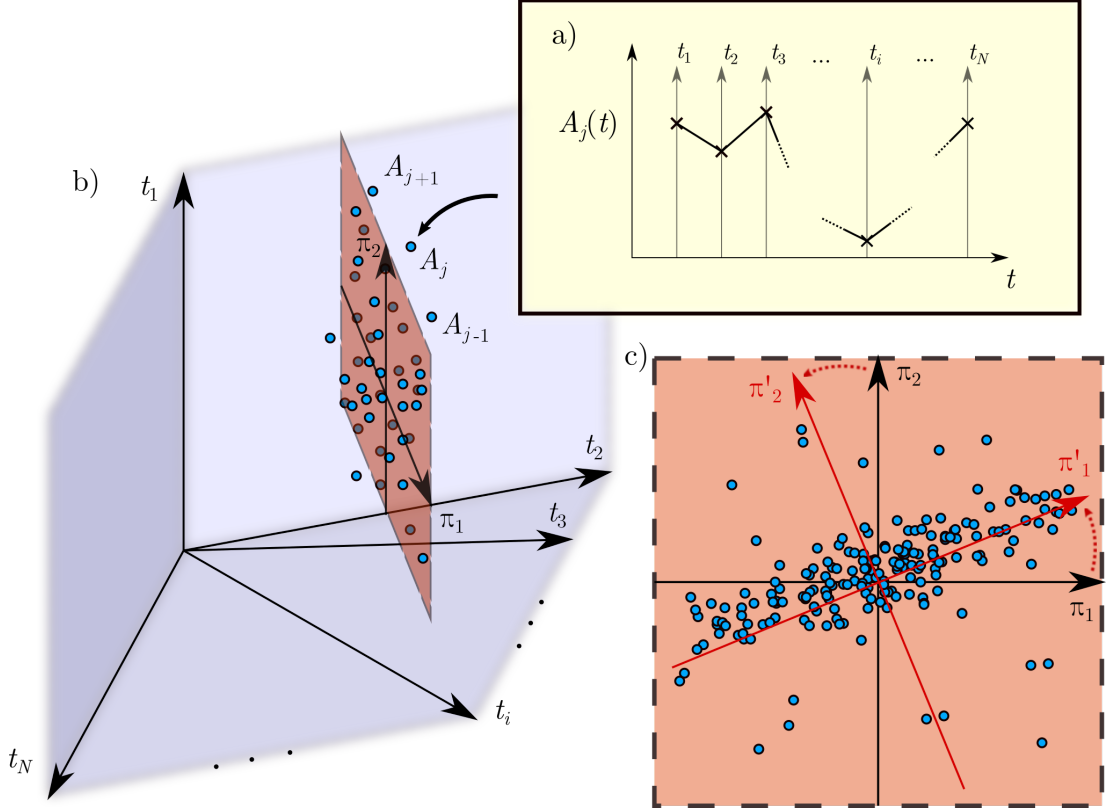


Figure 5.4: Graphical representation of PCA performed on μ SR data. a) The j -th asymmetry function in a set of data can be represented as a N dimensional vector. b) High-dimensional vector space of all asymmetry spectra. If the functions are linearly correlated, they will lie near smaller sub-space (represented by reddish plane). c) The PCA finds new orthogonal basis $(\pi_1, \pi_2) \rightarrow (\pi'_1, \pi'_2)$, which holds most covariance in least amount of basis vector in the data.

in 5.7:

$$\mathbf{E} = \begin{bmatrix} E_1(t_1^{\text{raw}}) & E_2(t_1^{\text{raw}}) & \dots & E_M(t_1^{\text{raw}}) \\ E_1(t_2^{\text{raw}}) & E_2(t_2^{\text{raw}}) & \dots & E_M(t_2^{\text{raw}}) \\ \vdots & \vdots & \ddots & \vdots \\ E_1(t_N^{\text{raw}}) & E_2(t_N^{\text{raw}}) & \dots & E_M(t_N^{\text{raw}}) \end{bmatrix}, \quad (5.8)$$

where $t_i^{\text{raw}}, i = 1, 2, \dots, N$ represents the initial time bins. There exists a way to incorporate weighted entries to PCA [97] but we have opted for re-binning i. e. changing t_i^{raw} to a new set of time intervals t_i^{new} , such that on average each time window holds the same amount of standard deviation. The difference between re-binning for PCA and regression analysis is that in the former case, all of the measurements have to be taken into account at the same time. This will work if the error grows similarly for all asymmetry curves.

The procedure of finding the correct time intervals starts with establishing the desirable amount of error. For the asymmetry curves from muon spectroscopy, this can be equal to the averaged initial error, since it is increasing due to decreasing muon counts

$$\bar{E}_1 = \frac{1}{M} \sum_{i=1}^M E_i(t_1^{\text{raw}}). \quad (5.9)$$

The next step is to iterate over the rest of the original time windows and compare the average

error with $\bar{E}_1$. For each new bin t'_j starting from old window t_k we evaluate the ratio

$$\frac{\bar{E}_1}{\bar{E}_k}, \quad (5.10)$$

where $\bar{E}_k = \frac{1}{M} \sum_{i=1}^M E_{t_k}$. This ratio in general will be smaller than 1, and so we add the next time window t_{k+1} to the new one $t'_j \rightarrow \frac{t_k + t_{k+1}}{2}$. Next ratio is calculated as

$$\frac{2\bar{E}_1}{\sqrt{(\bar{E}_k)^2 + (\bar{E}_{k+1})^2}}, \quad (5.11)$$

where the factor of 2 in the numerator is due to the definition of standard deviation of averaged quantity. Time windows are added until

$$\frac{L_j \bar{E}_1}{\sqrt{\sum_{l=0}^{L_j-1} (\bar{E}_{k+l})^2}} \approx 1, \quad (5.12)$$

and the final new time bin is set to $t'_j = \frac{1}{L_j} \sum_{l=0}^{L_j-1} t_{k+l}$.

In the above example, the $\bar{E}_1$ is defined through the averaged initial error, but in principle it can be set to any amount. This can be thought of as a hyperparameter of this ML method since it will affect the final analysis. For example, if the asymmetry spectra feature fast oscillation, the re-binning process can remove this and affect the principal components (PCs) and PC scores. It is important to note that regression analysis is still affected by similar issue if re-binning is performed.

5.2.2 Results - simulated data

We first tested the method on simulated data to check the effect of the error on the analysis. The decaying Kubo-Toyabe functions (5.6) were generated for various values of parameters Δ and λ . Additionally the Gaussian noise was added with standard deviation changing as a function of time in a following way

$$\Sigma(t) = R(a^t + b), \quad (5.13)$$

with a and b being fixed parameters. This reflects the noise in the data, which increases due to fewer number of muons in the sample. The parameter R changes the overall scale of the error and therefore can be used as a factor changing the quality of the analysis.

Figure 5.5 shows the results of applying PCA to the simulated data for different values of the added error. The generated Kubo-Toyabe functions have a fixed value of λ and the parameter $\Delta(T)$ has a linear dependency on the temperature below the “phase transition” $T < T_c$, and a constant value above $T_c = 2.1 K^2$. The 1st PC scores (second row) seem to be a good indicator of the phase throughout the different error values. The 2nd PC scores are much more susceptible to noise, and the 2nd PC curves lost their shape almost entirely by $R = 0.0025$. The first PC vector remains similar to the one obtained with no error, even though the time bins are changing.

Here, the analysis was performed on Kubo-Toyabe functions and the results suggest that PCA can be used to detect phase transitions, even for noisy data. This motivates using the method on data from real muon spectroscopy, although one must be careful because we suspect that at least for some of the materials studied, the Kubo-Toyabe is not a correct fitting function.

²This behaviour is taken from regression analysis of one of the materials that we have experimental data on.

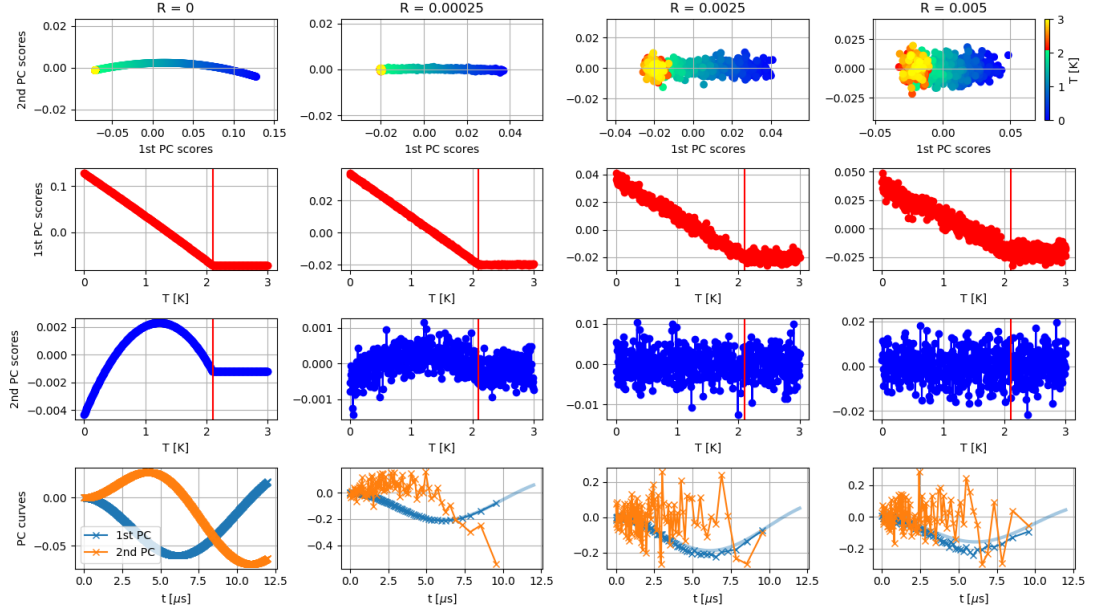


Figure 5.5: PCA performed on simulated Kubo-Toyabe functions. Each column represents data with different amount of error, with third column ($R = 0.0025$) corresponding to error comparative to the experimental data. First row shows the 1st vs 2nd most important PC scores, with colormap corresponding to change in temperature. On Second and third row, the individual changes in 1st and 2nd PC scores can be seen as a function of temperature. The fourth column shows changes in PC vectors for different error. The faded teal plot on second, third and fourth column indicates the 1st PC vector without the error.

5.2.3 Results - experimental data

We tested the PCA method of analysing μ SR data on measurements of different magnetic materials. Among those were suspected spin-liquid (LuCuGaO_4), antiferromagnet (iron oxy-selenide $\text{BaFe}_2\text{Se}_2\text{O}$) and TRSB superconductors (LaNiGa_2 , $\text{LaNi}_{1-x}\text{Cu}_x\text{C}_2$ for $x = 0, 0.1, 0.25$). The latter materials were the primary aim of our research, since they were particularly hard to analyse using the usual methods. This is due the fact that the change in intrinsic magnetic field can be very subtle, but in some theories, the magnetic field is predicted to emerge spontaneously at the same time as the superconducting order [98, 99, 100, 95]. If the correlation between sudden change in muon spectroscopy curves and the critical temperature can be detected, it would be a major proof of TRSB behaviour. The unsupervised ML technique might provide a better insight into the internal magnetic field change.

Before showing full results of PCA for muon spectroscopy, it is worth to see how the analysis works in detail. In Figure 5.6, we show an example that illustrates how the method detects changes in the shape of a set of experimentally measured asymmetry functions, obtained for a single material at different temperatures. The way in which those functions differ from each other is reflected in their respective scores for the 1st and 2nd PCs. Both high temperature (a) and low temperature (c) measurements have almost linear shape and they only differ in the values for the first PC score. Looking at the 1st PC shape (panel (e), blue curve), we can see that it is also almost linear and when multiplied by a large negative value—as it is for high temperature asymmetry function—and then added to the average (d), it increases the overall slope. For the low temperature curve, it is added with positive sign, which means that it will instead decrease the slope. We can see that it is exactly the difference between two

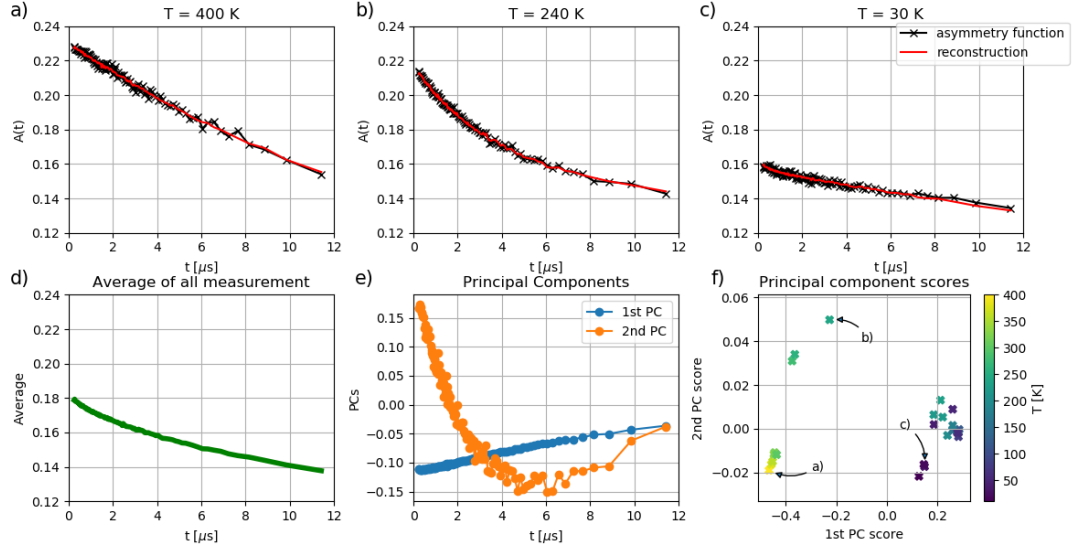


Figure 5.6: An illustration of how PC analysis can be used to reduce the dimensionality of a muon data set. The set consists of a sizeable number of experimentally-obtained muon asymmetry functions $A(t)$. The black curves in panels (a)–(c) present three particular examples. Each curve has 110 time stamps and therefore constitutes a point in a 110-dimensional space. PCA yields a small number of PCs which, through linear combination, can accurately describe any curve in the data set. In our case, we find the two most important PCs shown in panel (e). The reconstruction of the original data using the PCs and the average (d) (see Section 4.4.1) can be obtained by the formula “reconstruction = average + 1st PC score $\times$ 1st PC + 2nd PC score $\times$ 2nd PC + (higher PCs that were omitted)”. From that we can interpret the PCs as the most common deviations from the average curve. The reconstructions are shown, for our three examples, by the red curves in panels (a)–(c). Using only the first two PCs gives an accurate reconstruction in this case and therefore enables us to represent each curve by a single point on a two-dimensional plane (f). For this example we used 25 $A(t)$ curves for the material $\text{BaFe}_2\text{Se}_2\text{O}$ obtained at 25 different temperatures.

asymmetry functions (a) and (c). On the other hand the middle curve (b) differs mostly in second PC scores from the other two. When the second PC vector ((e), orange curve) is multiplied by a positive value and added to the average, it creates a more convex curve, which is reflected by the shape of the corresponding asymmetry function (b). This example illustrates that the method gives us more information about the data than for example a time-integrated asymmetry. Some important changes can be represented by PCs which have a mean value close to zero (like 2nd PC in Figure 5.6), and therefore they would not be seen in an average asymmetry.

For each material, there are between 8 and around 40 asymmetry functions taken at different temperatures (all measurements are ZF μSR). In the first attempt to perform PCA on the experimental data, we applied the analysis for each set of data separately, meaning that for each material separate matrices $\mathbf{A}_{\text{mat}}$ and $\mathbf{E}_{\text{mat}}$ were created (the columns of $\mathbf{A}_{\text{mat}}$ and $\mathbf{E}_{\text{mat}}$ correspond to asymmetry functions taken at different temperatures, see (5.7) and (5.8)). The results of this approach can be seen in Figure 5.7. Each of different material is presented in separate column. The accuracy of the PC scores can be read from the scree plots (last row), where the more the first PCs contribute to the overall covariance, the more information is held in those PCs. This means that if the scree plots are close to flat, much information is lost, and the phase transition might not be visible in the PC scores vs temperature plots. Judging

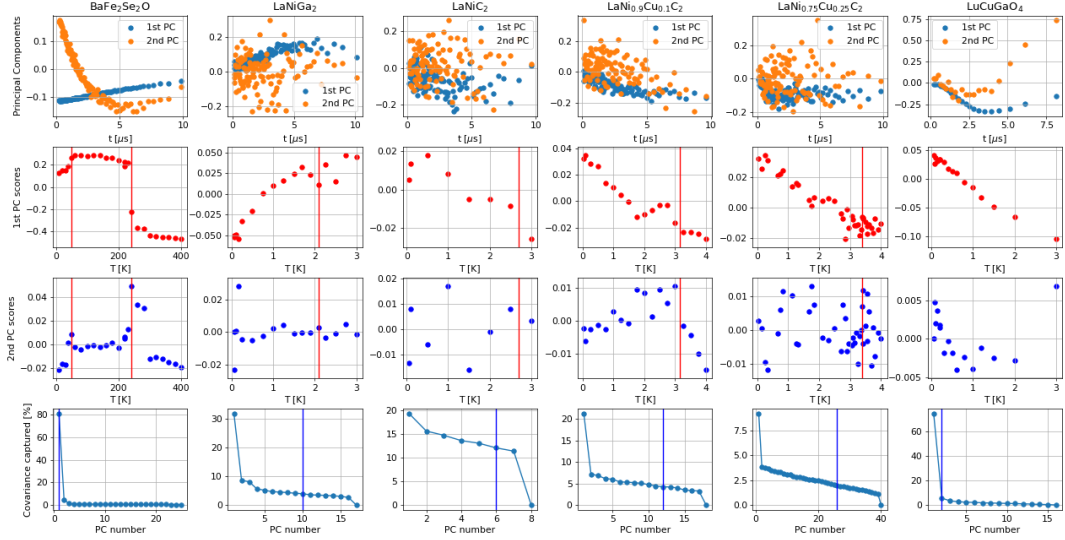


Figure 5.7: Results of PCA performed separately for all sets of data. Each column represents experimental data coming from a given magnetic material. First row shows first and second PC curves. In Second/third row the 1st/2nd PC score dependency on temperature is plotted, and the last row shows the scree plots for each set of data.

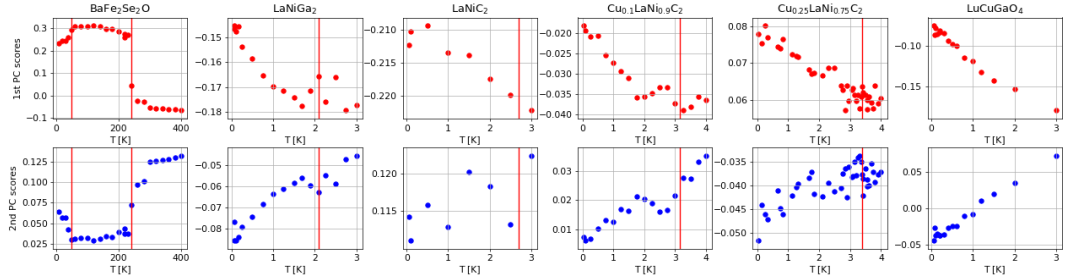


Figure 5.8: Results of joint PCA of all of the magnetic materials under study. First/second row shows the dependence of 1st/2nd PC scores on temperature correspondingly. Red vertical lines signify values of phase transitions taken from [7, 8].

from the plots 5.7, the superconducting materials are approximately in that category. Nevertheless, in most of them, the first PC holds a large fraction of the covariance, and some change of behaviour can be seen where expected. In the case of LuCuGaO_4 and $\text{BaFe}_2\text{Se}_2\text{O}$, only one or two first PCs hold above 80% of covariance.

The flat shape of the skree plots in 5.7 is not necessarily a result of the nonlinear/no correlation among the measurements. This can also be explained by the error being larger than the average distance between two points in the vector space of asymmetry functions (see Figure 5.4). If that is true, the larger number of asymmetry functions would help to find better PCs. Normally, that would require additional measurements, so we tried performing PCA simultaneously on all of the materials. That means writing all of the asymmetry vectors in the same matrix $\mathbf{A}_{\text{all}}$, and the errors in $\mathbf{E}_{\text{all}}$. Although the asymmetry functions from different measurements had vastly different shapes, we expected that there is enough common features for the technique to work.

The raw data from different materials did not always have the same time windows with the same resolution. Since there seemed to be no high-frequency oscillations in all measurements, the initial re-binning to establish the same time-bins could be performed.

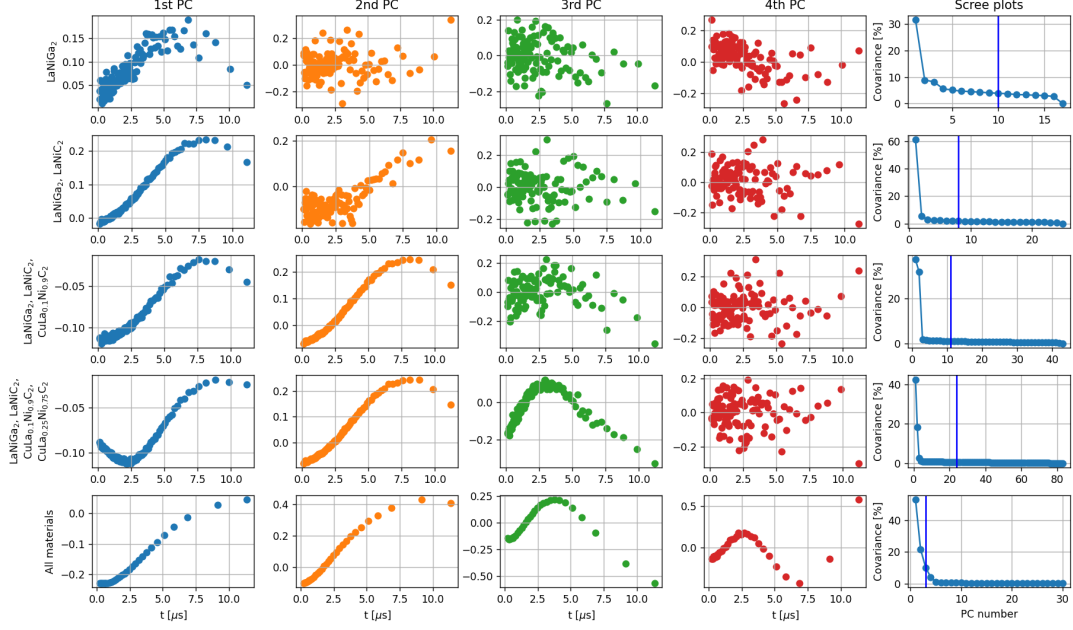


Figure 5.9: The change of PC vectors with addition of new batches of data. Each column represents different PC curve, with the last being the scree plot of PCA. In the top row, only asymmetry spectrums of LaNiGa_2 data are provided, and the data from different materials are gradually added in following rows (as indicated on left of the plots). The blue vertical line in scree plots indicate the amount of PCs needed to hold 80% of covariance.

The results of our second approach can be seen in Figure 5.8. Perhaps the most interesting conclusion is that the dependency on temperature of the first PC score has not changed by much for all materials. In contrast, the 2nd PC score has changed, and it appears to be a better indicator of phase transition, at least for LaNiGa_2 and $\text{LaNi}_{0.75}\text{Cu}_{0.25}\text{C}_2$. However, the transition at $T_c = 3.1\text{K}$ for $\text{LaNi}_{0.9}\text{Cu}_{0.1}\text{C}_2$ seems to be sharper for the previous analysis. Overall, although the materials studied have vastly different magnetic properties, the PCA performed on all of them simultaneously appears to find the common correlations and to be a good analysis tool for finding sudden changes with respect to temperature. It is worth noting that in both versions of the analysis, the largest variance in PC scores are seen for the antiferromagnet $\text{BaFe}_2\text{Se}_2\text{O}$, where the underlying experimental data spans the largest T range. From this, we can infer that PCA works best with data with larger differences in temperature, and this can be expected, as large changes in magnetic order cause large differences in asymmetry functions. More subtle changes, as in TRSB superconductors, will be harder to detect, but the relative differences are still visible enough. For example, in the separate analysis (Figure 5.7), both first and second PC scores go through positive and negative values, meaning that the contribution to reconstruction from PCs inverts with change in temperature.

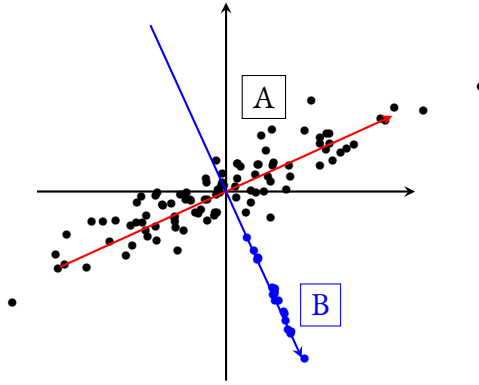
The additional positive of using all available data at once is that it usually improves the PC vectors and skree plots, giving more confidence in the analysis. Figure 5.9 shows the gradual improvement in vector shapes, when more asymmetry functions are added to the analysis, even for different materials. The PC curves become smoother the more data is provided, and additionally the scree plots have a much more pronounced “exponentially decayed” shape. In the last row, only the three first PCs capture over 80% of covariance, as opposed to the initial 10 PCs.

5.3 Extensions

The previous section summarized the research previously published in the special issue of Journal of Physics: Condensed Matter [1]. Since then, there have been some additional results regarding the use of the method in muon spectroscopy. These were later published as a proceedings article [2] and in the article discussing magnetic properties of the layered kagome superconductor [9], where we introduced the error calculation for PC scores.

5.3.1 Cumulative principal component score

The introduction of simultaneous PCA for materials with different physical properties has led to a further study of the capacity of PCs to reconstruct the data coming from a given material. It is possible that when PCA is performed on data combined from all materials, the first PCs will not contribute more to the reconstruction of a particular set of data, which is significantly different from the other asymmetry functions. The 2-D example of this would look like this



The two clouds of points represent different sets of data. Since there are many more points in the A category, if we perform PCA on all of the data the rotated set of coordinates would indicate that most of the covariance is held along the red line (1st PC). However, if we were to look just at batch B, the direction that most prominently reflects the change in this dataset is along the blue line (2nd PC).

The simplest metric that can indicate the relevance of a given PC for a part of the data can be constructed by summing the projections of that cluster unto the PC vectors. In the work described in this article [2], we define it as a cumulative principal component score (cPCS). For n -th PC and i -th set of measurements it can be written as

$$cPCS(i; n) = \sum_{j=1}^{N_i} |PCS_n^i(j)|, \quad (5.14)$$

where j runs over all data-points in i -th cluster. The absolute value reflects the fact that, in PC reconstruction, the negative and positive contributions are equally relevant. The cPCS can be thought of as a measure of contribution to the PC reconstruction.

Figure 5.10 shows the first four PC scores for the same magnetic materials as in Figure 5.8. Alongside are the cPCS for each material. As expected, cPCSs indicate that for most data sets, the 1st and 2nd PCs are the most important, and the two exceptions are LuCuGaO_4 and $\text{LaNi}_{0.9}\text{Cu}_{0.1}\text{C}_2$, where 3rd PCs hold more covariance than 2nd. In the case of the latter, the contribution from third PC is almost exactly the amount of the first. This is interesting

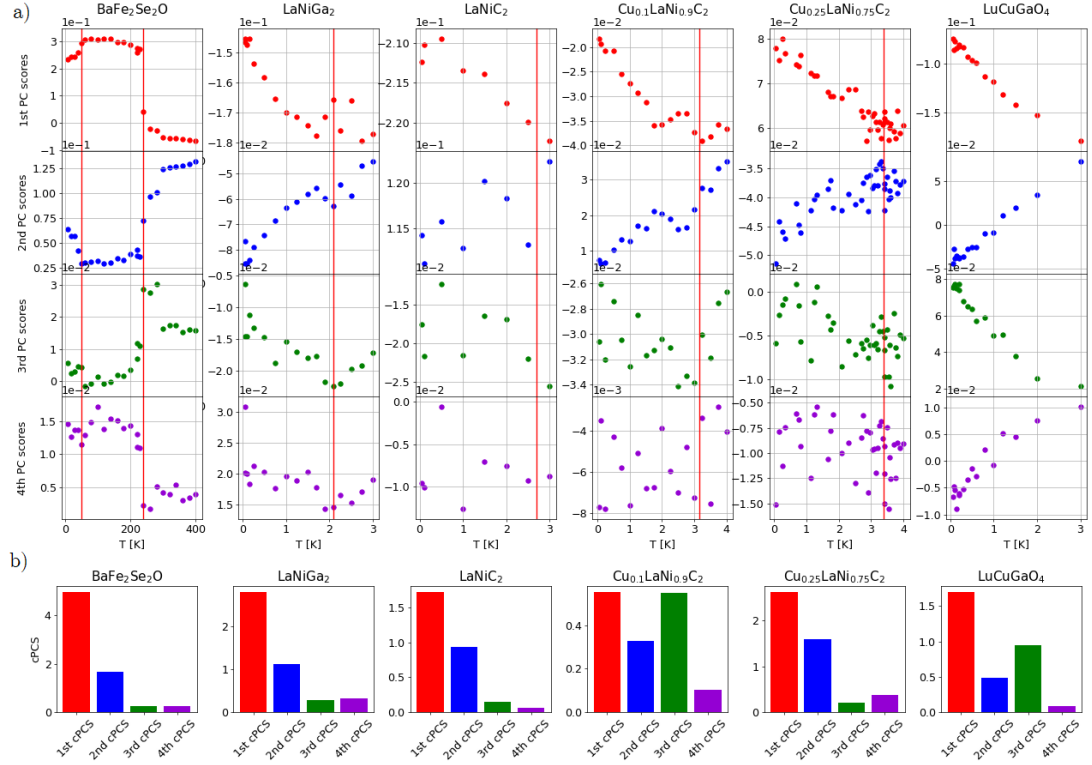


Figure 5.10: a) The temperature dependence of first four PC scores for asymmetry functions of magnetic materials. Red vertical lines indicate phase transitions [7, 8]. b) CPCS of first four PCs for corresponding materials.

especially when comparing the temperature dependency of 3rd PC of LaNi_{0.9}Cu_{0.1}C₂ in the joint analysis (5.10) and its 2nd PC in separate PCA (5.7). The shape of both of them is similar³, and shows visible kink near the superconducting phase transition. This confirms that although joint analysis can be used to improve PCA, greater care is needed in evaluating PC scores for individual batches of data.

5.3.2 Principal component score - calculating error

In the standard analysis, which includes the regression to a function, the margins of error of the fitting parameters can be calculated straightforwardly. In the previous figures, no error estimate was provided for PC scores, since it is not standard practice in PCA. However, when looking at the PC scores from the experimental data, it seems obvious that they do not represent the true values of the projections to PCs. This is also seen in the results for simulated data (see Figure 5.5), where the introduction of noise in the asymmetry functions eventually causes a loss of information in the PC scores. For better comparison to regression analysis, in PCA the standard deviation can be calculated for PC scores by using the definition

$$PC_{\text{score}}^{[n]}(T_i) = \sum_j^M PC^{[n]}(t_j) \times A(T_i, t_j), \quad (5.15)$$

³One is a vertical mirror reflection of the other but it is not important, since the simultaneous sign change of PC scores and PC vectors gives the same result (see Chapter 4)

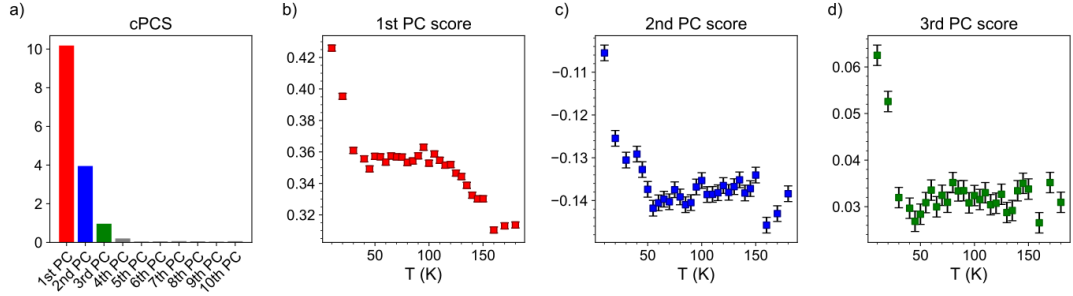


Figure 5.11: PCA of muon spectroscopy of CsV_3Sb_5 . PCA was performed jointly with other magnetic materials mentioned in this chapter. a) The cPCS for CsV_3Sb_5 . b-d) The three most important PC scores with error estimations. Figure taken from [9].

Principal component	T_1	T_2	T_3	...	T_N
1st	0.001912	0.001913	0.001914	...	0.001915
2nd	0.001829	0.001831	0.001831	...	0.001832
3rd	0.002189	0.002191	0.002192	...	0.002189

Table 5.1: Error margins for first three PCs for CsV_3Sb_5 data.

where $PC^{[n]}(t_j)$ is n -th PC vector and $A(T_i, t_j)$ is a given value of experimental asymmetry function. Assuming that the error is coming strictly from the noise in the data collection, one can define the standard deviation of $PC_{\text{score}}^{[n]}(T_i)$ as

$$\text{SD} [PC_{\text{score}}^{[n]}(T_i)] = \sqrt{\sum_j^M (PC^{[n]}(t_j) \times E(T_i, t_j))^2}, \quad (5.16)$$

where $E(T_i, t_j)$ is the standard deviation of i -th measurement at j -th time window.

This estimation was used in the article discussing the magnetic properties of kagome superconductor CsV_3Sb_5 [9]. Figure 5.11 shows the results. The error margins seem to be growing with less important PCs, but that is due to the smaller values of PC scores overall. The absolute value of errors does not change, as demonstrated in the table 5.1. This is a consequence of the normalisation of PC vectors and the re-binning of time-bins (as explained in 5.2.1). Because of the re-binning, the error part of equation 5.16 is more or less constant, and the standard deviation can be approximated

$$\text{SD} [PC_{\text{score}}^{[n]}(T_i)] \approx \bar{E}(T_i) \sqrt{\sum_j^M PC^{[n]}(t_j) = \bar{E}(T_i)}, \quad (5.17)$$

where the $\bar{E}(T_i)$ is the approximate error held by each time-window after re-binning.

Chapter 6

Towards quantum state tomography from neutron-scattering functions

In order to develop a machine learning procedure to learn entanglement from the experimental data, we had to first establish which observables are the most suitable for the training process. While initially we opted to try the dynamical structure factors, the recent work by one of the supervisors of this thesis [101] proved the existence of a bijection between static two-point correlators and the coefficients of a general Heisenberg-like Hamiltonian¹. Following that discovery, we worked to extend the proof for finite temperatures and to include the Hamiltonians with external magnetic field. Moreover, we explored the fitness landscape of neutron-scattering functions as a function of deviations from the initial state. This showed a strong suggestion that the relationship (experimental observables) $\rightarrow$ (Hamiltonian parameters, temperature) exists in this case, which also implies that the experimental observables can be used to detect entanglement². This chapter contains the results of our investigation into the possibility of quantum state tomography from two-point correlators and neutron-scattering functions³.

In the first section, we will present the background for the problem of quantum state tomography and cite the results of the two theorems that came from the preceding article (which is not an original work for this thesis as all the work was done by the supervisor of this thesis). In the following section, we will show a proof that those theorems also extend to the Hamiltonians with externally applied field, and a separate proof for finite-temperature states. Then we explain the methodology for obtaining numerical results for the fitness landscape of neutron scattering for state deviations. The presentation and discussion of those results are left at the end of this chapter.

6.1 Background

The large number of degrees of freedom in condensed matter systems can make obtaining detailed microscopic insights from experimental data challenging. Usually a model Hamiltonian $\hat{H}$ is used to predict the state of the system in the form of a ground state wave function ψ_0 (or more generally a thermal density matrix ρ). Once the state is known, the expectation value of an observable $\hat{O}$ can be evaluated using a known ground state functional,

¹Which includes number of interesting magnetic systems. Some of which undergo an ET.

²Since entanglement is a function of the quantum state, which we can infer in full.

³The results are currently published as a preprint [3]

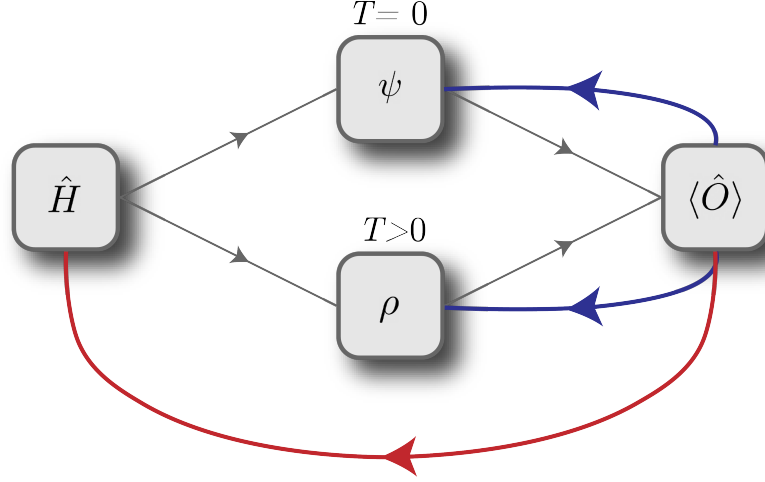


Figure 6.1: Quantum state tomography and Hamiltonian learning. The way to obtain observables from a Hamiltonian is straightforward. Ground state (or thermal density matrix for $T > 0$) is obtained by a suitable method, such as numerical ED. Then the expectation value of an observable $\langle \hat{O} \rangle$ is calculated from $\langle \psi | \hat{O} | \psi \rangle$ (or $\text{Tr} [\rho \hat{O}]$). The reverse function generally does not exist. The process of deducing a state from experimental observables is called quantum state tomography (blue arrows). We call obtaining initial Hamiltonian from observables Hamiltonian learning (red arrow).

$\langle \hat{O} \rangle = \langle \psi_0 | \hat{O} | \psi_0 \rangle$ (more generally $\langle \hat{O} \rangle = \text{Tr} \rho \hat{O}$). The comparison with the experiment is then used to refine the model $\hat{H}$. In general, however, the relationships $\hat{H} \rightarrow \langle \hat{O} \rangle$ and $\psi_0 \rightarrow \langle \hat{O} \rangle$ ($\rho \rightarrow \langle \hat{O} \rangle$) are not invertible: there is no guarantee of a unique model $\hat{H}$, or even a unique ground state (or density matrix) compatible with a given observation. Finding the reverse relationship $\langle \hat{O} \rangle \rightarrow \psi_0$ ($\langle \hat{O} \rangle \rightarrow \rho$) is usually called quantum state tomography (see Figure 6.1). There is no established term for inferring the Hamiltonian terms from the observables. For convenience, in this work we call it “Hamiltonian learning”.

One of the theoretical justifications for quantum tomography is provided by Hohenberg and Kohn’s theorem, stating that knowledge of the density distribution uniquely determines the ground state of the system [102], providing the latter is nondegenerate. Generalising this point of view, the prefactors of any unknown terms in the Hamiltonian are fully constrained if the corresponding ground state or thermal correlation functions are known [103]. Recently, one of the supervisors of this thesis has highlighted that this enables the learning of Hamiltonian parameters of systems with exclusively pairwise interactions of spins where the observable is the ground-state magnetic structure factor $S_{\alpha,\beta}(\mathbf{q})$ [101], as well as proving that the many-body wave function can also be inferred directly from the ground-state correlators. To cite the results in reference [101], it proves two theorems. These theorems state that for the general Heisenberg-like Hamiltonians of type:

$$H_J = \sum_{i,j} \sum_{\alpha,\beta} J_{i,j}^{\alpha,\beta} S_i^\alpha S_j^\beta, \quad (6.1)$$

and the connected magnetic two-point correlators at zero temperature:

$$\rho_{i,j}^{\alpha,\beta}[\Psi] = \langle \Psi | S_i^\alpha S_j^\beta | \Psi \rangle, \quad (6.2)$$

the following statements are true [101]:

Theorem 1 *Two Hamiltonians $\hat{H}, \hat{H}'$ cannot have the same ground-state correlator without sharing the corresponding ground states.*

Theorem 2 *Any N -qubit wave function Ψ that can reproduce the ground-state correlator of $\hat{H}$ represents a ground state of $\hat{H}$.*

Crucially, the magnetic structure factor⁴ is experimentally accessible via diffuse magnetic neutron scattering. This offers the prospect, *in principle*, of using diffuse magnetic scattering on a quantum magnet to completely determine its wave function or its Hamiltonian. Compared to other schemes aiming to determine states [104, 105] or Hamiltonians [106, 107, 108, 109, 110] from local measurements that have been discussed in the context of quantum simulators this has the advantage that it does not require the full covariance matrix [106, 107, 108, 109] and that all the necessary correlators are determined simultaneously in the neutron-scattering measurement so there is no scaling of experimental complexity [110] with the size of the system or the range of interactions.

The main purpose of the present work is to determine whether the above scheme could work *in practice*. In order to address this question we carry out a numerical examination of the fitness landscapes for quantum state tomography and Hamiltonian learning in the ground state and at finite temperature. We also generalise previous discussions [101, 111] to the case where there are applied magnetic fields and provide further evidence of the validity of the theorems at finite temperature.

6.2 Generalisations

6.2.1 Systems at finite temperature

Although the extension of Theorem 1 and a weaker version of Theorem 2 were proven in a recent article [111], we have also independently established the generalisation for Theorem 1. Similarly to the proof of the corresponding theorem for zero-temperature systems, we proceed by *reductio ad absurdum*. Let us suppose that there are two distinct Hamiltonians H_A, H_B of type (6.1), but all of the resulting correlation functions $C_{i,j}^{\alpha,\beta}$ are identical, with

$$\begin{aligned} C_{i,j}^{[A];\alpha,\beta} &= \text{Tr} \left[\rho_A S_i^\alpha S_j^\beta \right], \\ C_{i,j}^{[B];\alpha,\beta} &= \text{Tr} \left[\rho_B S_i^\alpha S_j^\beta \right], \end{aligned} \quad (6.3)$$

$$C_{i,j}^{[A];\alpha,\beta} = C_{i,j}^{[B];\alpha,\beta} \quad \text{for all } i, j, \alpha, \beta, \quad (6.4)$$

where ρ_A, ρ_B are thermal density matrices of systems A and B respectively:

$$\begin{aligned} \rho_A &= \frac{e^{-\beta H_A}}{\text{Tr} [e^{-\beta H_A}]} \\ \rho_B &= \frac{e^{-\beta H_B}}{\text{Tr} [e^{-\beta H_B}]}. \end{aligned} \quad (6.5)$$

⁴Although the theorems are statements about the relationship between Hamiltonians and two-point correlators, the relationship between Hamiltonians and structure factor is implied, since the latter is a Fourier transform of two-point correlators [101]

⁵Here we replaced symbol for correlation functions from $\rho_{i,j}^{\alpha,\beta}$ to $C_{i,j}^{\alpha,\beta}$ to avoid confusion with thermal density matrices $\rho(T)$.

Let us consider the Helmholtz free energy. If the quantum system is connected to the thermal bath, the free energy has a form [III]

$$F_0 = \langle H \rangle_\rho - TS_\rho = \text{Tr}[\rho H] + \frac{1}{\beta} \text{Tr}[\rho \ln \rho]. \quad (6.6)$$

The Gibbs-Bogoliubov inequality [II2] states that the free energy is minimised by the thermal density matrix coming from H at a given temperature. Meaning that

$$F_0 \leq F[\rho'] = \text{Tr}[\rho' H] + \frac{1}{\beta} \text{Tr}[\rho' \ln \rho']. \quad (6.7)$$

The equality in the above expression holds only for the case of infinite temperature $\beta \rightarrow 0$ or when $\rho = \rho'$. The first condition can be dismissed, since we are not interested in systems at infinite temperature. The second condition can only happen in our example if the thermal density matrices coming from the two distinct Hamiltonians are the same. However, this is not possible, because of the form of the thermal matrix (6.5).

Using the Gibbs-Bogoliubov property, we can write a hard inequality for the free energy coming from system A, and juxtapose it against the state ρ_B :

$$\begin{aligned} F_0^{[A]} &= \text{Tr}[\rho_A H_A] + \frac{1}{\beta} \text{Tr}[\rho_A \ln \rho_A] < \text{Tr}[\rho_B H_A] + \frac{1}{\beta} \text{Tr}[\rho_B \ln \rho_B] \\ &= \text{Tr}[\rho_B H_A] - \text{Tr}[\rho_B H_B] + \text{Tr}[\rho_B H_B] + \frac{1}{\beta} \text{Tr}[\rho_B \ln \rho_B] \\ &= \text{Tr}[\rho_B H_A] - \text{Tr}[\rho_B H_B] + F_0^{[B]}. \end{aligned} \quad (6.8)$$

On the other hand, if we perform the same calculations coming from the free energy of system B we get:

$$\begin{aligned} F_0^{[B]} &= \text{Tr}[\rho_B H_B] + \frac{1}{\beta} \text{Tr}[\rho_B \ln \rho_B] < \text{Tr}[\rho_A H_B] + \frac{1}{\beta} \text{Tr}[\rho_A \ln \rho_A] \\ &= \text{Tr}[\rho_A H_B] - \text{Tr}[\rho_A H_A] + \text{Tr}[\rho_A H_A] + \frac{1}{\beta} \text{Tr}[\rho_A \ln \rho_A] \\ &= \text{Tr}[\rho_A H_B] - \text{Tr}[\rho_A H_A] + F_0^{[A]}. \end{aligned} \quad (6.9)$$

Adding those two inequalities yields

$$F_0^{[A]} + F_0^{[B]} < F_0^{[A]} + F_0^{[B]} + \text{Tr}[\rho_B H_A] + \text{Tr}[\rho_A H_B] - \text{Tr}[\rho_A H_A] - \text{Tr}[\rho_B H_B]. \quad (6.10)$$

Because of the assumed form of the Hamiltonian, the traces in the inequality can be changed to weighted averages of correlation functions. For example, the $\text{Tr}[\rho_A H_A]$:

$$\begin{aligned} \text{Tr}[\rho_A H_A] &= \text{Tr} \left[\rho_A \sum_{i,j} \sum_{\alpha,\beta} J_{i,j}^{[A];\alpha,\beta} S_i^\alpha S_j^\beta \right] = \sum_{i,j} \sum_{\alpha,\beta} J_{i,j}^{[A];\alpha,\beta} \text{Tr}[\rho_A S_i^\alpha S_j^\beta] \\ &= \sum_{i,j} \sum_{\alpha,\beta} J_{i,j}^{[A];\alpha,\beta} C_{i,j}^{[A];\alpha,\beta}. \end{aligned} \quad (6.11)$$

Inserting these into inequality (6.10) and removing the same free energies from the left and the right side gives

$$\begin{aligned}
0 &< \sum_{i,j} \sum_{\alpha,\beta} \left(J_{i,j}^{[A];\alpha,\beta} C_{i,j}^{[B];\alpha,\beta} + J_{i,j}^{[B];\alpha,\beta} C_{i,j}^{[A];\alpha,\beta} - J_{i,j}^{[A];\alpha,\beta} C_{i,j}^{[A];\alpha,\beta} - J_{i,j}^{[B];\alpha,\beta} C_{i,j}^{[B];\alpha,\beta} \right) \\
&= \sum_{i,j} \sum_{\alpha,\beta} \left(J_{i,j}^{[A];\alpha,\beta} - J_{i,j}^{[B];\alpha,\beta} \right) \left(C_{i,j}^{[B];\alpha,\beta} - C_{i,j}^{[A];\alpha,\beta} \right).
\end{aligned} \tag{6.12}$$

But the second term $\left(C_{i,j}^{[B];\alpha,\beta} - C_{i,j}^{[A];\alpha,\beta} \right)$ must be equal to zero for all i, j, α, β , since the equivalence of two-point correlators was our assumption. Hence we get a contradiction:

$$0 < 0 \quad \text{Q.E.D.} \tag{6.13}$$

The extension of Theorem 1 to systems at finite temperatures is even slightly stronger than the original claim, since there is no additional restriction on the similarity of the ground states of two Hamiltonians. This is due to the fact that, by definition, thermal density matrices contain information about the ground state, as well as all of the excited states of the Hamiltonian, and therefore are uniquely defined by it.

6.2.2 Systems with linear spin terms

The general form of the Hamiltonian H_J in Eq. (6.1) encompasses many different magnetic systems including the Ising model [113], XXZ model [114], Kitaev model [115]), and many models with long-range interactions. However, the lack of linear spin terms excludes the effects of external (or local) magnetic fields. A more general model would take the form:

$$H_{J+h} = \sum_{i,j} \sum_{\alpha,\beta} J_{i,j}^{\alpha,\beta} S_i^\alpha S_j^\beta + \sum_i \sum_\alpha h_i^\alpha S_i^\alpha, \tag{6.14}$$

As we anticipated above, we will now show that the two classes are equivalent, provided that we allow the $J_{i,j}^{\alpha,\beta}$ coefficients in (6.1) to be complex⁶ We will do so by deriving an explicit transformation between H_J and H_{J+h} .

Our starting point are the commutation relations for the three components of the spin at a given site j : $\left[S_j^\alpha S_j^\beta \right] = i\hbar \sum_{\gamma=x,y,z} \epsilon_{\alpha\beta\gamma} S_j^\gamma$, where $\epsilon_{\alpha\beta\gamma}$ is the Levi-Civita symbol. Taking $\alpha\beta\gamma = yzx, zxy$ and xyz , the right-hand side takes the form of the constant $i\hbar$ multiplied by S_j^x, S_j^y , and S_j^z , respectively. We can use these relations to recast the spin operators in the second term on the right-hand side of (6.14) as sums of bilinears, which can be absorbed into the pre-existing bilinear terms. The result reads

$$H_{J+h} = \sum_{i,j} \sum_{\alpha,\beta} \bar{J}_{i,j}^{\alpha,\beta} S_i^\alpha S_j^\beta, \tag{6.15}$$

where we have introduced the new, modified exchange constants

$$\bar{J}_{i,j}^{\alpha,\beta} \equiv J_{i,j}^{\alpha,\beta} + i\delta_{i,j} \epsilon_{\alpha\beta\gamma} \hbar^{-1} h_j^\gamma. \tag{6.16}$$

Here $\alpha, \beta = x, y, z$ and $\alpha \neq \beta$, while γ takes the value that makes $\alpha\beta\gamma$ a permutation of xyz . Importantly, the form of complex coefficients (6.16) preserves hermiticity of the Hamiltonian H_{J+h} , which is to be expected, since it maps to the real Hamiltonian 6.14.

⁶This is fine, because proofs of Theorem 1, Theorem 2 and the generalisation to thermal densities do not rely on Hamiltonian coefficients being real.

6.3 General framework

The original work [101] and the generalisations described above motivated us to examine whether these results can be used *in practice* to determine the quantum state of Hamiltonian of the form (6.14) from experimentally available observables. The results presented here are limited to probing the fitness landscape for state tomography and not Hamiltonian learning. However, it would be the next logical step to investigate Hamiltonian learning as well. For quantum state tomography, we chose neutron-scattering functions [116] as target observables, since they are directly connected to two-point correlators and are routinely measured in neutron-scattering experiments. Below is a brief introduction to the neutron-scattering experiment, and the explanation of our fitness landscape searching method.

6.3.1 Neutron scattering functions

Diffuse neutron scattering is a high-energy experimental technique, implementing beam of neutrons to probe the internal structure of a solid-state material [116]. Since neutrons are charge neutral but have an intrinsic spin, their interactions with magnetic materials are mostly influenced by the electronic spin of molecules within those materials. The distribution of scattering of the beam of neutrons will therefore contain information about the magnetic structure of the probe. A full explanation of the experimental setup and the calculations of the scattering cross section can be found in [117, 118]. For us, the importance of neutron scattering is that the experimental results can be connected to two-point spin correlators of the materials wave function through integrated scattering function $\mathcal{S}(\mathbf{q})$ defined as⁷

$$\mathcal{S}(\mathbf{q}) = \sum_{\alpha, \beta} (\delta_{\alpha, \beta} - \hat{\mathbf{q}}_{\alpha} \hat{\mathbf{q}}_{\beta}) \mathcal{S}_{\alpha, \beta}(\mathbf{q}), \quad (6.17)$$

where $\mathbf{q}$ is a vector difference between incoming and scattered momentum of incident neutrons [119], α and β go through all the directions $\alpha, \beta = x, y, z$. The term $\delta_{\alpha, \beta} - \hat{\mathbf{q}}_{\alpha} \hat{\mathbf{q}}_{\beta}$ in (6.17) signifies that the only contribution to the integrated scattering function comes from the magnetic moments of the ions perpendicular to the change of momentum $\mathbf{q}$ [119]. $\mathcal{S}_{\alpha, \beta}(\mathbf{q})$ is the orientation resolved scattering function, connected to two-point spin correlation function by:

$$\mathcal{S}_{\alpha, \beta}(\mathbf{q}) = \frac{1}{L\hbar} \sum_{i, j} e^{i\mathbf{q}(\mathbf{R}_i - \mathbf{R}_j)} \times \sum_{\mu, \nu} \Lambda_{\alpha, \mu}(\phi_i) \Lambda_{\beta, \nu}(\phi_j) \langle S_i^{\mu} S_j^{\nu} \rangle, \quad (6.18)$$

where L is the number of spin sites, $\mathbf{R}_i$ and $\mathbf{R}_j$ are the positions of i -th and j -th sites, $\langle S_i^{\mu} S_j^{\nu} \rangle$ are the two-point magnetic correlation functions. The Lambda matrices serve as a transformation from a local axis of each individual spin site to the global axis through

$$\Lambda(\phi) = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (6.19)$$

where ϕ is the angle between local and experimental axes. The expression (6.18) works only for a setup in which we restricted spin sites to the x-y plane. In our investigation, we wanted

⁷Usually, neutron-scattering function refers to the time-frequency dependent function $\mathcal{S}(\mathbf{q}, \omega)$ [119], but since we are interested to the connection for static two-point correlators, we choose the integrated version $\mathcal{S}(\mathbf{q}) = \int d\omega \mathcal{S}(\mathbf{q}, \omega)$.

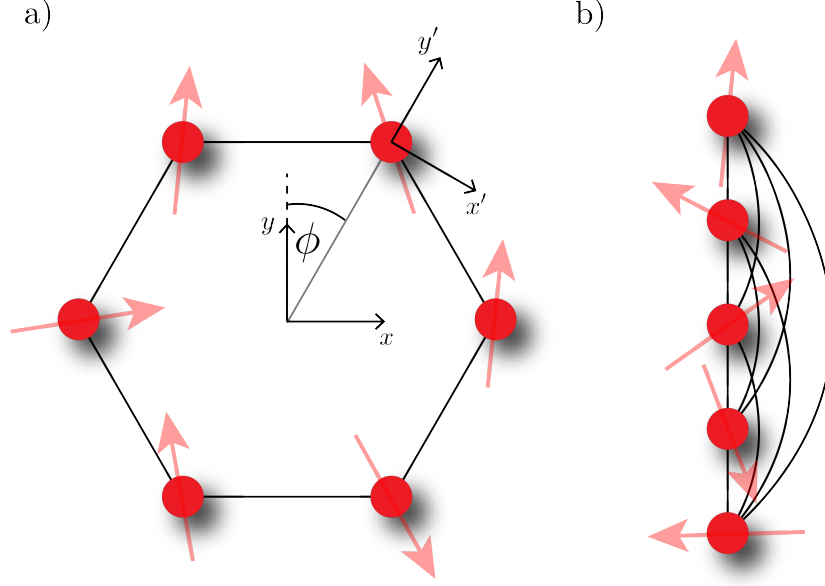


Figure 6.2: Geometries of magnetic systems used in simulating scattering functions: a) Magnetic molecule with a ring geometry. To preserve the symmetry of local coordinates for all spin sites, the transformation from local $\{x', y'\}$ to global $\{x, y\}$ coordinates has to be performed (according to the equation (6.18)), with the ϕ angle given by the deviation from global coordinates; b) Linear molecule with long range interactions. The target Hamiltonian describing that system in our research had non-decaying interactions for every pair of spin sites, technically making it a non-dimensional system. We still refer to it as a linear geometry, due to the fact that every local coordinate system is the same as the global one, which affects the calculated scattering functions.

to focus on simulating data from molecular magnets [120]. They can be described as spin chains on a ring (see Figure 6.2), which satisfy that condition (geometry in the x-y plane). In that configuration, the position of spin sites are dependent on the angle ϕ_j as follows

$$\mathbf{R}_j = R (\sin \phi_j, \cos \phi_j), \quad (6.20)$$

where R is the scaling factor.

6.3.2 Fitness landscape of scattering functions

Testing the fitness landscape of scattering functions in a space of wave functions requires a sampling process of variations from the initial wave function. Below is a detailed explanation of the procedure that we have chosen. For the sake of clarity, we explain the procedure for $T = 0$ case. The finite temperature case follows similar steps, and, if relevant, the differences will be mentioned when we discuss $T > 0$ results.

1. A reference Hamiltonian H is selected that describes a chain of L sites of $S = \frac{1}{2}$ spins. For the most part, we will assume the geometry shown in Figure 6.2: L sites forming a planar ring molecule with C_L rotational symmetry (and hence Hamiltonian parameters invariant under translations of the site indices i, j). We will also consider, when noted, the case of a linear molecule with open boundary conditions. The target ground state ψ can be obtained through exact diagonalization (ED). Once the state is

known, the two-point spin correlation functions $\langle S_i^\alpha S_j^\beta \rangle$ can be obtained, and from them the instantaneous magnetic structure factor $\mathcal{S}_{\alpha,\beta}(\mathbf{q})$ can be calculated. Following that, the integrated scattering function $\mathcal{S}(\mathbf{q})$ can be obtained, according to the equation (6.17)

2. We generate random states ψ' that are different from the ground state ψ of H . This can be achieved by adding a random variation to the ground state $\psi' = \psi + \delta\psi$, and normalize the newly created state ψ' . If we allow $\delta\psi$ to be random without any constraints, the state can either be a nonphysical one or coming from a Hamiltonian that is not of the Heisenberg-like form (6.14).
3. We can then define a distance $\Delta\mathcal{S}$ between two scattering functions $\mathcal{S}, \mathcal{S}'$ and compare it to the distances between the ground state ψ of our reference Hamiltonian and the random state ψ' .

To quantify the distance between scattering functions we use the root mean square deviation

$$\Delta\mathcal{S} = \frac{1}{\mathcal{N}} \left\{ \int d\mathbf{q} [\mathcal{S}(\mathbf{q}) - \mathcal{S}'(\mathbf{q})]^2 \right\}^{1/2}. \quad (6.21)$$

In order to make the absolute values of $\Delta\mathcal{S}$ meaningful, a judicious choice of the normalisation factor $\mathcal{N}$ is necessary. We take

$$\mathcal{N} = \left\{ \int d\mathbf{q} [\mathcal{S}(\mathbf{q}) - \bar{\mathcal{S}}]^2 \right\}^{1/2} \quad (6.22)$$

where $\bar{\mathcal{S}}$ is the average of $\mathcal{S}(\mathbf{q})$ over the momentum transfer $\mathbf{q}$ ⁸. Thus, a value of $\Delta\mathcal{S} \sim 1$ indicates that the differences between the target scattering function $\mathcal{S}(\mathbf{q})$ and the trial function $\mathcal{S}'(\mathbf{q})$ are appreciable on the scale of the features of the target function itself. In this case, the precision required in the experimental measurement of $\mathcal{S}(\mathbf{q})$ would not be greater than in a normal neutron-scattering experiment. A similar level of precision would be needed for $\mathcal{S}'(\mathbf{q})$ (for instance, using a noisy quantum computer). As it turns out, with this choice of normalisation we obtain a remarkable independence of our main result from the system size L .

In the state tomography problem, the distance $\Delta\mathcal{S}$ between scattering functions needs to be compared to the distance between the ground state ψ and the target state ψ' (or between density matrices ρ and ρ'). For pure states it is straightforward to define a metric for states as

$$\text{dist}(\psi, \psi') = 1 - |\langle \psi | \psi' \rangle|^2. \quad (6.23)$$

6.4 State tomography

Having established the method for probing the fitness landscape of scattering functions, let us first start by investigating how different functions look when generated from randomly sampled ground state wave functions ($T = 0$). The sampling process was as follows:

⁸We note that the quantity $\mathcal{N}$ is defined using the target scattering function $\mathcal{S}(\mathbf{q})$. If the domain of integration covers all of the $\mathbf{q}$ -space, the exact sum rule $\mathcal{S} = s(s+1)$ holds [116] and our expressions simplify. This value is the same regardless of whether $\mathcal{S}(\mathbf{q})$ or $\mathcal{S}'(\mathbf{q})$ is used to evaluate the RHS of Eq. (6.22) ensuring that the value of $\Delta\mathcal{S}$ is independent of the order in which we write the scattering functions. However, in practice, we often have experimental information over a restricted domain given, for instance, by the detector coverage of the neutron-scattering instrument.

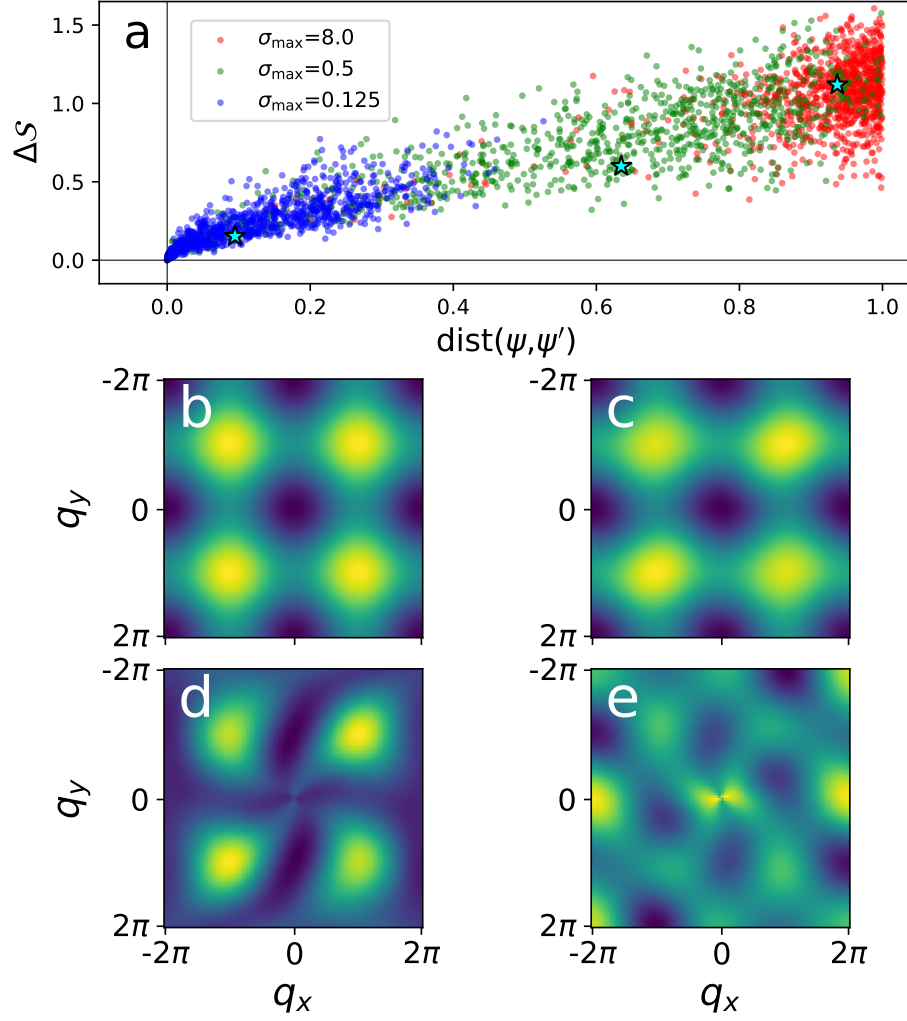


Figure 6.3: Fitness landscape for quantum state tomography using the ground-state, diffuse magnetic neutron-scattering function. The target state ψ is the ground state of a spin-1/2 ring tetramer with nearest-neighbour, AFM Heisenberg interactions. This corresponds to the zero-field Hamiltonian (6.1) with all coupling constants equal to zero except $J_{j,j+1}^{\alpha,\alpha} = 1$ for all $j = 1, 2, 3, 4$ and $\alpha = x, y, z$. (a) Cost function ΔS [Eq. (6.21)] as a function of the distance $\text{dist}(\psi, \psi')$ between the target ground state and a trial state vector ψ' [Eq. (6.23)]. We used three sets of 1000 random state vectors each with amplitudes from drawn the probability distribution centred on ψ described in the main text. Each individual state vector is shown by a filled circle. Each set had a different value of the parameter $\sigma_{\max}$, which gives the upper limit to the standard deviation sampled from uniform distribution (see explanation in section 6.4) (b) The target diffuse magnetic scattering function $S(\mathbf{q})$ obtained by ED. (c-e) Three scattering functions obtained from the three random state vectors highlighted with stars on panel (a), with c corresponding to the leftmost (least deviation) and e to the rightmost star (most deviation).

1. Draw real parts and imaginary parts for the components of R_j according to the distribution $p[\Re(R_j)] = p[\Im(R_j)] = N(\sigma, 0)[x]$, for $j = 1, \dots, \dim(\mathcal{H})$, with $N(\sigma, \mu)$ the normal distribution⁹ of standard deviation σ and mean μ , and we denote the Hilbert space dimension of the problem as $\dim(\mathcal{H})$.
2. Construct the unnormalised perturbed state with components $\tilde{\psi}'_j = \psi_j + R_j$.
3. Normalize the resulting wave function $\psi'_j = \mathcal{N} \tilde{\psi}'_j$, choosing $\mathcal{N}$ to enforce $\sum_j \psi'_j = 1$, yielding the components of the new ket $|\psi'\rangle$.

Normalisation allows for the distance metric (6.23) to be a well-defined measure. Since we have noticed that the normal distribution with a fixed value of σ tends to generate states that are tightly-grouped around certain values of $\text{dist}(\psi, \psi')$, we have opted to randomly generate the standard deviation for each sample, from a uniform distribution $\sigma = [0, \sigma_{\max}]$. The maximum value $\sigma_{\max}$ is set so that the samples cover most of the metric values.¹⁰

Figure 6.3 shows the numerical results for the reference ground state of the spin-1/2 anti-ferromagnetic (AFM) Heisenberg tetramer ($L = 4$) with nearest-neighbour interactions of fixed strength J . All of the dots in (a) represent different random samples, generated as dictated above. Three different values of $\sigma_{\max}$ were used for comparison. Since the $\text{dist}(\psi, \psi')$ measure can only have values from 0 to 1, we can be reasonably sure that this sampling covers a wide range of states. To investigate that further, we plotted three different scattering functions directly (c-e), and compared them to the reference one (b). The differences in them are clearly appreciable by eye, even for the state that is closer to the target. Note that the latter scattering function is about 15% from the target, while the wave function overlap is around the 5% level. This suggests that it is possible to determine the system's ground state with considerable precision using realistically noisy neutron-scattering data. We also note that the candidate function showing the closer match has, to a very good approximation, the same four-fold symmetry as the target function, in contrast to the other two candidates, which look qualitatively different. This is a trend we see in all our simulations. This again has practical implications: on the one hand, when symmetry constraints or other qualitative features of the data are known, including them can accelerate the search for the target state and improve its accuracy; but, on the other hand, our results show that an unbiased search through the Hilbert space may yield states with the right features even if these are not known *a priori* and the experimental and target scattering functions can only be determined with limited accuracy.

Figure 6.4 (a) shows the fitness landscape for different values of $L = 4, 6, 8, 10, 12, 14$, with the same AFM model as described above (ring geometry). As L increases, the data appear to collapse onto a single curve. Remarkably, a simple linear relationship with coefficient equal to unity appears to describe this curve very well:

$$\Delta\mathcal{S} = \text{dist}(\psi, \psi'). \quad (6.24)$$

Our results indicate that trial states ψ' approaching the target ψ are more likely to obey the remarkably simple law (6.24). This indicates an easy-to-navigate fitness landscape: once we are in the vicinity of the target scattering function, we just need to follow the slope of this curve to hit the target state. This is a remarkable result and constitutes our main finding. The

⁹For the sake of completeness, we also tried uniform distribution, but since the results are for our purpose indistinguishable, we are showing just the results obtained from the normal distribution.

¹⁰In the cases where we want to compare data from multiple system sizes, the scaling $\sigma_{\max}(L) = \sigma_{\max} \times 2^{-\frac{L}{2}}$ was used.

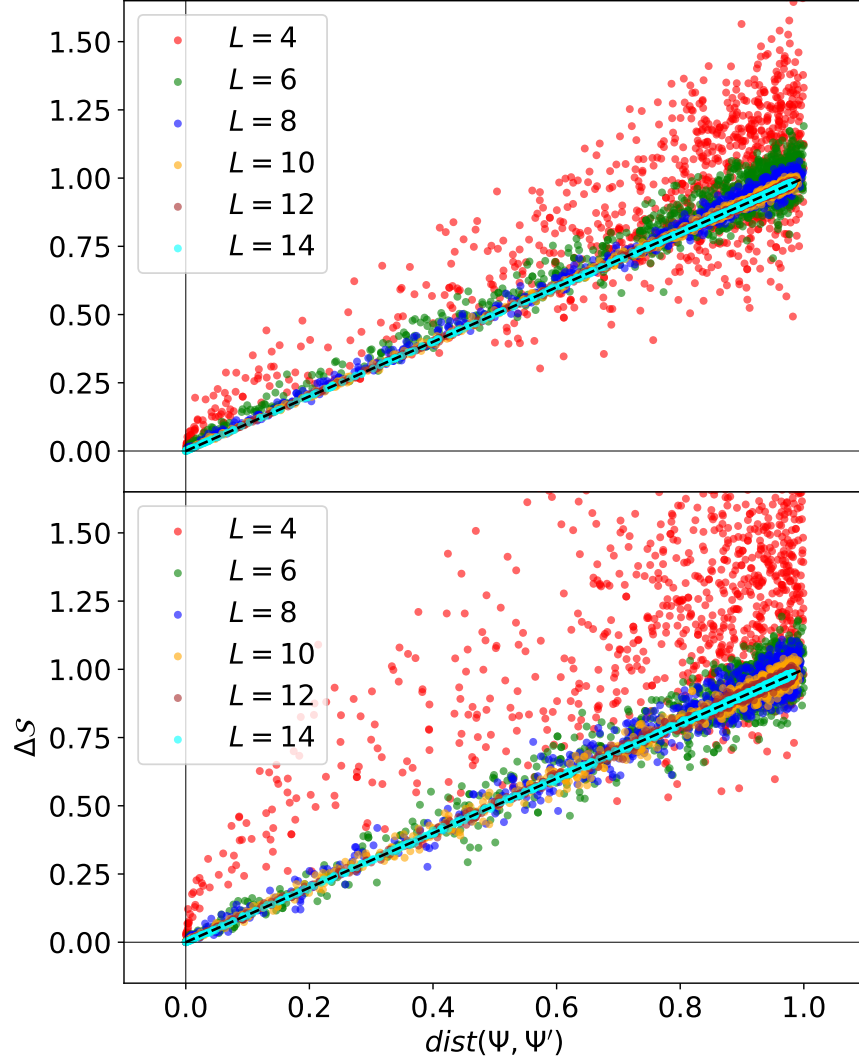


Figure 6.4: Dependence of the fitness landscapes on the number of spins L in the cluster. For each L , 1000 target states were sampled with normal-distributed perturbations around the target wave function (see main text). Target states were chosen as ground states of: (a) A spin-1/2, nearest-neighbour anti-ferromagnetic Heisenberg model in a ring geometry, i.e., the same model as in Figure 6.3, but with varying L . (b) A random long-range spin-1/2 Hamiltonian in a linear open chain geometry, given by Equation (6.14) with random exchange constants $J_{i,j}^{\alpha,\beta}$ and a random global magnetic field with components $h_i^\alpha = h^\alpha$. For each L , Hamiltonian parameters $J_{i,j}^{\alpha,\beta}$ and h^α were drawn from a normal distribution with zero average and standard deviation equal to J (the nearest-neighbour coupling of the Heisenberg Hamiltonian used in panel (a)). Dashed lines in (a,b) show the linear relation in Equation (6.24).

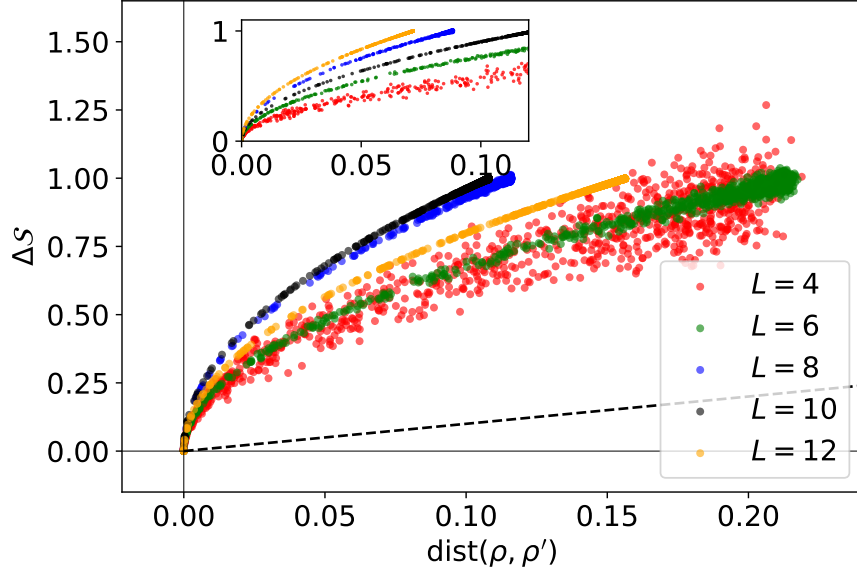


Figure 6.5: Dependence on L (see legend) of the fitness landscape for finite-temperature quantum tomography of a random Hamiltonian. The Hamiltonian was obtained in the same way as in Figure 6.4 (b). The target density matrix ρ is the thermal equilibrium density matrix of that Hamiltonian at temperature $T = J$. 1000 trial states (random density matrices) were tested against this target state. The distance between random density matrices and the target is measured on the horizontal axis and defined by equation (6.25). For $T = 0$ this would have coincided with the distance $\text{dist}(\psi, \psi')$ between ground states. The dashed line represents equation (6.24), for comparison. For the calculation of the scattering functions the atoms were assumed to be on a straight line. The inset shows similar output with different random targets at each size.

fact that the “empirical” law (6.24) is independent of L encourages us to conjecture that the relationship will hold even in the thermodynamic limit.

In the above discussion, we used the AFM Heisenberg model with nearest-neighbour interactions to generate our target ground state. However, our conclusions are robust to variations in the model and geometry. To illustrate this, Figure 6.4 (b) shows results for a linear molecule with randomly-chosen coupling constants and an externally-applied field of arbitrary strength and direction as a reference ground state. In addition to the different geometry, such model lacks $\text{SU}(2)$ and time-reversal symmetries. Moreover, the couplings are defined on a fully-connected graph with no natural length scale. Thus, as L grows, the model can no longer be considered 1-D. Finally, we note that such random Hamiltonians typically have competing ferromagnetic and AFM couplings. In spite of all these differences, we obtain a very similar behaviour. These results show that the arguments in the article [101] can be extended to the case where there is an applied magnetic field. Moreover, they suggest that the relationship (6.24) describing the fitness landscape for quantum tomography in the vicinity of the target state applies universally.

6.5 Finite temperature

In the previous section, we showed a fitness landscape of scattering functions at $T = 0$. As we explained earlier, the extension of Theorem 1 to $T > 0$ can be shown to be true. To test

whether the finite-temperature fitness landscape in the space of thermal states behaves well, we sampled random, positive definite density matrices in the vicinity of the thermal equilibrium density matrix of a given target Hamiltonian with normal-distributed long-range couplings. The procedure of scanning the fitness landscape is similar as for the ground state case (see Section 6.3.2), with the obvious difference that rather than adding a random deviation to ground state ψ , we now perturb the full density matrix $\rho(T)$. We define the distance between two different density matrices as

$$\text{dist}(\rho, \rho') = \frac{1}{2} \text{Tr} \left[(\rho - \rho')^\dagger (\rho - \rho') \right], \quad (6.25)$$

which reduces to (6.23) when $\rho = |\psi\rangle\langle\psi|$ and $\rho' = |\psi'\rangle\langle\psi'|$.

Our sampling approach for the finite temperature case was as follows:

1. Draw real and imaginary parts for the matrix $\mathbf{R}$ of the same size as the initial thermal density matrix $\rho(T)$, according to the normal distribution $p[\Re(R_{jk})] = p[\Im(R_{jk})] = N(\sigma, 0)[x]$, for $j, k = 1, \dots, \dim(\mathcal{H})$.
2. Ensure that the density matrix is positive-definite by multiplying the deviation by its hermitian conjugate: $\mathbf{R}^\dagger \mathbf{R}$.
3. The deviation is added to the initial density matrix and normalised accordingly:

$$\rho' = \mathcal{N}(\rho + \mathbf{R}^\dagger \mathbf{R}), \quad (6.26)$$

so that the $\text{Tr}[\rho'] = 1$.

Since the deviation generates even more randomness than in the ground state case, we expected that the system size scaling factor should be of the order $\propto (\dim H(L))^{-\frac{1}{4}}$. It turned out that in order to correctly compare the system scaling, an additional prefactor was needed. Therefore, for each sample, the standard deviation for the normal distribution was drawn from a uniform distribution $\sigma \in [0, c_L \sigma_{\max} \times 2^{-\frac{L}{4}}]$, with the c_L factor chosen empirically.

The resulting data is shown in Figure 6.5, numerically supporting the extension of Theorem 1 to finite temperature systems. For the generation of the data, we used the empirical scaling: $c_4 = c_6 = c_8 = 0.3$, $c_{10} = 0.1$ and $c_{12} = 0.03$. The most salient feature, compared to Figures 6.4 (a,b), is the much steeper (sub-linear) increase in $\Delta\mathcal{S}$ with increasing inter-state distance (6.25). This is due to the fact that the space of trial density matrices ρ' is of higher dimensionality than that of trial pure states ψ' . Indeed, even if we choose the target state to be arbitrarily close to the ground state (by choosing the target temperature to be very low, compared to all the Hamiltonian parameters), we still find this markedly sub-linear behaviour as the ρ' matrices almost never represent pure states like those considered earlier.

6.6 Discussion and Conclusions

This chapter contains all the evidence we have gathered for the prospect of quantum state tomography in magnetic systems based on diffuse magnetic neutron scattering. Our results allow us to assess how viable those kinds of experiments would be in practice. We have also included magnetic models with applied or internal fields, making the analysis more relevant to real experimental conditions. Our results for the fitness landscape indicate that the cost functions for quantum tomography are well behaved, both in the ground state and at finite temperature, with no evidence of barren plateaus or local minima.

The linear dependency of scattering functions in the state space with simple size scaling is a surprising result, although more calculations are needed to test its universality. We have not yet developed a theoretical justification for it. With our particular choice of normalisation, we set the slope of this relationship to be one and ensure collapse across all system sizes. However, independently of the normalisation, data for each system size tend to cluster more sharply around a linear relationship as the system size grows. Additionally, the normalisation factor is only a function of the target state, not the randomly generated one. The number of terms in the sum in the definition of $S(\mathbf{q})$ grows quadratically with size L and that may explain the different typical scale of deviation from the average that is the basis of the normalization factor (see Equation (6.22)). In any case, the result warrants further investigation.

The case of finite temperature fitness landscape also deserves special attention. The fact that the connection between distances and scattering function deviations is sub-linear may seem surprising at first, but it could be connected to the contributions from all of the spectrum to the cost function, not just the ground state. Furthermore, the proofs of the one-to-one correspondences between Hamiltonians and correlators (Theorem 1) and between wave functions and correlators (Theorem 2) are based on the Rayleigh-Ritz variational principle [121], which is used by many numerical techniques, such as density functional theory [122], Variational Monte Carlo methods [10] or recent artificial intelligence methods [11]. The principle states that given a Hamiltonian H , the ground state will minimise the energy globally, which means that $E(\tilde{\psi}) = \langle \tilde{\psi} | H | \tilde{\psi} \rangle / \langle \tilde{\psi} | \tilde{\psi} \rangle$ is a good optimization function for finding ground states. Their generalisations to systems at finite temperatures is the Gibbs-Bogoliubov inequality 6.7. The free energy is minimised for the thermal density matrix $\tilde{\rho} = e^{-\beta H} / \text{Tr} [e^{-\beta H}]$. It turns out that the entropy term does not alter the reasoning of the proof of Theorem 1, as shown by Murta and Fernández-Rossier [111], and is repeated here in Section 6.2.1. However, as noted in that reference, that does not carry over to Theorem 2: one cannot discard the possibility that there are density matrices with lower entropy than the real one which reproduce the experimental data. However, the systematic numerical exploration presented here did not provide any examples of this, giving strong indication that such cases, if they exist, must be special, at least for finite-size systems.

Our results suggest that the variational approach to quantum magnets proposed in reference [101], where an efficient representation of the wave function of a magnet is optimized to reproduce experimentally obtained neutron-scattering data, may be feasible, at least for magnets formed of finite-size clusters. Additional studies will be necessary to ascertain whether this conclusion carries over to fully connected systems as well as the effect of noisy and incomplete data sets. Effective ways to represent the wave function, for instance, using neural networks, tensor networks or quantum circuits (and the potential of the latter to achieve scaling advantages when implemented on noisy, intermediate-scale quantum hardware) will also need to be developed.

As was explained at the beginning of this chapter, the studies described here helped to choose the observables needed to obtain the information about the entanglement for the entanglement learning scheme described in the next chapter. Regardless, we think that these results stand on their own and are worth pursuing in further research.

Chapter 7

Entanglement learning

Having established the observables $\hat{O}$ that can serve as an input for machine learning (ML) to learn entanglement, we proceed to study how well convolutional neural networks (CNNs) can be trained to serve as a function that predicts some measure of entanglement $F(\langle\hat{O}\rangle)$. A sketch of the framework we have followed for this task can be seen in Figure 7.1. We first choose a spin chain Hamiltonian $\hat{H}$ with interesting entanglement properties, such as the XY model discussed in Chapter 2. Then we calculate the imaginary time evolution of a thermal density matrix, using tensor network (TN) methods described in Chapter 3. At each inverse temperature step we calculate observables and entanglement entropy. Finally, we use supervised ML with calculated observables as input and the entanglement as labels.

There are many potential approaches to train NN to learn the entanglement. The two main goals that are in the scope of this thesis are: a) To show whether the CNN ML algorithm can be used to fit the entanglement entropy as a function of observables easily. b) To see how well the CNN structure can predict the entanglement entropy of data not used in the training session, i.e. predicting other sets of temperature T or magnetic fields h within one model, or predicting data from other Hamiltonians. This chapter is a collection of results that serves to answer these questions.

The following text is divided into three sections. The first part focuses on a more detailed explanation of the spin $\frac{1}{2}$ chain Hamiltonians used to gather data, the input and cost functions for the ML algorithm, and the CNN architecture used to learn the entanglement. The second section shows the results of predicting the entanglement entropy from CNNs and describes a method to obtain insight about the learning process, which in turn can tell us something about the connection between entanglement and observables. It also shows the application of CNN as a classification tool, when two classes (different physical phases) are known. The final section is a summary and brief discussion of the results shown in the previous section.

7.1 General framework

7.1.1 Toy models

The results of the previous chapter suggested that the models in the class of Heisenberg-like magnetic Hamiltonians are uniquely determined by experimentally available observables (except for the zero-temperature case when two different Hamiltonians have the same ground states).

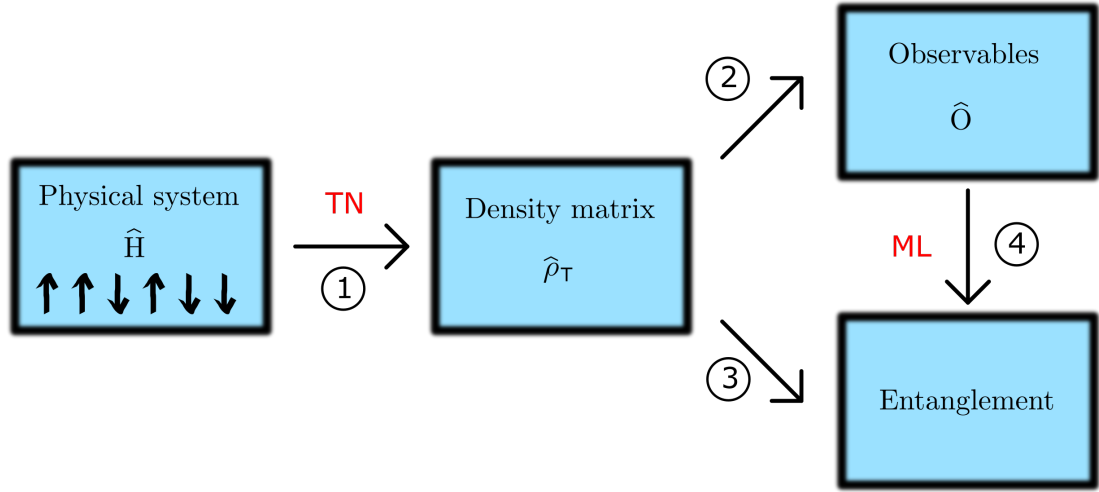


Figure 7.1: Framework for testing entanglement learning. Simulated thermal density matrix for a model Hamiltonian $\hat{H}$ can be obtained using TNs (1). From density matrix one can calculate both observables (2) and entanglement entropy (3), which can then be fed to ML algorithm in order to reproduce the entanglement (4).

Anisotropic XY model

The entanglement features of the 1-D spin- $\frac{1}{2}$ anisotropic XY model are explained in the final section of Chapter 2. To summarize, in the thermodynamic limit, the system has following properties:

- The Hamiltonian of the system have two parameters —anisotropy coefficient γ and external magnetic field h

$$H_{XY} = \sum_j (1 + \gamma) S_j^x S_{j+1}^x + (1 - \gamma) S_j^y S_{j+1}^y + h S_j^z. \quad (7.1)$$

- The system has a critical point independent of the anisotropy coefficient — $h = h_c = 1$, where the ground state changes from the antiferromagnetic (AFM) long-range order (LRO) ($h < 1$) to paramagnetic ($h > 1$) phase.
- If the anisotropy coefficient is nonzero, the ground state becomes completely factorized (classical) at $h = h_f = \sqrt{1 - \gamma^2}$. This entanglement transition (ET) always occurs within the ordered phase.

Anisotropic XYZ model

More generally, we can choose the spin chain anisotropic XYZ model, which is described by the Hamiltonian

$$H_{XYZ} = \sum_j (1 + \gamma) S_j^x S_{j+1}^x + (1 - \gamma) S_j^y S_{j+1}^y + \Delta S_j^z S_{j+1}^z + h S_j^z. \quad (7.2)$$

We choose this form of a general XYZ model because it follows from the previously introduced XY model (7.1)¹. The additional interaction term along the z axis changes the be-

¹This form can reproduce any XYZ model of the kind $H = \sum_j J_x S_j^x S_{j+1}^x + J_y S_j^y S_{j+1}^y + J_z S_j^z S_{j+1}^z + h S_j^z$, except the classical case: $J_x = 0, J_y = 0, J_z \neq 0$, which is not very interesting for exploring quantum entanglement.

haviour of the system. However, the phase of AFM LRO remains for small external fields, and quantum criticality is observed. Quantum critical point is not known for general XYZ system, but authors of the article [123] postulate the lower and upper bounds for it to be:

$$h_l = \sqrt{(1 + \Delta)^2 - \gamma^2}, \quad h_u = 1 + \Delta + |\gamma|, \quad h_l \leq h_c \leq h_u. \quad (7.3)$$

The factorisation field is known and it is equal to

$$h_f = \sqrt{(1 + \Delta)^2 - \gamma^2}. \quad (7.4)$$

Importantly, the lower bound for critical transition is the same as the factorisation field, which suggests that the classical point happens always within the AFM phase or at the critical point.

Entanglement transition and quantum critical point for finite size system

It is worth mentioning that although in the thermodynamic limit the transition from AFM to ferromagnetic dominance happens at the quantum critical point, for the finite size calculations performed here, the transition from AFM spin configuration to a ferromagnetic one happens at the factorisation field. For finite size system, the gap between the ground state and the first excited state closes at different values of field $h \leq h_f$. For the first $N - 1$ (N is the total number of ground state level crossings and is given by $N = \frac{L}{2}$ for system size L with even number of spins) level crossings, the ground state on either side retains predominantly AFM correlations. By contrast, for the last level crossing at the factorisation field, the ground state changes to become ferromagnetic at $h > h_f$. Therefore, the appearance of a ferromagnetic peak in the correlator can be seen as an indicator of the ET, but this is coincidental. As mentioned before, at the factorisation field, for a finite system, the ground state becomes two-fold degenerate, and therefore any mixture of those two states $|\psi\rangle = A|\psi_A\rangle + B|\psi_B\rangle$ is a ground state as well. It turns out that one can choose coefficients A and B , so that the resulting state $|\psi\rangle$ becomes completely factorised, and this only happens at the factorisation field, not the other ground state level crossings [120]. A detailed analysis of the finite-size XY anisotropic model, its ET and the way the system approaches quantum phase transition in the thermodynamic limit in finite-size magnetic molecules, can be found in [120].

Although we have been using finite-size calculations in this chapter, we have not been able to see the indicator of the ET in our data (see Figure 7.2). This is due to the fact that our calculations were carried out at finite temperature, which not only adds excited states to the thermal state, but also adds two ground states with equal weight, which almost certainly would not create a factorised state. It would be interesting to see how ML handles such factors as the ET (this is why XY and XYZ models were chosen for this study), but we decided to stick to our method of obtaining the data, and see if the ML can reconstruct entanglement entropy correctly in the first place.

7.1.2 Observables

Training CNNs in a supervised way requires both inputs and outputs. As discussed in previous chapters, the static two-point correlators $\langle S^\alpha S^\beta \rangle$ could serve as a good input for the algorithm to learn the entanglement. Chapter 3 explains how finite-temperature static correlators can be obtained using TNs with desired precision. For the “cooling process” we tested both the time-evolving block-decimation and matrix product operator (MPO) evolution algorithms. Since there was no apparent difference for the sizes of systems that we opted for,

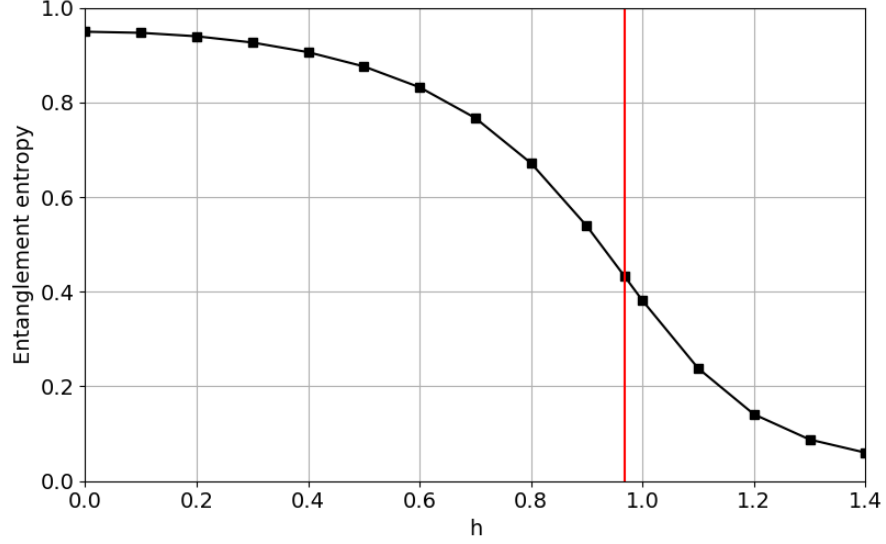


Figure 7.2: Entanglement entropy as a function of magnetic field, for the lowest achieved temperature $T = 0.1$, for XY model with anisotropy $\gamma = 0.25$. Red vertical line indicates the factorisation field. Clearly, there is no discontinuity at $h = h_f$.

the results presented here were obtained using MPO evolution (see Section 3.4.2). For each set of values of the Hamiltonian (either (γ, h) for the XY model or (γ, Δ, h) for the XYZ model), the infinite-temperature wave function $|\psi(T = \infty)\rangle$ was initialised and evolved to a desired value of $T = \frac{1}{\beta}$. At each step, both the correlation function and the entanglement measures (more about them in the next Section 7.1.4) were calculated. We have used TeNPy package [6] to perform the calculations related to both observables and the entanglement.

An example of static correlation functions for the XY model is shown in Figure 7.3. Correlators show clear long-range AFM correlations in dominating $S^x S^x$ interactions, weaker AFM order in $S^y S^y$, and a small ferromagnetic correlation in $S^z S^z$ for a weak magnetic field. The long range of the AFM order is also weakened by thermal fluctuations for systems at small β , but it does not affect the long range of $S^z S^z$, merely decreasing its overall value for higher magnetic fields. Close to critical field (displayed here at factorisation field $h = h_f \approx 0.968h_c^2$) spins start to align in the z direction at a strength around half of the maximal 0.25, eventually suppressing the AFM order in x direction completely. Notable are also the boundary effects on $S^z S^z$, which occur due to open boundary conditions, usual for TN calculations. The static correlators were calculated for a spin chain with a number of spins $L = 30$. All combinations of different orientations $S^\alpha S^\beta$, $\alpha, \beta = x, y, z$ were used in the calculations.

²Since all the data is gathered for finite temperature, we do not expect to find traces of fully factorized ground state for h_f .

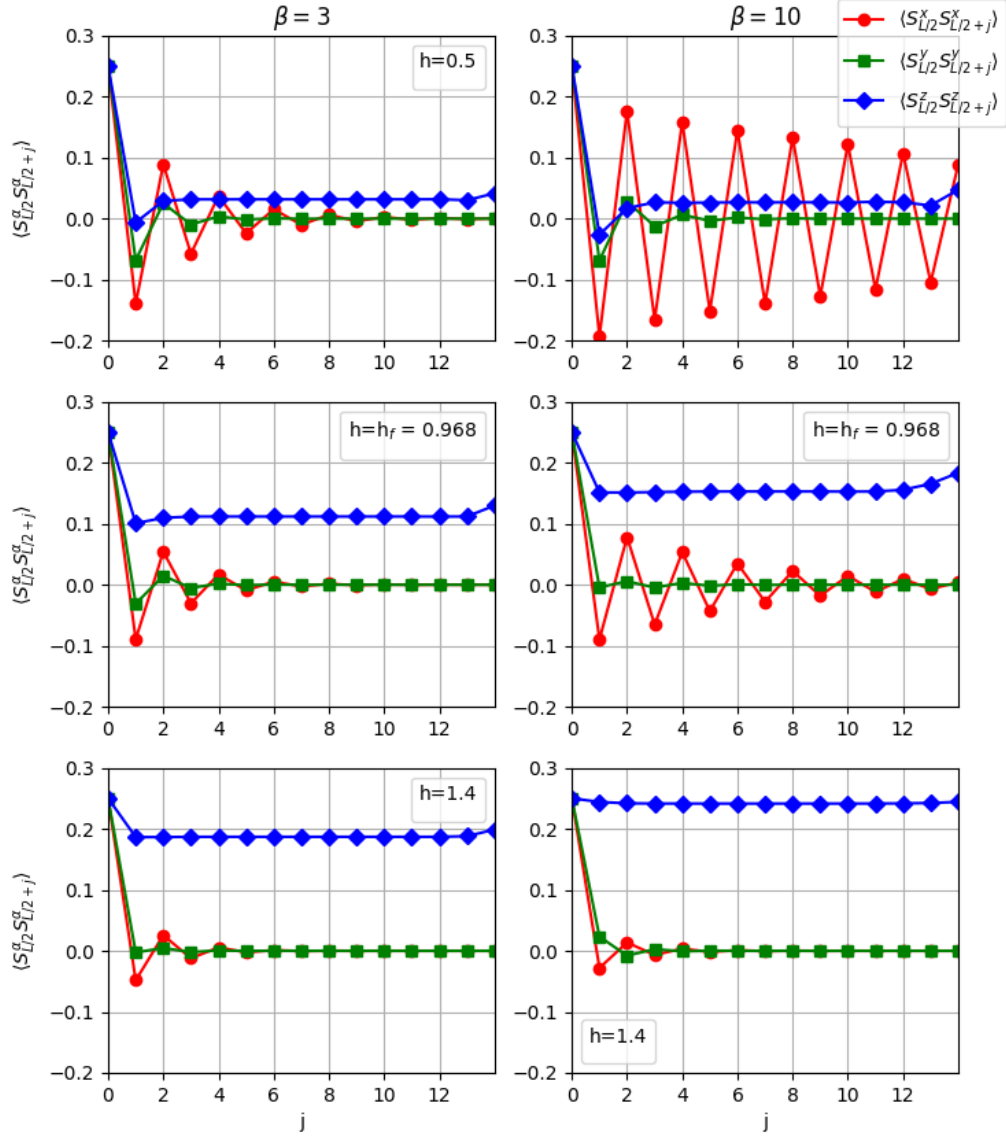


Figure 7.3: Examples of static two-point correlation functions $\langle S^x S^x \rangle$, $\langle S^y S^y \rangle$ and $\langle S^z S^z \rangle$ from the XY model (7.1) with anisotropy $\gamma = 0.25$, as a function of distance j . The figures are drawn for two different values of β (left column $\beta = 3$, right column $\beta = 10$), and three different values of magnetic field: long-range AFM (top), at factorisation field (middle), after factorisation and critical fields (bottom).

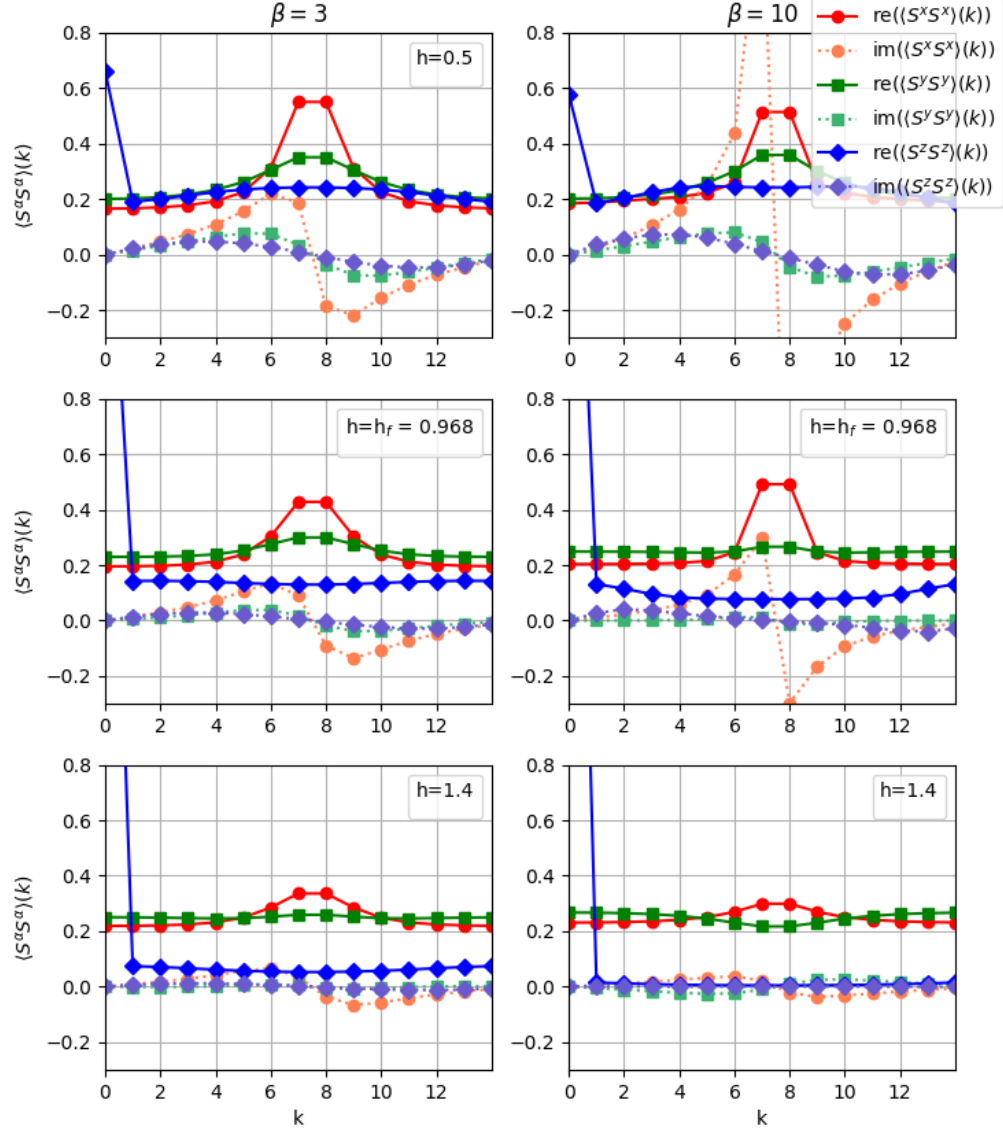


Figure 7.4: Close up of structure factors $\langle S^x S^x \rangle$, $\langle S^y S^y \rangle$ and $\langle S^z S^z \rangle$ from the XY model (7.1) with anisotropy $\gamma = 0.25$, as a function of frequency domain index $k = \frac{qL}{2\pi}$ (outlier zero modes for $\langle S^z S^z \rangle$ are not shown to focus on the features of the rest of structure factors). The figures are drawn for two different values of β (left column $\beta = 3$, right column $\beta = 10$), and three different values of magnetic field: long-range AFM (top), at factorisation field (middle), after factorisation and critical fields (bottom).

In the following results, we have also tested static structure factors $\langle S^\alpha S^\beta \rangle(k)$ as training data, since they were straightforwardly obtained from two-point correlators, by performing Fourier transform. Examples of those are shown in Figure 7.4, for the same Hamiltonian as used in Figure 7.3. Again, the AFM phase is seen in the peak of $S^x S^x$ for $q = \pi$, which seems to retain its real amplitude until the critical field. Although the frequency domain in solid state physics usually gives us a better summary of the features of the model, the ML algorithm can be better trained on more complex data, like two-point correlations in the XY model. For example, the large zero frequency mode in $S^z S^z$ can dominate all the other inputs, making the small but important changes in them irrelevant for a classifying task.

Although in Chapter 6 we found out that static correlators contain enough information to reconstruct entanglement, the dynamic correlators $\langle S_{L/2+j}^\alpha(t) S_{L/2}^\alpha(0) \rangle$ and the dynamical structure factors $S^{\alpha,\alpha}(k, \omega)$ were also used as an input for entanglement learning. The definition of $\langle S_{L/2+j}^\alpha(t) S_{L/2}^\alpha(0) \rangle$ for finite temperatures and how they were obtained is explained in Chapter 3. The dynamical structure factors $S^{\alpha,\alpha}(k, \omega)$ are Fourier transforms of dynamical correlators in spatial and time dimensions. Contrary to the static observables, the data points in both $\langle S_{L/2+j}^\alpha(t) S_{L/2}^\alpha(0) \rangle$ and $S^{\alpha,\alpha}(k, \omega)$ are correlated in two dimensions, meaning that 2-D convolutional layers are more effective. Due to the difficulty of gathering observables that were obtained by both imaginary and real evolution, the system size used for calculations was smaller than the static correlators case ($L = 20$), and only for the same-orientation correlators $\langle S^\alpha S^\alpha \rangle$, $\alpha = x, y, z$.

7.1.3 Entanglement of purification

The von Neumann entropy (2.10) is defined only for a density matrix that comes from tracing out a part of a full system. This system is described by a pure wave function. However, if the system is in a mixed state, the von Neumann entropy cannot be defined in the same way. Since we used thermal density matrix for the calculations here, we could not set the von Neumann entropy as our target for the CNN to predict. Instead we have used a natural measure of entanglement for purification matrix product states (MPSs) called entanglement of purification [124]. It is not usually considered a measure of entanglement, specifically because it does not necessarily vanish when the state is completely disentangled. However, it is still connected to entanglement entropy and in the limit $T \rightarrow 0$, it becomes identical to the von Neumann entropy [53].

The way the entanglement of purification is calculated is depicted in Figure 7.5. Although the auxiliary indices q are not the same as the physical ones p , they can be joined together to form a TN which looks like a pure state MPS. The “physical” dimensions of this new TN are the dimension of the initial Hilbert space squared. Because this TN has now the same form as regular MPS, the entanglement entropy can be calculated normally, but the result will vary depending on the auxiliary space, to which a unitary operator can be attached without changing the thermal density matrix³. It turns out that the minimal value of this entanglement entropy is the entanglement of purification [54]. The minimal value can be achieved by applying disentanglers, which happens to be also a good optimisation procedure for TNs. In this work, we have used backward time evolution disentanglers, which are proven to be a good optimising method for approaching the minimal entanglement [53]. In the later part of this chapter, we still refer to our target measure as entanglement entropy

³This is because the thermal density matrix is obtained by tracing out the auxiliary space, and the unitary operation will be cancelled by its hermitian conjugate. See Section 3.4.3 for more details.

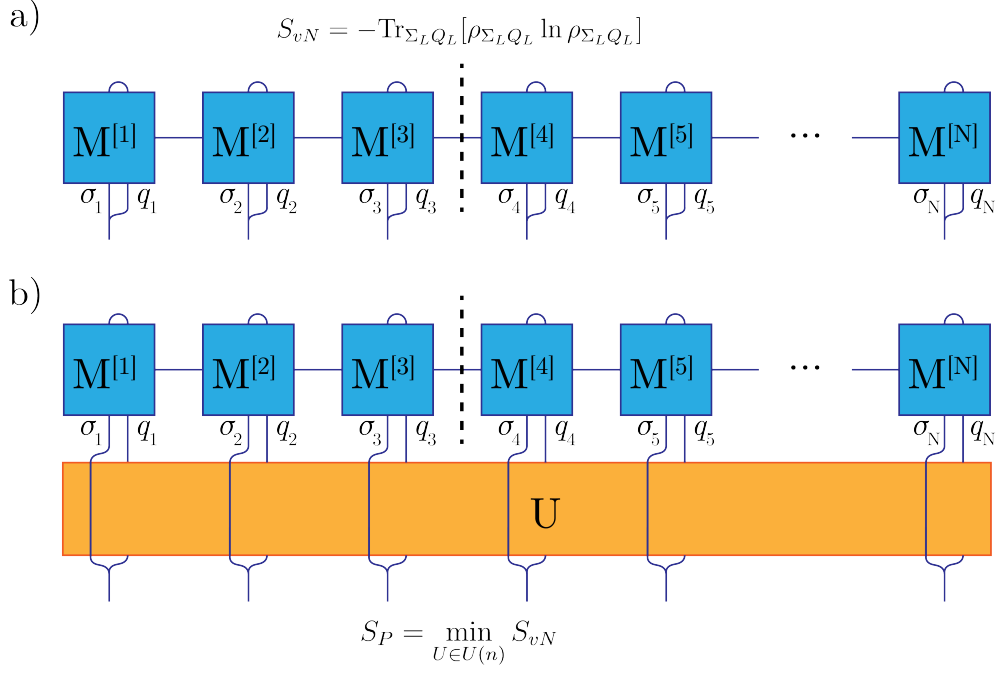


Figure 7.5: Entanglement of purification for 1D purification MPS. a) After joining physical indices σ with the auxiliary q , the von Neumann entropy of any bisection can be calculated as with regular MPS according to formula depicted, where Σ_L and Q_L are physical and auxiliary Hilbert spaces of the left section accordingly. Density matrix $\rho_{\Sigma_L Q_L}$ is given by tracing out the right section. b) Due to the fact that the thermal density matrix is invariant to any unitary matrix U applied to auxiliary indices, but the von Neumann entropy for the purification MPS will change as a function of U , a well defined measure is the minimum of von Neumann entropy for all different unitary operations U , which defines entanglement of purification.

or even von Neumann entropy⁴, but it is important to emphasise that this is a value that approaches purification of entanglement, rather than being directly connected to pure state von Neumann entropy.

7.1.4 Cost function

We established the input for the neural network (NN) which we will build in the previous section. It is also crucial to choose the right output (labels). It would be possible to set it as any of the entanglement measures mentioned in Chapter 2, such as 1-tangle, concurrence or quantum fisher information. However, since we want to train NN to predict entanglement, and the TN calculation gives the entanglement spectrum directly anyway, it was straightforward to set entanglement entropy or entanglement spectrum as the output. From those, the simplest output is just the single number — entanglement entropy averaged over all possible bisections of the 1-D chain model. The cost function (see Section 4.2.2) for the training process can then be for example:

$$C^{[1]}(\text{training data}) = (y(\text{training data}) - S_{av}(\beta, h, \gamma, \Delta))^2, \quad (7.5)$$

where $y(\text{training data})$ is the final output of NN, $S_{av} = \frac{1}{L-1} \sum_j^{L-1} S_j(\beta, h, \gamma, \Delta)$, and the $S_j(\beta, h, \gamma, \Delta)$ is the von Neumann entropy (2.12) for a bisection after $j - th$ spin site on a

⁴As it is commonly referred to by those names in the TN literature [54].

chain of size L of a Hamiltonian with parameters h, γ, Δ^5 , at inverse temperature β .

In order to increase potential information about entanglement, NN output could be predicting the entanglement entropy for each bisection. The cost function then could be:

$$C^{[2]}(\text{training data}) = \sum_j^{L-1} (y_j(\text{training data}) - S_j(\beta, h, \gamma, \Delta))^2. \quad (7.6)$$

For the relatively small systems studied here, the difference between bisection close to edges of the system and in the bulk will add relevant information about the entanglement and that can help with NN training. For larger, translationally invariant systems most of the output neurons y_j should be trained to predict the average entanglement entropy S_{av} and the edge cases will become less relevant.

7.1.5 Neural network architecture

Although the input and the output for NN learning are relatively small for our case, which could justify using fully connected layers, we opted to use CNN (see Section 4.4.2), since the observables in the input are spatially correlated. We have used keras library for the calculations [125]. For most of the results, we have used 2-5 initial convolutional layers, followed by a global averaged pooling layer and a couple of fully connected layers. Although this does not qualify our method as a deep NN, we have seen that the performance does not change significantly after adding more layers. If not otherwise mentioned: the activation functions are ReLU for the convolutional layers and linear for fully connected ones; the cost function is the mean square error (as explicitly written in (7.5) and (7.6)); and the training optimiser is ADAM [126]. Other options for the hyperparameters were explored, but improvements, if any, were minimal.

7.2 Results

To quantify how close the NN can predict the entanglement, we measure the accuracy by $D(\text{prediction})$, which can be defined as

$$D(\text{prediction}) = \frac{\|\mathbf{S}_{\text{true}} - \mathbf{S}_{\text{prediction}}\|}{\|\mathbf{S}_{\text{true}}\|} = \frac{\sqrt{\sum_{j,k} (S_{\text{true}}[h_j, \beta_k] - S_{\text{prediction}}[h_j, \beta_k])^2}}{\sqrt{\sum_{j,k} (S_{\text{true}}[h_j, \beta_k])^2}}, \quad (7.7)$$

where $\mathbf{S}_{\text{true}}$ is some entanglement measure like average entanglement entropy or entanglement entropy at a specific bisection of the chain, while $\mathbf{S}_{\text{prediction}}$ is the result obtained by the NN. The measure $D(\text{prediction})$ is close to 0 when prediction is accurate and close to 1 when an average of a single point (h_j, β_k) predictions is off by 100% of true entanglement measure.

A sketching of the procedure of entanglement learning is shown in Figure 7.6. First, a subset of the total data (from the full range of magnetic fields and temperatures within one model, or from data coming from multiple models) is chosen as a training set, while the rest remains as a validation set. Each set of correlators or structure factors $\langle S^\alpha S^\alpha \rangle$ coming from a specific value of $(h_j, \beta_k, \gamma_n, \Delta_n)$, where n is an index specifying concrete model, is

⁵That is for the XYZ model (7.2), XY model is included in that definition by setting $\Delta = 0$.

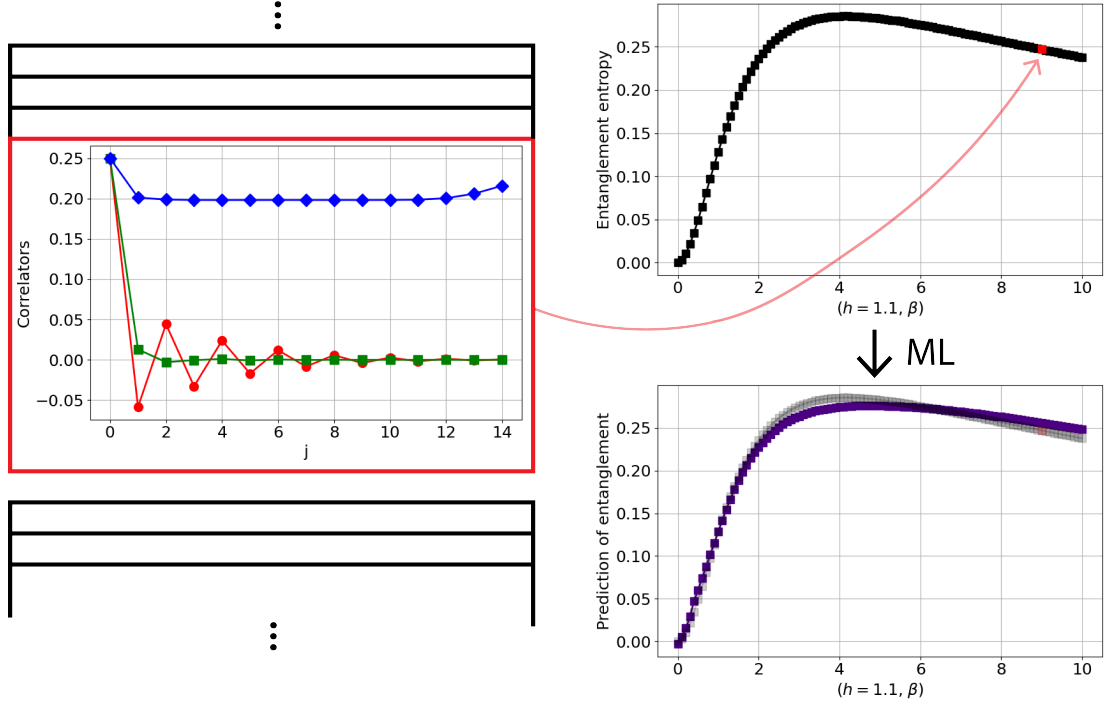


Figure 7.6: The general scheme for entanglement learning, presented using the example of the XY model with anisotropy $\gamma = 0.25$. A portion of the entire data is chosen as the training set. Each data point consists of spatial correlation functions (or structure factors) and the entanglement entropy obtained from the same state as those observables (red arrow pointing to the specific value of entanglement entropy). For clarity, only $\langle S^x S^x \rangle$, $\langle S^y S^y \rangle$ and $\langle S^z S^z \rangle$ are displayed in the left panel. The colours indicate the same observables as in Figures 7.3 and 7.4. Entanglement entropy was plotted for a single value of $h = 1.1$ and entire range of β in right panels. Using CNN with the correlators as an input and the entanglement entropy as an output, the estimation of entanglement entropy (depicted by the violet points in the right bottom plot, the greyed out points are the actual entanglement entropy, as also depicted on the right top plot) can be calculated after training.

connected to the corresponding value of entanglement measure. CNN is trained to predict this value of entanglement, when presented with the set of correlators as an input. After some training sessions, the loss function of the validation set will start to increase, which is a sign of CNN overfitting to the training data (see Section 4.1). At this point a good prediction of entanglement measure should be achieved.

7.2.1 Learning entanglement landscape within given toy model

As a first result, let us study how well NNs can predict the average entanglement landscape $S_{av}(\beta, h)$ for a given toy model, when presented with only the fraction of the data as the training set. The remainder of the entanglement landscape serves as the validation set. To gain interesting insight into how much of the data is required to train NN, different “strategies” for training were selected, meaning both the percentage of the training data as compared to the entire set, and which part of the landscape was chosen. Strategies studied in this thesis broadly fall into two categories: a) For a given inverse temperature β_k , scan the entire magnetic field range $\{h_j\}_{\beta_k}$ and add the input and the average entropy to the training set. Perform those scans for many different β_k ; b) For a given magnetic field h_j , scan the entire

range of inverse temperatures $\{\beta_k\}_{h_j}$. Do those scans for many different h_j ⁶. Since the chosen entanglement measure was the average entanglement entropy, the cost function for the NN was $C^{[1]}$ (equation (7.5)). For comparison, we have studied both observables as inputs — two-point correlation functions and structure factors.

The results are shown in Figures 7.7-7.10. Each figure depicts entanglement learning for two different XY models with anisotropy $\gamma = 0.25$ and $\gamma = 0.5$, for which the same strategies have been used. For each type of scans, five different training sets were chosen, which are depicted by the red vertical or horizontal lines. Training sets were always much smaller than validation data. Overall, even the small NN that was used for this problem handled the task particularly well, if the chosen input was two-point correlation functions (Figures 7.7 and 7.8). The best reconstruction of the average entanglement landscape, not surprisingly, were the evenly-spaced scans that covered entire landscapes. Interestingly, the second best performing strategies were usually the ones that took data from both ends of the range of either β or h (bottom left strategy for h scans and top right strategy for β scans), even if the training set was very small as compared to the entire landscape. For example, the XY model with anisotropy $\gamma = 0.25$, for just two scans of low and high β (Figure 7.7, second row, first column) the NN can reasonably well reproduce the features in the bottom right parts of the entanglement entropy, which are far apart from the training set. Additionally, from the scans of magnetic field, we can see that the algorithm is much better trained on low than high temperature data, which seems logical since low temperature input is much more feature rich, as can be seen from correlation functions (see Figure 7.3).

Looking at Figures 7.9 and 7.10, which represent the entanglement learning from structure factors, it seems obvious that this input is much harder to train the NN on. The best prediction was achieved for the temperature scans at evenly spaced magnetic fields, which, although comparable to the results for correlation functions in the measure D (prediction), give much less smooth predictions for the landscape.

⁶Those strategies were chosen to be similar to an experimental setup. It is worth mentioning that performing temperature scans for a set external magnetic field is easier in experiment than performing magnetic field scans at set temperature. That is because changing external magnetic field affects the temperature of the sample [127].

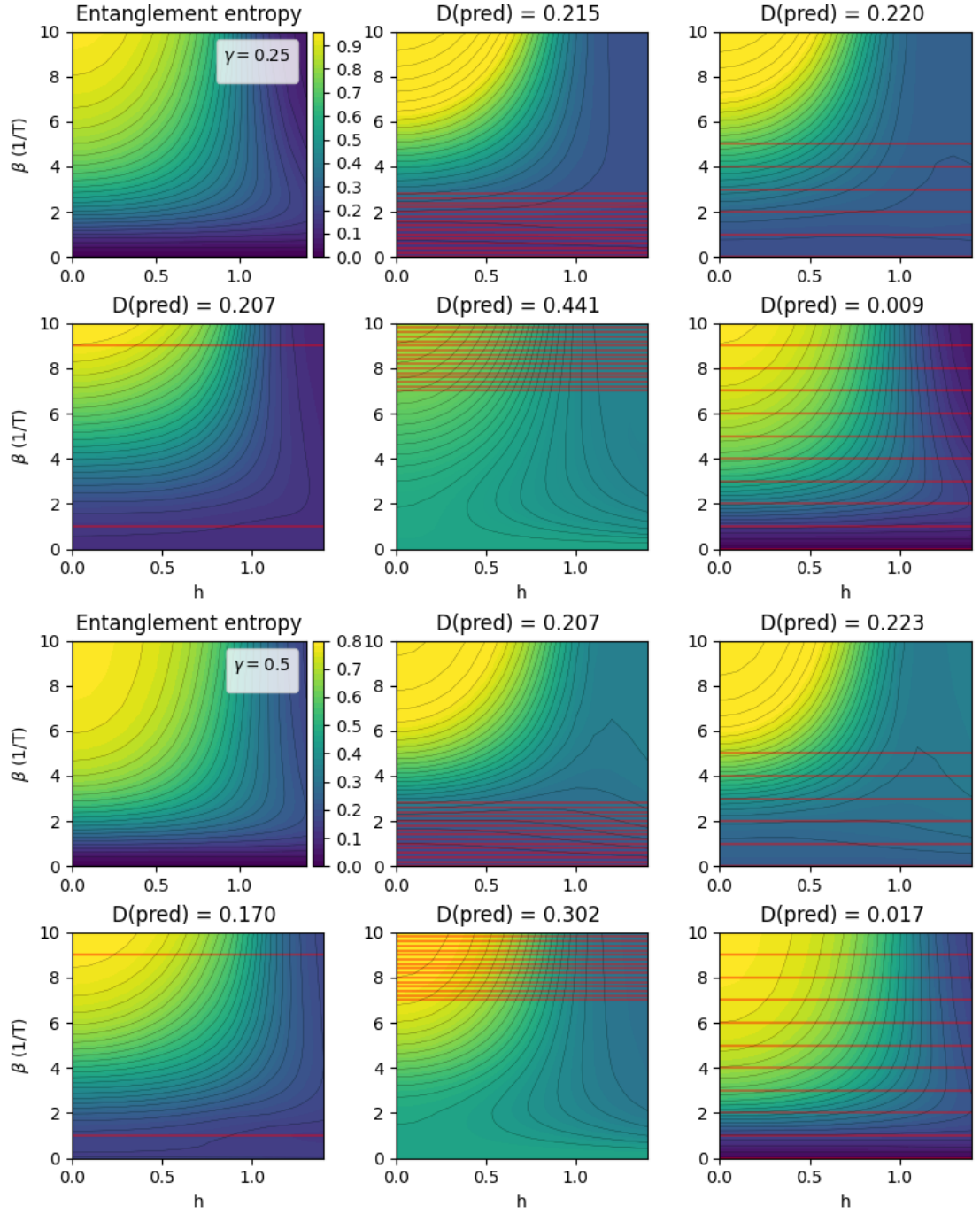


Figure 7.7: Different entanglement learning strategies for training on two-point correlation functions. Scans of external magnetic fields h at chosen inverse temperature β . The data is shown for XY model with $\gamma = 0.25$ (top 6 plots) and XY model with $\gamma = 0.5$. (bottom 6 plots). Red horizontal lines indicate data that was used as a training set. The true entanglement entropy is displayed in the top left plot for each model. The contour lines show the values of prediction for each training strategy, while the number at the top is the measure of fitness defined in (7.7).

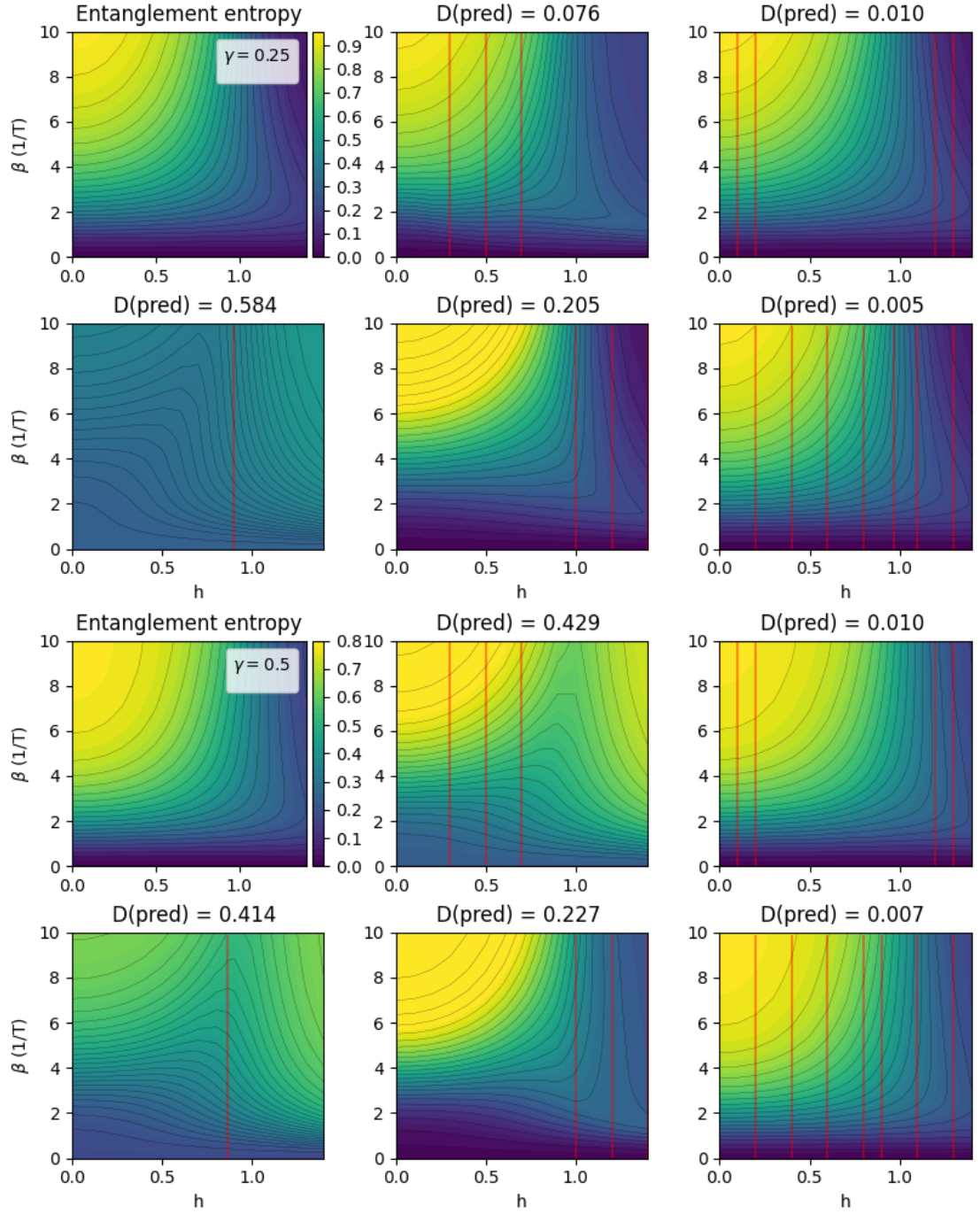


Figure 7.8: Different entanglement learning strategies for training on two-point correlation functions. Scans of inverse temperature β at chosen external magnetic fields h . The data is shown for XY model with $\gamma = 0.25$ (top 6 plots) and XY model with $\gamma = 0.5$. (bottom 6 plots). Red horizontal lines indicate data that was used as a training set. The true entanglement entropy is displayed in the top left plot for each model. The contour lines show the values of prediction for each training strategy, while the number at the top is the measure of fitness defined in (7.7).

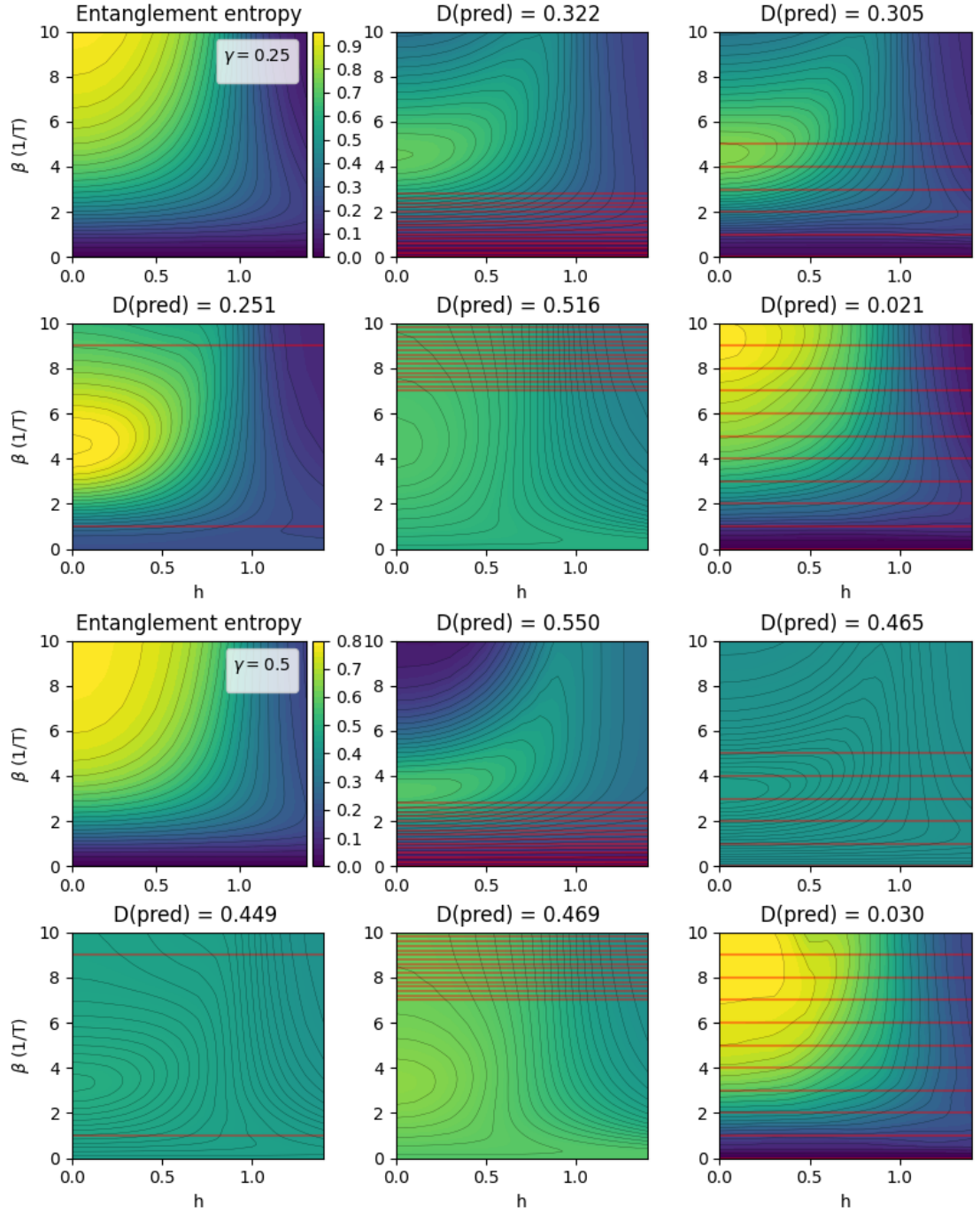


Figure 7.9: Different entanglement learning strategies for training on structure factors. Scans of external magnetic fields h at chosen inverse temperature β . The data is shown for XY model with $\gamma = 0.25$ (top 6 plots) and XY model with $\gamma = 0.5$. (bottom 6 plots). Red horizontal lines indicate data that was used as a training set. The true entanglement entropy is displayed in the top left plot for each model. The contour lines show the values of prediction for each training strategy, while the number at the top is the measure of fitness defined in (7.7).

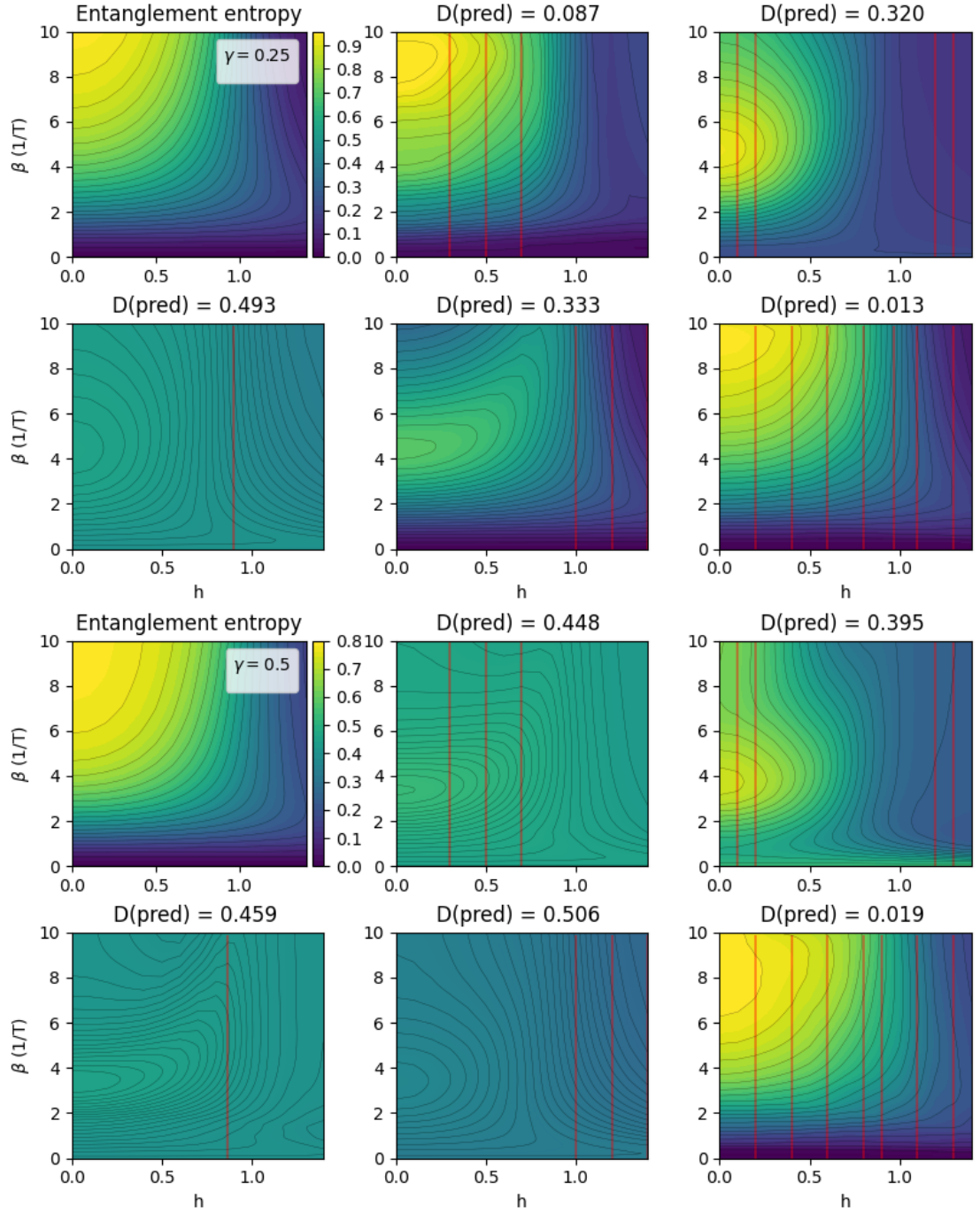


Figure 7.10: Different entanglement learning strategies for training on structure factors. Scans of inverse temperature β at chosen external magnetic fields h . The data is shown for XY model with $\gamma = 0.25$ (top 6 plots) and XY model with $\gamma = 0.5$. (bottom 6 plots). Red horizontal lines indicate data that was used as a training set. The true entanglement entropy is displayed in the top left plot for each model. The contour lines show the values of prediction for each training strategy, while the number at the top is the measure of fitness defined in (7.7).

7.2.2 Learning entanglement landscape between two models

Having explored the entanglement learning within single Hamiltonian, the next logical step is to try to predict entanglement of one model, having trained NN on another. Here we take all of the data from one model to be the training set. While NN is trying to fit the data from one model, we test the prediction for the other. We also decided to use entanglement entropy at all bisections as an output, trying to see how well NN can handle more complicated cost function for the entanglement learning.

The results for learning between two XY Hamiltonians with different anisotropy coefficients $\gamma = 0.25$ and $\gamma = 0.5$ are shown in Figures 7.11 and 7.12. The entanglement shown is for the bisection in the middle of the $L = 30$ spin chain. While it is not surprising that the prediction of the entanglement using two-point correlators is almost exact for the training set, NN also gives a good prediction for the other model. The features that are the signs of ET⁷, such as the “kink” at some value of h , where for the lower external magnetic field the entanglement constantly grows with temperature, while at the higher field, entanglement has reached its highest point and is lowering. The field which separates those two features of the entanglement landscape is pretty well identified by the ML algorithm for the two-point correlators (Figure 7.11). The data from training NN on structure factors (Figure 7.12) again shows that it is not obvious that ML can easily extrapolate to other models. While, for example, it is trained on $\gamma = 0.25$ Hamiltonian and set to predict the $\gamma = 0.5$ (Figure 7.12, second row) model, the prediction is a smooth function, but completely fails at high β and low h .

Additionally, we checked if the same prediction is possible even for more distant models. For this, we set either a XY model with $\gamma = 0.25$ or a XYZ model with parameters $\gamma = 0.5$ and $\Delta = 0.25$ as a training set and tried to predict the other model from trained NN. The results displayed in Figures 7.13 and 7.14 are for training on two-point correlators and structure factors correspondingly. Again, training on the two-point correlators clearly outperforms training on structure factors. Similar results were observed for other bisections of the systems, not shown (since the output in this case was a 1-D vector of entanglement entropy from all bisections of the system, NN is trained on all of them simultaneously). Overall, the NN can predict entanglement for systems with significantly deviated interaction strengths from the Hamiltonian from which the NN have learned.

⁷Because of the finite temperature and a relatively small size of the system, we do not see the entanglement entropy go to 0, as we would expect at factorisation field.

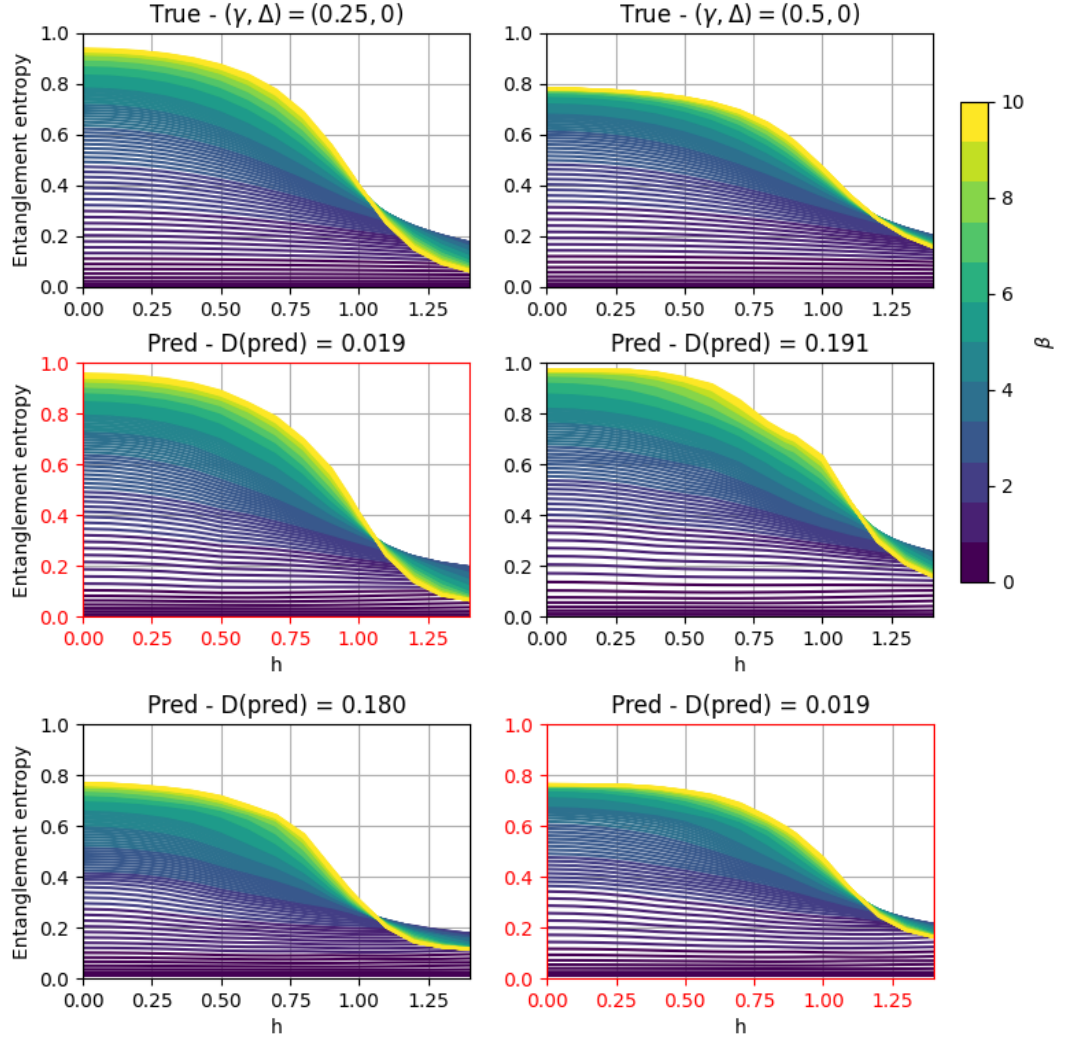


Figure 7.11: Entanglement learning between two different XY Hamiltonians, trained on two-point correlators. Left column shows the true and predicted entanglement entropy at the bi-section in the middle of $L = 30$ spin chain for a XY model with anisotropy $\gamma = 0.25$. Right column shows the same for XY model with anisotropy $\gamma = 0.5$. The plots are functions of external magnetic field with a range of β values. Top two figures show the true entanglement entropy. Two middle figures show predictions when trained on two-point correlators from just the $\gamma = 0.25$ model. The bottom two figures show predictions when trained on two-point correlators from just the $\gamma = 0.5$ model. Red border and axes indicates which of the prediction is done on the model that served as the training set for the NN.

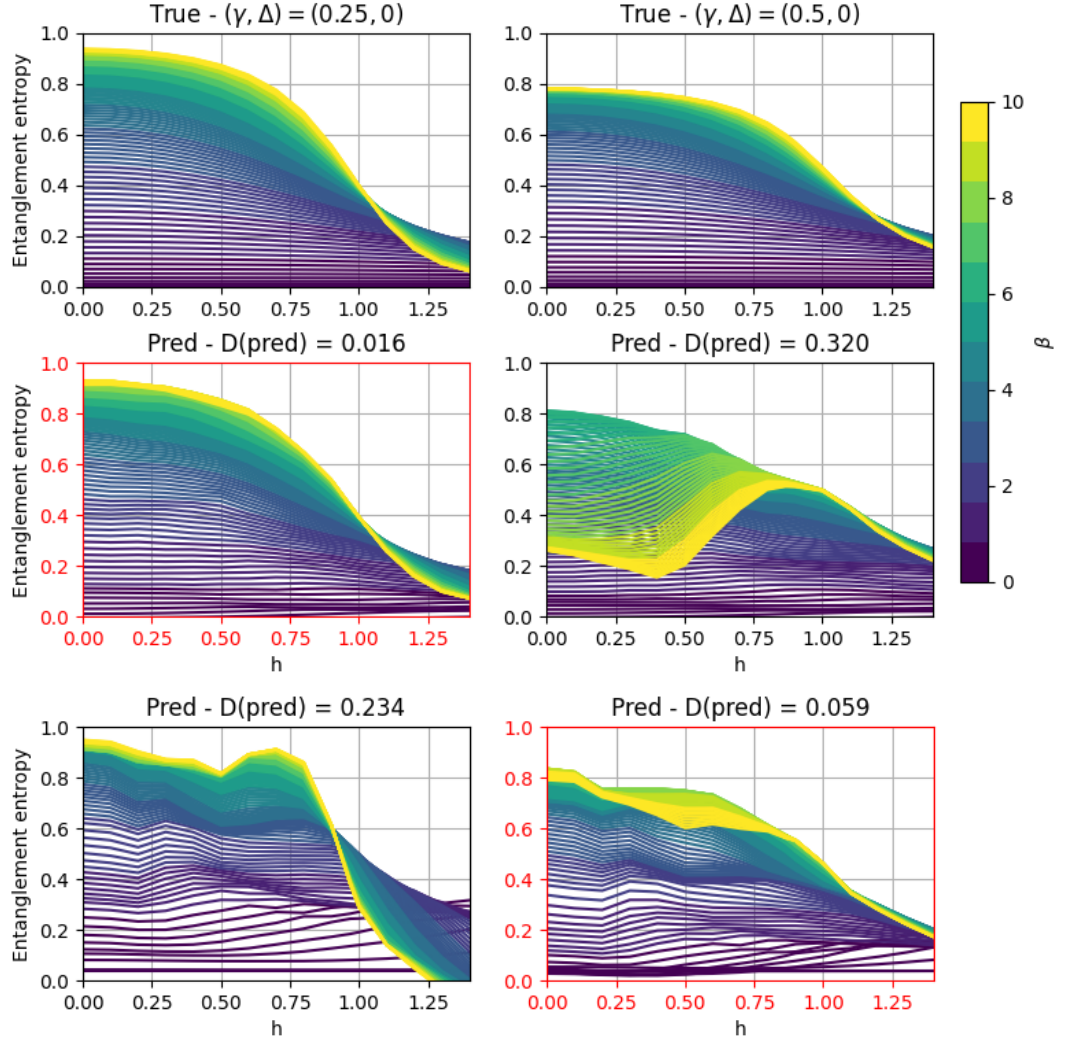


Figure 7.12: Entanglement learning between two different XY Hamiltonians, trained on structure factors. Left column shows the true and predicted entanglement entropy at the bi-section in the middle of $L = 30$ spin chain for a XY model with anisotropy $\gamma = 0.25$. Right column shows the same for XY model with anisotropy $\gamma = 0.5$. The plots are functions of external magnetic field with a range of β values. Top two figures show the true entanglement entropy. Two middle figures show predictions when trained on structure factors from just the $\gamma = 0.25$ model. The bottom two figures show predictions when trained on structure factors from just the $\gamma = 0.5$ model. Red border and axes indicates which of the prediction is done on the model that served as the training set for the NN.

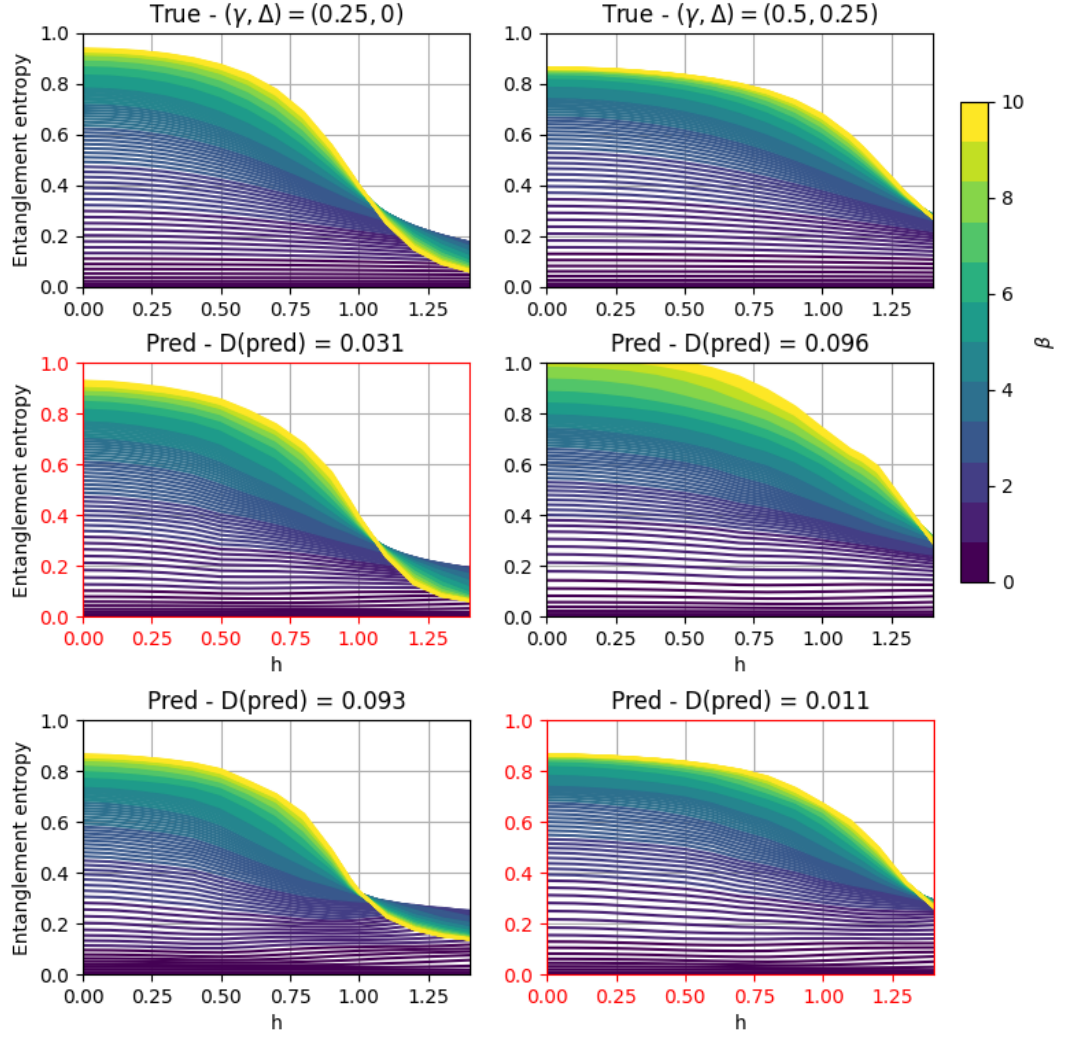


Figure 7.13: Entanglement learning between XY and XYZ Hamiltonians, trained on two-point correlators. Left column shows the true and predicted entanglement entropy at the bisection in the middle of $L = 30$ spin chain for a XY model with anisotropy $\gamma = 0.25$. Right column shows the same for XYZ model with anisotropy $\gamma = 0.5$ and $\Delta = 0.25$. The plots are functions of external magnetic field with a range of β values. Top two figures show the true entanglement entropy. Two middle figures show predictions when trained on two-point correlators from just the $(\gamma, \Delta) = (0.25, 0)$ model. The bottom two figures show predictions when trained on two-point correlators from just the $(\gamma, \Delta) = (0.5, 0.25)$ model. Red border and axes indicates which of the prediction is done on the model that served as the training set for the NN.

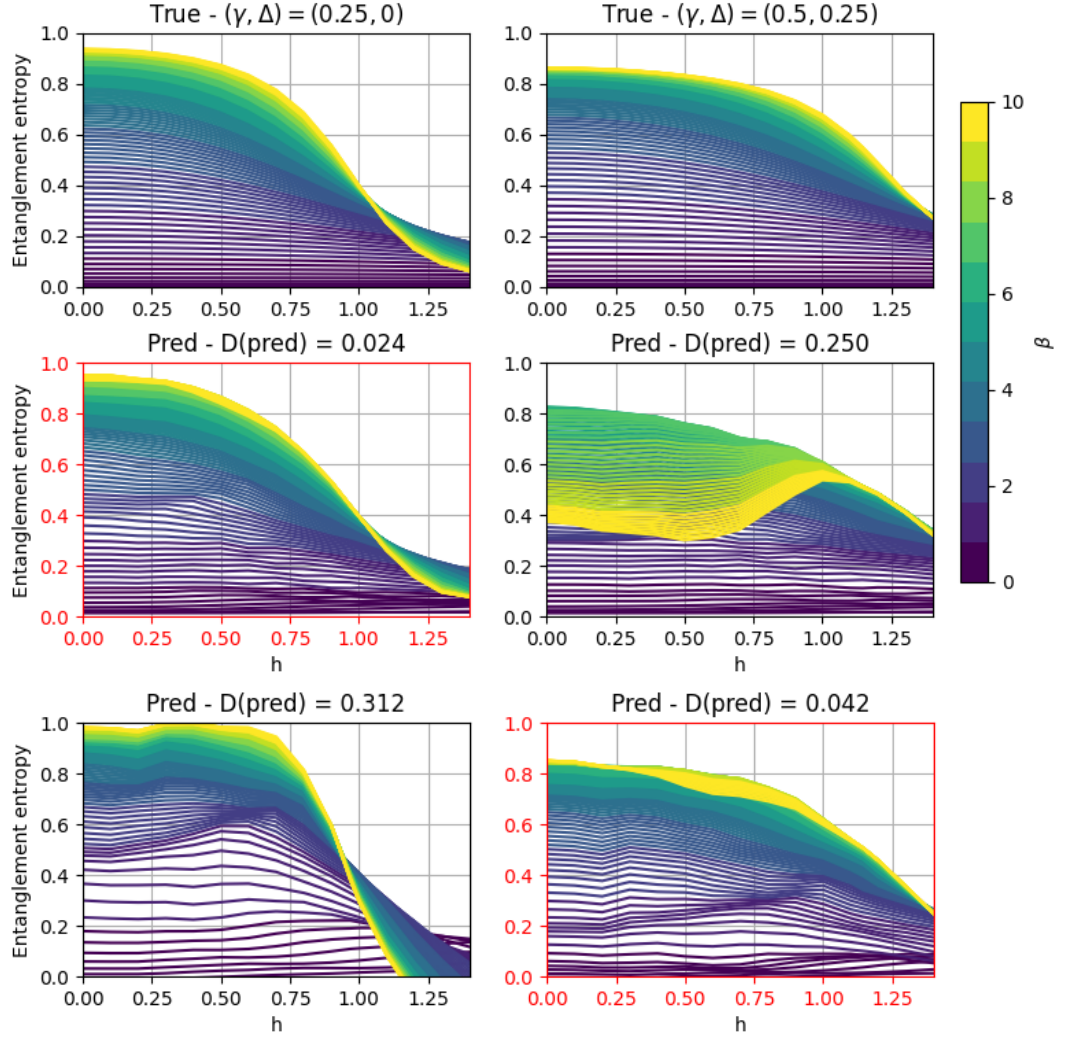


Figure 7.14: Entanglement learning between XY and XYZ Hamiltonians, trained on structure factors. Left column shows the true and predicted entanglement entropy at the bisection in the middle of $L = 30$ spin chain for a XY model with anisotropy $\gamma = 0.25$. Right column shows the same for XYZ model with anisotropy $\gamma = 0.5$ and $\Delta = 0.25$. The plots are functions of external magnetic field with a range of β values. Top two figures show the true entanglement entropy. Two middle figures show predictions when trained on structure factors from just the $(\gamma, \Delta) = (0.25, 0)$ model. The bottom two figures show predictions when trained on structure factors from just the $(\gamma, \Delta) = (0.5, 0.25)$ model. Red border and axes indicates which of the prediction is done on the model that served as the training set for the NN.

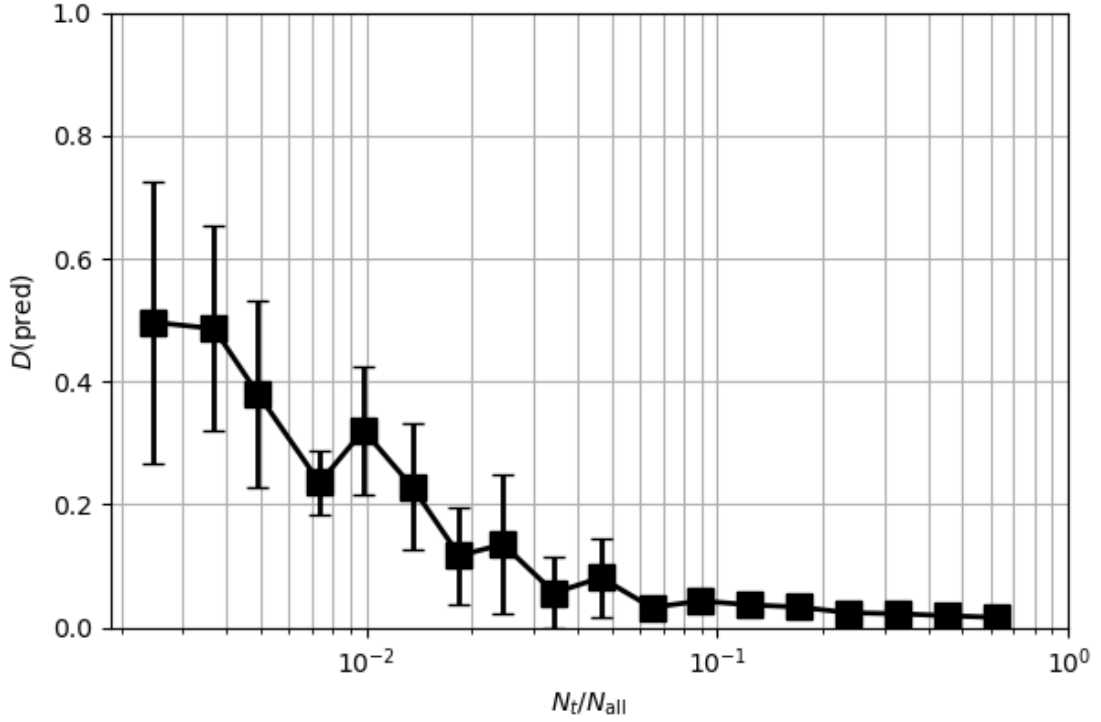


Figure 7.15: Entanglement learning for the XY Hamiltonian with anisotropy $\gamma = 0.25$, trained on dynamic two-point correlators. The ratio N_t/N_{all} is the amount of data in the training set as compared to the total number of data points. $D(\text{pred})$ is calculated according to Equation (7.7) for all data points. The error bars show the variance between different training sets in multiple runs. The training sets were chosen at random.

7.2.3 Learning from dynamical observables

While the previous results were obtained from learning on static correlators, we have also performed entanglement learning for dynamical two-point correlation functions and structure factors. Rather than repeating the training strategies described in Section 7.2.1, we have opted for randomly choosing data points (β, h) from the whole set of inverse temperatures and external magnetic fields, to serve as the training set. After training our CNN to learn the average entanglement entropy, we calculated $D(\text{prediction})$ for that random sample of training data points. To see how much data is required to correctly learn the entanglement, we trained CNN for multiple different fractions of the whole data set $\frac{N_t}{N_{\text{all}}}$, where N_t is the number of randomly chosen samples and N_{all} is the number points (β, h) in whole training set. We used only data from a single anisotropic XY Hamiltonian, with $\gamma = 0.25$. To account for the randomness of choosing data points (especially for low N_t), we performed $M = 10$ different samplings for each fraction $\frac{N_t}{N_{\text{all}}}$.

Figure 7.15 shows the fitness function $D(\text{prediction})$ with respect to the fraction $\frac{N_t}{N_{\text{all}}}$, when trained on dynamical two-point correlators. Figure 7.16 shows the same when CNN was trained on dynamical structure factors. Although it is hard to compare those results with the previous ones shown in Section 7.2.1 (because both the input and the training strategy is different), overall it seems that dynamical correlators perform better as the input observables than the static ones. This feels reasonable, since the former contain much more information and features than the latter. It takes very little training data for CNN to predict the entanglement well. At around 6% for the two-point correlators and 3% of structure factors, we

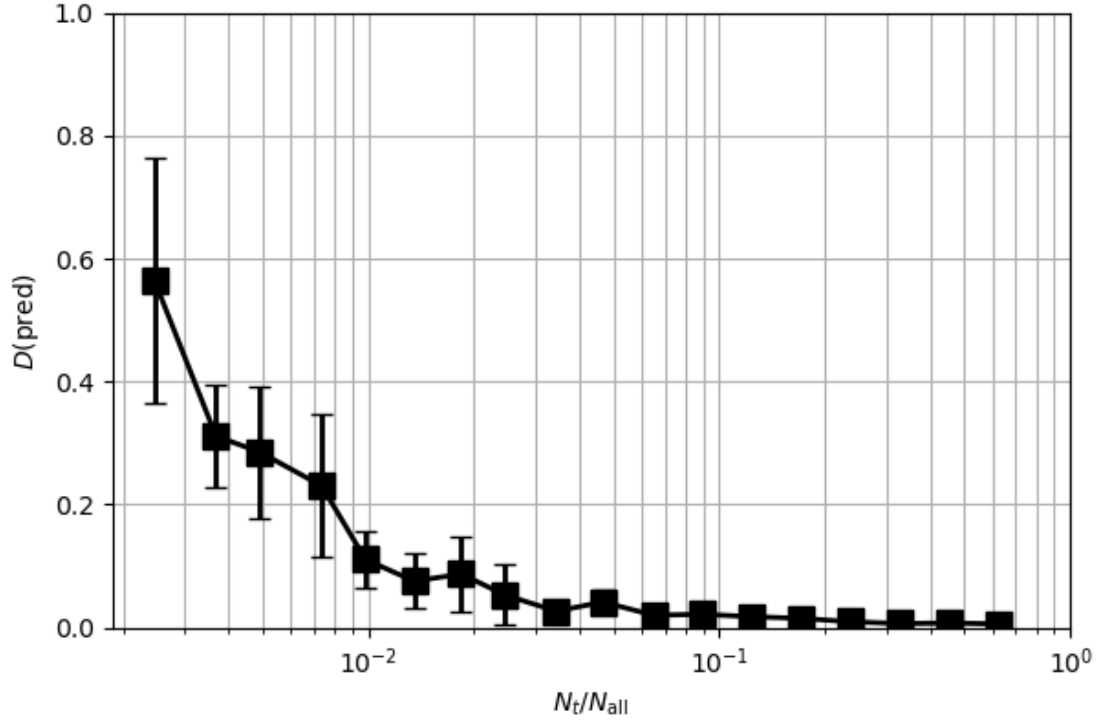


Figure 7.16: Entanglement learning for the XY Hamiltonian with anisotropy $\gamma = 0.25$, trained on dynamic structure factors. The ratio N_t/N_{all} is the amount of data in the training set as compared to the total number of data points. $D(\text{pred})$ is calculated according to Equation (7.7) for all data points. The error bars show the variance between different training sets in multiple runs. The training sets were chosen at random.

see that $D(\text{prediction})$ drops almost to zero and further improvement is much smaller. Furthermore, the variation in $D(\text{prediction})$ is negligible from that point on, suggesting that any random sampling at that number N_t collects enough information about the entanglement landscape. Surprisingly, the CNN trained on dynamical structure factors performs better than the one trained on two-point correlators, opposite to the static observables case. This is seen both in the smaller ratio that is needed to reach good prediction values (as mentioned before around 6% for the two-point correlators and 3% for structure factors) and in smaller values of $D(\text{prediction})$ at larger ratios.

7.3 Class activation maps

NNs are often referred to as “black boxes”, meaning that even though we understand the individual actions of neurons (or units) and learning procedures, the complexity that arises from many neurons acting together is usually very hard to trace. That means that we cannot tell anything interesting about the input data just by looking at weights of NNs. For physical applications, it would be extremely useful to have some insight into the learning procedure, because that could highlight the correlations in the data which NNs treat as important in the classification process. Class activation maps (CAMs) [128] are a simple method which shows exactly that. If the CNNs that we are using have k units in its last convolutional layer and we perform global average pooling (GAP) to a fully connected layer (FCL) such that the

activation in FCL for neuron k is

$$F_k = \sum_{x,y} f_k(x, y), \quad (7.8)$$

where $f_k(x, y)$ is the activation of k unit in the last convolutional layer at position x, y . Then the total input to the final class categorization layer is

$$S_c = \sum_k w_k^c \sum_{x,y} f_k(x, y) = \sum_{x,y} \sum_k w_k^c f_k(x, y). \quad (7.9)$$

The overall contribution from spatial coordinates x, y to the class score S_c is called class activation map

$$M_c(x, y) = \sum_k w_k^c f_k(x, y). \quad (7.10)$$

This metric tells us for each image which neurons are important in recognising this image as belonging to a given class c . It was shown to be very successful at recognising important features in RGB pictures.

As a specific application, we examine how the CAMs help us diagnose which features of the chosen observables are critical in determining which phase the system is in. As we mentioned in Section 7.1.1, the factorisation field h_f coincides with the change between dominance of the AFM and ferromagnetic ground states. Although the change in correlations at the level crossing should still be visible if our data was collected for zero-temperature case and a mixture of two-fold degeneracy that factorises the state, this still should be thought of at most as an indirect indicator of the ET.

We have decided to use the scattering function data of a six site molecular magnet for this application of NN, since unlike the previous two-point correlators and structure factors, scattering functions are a 2-D input, and therefore are better suited for a technique used originally on 2-D images. A large amount of data for that task was collected using ED, rather than TN methods. The example of eight scattering functions for a XY hexagonal ring, with an anisotropy parameter $\gamma = 0.4$ is presented in Figure 7.17. The left column represents images with magnetic field values less than factorisation field $h < h_f$ and the right column for $h > h_f$. One can clearly see the transition changing the AFM correlations (peaks at $\mathbf{q} = 2\pi \left(\cos \left[\frac{2\pi}{6} \left(n + \frac{1}{2} \right) \right], \sin \left[\frac{2\pi}{6} \left(n + \frac{1}{2} \right) \right] \right), n = 0, 1, 2, \dots, 5$) to ferromagnetic correlations (peaks at $\mathbf{q} = \mathbf{0}$ and $\mathbf{q} = 2\pi \frac{1}{\cos[\frac{\pi}{6}]} \left(\cos \left[\frac{2\pi}{6} n \right], \sin \left[\frac{2\pi}{6} n \right] \right), n = 0, 1, 2, \dots, 5$).

The results of applying CNN learning and CAM analysis for a six site molecular magnet of $S = \frac{1}{2}$ XY Hamiltonian (with anisotropy parameter $\gamma = 0.4$) with perpendicular field are shown in Figures 7.18 and 7.19. The input data are the scattering functions shown in Figure 7.17. The system goes through ET at $h_f \approx 0.916$. In order to use CAMs, we use CNN as a classifier, meaning that rather than learning of the entanglement entropy like in previous section, we simply tell whether an image at a given h is below or above h_f and that is the label for that image. We generated 1000 images for h randomly sampled from an uniform distribution $h \in [0, 1]$. We then trained NN to recognise if the image is in the class of “below h_f ” or “above h_f ”. After successful training, we looked at the CAMs.

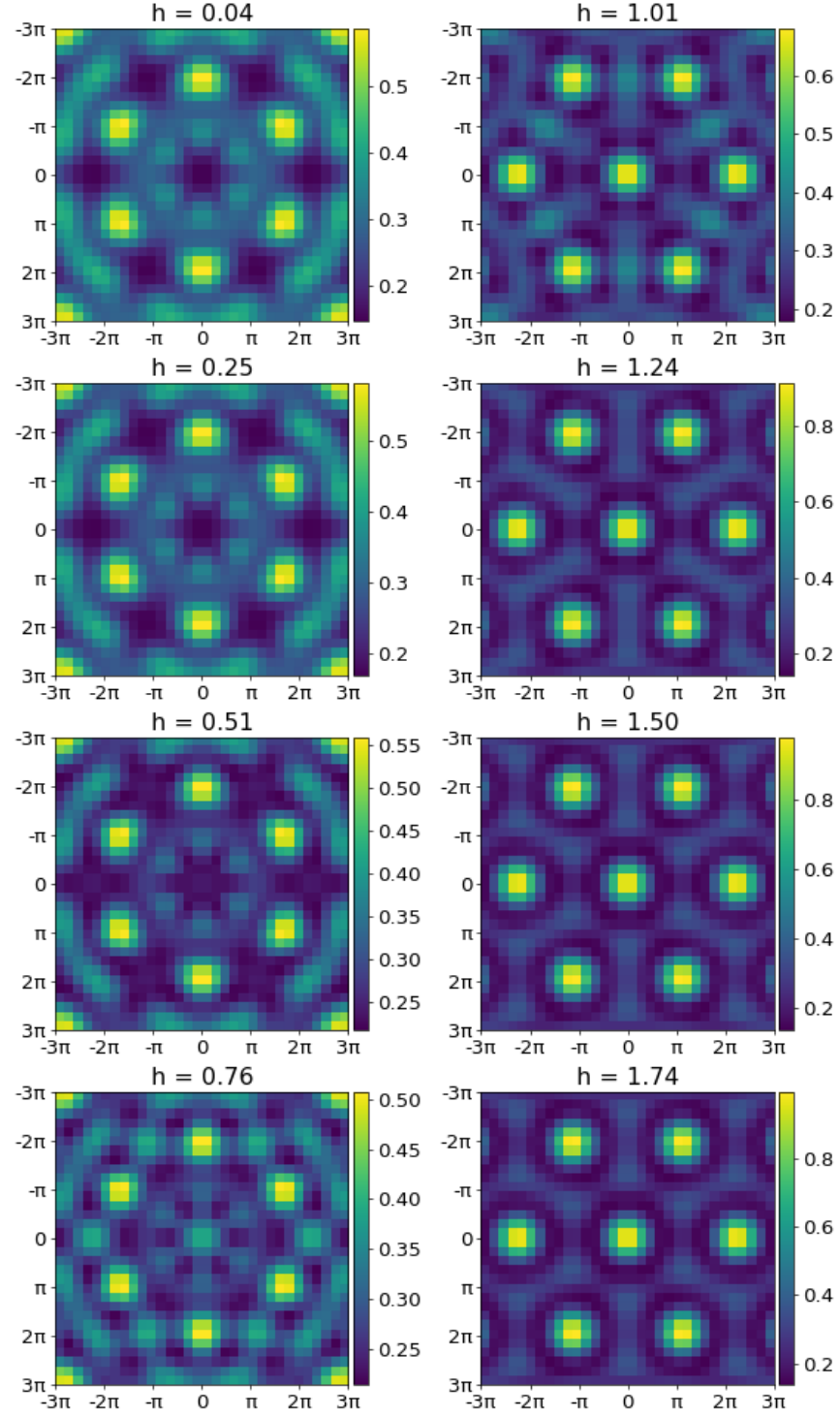


Figure 7.17: Example of 2-D scattering functions for an hexagonal molecular magnet with the XY Hamiltonian (7.1), with anisotropy parameter $\gamma = 0.4$. Left column show scattering functions before h_f , and right columns are the scattering functions after h_f .

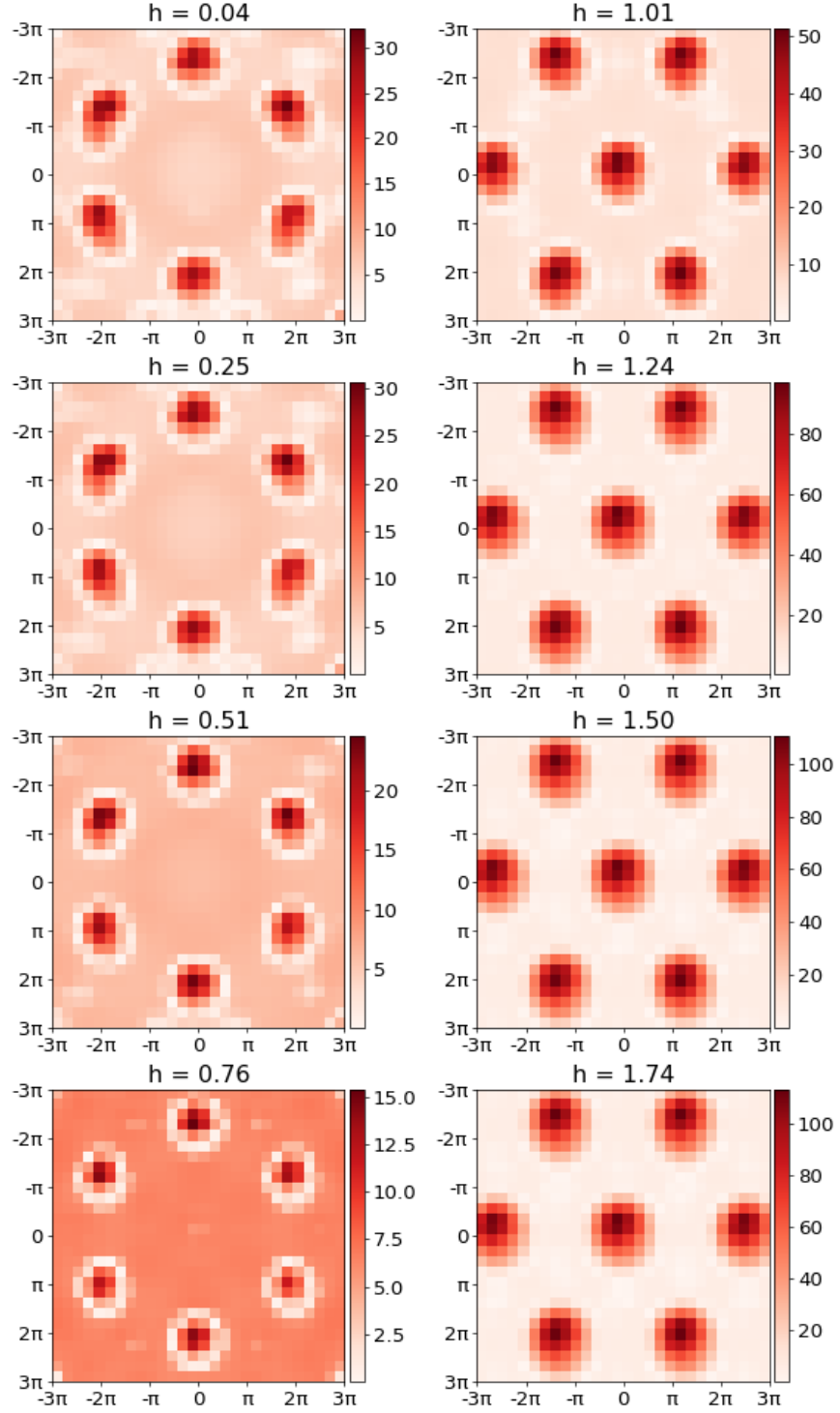


Figure 7.18: CAMs for identifying the phase below ET. Left column shows the CAMs from a scattering functions of a anisotropic XY model (7.1) with anisotropy parameter $\gamma = 0.4$ with external magnetic fields smaller and close to h_f . Right column shows CAMs for the same model h_f .

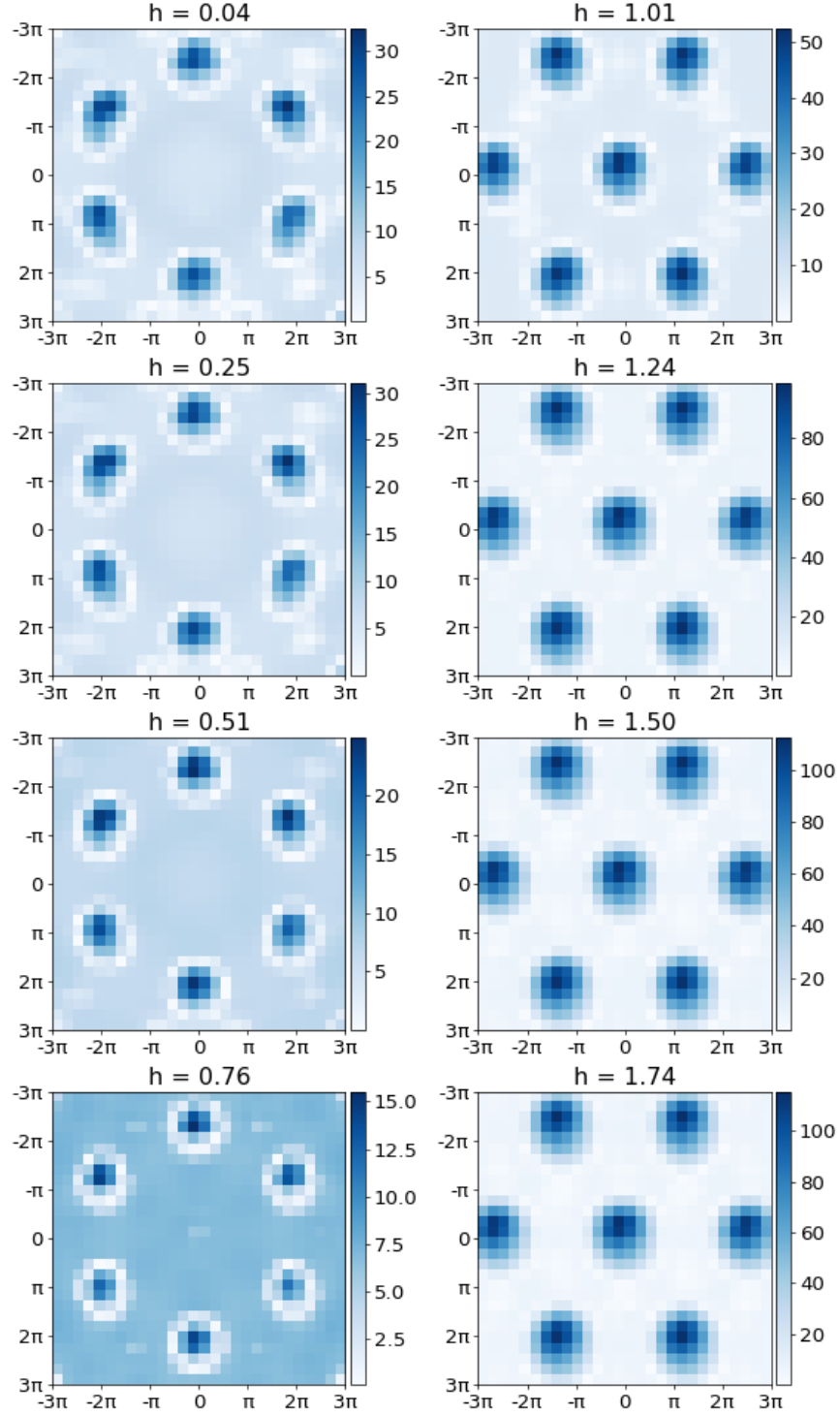


Figure 7.19: CAMs for identifying the phase after ET. Left column shows the CAMs from a scattering functions of a anisotropic XY model (7.1) with anisotropy parameter $\gamma = 0.4$ with external magnetic fields smaller and close to h_f . Right column shows CAMs for the same model after h_f .

The Figure 7.18 shows the CAM for phase below and after h_f , for an example of 8 images with different external magnetic field h . CAMs can be interpreted as the importance of the neurons in the final CNN layer⁸ (before average pooling) in deciding whether that image is in a given class or not. So the information in Figure 7.18 is telling us that in order to classify the images correctly as “below h_f ” (left column), it looks at the AFM peaks at $\mathbf{q} = 2\pi \left(\cos \left[\frac{2\pi}{6} \left(n + \frac{1}{2} \right) \right], \sin \left[\frac{2\pi}{6} \left(n + \frac{1}{2} \right) \right] \right)$, $n = 0, 1, 2, \dots, 5$, and if the images are classified correctly as not being “below h_f ” it looks at the ferromagnetic peaks at $\mathbf{q} = \mathbf{0}$ and $\mathbf{q} = 2\pi \frac{1}{\cos[\frac{\pi}{6}]} \left(\cos \left[\frac{2\pi}{6} n \right], \sin \left[\frac{2\pi}{6} n \right] \right)$, $n = 0, 1, 2, \dots, 5$. In other words, the NN will try to find the AFM peaks and deduce that as an indication of being “below h_f ”, but it will also look at the ferromagnetic peaks and if it finds any, that is an indication of not being in the class of “below h_f ”. Interestingly, the same result is obtained when NN classify objects as being or not being in the class of “above h_f ” (Figure 7.19). It looks at the same values of $\mathbf{q}$ for that classification. Interestingly, near the transition point, in the AFM phase, NN seems to look at the entire range of $\mathbf{q}$ vector when deciding whether it is in one phase or not, rather than just the peaks.

7.4 Conclusions

As mentioned in the introduction to this chapter, we set out to answer whether it is possible to train NNs to predict the entanglement entropy directly, and see how well the limited training data on a single Hamiltonian can be used to extrapolate to other systems. We showed that in the case of learning (two-point correlation function) $\rightarrow$ (entanglement entropy), for the anisotropic XY and XYZ models, a NN can be very successful. It reconstructs the features of the entanglement landscape from learning on small subset of all data from a single model. It also can reconstruct entanglement from a Hamiltonian, when trained on another model. While training (static two-point correlation function) $\rightarrow$ (entanglement entropy) proved to be successful, training (static structure factors) $\rightarrow$ (entanglement entropy) resulted in less accurate entanglement predictions, both for the training set and the entire entanglement landscape. The exact opposite result was observed when training CNN on dynamic observables. We have shown that for this more feature-rich data, the entire entanglement landscape could be correctly predicted when using only around 3% of the entire data as the training set, within one Hamiltonian model. This tells us that for the real-life application of CNN as entanglement detection, we potentially would need very few labeled data points⁹ to have a useful entanglement measure.

To gain additional insight about the learning process, we also tried to use NN to classify whether a given data is below or above factorisation field h_f , with scattering functions as an input. By using CAMs, this allowed us to see which features of the input data are important for the decision process. The NN learns to look mainly at the peaks that belong either to AFM or ferromagnetic correlations. Closer to the factorisation field, the algorithm also seems to focus more on all of the neurons, not only those that represent peaks. The previous work by Hannah Irons et al. [120] showed that the $k = (0, 0)$ point can be a good order parameter for the ET. Using CAMs in CNNs provide a broader interpretation of the change between the AFM and ferromagnetic orders.

⁸Which, because of the nature of CNN, extends to the neurons from the input layer.

⁹In order to train CNN to detect entanglement entropy the real-life data used for the training set would have to come from experimental methods in which Hamiltonian parameters are known and can be controlled like cold atoms in optical lattices.

Chapter 8

Conclusions

In this dissertation we explored machine learning (ML) applications to the study of magnetic materials. Specifically, we focused on how ML can be used to study phase transitions in time-reversal symmetry breaking (TRSB) superconductors, and how well convolutional neural networks (CNNs) are able to predict the entanglement in models that exhibit entanglement transition (ET). Each original result that is within the scope of this thesis can be summarised as follows:

- Chapter 5 — We showed that principal component analysis (PCA) can be successfully used to determine TRSB superconducting phase transitions in a range of materials, from a muon spectroscopy data. We first tested the effects of the experimental error using simulated Kubo-Toyabe functions with Gaussian noise, and came up with a method of representing data that came from differently binned experiments, so that the overall error is similar for each new bin. This was necessary for the PCA to work on large data set. We then showed both separate (independent PCA for each of the material) and joint PCA analysis of materials, which included TRSB superconductors, proposed spin liquid and an antiferromagnet. Furthermore, we established a method of calculating error in principal component (PC) scores that is coming from the experimental error, which to our knowledge has not been done before. When we discovered that the PCA is improved upon adding more muon spectroscopy data, even from materials that have vastly different magnetic properties, we shown that a simple metric (cumulative principal component score) can be used to determine which of the PCs are important for a given material, when performing joint analysis.
- Chapter 6 — Following on previous results of one of the supervisors of this dissertation [101], we extended a proof of the direct connection between neutron-scattering functions and Heisenberg-like Hamiltonians/ground states to the same systems with the addition of external magnetic field. Although the extension to the scattering data at finite temperature was published first by Murta et al. [111], we have also independently developed that proof. We then studied the fitness landscape of scattering functions by randomly sampling variations from a target state, both for the ground state and for thermal density matrix. We found that for the zero-temperature calculations, when properly normalised, the scattering functions follow a simple linear trend up to the size of the system $L = 14$, meaning that the further the distance between two states, the further are the two scattering functions that come from those states. Surprisingly, the fitness landscape of scattering functions at finite temperatures showed sub-linear behaviour as a function of the distance between two density matrices. Over-

all, we showed encouraging evidence that the connection (Observables) $\rightarrow$ (Quantum state) for magnetic materials can be utilised in practice. Most relevantly in this research, for entanglement learning.

- Chapter 7 — We have tested the possibility for CNNs to predict the entanglement entropy for a range of magnetic models. We obtained two-point correlators, structure factors and entanglement measures from tensor network (TN) calculations, which allowed us to get finite-temperature data for system sizes larger than it is possible using exact diagonalization. Then, observables were used as an input and entanglement measure as an output for CNNs. We showed that CNNs can reproduce the entanglement landscape (as a function of inverse temperature β and external magnetic field h) of a single XY model by using a fraction of the data as a training set. By using multiple training strategies, we have shown that CNNs are learning specifically well when trained on border cases (low β , high β or low h , high h) or evenly separated data. Then, we have also shown that learning entanglement is possible between two different Hamiltonians. CNN can predict important features of the entanglement landscape of one model, while being trained on another. For both of those tasks we found that although static two-point correlators and static structure factors contain the same amount of information, the particular CNN architecture we employed performs better when trained on the two-point correlators. When CNN was trained on time-dependent observables however, the opposite was true. The dynamical structure factors needed smaller ratios of the training versus whole dataset to achieve good predictions for the entanglement. We approximated that this ratio is around 3% for dynamical structure factors and 6% for dynamical correlators. Finally, we showed how neurons in CNNs activate when determining two different phases of a magnetic material from scattering functions and learned that it looks at the important features of each corresponding phase.

8.1 Future work

The ML methods will almost certainly find more applications in the study of condensed matter systems. While this work mostly focused on the experimental data analysis, there has been many more ideas developed on how the field of ML and quantum physics can interact in the future. Especially worth mentioning here are the recent techniques that combine ML with TNs [129, 63], as both were used in obtaining the results of this thesis. I hope that the work described in this dissertation will motivate further research into ML methods in magnetic materials. As a final note, I would like to describe possible directions of future work that can be undertaken for each chapter that contained original work.

The successful application of PCA to muon spectroscopy data suggest that the technique could be a possible addition to other established analytical tools for that field. It would be especially helpful for studying materials for which functions like Kubo-Toyabe or an exponential decay do not describe the interactions between magnetic field of the probe and muons. Since its performance increases with more data added, an integration with the database of previous experiments could vastly improve on the method, and therefore be an interesting project to take on. Another extension would be to replace PCA with autoencoders (described here briefly in Chapter 4), which has an advantage of detecting nonlinear correlations in the data as well.

The logical follow up from the work described in Chapter 6 is to actually develop a

method of finding the ground state (or thermal state) of a material from neutron-scattering functions. This would possibly require finding an algorithm that would minimise distance between true scattering function and one that is obtained from a trial state, since we know that there is a one-to-one correspondence between those two quantities. Once a computational method of quantum state tomography is established, a possible experimental confirmation could involve testing the technique on cold atom lattices, for which the parameters of the Heisenberg-like Hamiltonian could be changed easily. One could also check the fitness landscape's resistance to experimental noise by introducing Gaussian noise to the target state.

There are many possible directions from the results of entanglement learning described in Chapter 7. For example drawing random samples of XYZ Hamiltonians with different parameters (γ, Δ) , choosing to train CNN on just a few of them and see how well it can predict other random Hamiltonians. One could also explore other types of magnetic Hamiltonians for entanglement learning. We also have not introduced any kind of error analysis (how well CNN can train with the introduction of noise), which would be useful if the method should be introduced to study experimental data. Following that, the application of the method to real life experiment, to detect entanglement entropy directly from the data and compare it to some other entanglement measure (like I-tangle, concurrence or quantum fisher information), would make the method more applicable as a useful tool of analysis. As a final potential branch of inquiry, rather than predicting the entanglement entropy, the CNN could be trained to predict entanglement spectrum.

Bibliography

- [1] T. Tula, G. Möller, J. Quintanilla, S. R. Giblin, A. D. Hillier, E. E. McCabe, S. Ramos, D. S. Barker, and S. Gibson, “Machine learning approach to muon spectroscopy analysis,” *Journal of Physics: Condensed Matter*, vol. 33, p. 194002, 5 2021. (document), 5.3
- [2] T. Tula, G. Möller, J. Quintanilla, S. R. Giblin, A. D. Hillier, E. E. McCabe, S. Ramos, D. S. Barker, and S. Gibson, “Joint machine learning analysis of muon spectroscopy data from different materials,” *Journal of Physics: Conference Series*, vol. 2164, p. 012018, 3 2022. (document), 5.3, 5.3.1
- [3] T. Tula, J. Quintanilla, and G. Möller, “Fitness landscape for quantum state tomography from neutron scattering,” *arXiv:2412.10502*, 2024. (document), 3
- [4] ThingLab, “Writeful.” <https://www.writefull.com/>, 2016. (document)
- [5] U. Schollwöck, “The density-matrix renormalization group in the age of matrix product states,” *Annals of Physics*, vol. 326, no. 1, pp. 96–192, 2011. (document), 3.1, 3.1.3, 3.1.4, 3.1.4, 3.5
- [6] J. Hauschild and F. Pollmann, “Efficient numerical simulations with Tensor Networks: Tensor Network Python (TeNPy),” *SciPost Phys. Lect. Notes*, p. 5, 2018. (document), 3.1, 3.7
- [7] H. H. Sung, S. Y. Chou, K. J. Syu, and W. H. Lee, “The antipressure effect of Cu on Tc in superconducting $\text{La}(\text{Ni}_{1-x}\text{Cu}_x)\text{C}_2$ ($0 \leq x \leq 0.25$),” *Journal of Physics Condensed Matter*, vol. 20, 4 2008. (document), 5.8, 5.10
- [8] B. D. Coles, A. D. Hillier, F. C. Coomer, N. C. Bristowe, S. Ramos, and E. E. McCabe, “Spin interactions and magnetic order in the iron oxychalcogenides $\text{BaFe}_2\text{Q}_2\text{O}$ (Q= S and Se),” *Physical Review B*, vol. 100, 7 2019. (document), 5.8, 5.10
- [9] Z. Shan, P. K. Biswas, S. K. Ghosh, T. Tula, A. D. Hillier, D. Adroja, S. Cottrell, G.-H. Cao, Y. Liu, X. Xu, Y. Song, H. Yuan, and M. Smidman, “Muon spin relaxation study of the layered kagome superconductor CsV_3Sb_5 ,” *Phys. Rev. Research*, vol. 4, p. 033145, 2022. (document), 5.3, 5.11, 5.3.2
- [10] P. O. Scherer, *Computational Physics: Simulation of Classical and Quantum Systems*. Springer, 2010. 1.1, 6.6
- [11] G. Carleo and M. Troyer, “Solving the quantum many-body problem with artificial neural networks,” *Science*, vol. 355, no. 6325, pp. 602–606, 2017. 1.1, 4, 4.3, 4.3, 6.6
- [12] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt, “Proposed experiment to test local hidden-variable theories,” *Physical Review Letters*, vol. 23, 1969. 2, 2.1.3, 2.1.3

- [13] S. J. Freedman and J. F. Clauser, “Experimental test of local hidden-variable theories,” *Physical Review Letters*, vol. 28, 1972. 2, 2.1.2, 2.1.4
- [14] S. P. Neumann, A. Buchner, L. Bulla, M. Bohmann, and R. Ursin, “Continuous entanglement distribution over a transnational 248 km fiber link,” *Nature Communications*, vol. 13, 2022. 2
- [15] J. Joshi, M. Alimuddin, T. S. Mahesh, and M. Banik, “Experimental verification of many-body entanglement using thermodynamic quantities,” *Physical Review A*, vol. 109, 2024. 2
- [16] B. M. Terhal, “Detecting quantum entanglement,” *Theoretical Computer Science*, vol. 287, 2002. 2
- [17] L. Amico, R. Fazio, A. Osterloh, and V. Vedral, “Entanglement in many-body systems,” *Reviews of Modern Physics*, vol. 80, 2008. 2, 2.2.1, 2.2.2, 5
- [18] A. Einstein, M. Born, and H. Born, *The Born-Einstein Letters: Correspondence Between Albert Einstein and Max and Hedwig Born from 1916-1955, with Commentaries by Max Born*. Macmillan, 1971. 2.1.1
- [19] A. Einstein, B. Podolsky, and N. Rosen, “Can quantum-mechanical description of physical reality be considered complete?,” *Physical Review*, vol. 47, 1935. 2.1.1
- [20] D. W. Belousek, “Einstein’s 1927 unpublished hidden-variable theory: Its background, context and significance,” *Studies in History and Philosophy of Science Part B: Studies in History and Philosophy of Modern Physics*, vol. 27, 1996. 2.1.2
- [21] M. a. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. 2011. 2.1.3
- [22] A. Aspect, P. Grangier, and G. Roger, “Experimental tests of realistic local theories via bell’s theorem,” *Physical Review Letters*, vol. 47, 1981. 2.1.4
- [23] A. Aspect, J. F. Clauser, and A. Zeilinger, “The nobel prize in physics 2022,” *Nobel Prize Official Website*, 2022. 2.1.4
- [24] G. González-García, R. Trivedi, and J. I. Cirac, “Error Propagation in NISQ Devices for Solving Classical Optimization Problems,” *PRX Quantum*, vol. 3, p. 040326, Dec 2022. 2.2
- [25] D. Bouwmeester, J. W. Pan, K. Mattle, M. Eibl, H. Weinfurter, and A. Zeilinger, “Experimental quantum teleportation,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 356, 1998. 2.2
- [26] J. Yin, Y. H. Li, S. K. Liao, M. Yang, Y. Cao, L. Zhang, J. G. Ren, W. Q. Cai, W. Y. Liu, S. L. Li, R. Shu, Y. M. Huang, L. Deng, L. Li, Q. Zhang, N. L. Liu, Y. A. Chen, C. Y. Lu, X. B. Wang, F. Xu, J. Y. Wang, C. Z. Peng, A. K. Ekert, and J. W. Pan, “Entanglement-based secure quantum cryptography over 1,120 kilometres,” *Nature*, vol. 582, 2020. 2.2
- [27] L. Amico, R. Fazio, A. Osterloh, and V. Vedral, “Entanglement in many-body systems,” *Reviews of Modern Physics*, vol. 80, 2008. 2.2

- [28] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki, “Quantum entanglement,” *Reviews of Modern Physics*, vol. 81, pp. 865–942, 6 2009. 2.2
- [29] J. Eisert, “Entanglement and tensor network states,” *arXiv:1308.3318*, 2013. 2.2
- [30] H. Li and F. D. Haldane, “Entanglement spectrum as a generalization of entanglement entropy: Identification of topological order in non-abelian fractional quantum hall effect states,” *Physical Review Letters*, vol. 101, 2008. 3
- [31] M. Ozawa and N. Javerzat, “Perspective on physical interpretations of rényi entropy in statistical mechanics,” *EPL*, vol. 147, 7 2024. 2.2.1
- [32] L. Amico, A. Osterloh, F. Plastina, R. Fazio, and G. M. Palma, “Dynamics of entanglement in one-dimensional spin systems,” *Physical Review A - Atomic, Molecular, and Optical Physics*, vol. 69, 2004. 2.2.2, 2.2.2, 2.2.3
- [33] T. Roscilde, P. Verrucchi, A. Fubini, S. Haas, and V. Tognetti, “Entanglement and factorized ground states in two-dimensional quantum antiferromagnets,” *Physical Review Letters*, vol. 94, 2005. 2.2.2, 2.2.3
- [34] V. Coffman, J. Kundu, and W. K. Wootters, “Distributed entanglement,” *Physical Review A - Atomic, Molecular, and Optical Physics*, vol. 61, 2000. 2.2.3
- [35] T. J. Osborne and F. Verstraete, “General monogamy inequality for bipartite qubit entanglement,” *Physical Review Letters*, vol. 96, 2006. 2.2.3
- [36] T. L. Wang, L. N. Wu, W. Yang, G. R. Jin, N. Lambert, and F. Nori, “Quantum fisher information as a signature of the superradiant quantum phase transition,” *New Journal of Physics*, vol. 16, 2014. 2.2.4
- [37] P. Hyllus, W. Laskowski, R. Kriskchek, C. Schwemmer, W. Wiczorek, H. Weinfurter, L. Pezzé, and A. Smerzi, “Fisher information and multiparticle entanglement,” *Physical Review A - Atomic, Molecular, and Optical Physics*, vol. 85, 2012. 2.2.4
- [38] P. Hauke, M. Heyl, L. Tagliacozzo, and P. Zoller, “Measuring multipartite entanglement through dynamic susceptibilities,” *Nature Physics*, vol. 12, 2016. 2.2.4, 2.2.4
- [39] J. Lambert and E. S. Sørensen, “Estimates of the quantum fisher information in the $s=1$ antiferromagnetic heisenberg spin chain with uniaxial anisotropy,” *Physical Review B*, vol. 99, 2019. 2.2.4, 2.2.4
- [40] M. S. Sarandy, T. R. D. Oliveira, and L. Amico, “Quantum discord in the ground state of spin chains,” *International Journal of Modern Physics B*, vol. 27, 2013. 2.3
- [41] S. Campbell, J. Richens, N. L. Gullo, and T. Busch, “Criticality, factorization, and long-range correlations in the anisotropic xy model,” *Physical Review A - Atomic, Molecular, and Optical Physics*, vol. 88, 2013. 2.3
- [42] O. Marty, M. Epping, H. Kampermann, D. Bruß, M. B. Plenio, and M. Cramer, “Quantifying entanglement with scattering experiments,” *Physical Review B - Condensed Matter and Materials Physics*, vol. 89, 2014. 2.3
- [43] S. R. White, “Physical review letters density matrix formulation for quantum renormalization groups,” vol. 69, 1992. 3

- [44] R. Orús, “A practical introduction to tensor networks: Matrix product states and projected entangled pair states,” *Annals of Physics*, vol. 349, 2014. 3, 3.1, 3.1.4
- [45] F. Arute, K. Arya, and R. B. et al., “Quantum supremacy using a programmable superconducting processor,” *Nature*, vol. 574, 2019. 3
- [46] T. Ayral, T. Louvet, Y. Zhou, C. Lambert, E. M. Stoudenmire, and X. Waintal, “Density-matrix renormalization group algorithm for simulating quantum circuits with a finite fidelity,” *PRX Quantum*, vol. 4, 2023. 3
- [47] K. G. Wilson, “The renormalization group: Critical phenomena and the kondo problem,” *Reviews of modern physics*, vol. 47, no. 4, p. 773, 1975. 3.1
- [48] S. R. White, “Density-matrix algorithms for quantum renormalization groups,” *Physical Review B*, vol. 48, 1993. 3.1
- [49] R. M. Wald, *General relativity*. University of Chicago press, 2010. 1
- [50] I. Affleck, T. Kennedy, E. H. Lieb, and H. Tasaki, “Rigorous results on valence-bond ground states in antiferromagnets,” *Physical Review Letters*, vol. 59, 1987. 3.1.3
- [51] M. Zwolak and G. Vidal, “Mixed-state dynamics in one-dimensional quantum lattice systems: A time-dependent superoperator renormalization algorithm,” *Physical Review Letters*, vol. 93, 11 2004. 3.3
- [52] E. M. Stoudenmire and S. R. White, “Minimally entangled typical thermal state algorithms,” *New Journal of Physics*, vol. 12, 5 2010. 3.3
- [53] C. Karrasch, J. H. Bardarson, and J. E. Moore, “Reducing the numerical effort of finite-temperature density matrix renormalization group calculations,” *New Journal of Physics*, vol. 15, 2013. 3.3.2, 3.4.3, 7.1.3
- [54] J. Hauschild, E. Leviatan, J. H. Bardarson, E. Altman, M. P. Zaletel, and F. Pollmann, “Finding purifications with minimal entanglement,” *Physical Review B*, vol. 98, 2018. 3.3.2, 3.4.3, 7.1.3, 4
- [55] G. Vidal, “Efficient simulation of one-dimensional quantum many-body systems,” *Physical Review Letters*, vol. 93, 2004. 3.4.1
- [56] B. C. Hall and B. C. Hall, *Lie groups, Lie algebras, and representations*. Springer, 2013. 3.4.1
- [57] M. P. Zaletel, R. S. Mong, C. Karrasch, J. E. Moore, and F. Pollmann, “Time-evolving a matrix product state with long-ranged interactions,” *Physical Review B - Condensed Matter and Materials Physics*, vol. 91, 4 2015. 3.4.2
- [58] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006. 4
- [59] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., “Imagenet large scale visual recognition challenge,” *International journal of computer vision*, vol. 115, pp. 211–252, 2015. 4

- [60] N. Fatima, A. S. Imran, Z. Kastrati, S. M. Daudpota, and A. Soomro, "A systematic literature review on text generation using deep neural network models," *IEEE Access*, vol. 10, pp. 53490–53503, 2022. 4
- [61] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019. 4
- [62] K. Roose, "An A.I. Generated Picture Won an Art Prize. Artists Aren't Happy.," *New York Times*, 2022. 4
- [63] E. M. Stoudenmire and D. J. Schwab, "Supervised learning with tensor networks," in *Advances in Neural Information Processing Systems*, 2016. 4, 8.1
- [64] N. Irikura and H. Saito, "Neural-network quantum states at finite temperature," *Physical Review Research*, vol. 2, 2020. 4
- [65] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.", 2022. 4.1
- [66] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of Mathematical Biophysics*, vol. 5, 1943. 4.2
- [67] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, 1958. 4.2, 4.2.1
- [68] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition," *Biometrika*, vol. 71, no. 599-607, p. 6, 1986. 4.2, 4.2.2, 4.2.2, 4.2.2
- [69] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. V. Essen, A. A. Awwal, and V. K. Asari, "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, 2019. 4.2
- [70] S. Sonoda and N. Murata, "Neural network with unbounded activation functions is universal approximator," *Applied and Computational Harmonic Analysis*, vol. 43, 2017. 4.2, 4.2.1
- [71] D. O. Hebb, *The organization of behavior: A neuropsychological theory*. John Wiley & Sons, 1949. 4.2.1
- [72] M. Minsky and S. A. Papert, *Perceptrons, reissue of the 1988 expanded edition with a new foreword by Léon Bottou: an introduction to computational geometry*. MIT press, 2017. 4.2.1
- [73] L. Lu, Y. Shin, Y. Su, and G. E. Karniadakis, "Dying ReLU and initialization: Theory and numerical examples," *Communications in Computational Physics*, vol. 28, 2020. 6
- [74] A. Senior, G. Heigold, M. Ranzato, and K. Yang, "An empirical study of learning rates in deep neural networks for speech recognition," 2013. 4.2.2

- [75] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *Ussr computational mathematics and mathematical physics*, vol. 4, no. 5, pp. 1–17, 1964. 4.2.2
- [76] N. Zhang, S. Ding, J. Zhang, and Y. Xue, "An overview on restricted boltzmann machines," *Neurocomputing*, vol. 275, 2018. 4.3
- [77] S. Sorella, M. Casula, and D. Rocca, "Weak binding between two aromatic rings: Feeling the van der waals attraction by quantum monte carlo methods," *Journal of Chemical Physics*, vol. 127, 2007. 9
- [78] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010. 4.4.1
- [79] R. B. Cattell, "The scree test for the number of factors," *Multivariate Behavioral Research*, vol. 1, 1966. 4.4.1
- [80] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in *Proceedings of ICML workshop on unsupervised and transfer learning*, pp. 37–49, JMLR Workshop and Conference Proceedings, 2012. 4.4.1
- [81] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural networks*, vol. 2, no. 6, pp. 459–473, 1989. 4.4.1
- [82] D. Marr and T. Poggio, "From understanding computation to understanding neural circuitry," tech. rep., USA, 1976. 4.4.2
- [83] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, p. 106, 1962. 4.4.2
- [84] B. Blumenfeld, D. Bibitchkov, and M. Tsodyks, "Neural network model of the primary visual cortex: from functional architecture to lateral connectivity and back," *Journal of computational neuroscience*, vol. 20, pp. 219–241, 2006. 4.4.2
- [85] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, 1980. 4.4.2
- [86] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, 1998. 4.4.2
- [87] S. H. Neddermeyer and C. D. Anderson, "Note on the nature of cosmic-ray particles," *Physical Review*, vol. 51, no. 10, p. 884, 1937. 5.1
- [88] S. L. Lee, R. Cywinski, and S. Kilcoyne, *Muon science: Muons in physics, chemistry and materials*, vol. 51. CRC press, 1999. 5.1, 5.1, 5.1
- [89] A. Yaouanc and P. D. De Reotier, *Muon spin rotation, relaxation, and resonance: applications to condensed matter*. No. 147, Oxford University Press, 2011. 5.1, 5.1, 1
- [90] Y. Kuno and Y. Okada, "Muon decay and physics beyond the standard model," *Reviews of Modern Physics*, vol. 73, 2001. 5.1

- [91] J. Bailey, K. Borer, F. Combley, H. Drumm, F. Krienen, F. Lange, E. Picasso, W. Von Ruden, F. Farley, J. Field, *et al.*, “Measurements of relativistic time dilatation for positive and negative muons in a circular orbit,” *Nature*, vol. 268, no. 5618, pp. 301–305, 1977. 5.1
- [92] B. Rossi and D. B. Hall, “Variation of the rate of decay of mesotrons with momentum,” Feb 1941. 5.1
- [93] R. Kadono, J. Imazato, T. Matsuzaki, K. Nishiyama, K. Nagamine, T. Yamazaki, D. Richter, and J.-M. Welter, “Quantum diffusion of positive muons in copper,” *Phys. Rev. B*, vol. 39, pp. 23–41, 1989. 5.1
- [94] I. J. Onuorah, P. Bonfã, and R. De Renzi, “Muon contact hyperfine field in metals: A dft calculation,” *Phys. Rev. B*, vol. 97, p. 174414, 2018. 5.1
- [95] A. Tsuruta, S. Yukawa, and K. Miyake, “Intrinsic angular momentum and intrinsic magnetic moment of chiral superconductor on two-dimensional square lattice,” *Journal of the Physical Society of Japan*, vol. 84, no. 9, p. 094712, 2015. 5.1, 5.2.3
- [96] N. R. Draper and H. Smith, *Applied regression analysis*, vol. 326. John Wiley & Sons, 1998. 5.2.1
- [97] L. Delchambre, “Weighted principal component analysis: a weighted covariance eigendecomposition approach,” *Monthly Notices of the Royal Astronomical Society*, vol. 446, pp. 3545–3555, 12 2014. 5.2.1
- [98] S. K. Ghosh, M. Smidman, T. Shang, J. F. Annett, A. D. Hillier, J. Quintanilla, and H. Yuan, “Recent progress on superconductors with time-reversal symmetry breaking,” *Journal of Physics: Condensed Matter*, vol. 33, no. 3, p. 033001, 2020. 5.2.3
- [99] A. D. Hillier, J. Quintanilla, B. Mazidian, J. F. Annett, and R. Cywinski, “Nonunitary triplet pairing in the centrosymmetric superconductor LaNiGa_2 ,” *Phys. Rev. Lett.*, vol. 109, p. 097001, 2012. 5.2.3
- [100] A. D. Hillier, J. Quintanilla, and R. Cywinski, “Evidence for time-reversal symmetry breaking in the noncentrosymmetric superconductor LaNiC_2 ,” *Phys. Rev. Lett.*, vol. 102, p. 117007, 2009. 5.2.3
- [101] J. Quintanilla, “Relationship between the wave function of a magnet and its static structure factor,” *Physical Review B*, vol. 106, no. 10, p. 104435, 2022. 6, 6.1, 6.1, 6.1, 4, 6.3, 6.4, 6.6, 8
- [102] P. Hohenberg and W. Kohn, “Inhomogeneous Electron Gas,” *Physical Review*, vol. 136, no. 3B, pp. B864–B871, 1964. 6.1
- [103] T. K. Ng, “Generalized Hohenberg-Kohn theorem and effective Hamiltonian of a quantum system,” *Physical Review B*, vol. 44, no. 5, pp. 2407–2409, 1991. 6.1
- [104] B. Swingle and I. H. Kim, “Reconstructing quantum states from local data,” *Physical Review Letters*, vol. 113, no. 26, p. 260501, 2014. 6.1
- [105] M. Cramer, M. B. Plenio, S. T. Flammia, R. Somma, D. Gross, S. D. Bartlett, O. Landon-Cardinal, D. Poulin, and Y.-K. Liu, “Efficient quantum state tomography,” *Nature Communications*, vol. 1, no. 1, p. 149, 2010. 6.1

- [106] E. Chertkov and B. K. Clark, “Computational inverse method for constructing spaces of quantum models from wave functions,” *Physical Review X*, vol. 8, no. 3, p. 031029, 2018. 6.1
- [107] M. Greiter, V. Schnells, and R. Thomale, “Method to identify parent Hamiltonians for trial states,” *Physical Review B*, vol. 98, no. 8, p. 081113, 2018. 6.1
- [108] E. Bairey, I. Arad, and N. H. Lindner, “Learning a local Hamiltonian from local measurements,” *Physical Review Letters*, vol. 122, no. 2, p. 020504, 2019. 6.1
- [109] X.-L. Qi and D. Ranard, “Determining a local Hamiltonian from a single eigenstate,” *Quantum*, vol. 3, p. 159, 2019. 6.1
- [110] A. Anshu, S. Arunachalam, T. Kuwahara, and M. Soleimanifar, “Sample-efficient learning of interacting quantum systems,” *Nature Physics*, vol. 17, no. 8, pp. 931–935, 2021. 6.1
- [111] B. Murta and J. Fernández-Rossier, “One-to-one correspondence between thermal structure factors and coupling constants of general bilinear Hamiltonians,” *Physical Review E*, vol. 105, no. 6, p. L062101, 2022. 6.1, 6.2.1, 6.2.1, 6.6, 8
- [112] J.-G. Liu, L. Mao, P. Zhang, and L. Wang, “Solving quantum statistical mechanics with variational autoregressive networks and quantum circuits,” *Machine Learning: Science and Technology*, vol. 2, no. 2, p. 025011, 2021. 6.2.1
- [113] E. Ising, *Beitrag zur theorie des ferro-und paramagnetismus*. PhD thesis, Grefe & Tiedemann, 1924. 6.2.2
- [114] S.-J. Gu, G.-S. Tian, and H.-Q. Lin, “Ground-state entanglement in the XXZ model,” *Physical Review A*, vol. 71, no. 5, p. 052322, 2005. 6.2.2
- [115] A. Kitaev, “Anyons in an exactly solved model and beyond,” *Annals of Physics*, vol. 321, no. 1, pp. 2–111, 2006. 6.2.2
- [116] C. Broholm, “Neutron scattering, correlation functions & linear response functions.” (Lecture notes, 2010). 6.3, 6.3.1, 8
- [117] S. W. Lovesey, *Theory of neutron scattering from condensed matter*. Oxford University Press, 1984. 6.3.1
- [118] D. S. Sivia, *Elementary scattering theory: for X-ray and neutron users*. Oxford University Press, 2011. 6.3.1
- [119] H. Irons, *Experimental implications of the entanglement transition in clustered quantum materials*. PhD thesis, 2016. 6.3.1, 7
- [120] H. R. Irons, J. Quintanilla, T. G. Perring, L. Amico, and G. Aeppli, “Control of entanglement transitions in quantum spin clusters,” *Phys. Rev. B*, vol. 96, p. 224408, 2017. 6.3.1, 7.1.1, 7.4
- [121] F. Becca and S. Sorella, *Quantum Monte Carlo approaches for correlated systems*. Cambridge University Press, 2017. 6.6

- [122] K. Capelle, “A bird’s-eye view of density-functional theory,” *Brazilian journal of physics*, vol. 36, pp. 1318–1343, 2006. 6.6
- [123] J. Kurmann, H. Thomas, and G. Müller, “Antiferromagnetic long-range order in the anisotropic quantum spin chain,” *Physica A: Statistical Mechanics and its Applications*, vol. 112, 1982. 7.1.1
- [124] B. M. Terhal, M. Horodecki, D. W. Leung, and D. P. DiVincenzo, “The entanglement of purification,” *Journal of Mathematical Physics*, vol. 43, no. 9, pp. 4286–4298, 2002. 7.1.3
- [125] F. Chollet *et al.*, “Keras.” <https://keras.io>, 2015. 7.1.5
- [126] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015. 7.1.5
- [127] V. Pecharsky and K. Gschneidner, “Magnetocaloric effect,” in *Encyclopedia of Condensed Matter Physics* (F. Bassani, G. L. Liedl, and P. Wyder, eds.), pp. 236–244, Oxford: Elsevier, 2005. 6
- [128] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016. 7.3
- [129] W. Huggins, P. Patil, B. Mitchell, K. B. Whaley, and E. M. Stoudenmire, “Towards quantum machine learning with tensor networks,” *Quantum Science and Technology*, vol. 4, 2019. 8.1