Department of Physics and Astronomy

University of Heidelberg

Master thesis

in Physics

submitted by

Felix Behrens

born in Ratzeburg

2020

Neural Network Representation of Quantum Many-body States and Time Evolution in the Transverse Field Ising Model by Positive Operator Valued Measure

This Master thesis has been carried out by Felix Behrens

at the

Kirchhoff Institut für Physik

under the supervision of

Herrn Prof. Thomas Gasenzer

Neuronale Netzwerkrepräsentation von Quantenvielteilchensystemen und Zeitentwicklung im Isingmodell mit transversalem Feld mithilfe von Positive Operator Valued Measure:

Neuronale Netzwerkquantenzustände (NQZ) ziehen viel Aufmerksamkeit an, da sie das Potential haben, als sehr ausdrucksstarker Variationsansatz für Quantenvielteilchensysteme zu dienen. In dieser Arbeit präsentieren wir die Machbarkeit dessen, dass NQZ basierend auf reellen Zahlen, im Gegensatz zu komplexen, unitäre Zeitentwicklung lösen kann. Dafür verwenden wir ein neuronales Netzwerk als generatives Modell, um sowohl Grundzustände als auch hoch verschränkte Zustände darzustellen. Die Zeitentwicklung unter dem Isingmodell mit transversalem Feld (TFI) nach einer plötzlichen Änderung des TFI-Parameters wird mithilfe von 'positive operator valued measure' (POVM) in die Sprache des maschinellen Lernens übersetzt. Wir vergleichen unsere Ergebnisse mit denen des exakten Diagonalisierungsverfahrens und finden heraus, dass die Genauigkeit begrenzt und in unserer Anwendung der NQZ-Löser eventuell instabil ist. Zudem ist der Rechenaufwand bei kleinen Systemgrößen viel größer als beim exakten Diagonalisierungsverfahren.

Neural Network Representation of Quantum Many-body States and Time Evolution in the Transverse Field Ising Model by Positive Operator Valued Measure:

Neural-network quantum states (NQS) attract a lot of attention due to their potential to serve as a very expressive variational ansatz for quantum many-body systems. In this work, we present a proof of principle that NQS based on real numbers, in contrast to complex ones, can solve the unitary time evolution. We use a neural-network as a generative model to represent both ground states and highly entangled ones. The time evolution under the transverse-field Ising (TFI) Hamiltonian after a sudden change of the TFI parameter is transferred into the language of machine learning by the formalism of positive operator valued measure. We compare our results to exact diagonalization and find that precision is limited, that in our application the NQS solver is eventually unstable and that computational cost is much larger than for exact diagonalization for small system sizes.

Contents

1 Introduction

Although quantum mechanics is a century old discipline in physics, it still presents challenging and interesting problems. The Hilbert space description of quantum states is very simple to grasp in terms of its mathematical description of states as vectors and operators as matrices but brings the problem of being very inefficient. The dimension of the Hilbert space scales exponentially with system size. There are two key aspects of the exponential scaling, the first one concerning the quantum state itself and the second one concerning the time evolution. From an experimental point of view, to extract the full information about a generic unknown state, one has to perform exponentially many different measurements for linearly many particles. From the theoretical point of view, one needs exponentially scaling amount of memory to store the state. To determine the time evolution theoretically in a standard straight-forward approach, e.g. from the Schrödinger equation, one has to diagonalize an exponentially large matrix.

To circumvent this issue, there are multiple approaches to deal with the exponential scaling with system size, but all efficient approximation schemes turn out to struggle in different regimes. We want to state just a few.

Quantum Monte-Carlo methods, which sample a finite number of physically relevant configurations, often suffer from the sign problem when approximating infinite sums with positive and negative contributions in the classical representation of quantum states [Troyer and Wiese, 2005].

One-dimensional quantum spin systems with little entanglement can be very well captured by simulation methods based on matrix product states (MPS), such as the time-dependent density-matrix renormalization group (tDMRG) approach [Schollwöck, 2011]. Especially at quantum critical regimes where correlation lengths diverge, this method scales up its resources exponentially in system size.

Semi-classical methods like the discrete truncated Wigner-approximation show good agreement to the analytical solution when applying a sudden change to the transverse field Ising (TFI) model Hamiltonian, a quench, to the quantum critical regime. But they suffer from instabilities for long time evolution and deviations for quenches to intermediate distances from the quantum critical point [Czischek et al., 2018b].

Dimensional reduction and feature extraction known from machine learning [Hinton and Salakhutdinov, 2006] can be applied to wave functions [Carleo and Troyer, 2017]. Machine learning studies algorithms and statistical models that computers use to perform tasks without explicit instructions [Bishop, 2006]. These statistical models are designed to approximate high dimensional functions. The neural-network quantum state (NQS) is a graph that calculates the corresponding phase and amplitude of a wave function with a chosen number of internal parameters for exponentially many spin configurations. One can derive a learning scheme following the Schrödinger equation on the basis of feedback from variational principles [Carleo and Troyer, 2017].

A study of the regimes of validity for NQS [Czischek et al., 2018a] shows that quenches to the vicinity of the quantum critical point of the TFI model require a strongly increased number of network parameters. Also deviations to the exact time evolution are of similar size as semi-classical approaches.

A representation of quantum many-body states as real probabilities that is also valid for mixed states can be formulated through the connection between measurement probabilities and the density matrix, the positive operator valued measure [Carrasquilla et al., 2019b]. A description based on real probabilities has the advantage that standard machine learning algorithms can be applied to represent these states. Very recently, the application of one- and two-qubit quantum gates have been presented [Carrasquilla et al., 2019a].

The missing link and consequent next step is a machine learning solver based on real probabilities of the time evolution, which easily generalizes to open and mixed systems.

In this thesis the goal is a proof of principal for a synthesis of neural network quantum states based on positive real numbers and the unitary time evolution. Therefore we investigate a numerical method from machine learning, the Restricted Boltzmann Machine (RBM) [Smolensky, 1986], which is an especially simple architecture of neural networks, to approximate quantum states and exploit dimension reduction of the parameter space. We chose a real positive representation of density matrices in order to account for mixed states, circumvent the sign problem and to profit from standard learning schemes, especially the Contrastive Divergence algorithm (CD). To bring a general density matrix in the desired form of a real positive function, we must first apply the method of positive operator valued measure (POVM). The resulting function is then the probability of a set of measurement outcomes which can be used equivalently to the density matrix. This probability distribution determines the likelihood of any given measurement and can be represented by standard machine learning graphs like the RBM. The POVM description of the unitary time evolution under any Hamiltonian for the resulting probability density then arises quite naturally. It is an exact mapping and can be integrated through step-wise training of the RBM and sampling the distribution of the next time step. We observe that, in order to approximately solve the POVM equation of motion, large sample sizes are needed due to statistical sampling and learning errors.

This thesis is structured in the following way. First, we want to investigate the information content of a many-body quantum state. By recapitulating some basics of quantum mechanics (Section 3), we will introduce the concept of entanglement and state reconstruction from subsystems. This motivates POVM which is one of the two major concepts of this work, described by the following Section 4. The second major concept, RBM, will then be introduced and its applications discussed (Section 5). Bringing these two concepts together with the equation of motion creates a formalism for the time evolution of the neural-network representation of many-body quantum states based on positive real numbers (Section 6). For illustration, the results will be compared to exact diagonalization for small system sizes (Section 8).

2 Notation

Throughout this thesis we will use some standard notation.

The Pauli-matrices are

$$\begin{aligned}
\sigma &= (\sigma_x, \sigma_y, \sigma_z), \\
\sigma_x &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \\
\sigma_y &= \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \\
\sigma_z &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.
\end{aligned} \tag{2.0.1}$$

The TFI Hamiltonian with nearest-neighbour interaction in one spatial dimension is

$$H_{TFI} = -\sum_i \sigma_z^i \sigma_z^{i+1} - h_f \sum_i \sigma_x^i, \tag{2.0.2}$$

where the sum over i runs over all particles of the system. Boundary conditions are periodic, i.e. the first particle is nearest-neighbour to the last one. h_f is the transverse magnetic field, the parameter of the model. If this parameter is abruptly changed, this is called a quench and exhibits non-trivial time evolution.

We use natural units with Planck's constant $\hbar = 1$.

3 Information Content of a State

The information content of a state is the minimal set of parameters to fully represent it [Brody and Hughston, 2000]. RBMs are often used for dimension reduction of the parameter space. As a first application, we just want to represent some quantum states. Therefore we need to understand how much information is stored in certain quantum states. The standard way to do so is to investigate bipartite entanglement.

3.1 Bipartite Entanglement of Pure States

A bipartition of a closed system is a theoretical division into two subsystems A and B. If the Hilbert space of the whole state is $\mathcal{H}$, one can always find a basis such that it can be decomposed according to the bipartition $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$. $|\Psi_{AB}\rangle$ denotes a state in $\mathcal{H}$, a state in $\mathcal{H}_A$ is written as $|\Psi_A\rangle$ and analogue for subsystem B. In general, $|\Psi_{AB}\rangle$ is a linear combination of states in $\mathcal{H}_A$ and $\mathcal{H}_B$, thus

$$|\Psi_{AB}\rangle = \sum_{i,j} c_{i,j} \, |\Psi_A\rangle_i \, |\Psi_B\rangle_j \, . \tag{3.1.1}$$

In the special case in which only one coefficient $c_{i,j}$ is non-zero in Equation 3.1.1, the state is called separable. Separability is a defining quantity of entanglement: If the wave function is not separable, system A and system B are entangled.
As an example, let us take two spin-$\frac{1}{2}$ states:

$$\begin{aligned} |\Psi_1\rangle &= \frac{1}{\sqrt{2}} \left(|\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle \right) \\ |\Psi_2\rangle &= \frac{1}{2} \left(|\uparrow\uparrow\rangle + |\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle + |\downarrow\downarrow\rangle \right) . \end{aligned} \tag{3.1.2}$$

If not denoted otherwise, $|\uparrow\rangle$ and $|\downarrow\rangle$ are the eigenvectors of the Pauli-z-matrix σ_z defined in Equation 2.0.1:

$$\sigma_z \left|\uparrow\right\rangle = \left|\uparrow\right\rangle , \quad \sigma_z \left|\downarrow\right\rangle = - \left|\downarrow\right\rangle . \tag{3.1.3}$$

In the spirit of matrix product states (MPS) [Schollwöck, 2011], we investigate the separability of the states in Equation 3.1.2 by rewriting them according to the Schmidt decomposition. It states that one can always find two basis rotations for the bipartition such that Equation 3.1.1 can be written as $\sum_{i=1,2} s_i \left|\tilde{\Psi}_A^i \tilde{\Psi}_B^i\right\rangle$, where the tilde denotes the state in the rotated basis.

$$\begin{aligned}
\left|\Psi_1\right\rangle &= \sum_{i,j} c_{i,j} \left|\Psi_A\right\rangle_i \left|\Psi_B\right\rangle_j \\
&= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} \left|\uparrow\right\rangle \\ \left|\downarrow\right\rangle \end{bmatrix}_A \otimes \begin{bmatrix} \left|\uparrow\right\rangle \\ \left|\downarrow\right\rangle \end{bmatrix}_B \right) \\
&= \frac{1}{\sqrt{2}} \begin{bmatrix} \left|\uparrow\right\rangle & \left|\downarrow\right\rangle \end{bmatrix}_A \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \left|\uparrow\right\rangle & \left|\downarrow\right\rangle \end{bmatrix}_B \\
&= \sum_{i=1,2} s_i \left|\Psi_A^i \Psi_B^i\right\rangle \\
&= \frac{1}{\sqrt{2}} \left|\uparrow\uparrow\right\rangle + \frac{1}{\sqrt{2}} \left|\downarrow\downarrow\right\rangle ,
\end{aligned} \tag{3.1.4}$$

with $s_1 = s_2 = \frac{1}{\sqrt{2}}$. From the first to the second line, we made the coefficients $c_{i,j}$ explicit. In a next step we reshaped the dot product according to the bipartition and finally notice that only diagonal elements contribute. The Schmidt-rank is two for this state, meaning that the diagonal matrix connecting the bipartition has two non-zero elements. Thus, independent of the basis, there is entanglement between the bipartition. As a second example consider $|\Psi_2\rangle$:

$$\begin{aligned}|\Psi_2\rangle &= \sum_{i,j} c_{i,j} \left|\Psi_A\right\rangle_i \left|\Psi_B\right\rangle_j \\ &= \frac{1}{2}\begin{bmatrix}1 & 1 & 1 & 1\end{bmatrix}\left(\begin{bmatrix}|\uparrow\rangle \\ |\downarrow\rangle\end{bmatrix}_A \otimes \begin{bmatrix}|\uparrow\rangle \\ |\downarrow\rangle\end{bmatrix}_B\right) \\ &= \frac{1}{2}\begin{bmatrix}|\uparrow\rangle & |\downarrow\rangle\end{bmatrix}_A \begin{bmatrix}1 & 1 \\ 1 & 1\end{bmatrix}\begin{bmatrix}|\uparrow\rangle \\ |\downarrow\rangle\end{bmatrix}_B\end{aligned} \tag{3.1.5}$$

Applying singular value decomposition to the matrix in this case yields

$$\begin{aligned}\frac{1}{2}\begin{bmatrix}1 & 1 \\ 1 & 1\end{bmatrix} &= \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1 \\ 1 & -1\end{bmatrix}\begin{bmatrix}1 & 0 \\ 0 & 0\end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1 \\ 1 & -1\end{bmatrix} \\ &= USV^\dagger .\end{aligned} \tag{3.1.6}$$

The matrices U and $V^\dagger$ diagonalize the original matrix. The diagonal matrix has one non-zero entry, there is just a product, so there is only one singular value which means that there is no entanglement. U and $V^\dagger$ can now be applied to the bases of the bipartition. Our second example state looks very simple if we define the basis transformation given by U and $V^\dagger$: $(|\uparrow\rangle, |\downarrow\rangle) \to (\frac{1}{\sqrt{2}}(|\uparrow\rangle + |\downarrow\rangle), \frac{1}{\sqrt{2}}(|\uparrow\rangle - |\downarrow\rangle))$ and call the new basis the x-basis, denoted by an index x as it is the eigenvectors of the Pauli-x-matrix:

$$\begin{aligned}|\Psi_2\rangle &= \frac{1}{2}\begin{bmatrix}|\uparrow\rangle & |\downarrow\rangle\end{bmatrix}_A USV^\dagger \begin{bmatrix}|\uparrow\rangle \\ |\downarrow\rangle\end{bmatrix}_B \\ &= \begin{bmatrix}\frac{1}{\sqrt{2}}(|\uparrow\rangle + |\downarrow\rangle) & \frac{1}{\sqrt{2}}(|\uparrow\rangle - |\downarrow\rangle)\end{bmatrix}_A \begin{bmatrix}1 & 0 \\ 0 & 0\end{bmatrix}\begin{bmatrix}\frac{1}{\sqrt{2}}(|\uparrow\rangle + |\downarrow\rangle) \\ \frac{1}{\sqrt{2}}(|\uparrow\rangle - |\downarrow\rangle)\end{bmatrix}_B \\ &= |\uparrow\uparrow\rangle_x .\end{aligned} \tag{3.1.7}$$

Indeed, Ψ_2 is a product state in the x-basis and thus not entangled at all. The method of Schmidt decomposition is powerful because it rotates the basis locally into a coordinate system closest to a product state but leaves the correlations (up to a given order) between partitions unchanged.

3.2 Mixed States and Measuring Subsystems

In the last section, we investigated in two examples how two subsystems can be differently entangled. In this section we want to investigate the information content of the whole system by investigating the states of the subsystems. Therefore we need to generalize our notion of states.

In general states are mixed. A description of mixed states does not only capture the statistical features arising from quantum uncertainty but also includes the classical uncertainties. When writing an ensemble theory of quantum states, this defines the density operator or density matrix for finite Hilbert spaces. This object captures effects from thermal systems, open systems and all kinds of interaction with the environment. Especially when a bipartition of a closed system is considered, the two subsystems might be open if they are entangled. A density matrix can easily be constructed from pure states

$$\rho = \sum_i p_i \left|\Psi_i\right\rangle \left\langle\Psi_i\right| , \tag{3.2.1}$$

where p_i are non-negative coefficients that add up to one. Its interpretation is that the p_i are a classical probability of mixing different states $|\Psi_i\rangle$, thus the name mixed states.

A measurement of an operator O on the system that is described by a density matrix is given by Born's rule

$$\langle O\rangle = \mathrm{Tr}(O\rho), \tag{3.2.2}$$

where $\langle . \rangle$ denotes the expectation value.

To get an intuition for density matrices, let us consider the following two examples.

The first one is the outer product of the maximally entangled state $|\Psi_1\rangle \otimes \langle\Psi_1|$:

$$\rho_1 = \frac{1}{2}\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \tag{3.2.3}$$

the second example is a maximally mixed state with $p_i = \frac{1}{4}$ for all i in Equation 3.2.1

$$\rho_2 = \frac{1}{4}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.2.4}$$

If the density matrix of the whole system is denoted as ρ and we think of a bipartition A and B, the corresponding density matrices are defined as partial traces over the other subsystem:

$$\rho_A = Tr_B(\rho), \quad \rho_B = Tr_A(\rho). \tag{3.2.5}$$

When we look at the subsystems, we find for both cases $\rho_A = \rho_B = \frac{1}{2}\mathbb{I}_{2x2}$. Notably, a general density matrix cannot be reconstructed from its partial traces, as information is lost in the process of taking the trace. More information is needed. To restore the lost information, one can investigate correlation functions according to Equation 3.2.2, where the correlation $\langle\sigma^z\sigma^z\rangle$ denotes the expectation value of $\sigma_z \otimes \sigma_z$:

$$\begin{aligned} \langle\sigma^z\sigma^z\rangle_1 &= \mathrm{Tr}\left(\frac{1}{2}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}\right) = 1, \\ \langle\sigma^z\sigma^z\rangle_2 &= 0. \end{aligned} \tag{3.2.6}$$

In the first line, the first matrix is $\sigma_z \otimes \sigma_z$ in numbers and the second matrix is ρ_1. The index denotes which of the two examples we consider. Because the expectation values are zero, the two-point correlation equals the connected two-point correlation. The latter encodes the statistical correlation and the quantum entanglement. While it is one in the first example, it vanishes for the second one. The first state is maximally entangled whereas the second state is completely random as there is no correlation.

For a general density matrix

$$\rho = \begin{bmatrix} a & b & c & d \\ b^* & f & g & h \\ c^* & g^* & k & l \\ d^* & h^* & l^* & p \end{bmatrix} \tag{3.2.7}$$

the partial traces are given by

$$\begin{aligned} \rho_A = \mathrm{Tr}_B\, \rho &= \begin{bmatrix} a+f & c+h \\ c^*+h^* & k+p \end{bmatrix}, \\ \rho_B = \mathrm{Tr}_A\, \rho &= \begin{bmatrix} a+k & b+l \\ b^*+l^* & f+p \end{bmatrix}. \end{aligned} \tag{3.2.8}$$

If one measures the expectation value of the diagonal operator σ^z on both subsystems and the correlation, one can extract the full diagonal of the two-particle density matrix:

$$\begin{aligned} < \sigma^z_A > &= 2(a+f) - 1 \\ < \sigma^z_B > &= 2(a+k) - 1 \\ < \sigma^z_A \sigma^z_B > &= 1 - 2(f+k) \\ p &= 1 - a - f - k \end{aligned} \tag{3.2.9}$$

Measuring the other Pauli operators and their correlations, one can extract all free parameters of the full density matrix. A more systematic approach is given by positive operator valued measures. In this framework, which will be the topic of the next chapter, a minimal set of informationally complete measurements can be defined. As an extra feature we will demand these operators to be positive in the operator sense i.e. to have exclusively non-negative eigenvalues.

4 Positive Operator Valued Measure (POVM)

In the following section, we will present a method to systematically extract the complete information content of a general state and map it on positive real numbers. As the name already reveals, we will measure positive operators $M^{(a)}$ according to Born's rule. But there is more to it, so let us first consider one particle before generalizing to N particles.

4.1 One Particle

One particle with just one spin$-\frac{1}{2}$ degree of freedom is defined by its density matrix ρ as an element of the Lie algebra $\mathfrak{su}(2)$. This is the vector space of traceless unitary 2x2-matrices together with the commutator '[., .]' and has three generators. Thus, it can be spanned by the Pauli matrices. So, the density matrix is two-dimensional but has three free real parameters. Thus, one can find a three-dimensional representation in form of a vector $\vec{s}_\rho$, i.e. the Bloch representation:

$$\rho(\vec{v}) = \frac{1}{2} \left(\mathbb{I}_{2x2} + \vec{s}_\rho \cdot \vec{\sigma} \right) . \tag{4.1.1}$$

From this representation we know a possible way to construct unitary positive operators $M^{(a)}$ [Carrasquilla et al., 2019b]

$$M^{(a)} = \frac{1}{4} \left(\mathbb{I}_{2x2} + \vec{s}^{(a)} \cdot \vec{\sigma} \right) , \tag{4.1.2}$$

where we defined $a \in \{1, 2, 3, \ldots, n\}$ real valued three component vectors $\vec{s}$. This is not the only way to construct positive operators, just a very intuitive one.

We demand these operators to sum up to the identity

$$\sum_{a=1}^{n} M^{(a)} = \mathbb{I}_{2x2}, \tag{4.1.3}$$

because then the expectation values

$$P(a) = \mathrm{Tr}\left(M^{(a)} \cdot \rho\right) \tag{4.1.4}$$

define a probability distribution $P(a)$, which is normalized, as can be seen from explicit summation:

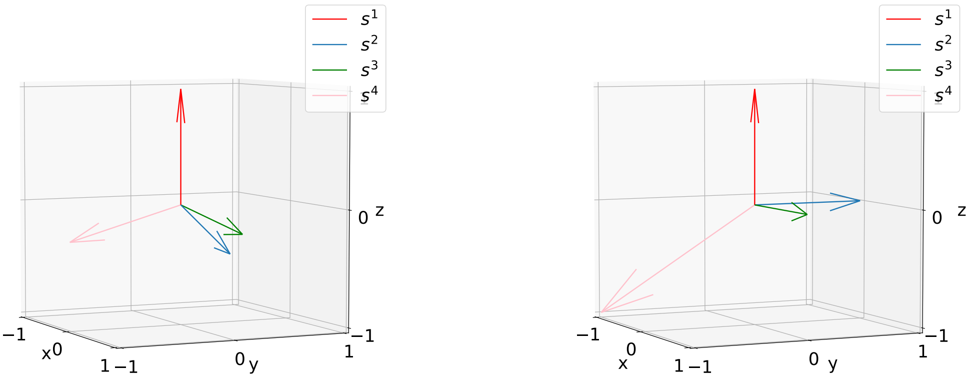
$$\begin{aligned} \sum_{a=1}^{n} P(a) &= \sum_{a} \mathrm{Tr}\left(M^{(a)} \cdot \rho\right) \\ &= \mathrm{Tr}\left(\rho \cdot \sum_{a=1}^{n} M^{(a)}\right) \\ &= \mathrm{Tr}(\rho) \\ &= 1. \end{aligned} \tag{4.1.5}$$

If the density matrix ρ is defined by a spin vector $\vec{s}_\rho$, i.e. $\rho = \frac{1}{2}\left(\mathbb{I}_{2x2} + \vec{s}_\rho \cdot \vec{\sigma}\right)$, the probability turns out to be

$$\begin{aligned} P(a) &= \mathrm{Tr}\big(M^{(a)} \cdot \rho\big) \\ &= \mathrm{Tr}\left(\frac{1}{8}(\mathbb{I}_{2x2} + \vec{s}_\rho \cdot \vec{\sigma})(\mathbb{I}_{2x2} + \vec{s}^{(a)} \cdot \vec{\sigma})\right) \\ &= \frac{1}{4}\left(1 + \vec{s}_\rho \cdot \vec{s}^{(a)}\right), \end{aligned} \tag{4.1.6}$$

where we used that σ is traceless and $\sigma^2 = \mathbb{I}_{2x2}$. It is quite instructive to see in Figure 4.1, that indeed $P(a)$ fulfills the conditions of a normalized probability distribution, i.e. non-negativity as $\vec{s}_\rho$ has maximal length one and the $\vec{s}^{(a)}$ sum up to zero.

The measurement is informationally complete (IC), if the $M^{(a)}$ span the complete vector space. Therefore we need at least four positive operators, as the vector space is three-dimensional and the operators need to fulfill the normalization condition in Equation 4.1.3.



(a) Tetrahedral set vectors defined in Equation 4.1.7

(b) Antenna set vectors defined in Equation 4.1.8

Figure 4.1: Two sets of vectors that each define a POVM measurement set of operators via Equation 4.1.2

Let's demonstrate such a measurement by three examples: the 'tetrahedral'

$$s^1 = (0,0,1), s^2 = (\frac{2\sqrt{2}}{3}, 0, -\frac{1}{3}), s^3 = (-\frac{\sqrt{2}}{3}, \sqrt{\frac{2}{3}}, -\frac{1}{3}), s^4 = (-\frac{\sqrt{2}}{3}, -\sqrt{\frac{2}{3}}, -\frac{1}{3}),$$

$$\begin{aligned} M^1 &= \begin{bmatrix} 0.5 & 0 \\ 0 & 0 \end{bmatrix}, \\ M^2 &= \begin{bmatrix} \frac{1}{6} & \frac{1}{\sqrt{18}} \\ \frac{1}{\sqrt{18}} & \frac{1}{3} \end{bmatrix}, \\ M^3 &= \begin{bmatrix} \frac{1}{6} & \frac{1}{6\sqrt{2}} - \frac{i}{\sqrt{12}} \\ \frac{-1}{6\sqrt{2}} + \frac{i}{\sqrt{12}} & \frac{1}{3} \end{bmatrix}, \\ M^4 &= \begin{bmatrix} \frac{1}{6} & \frac{-1}{6\sqrt{2}} + \frac{i}{\sqrt{12}} \\ \frac{-1}{6\sqrt{2}} - \frac{i}{\sqrt{12}} & \frac{1}{3} \end{bmatrix}, \end{aligned} \tag{4.1.7}$$

the second we dub 'antenna'

$$s^1 = (1,0,0), s^2 =(0,1,0), s^3 = (0,0,1), s^4 = -(1,1,1),$$

$$\begin{aligned}
M^1 &= \frac{1}{4}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \\
M^2 &= \frac{1}{4}\begin{bmatrix} 1 & -i \\ i & 1 \end{bmatrix}, \\
M^3 &= \frac{1}{2}\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \\
M^4 &= \frac{1}{4}\begin{bmatrix} 0 & -1+i \\ -1-i & 2 \end{bmatrix}
\end{aligned} \tag{4.1.8}$$

and the 'Pauli-4', which is constructed from the projections of the eigenvectors of the Pauli operators with positive eigenvalue and the fourth one such that the identity condition 4.1.3 is fulfilled:

$$\begin{aligned}
M^1 &= \frac{1}{3}\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \\
M^2 &= \frac{1}{6}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \\
M^3 &= \frac{1}{6}\begin{bmatrix} 1 & i \\ -i & 1 \end{bmatrix} \\
M^4 &= \frac{1}{6}\begin{bmatrix} 2 & -1-i \\ -1+i & 4 \end{bmatrix}.
\end{aligned} \tag{4.1.9}$$

The first two sets are positive by construction (Equation 4.1.2) as long as the vectors sum up to zero. Drawing the vectors in three dimensions gives the measurement sets their names (see Figure 4.1). The 'Pauli-4' set does not have a representation induced by a set of vectors (Equation 4.1.2). This shows that Equation 4.1.2 defines a subspace in the space of all positive measurement sets, which is not surprising seeing that Equation 4.1.2 fixes the trace to be $\frac{1}{2}$ as an additional constraint.

For later reference, we define the overlap matrices as

$$\begin{aligned} T^{a,a'} &= \mathrm{Tr}\left(M^{(a)} \cdot M^{(a')}\right) \\ &= M^{(a)}_{ij} \cdot M^{(a')}_{ji}, \end{aligned} \tag{4.1.10}$$

where we used the Einstein sum convention to make explicit that the trace is taken over the physical indices i and j of the operator M^a_{ij} and not over the POVM index a. The overlap matrices for the three POVM sets are

$$\begin{aligned} T_{\text{tetrahedral}} &= \frac{1}{12}\begin{bmatrix} 3 & 1 & 1 & 1 \\ 1 & 3 & 1 & 1 \\ 1 & 1 & 3 & 1 \\ 1 & 1 & 1 & 3 \end{bmatrix}, \\ T_{\text{antenna}} &= \frac{1}{8}\begin{bmatrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 1 & 2 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}, \\ T_{\text{Pauli}-4} &= \frac{1}{18}\begin{bmatrix} 2 & 1 & 1 & 2 \\ 1 & 2 & 1 & 2 \\ 1 & 1 & 2 & 2 \\ 2 & 2 & 2 & 12 \end{bmatrix}. \end{aligned} \tag{4.1.11}$$

As we will see in the next section, we need these overlap matrices to be invertible. One can construct POVM bases that do not fulfill this requirement [Carrasquilla et al., 2019b]. They are excluded in this work, the given examples are indeed invertible. There inverses are

$$
\begin{aligned}
T^{-1}_{\text{tetrahedral}} &= \begin{bmatrix} 5 & -1 & -1 & -1 \\ -1 & 5 & -1 & -1 \\ -1 & -1 & 5 & -1 \\ -1 & -1 & -1 & 5 \end{bmatrix}, \\
T^{-1}_{\text{antenna}} &= \begin{bmatrix} 6 & -2 & -2 & 0 \\ -2 & 6 & -2 & 0 \\ -2 & -2 & 6 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}, \\
T^{-1}_{\text{Pauli}-4} &= \begin{bmatrix} 14 & -4 & -4 & -1 \\ -4 & 14 & -4 & -1 \\ -4 & -4 & 14 & -1 \\ -1 & -1 & -1 & 2 \end{bmatrix}.
\end{aligned}
\tag{4.1.12}
$$

4.2 N Particles

A measurement on N qubits on a chain is then defined via the tensor product of the $M^{(a)}$ matrices $M^{(a_1)} \otimes M^{(a_2)} \otimes \ldots M^{(a_N)}$ by the 4^N possible combinations of the four (a) at each site. From now on, whenever we write a without an index, we mean the vector including four components at all sites. With an index i, the four components at site i are meant. This defines the probability distribution

$$P(a) = \text{Tr}\left(M^{(a} \cdot \rho\right), \tag{4.2.1}$$

now with the N-particle density matrix. This relation can be inverted. The reconstruction of the density matrix is defined by the following calculation, where we will multiply a one and recognize Equation 4.2.1:

$$\begin{aligned} P(a) &= P(a')\mathbf{1}_{a',a} \\ &= P(a')T^{-1}_{a',a''}\operatorname{Tr}\Big(M^{a''}M^{a}\Big) \\ &= \operatorname{Tr}\Big(P(a')T^{-1}_{a',a''}M^{a''}M^{a}\Big) \\ &= \operatorname{Tr}\Big(M^{a}P(a')T^{-1}_{a',a''}M^{a''}\Big) \\ &\equiv \operatorname{Tr}(M^{a}\rho), \end{aligned} \tag{4.2.2}$$

such that

$$\begin{aligned} \rho &= \sum_{a,a'} P(a)T^{-1}_{a,a'}M^{(a')} \\ &= \mathbb{E}_{a\sim P(a)}\left(\sum_{a'} T^{-1}_{a,a'}M^{(a')}\right), \end{aligned} \tag{4.2.3}$$

where $a \sim P(a)$ means that a follows $P(a)$ and $\mathbb{E}$ denotes an expectation value.

Operators O can be written as

$$O = \sum_{a} Q_O(a)M^{(a)}, \tag{4.2.4}$$

This relation can be solved for the coefficient Q_O of the operator in the POVM basis. First, we multiply by $M^{(a')}$ and trace out physical indices

$$O_{ij}M^{(a')}_{ji} = \sum_{a} Q_O(a)M^{(a)}_{ij}M^{(a')}_{ji} \tag{4.2.5}$$

and recognize the overlap matrix $T_{a,a'}$ on the right hand side. We multiply by its inverse and arrive at

$$Q_O(a') = \operatorname{Tr}\big(OM^{(a)}\big)\cdot T^{-1}_{a,a'}. \tag{4.2.6}$$

As it turns out when a complete set of POVM and its statistics are given (equivalent to $P(a)$ for large sample sizes), all expectation values can be calculated even without explicitly reconstructing the density matrix:

$$
\begin{aligned}
\mathrm{Tr}(O \cdot \rho) &= \mathrm{Tr}\left(\sum_{a} Q_O(a) M^{(a)} \sum_{a',a''} P(a') T^{-1}_{a',a''} M^{(a'')}\right) \\
&= \sum_{a,a',a''} Q_O(a) P(a') T^{-1}_{a',a''} \, \mathrm{Tr}\left(M^{(a)} M^{(a'')}\right) \\
&= \sum_{a} Q_O(a) P(a) \\
&= \mathbb{E}_{a \sim P(a)}(Q_O(a)).
\end{aligned}
\tag{4.2.7}
$$

Here, $\mathbb{E}_{a \sim P(a)}$ denotes an expectation value over samples a following the probability distribution $P(a)$. All information is stored in the probability distribution $P(a)$.

5 Restricted Boltzmann Machine (RBM)

We use a RBM to approximate a given probability distribution. In this case the RBM is referred to as a generative model. It has polynomially many model parameters $\theta = \{W, c, b\}$. An RBM can be represented by a graph (see Figure 5.1), which can be interpreted as a matrix multiplication. Each node is an entry of a vector whereas each line is an entry of the matrix W. The vectors are referred to as layers. The upper layer is called 'hidden', the lower one 'visible'. As a bilinear form, it defines the so called network energy

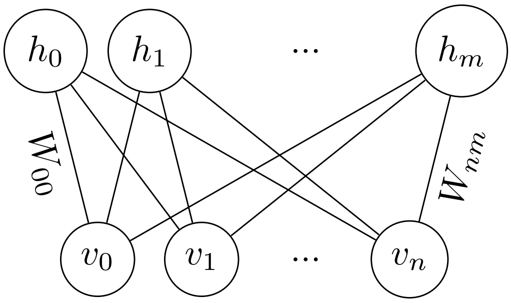


Figure 5.1: RBM graph representing the first term $\mathbf{v} \cdot W \cdot \mathbf{h}$ of the network energy $E(\mathbf{v}, \mathbf{h})$ in Equation 5.0.1, where $\mathbf{v}$ and $\mathbf{h}$ are the so called visible and hidden layer respectively and W the connecting weights. There is an all to all connection between the two layers but no connection within them. Thus, the model is called restricted.

$$E(\mathbf{v}, \mathbf{h}) = -\mathbf{v} \cdot W \cdot \mathbf{h} - \mathbf{c} \cdot \mathbf{v} - \mathbf{b} \cdot \mathbf{h}. \tag{5.0.1}$$

The W is the connection between visible $\mathbf{v}$ and hidden $\mathbf{h}$ layer. On both acts a bias $\mathbf{c}$ and $\mathbf{b}$ respectively, which can be interpreted as local external fields. With this quantity, the energy, at hand, one can define thermodynamic probabilities

$$P(\mathbf{v}) = \frac{1}{Z} \sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h})), \tag{5.0.2}$$

with the partition sum

$$Z = \sum_{\mathbf{h},\mathbf{v}} \exp(-E(\mathbf{v}, \mathbf{h})) \tag{5.0.3}$$

In most cases, the entries of the visible vector (v_i) and in all cases the entries of the hidden vector (h_i) are binary (take values zero or one).

A possible interpretation of the hidden layer is that it represents the microscopic degrees of freedom and, when summed over, give the macroscopic theory with effective interactions between the visible nodes. With this picture in mind one can define the network free energy $F(\mathbf{v})$

$$P(\mathbf{v}) = \frac{1}{Z} \sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h})) = \frac{1}{Z} \exp(-F(\mathbf{v})), \tag{5.0.4}$$

with the partition sum as before

$$Z = \sum_{\mathbf{v}} \exp(-F(\mathbf{v})). \tag{5.0.5}$$

We use that the hidden layer does not have any intra-layer connections so that the sum over $\mathbf{h}$ factorizes and that the hidden units only take binary values, i.e. are zero or one. Thus the sum over the hidden units can be calculated explicitly:

$$\begin{aligned}
\sum_{\mathbf{h}} \exp(\mathbf{v} \cdot W \cdot \mathbf{h} + \mathbf{c} \cdot \mathbf{v} + \mathbf{b} \cdot \mathbf{h}) &= \exp(\mathbf{c} \cdot \mathbf{v}) \sum_{\mathbf{h}} \exp(\mathbf{v} \cdot W + \mathbf{b} \cdot \mathbf{h}) \\
&= \exp(\mathbf{c} \cdot \mathbf{v}) \sum_{\mathbf{h}} \prod_{i=1}^{m} \exp((\mathbf{v} \cdot W + \mathbf{b})_i h_i) \\
&= \exp(\mathbf{c} \cdot \mathbf{v}) \prod_{i=1}^{m} \sum_{h_i=0}^{1} \exp((\mathbf{v} \cdot W + \mathbf{b})_i h_i) \\
&= \exp(\mathbf{c} \cdot \mathbf{v}) \prod_{i=1}^{m} (1 + \exp((\mathbf{v} \cdot W + \mathbf{b})_i)) \,.
\end{aligned} \tag{5.0.6}$$

Then the free energy can be written as

$$\begin{aligned} F(\mathbf{v}) &= -\log\left(\sum_{\mathbf{h}} \exp(-E(\mathbf{v},\mathbf{h}))\right) \\ &= -\mathbf{c}\cdot\mathbf{v} - \sum_j \log(1+\exp(v_i W_{ij} + b_j)). \end{aligned} \tag{5.0.7}$$

Given a set of model parameters θ, we can now easily calculate the probability for any input state $\mathbf{v}$:

$$P(\mathbf{v}) = \frac{1}{Z}\exp(\mathbf{c}\cdot\mathbf{v})\prod_{i=1}^{m}\left(1+\exp((\mathbf{v}\cdot W+\mathbf{b})_i)\right). \tag{5.0.8}$$

5.1 Thermalization

Introduced as a thermal model by defining a Boltzmann distribution of the network energy, why and how does the system converge to its thermal probability distribution? This question is important because we will make statistical approximations of the probability by drawing samples from the RBM. The key insight is, that if we define a Gibbs sampling procedure with the given probability distribution, this is a special case of Metropolis-Hastings which is a special case of a Markov process which guarantees that there is thermal equilibrium. So let us define the needed ingredients one by one. In this subsection we summarize from [Gelman et al., 2013].

5.1.1 Markov Process

A Markov process is a sequence of random variables, also called sample set, in which the distribution of the next sample only depends on the current value. Thus the process is uniquely defined by a positive transition probability $P(v'|v)$ for any two samples v,v' to go from v to v'. A sufficient condition for the existence of a stationary distribution $\Pi(v)$ is detailed balance $\Pi(v')P(v|v') = \Pi(v)P(v'|v)$. The stationary distribution is unique, if the Markov process is ergodic, that is if the number of steps to come back to any point has finite expectation value and non-zero variance.

5.1.2 Metropolis-Hastings

In a Metropolis process, the transition probability $P(v'|v)$ can be written as a product of proposal probability $g(v'|v)$ and acceptance probability $A(v', v)$. It is thus a special case of a Markov process. A new sample is proposed according to $g(v'|v)$, the probability of proposing v' while being at v. The acceptance is often chosen to be $A(v', v) = \min\left(1, \frac{P(v')}{P(v)}\right)$. The algorithm then works the following way:

- start from sample v,
- generate a candidate v' from $g(v'|v)$,
- draw a unitary random number $r \in [0, 1]$,
- if $r \leq A(v', v)$, accept v' as the next sample, else reuse v.

By this scheme, a more probable sample is always added to the sample set but a less probable state only by chance corresponding to the relative occurrence in the target distribution $P(v)$. The fact that the normalization of the target distribution cancels out in the acceptance $A(v', v)$ makes the scheme a powerful tool to approximate a probability distribution whose normalization is difficult to compute.

5.1.3 Gibbs Sampling

Gibbs sampling [Geman and Geman, 1984] introduces an alternating update scheme for the hidden and visible vector respectively, based on the conditional posterior distribution (conditional sampling). It is a special case of a Metropolis-Hastings process for the RBM. The new sample v' is proposed for each entry of the vector v separately (conditioned on the state of all others). As it turns out, for RBMs, the acceptance $A(v', v)$ is always one, thus the new sample is always accepted. Due to the structure of the RBM, which only allows interlayer connections but no intra-layer connections, the conditional probabilities of the visible entries, given one specific hidden vector, are independent of each other. For binary visible units, the conditional probability of taking the value one compared to taking the value zero is then given by the

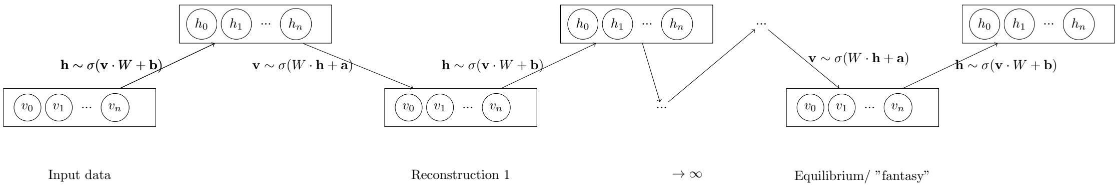

h_0 h_1 ··· h_n
$\mathbf{h} \sim \sigma(\mathbf{v} \cdot W + \mathbf{b})$
$\mathbf{v} \sim \sigma(W \cdot \mathbf{h} + \mathbf{a})$
v_0 v_1 ··· v_n
Input data
Reconstruction 1
$\to \infty$
Equilibrium/ ”fantasy”

Figure 5.2: Gibbs sampling. The RBM gets to its thermal equilibrium when alternately applying Equation 5.1.3 derived in Equation 5.1.2 to the visible and hidden layer. '∼' means that the layer takes new values following the given probability distribution. The first reconstruction is reached after sampling the hidden layer depending on the data and sampling the visible layer depending on the hidden layer.

exponential of the energy difference as the partition sum cancels out:

$$\begin{aligned}\frac{P(v_i = 1|\mathbf{h})}{P(v_i = 0|\mathbf{h})} &= \exp\left(-(E(v_i = 1|\mathbf{h}) - E(v_i = 0|\mathbf{h}))\right) \\ &= \exp(W_{ij}h_j + c_i)\end{aligned} \tag{5.1.1}$$

We use that the two probabilities $P(v_i = 1|\mathbf{h}), P(v_i = 0|\mathbf{h})$ add up to one to express the latter one by the first one. Then we reorder the terms and define the sigmoid function $\sigma(x) = \frac{1}{1+\exp(-x)}$ and the local energy $E_i = -W_{ij}h_j - c_i$:

$$\begin{aligned}\implies P(v_i = 1|\mathbf{h}) &= \exp(W_{ij}h_j + c_i)(1 - P(v_i = 1|\mathbf{h})) \\ \iff P(v_i = 1|\mathbf{h})(1 + \exp(W_{ij}h_j + c_i)) &= \exp(W_{ij}h_j + c_i) \\ \iff P(v_i = 1|\mathbf{h}) &= \frac{1}{1 + \exp(-W_{ij}h_j - c_i)} \\ &\equiv \sigma(-E_i).\end{aligned} \tag{5.1.2}$$

An analogue calculation is true for the conditional probability $P(h_i = 1|\mathbf{v})$ of the hidden layer given the visible one. This defines the proposal probabilities. So the evolution to equilibrium (see Figure 5.2) is given by alternately applying:

$$\begin{aligned}&\mathbf{h} \sim \sigma(\mathbf{v} \cdot W + \mathbf{b}) \\ &\mathbf{v} \sim \sigma(W \cdot \mathbf{h} + \mathbf{c}),\end{aligned} \tag{5.1.3}$$

where $\mathbf{h} \sim \sigma(\dots)$ means that $\mathbf{h}$ takes a value following the probability distribution given by $\sigma(\dots)$. One Gibbs reconstruction is defined as setting the hidden units to one according to the conditional probability $P(h_i = 1|\mathbf{v}) = \sigma(\mathbf{v} \cdot W + \mathbf{b})$ and zero otherwise for all i in the hidden layer and afterwards setting the visible units to one according to $P(v_i = 1|\mathbf{h}) = \sigma(W \cdot \mathbf{h} + \mathbf{c})$ for all i in the visible layer with the hidden units already updated.

5.2 Learning

Now we know that there is thermal equilibrium of the RBM, which is the Boltzmann distribution, and how to get there based on sampling. Given the weights that determine the network energy we can pick up samples from Gibbs sampling and approximate the probability distribution without calculating the normalization explicitly.

Let us turn around the question. If we want to reproduce a given probability distribution with the RBM, how do we find the weights? This process is called learning. An RBM can approximate every probability distribution with arbitrary precision if the number of hidden nodes is large enough, which might increase exponentially with the number of visible units. In practice we will not make use of this feature and stick to the polynomial approximation but keep in mind that the number of hidden nodes is closely linked to the representational power. Given a probability distribution $P(v)$ that we want to represent by the network P_{RBM} we want to find the minimum distance between them with respect to the weights $D(P(v), P_{RBM}(v))$. Typically, one chooses the Kullback-Leibler (KL) divergence as a distance measure [Hinton, 2002]. It approaches zero, as the two probability distributions approach each other, but is not symmetric, thus is not a metric. The KL divergence is defined in this context as the difference between two probability distributions P and Q:

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_v P(v) \log\left(\frac{P(v)}{Q(v)}\right). \tag{5.2.1}$$

Given some input data v following a probability distribution $P^0(v)$, the probability of visible layer after one step of Gibbs reconstruction is $P^1(v)$ and the equilibrium distribution is $P^\infty(v)$.

We will apply the Contrastive Divergence (CD) learning scheme [Hinton, 2002]. For a detailed calculation see Equation B.3 in the Appendix B.1. The variation of the distance function by one entry of the connecting weight is given by

$$\frac{\partial}{\partial W_{ij}} \big(D_{\mathrm{KL}}(P^0 \parallel P_\theta^\infty) - D_{\mathrm{KL}}(P_\theta^1 \parallel P_\theta^\infty)\big) = \langle v_i h_j \rangle_{P^0} - \langle v_i h_j \rangle_{P_\theta^1} \,, \tag{5.2.2}$$

where the expectation values of the correlation between the layers over the data distribution P^0 and the first reconstruction P_θ^1 enter. This expectation value is an averages over a 'mini-batch', a subsample of about 20 samples. The formation of a mini batch is to calculate expectation values efficiently. The interpretation of the chosen error function is the following: We want the first reconstruction to have the same distance from the network thermal distribution as the input data. Thus, if we have learned successfully and the data distribution is roughly the thermal distribution, then the generated distribution by one step of Gibbs sampling will also roughly be thermal, i.e. describe the data distribution. This yields the learning algorithm:

- Given a sample set,
- form mini batches,
- for a sample v in the mini batch:
  - set $\mathbf{h} = \sigma(\mathbf{v} \cdot W + \mathbf{b})$,
  - define $W_+ = \mathbf{v}\mathbf{h}^{\mathrm{T}}$,
  - sample $\mathbf{h} \sim \sigma(\mathbf{v} \cdot W + \mathbf{b})$,
  - sample $\mathbf{v} \sim \sigma(W \cdot \mathbf{h} + \mathbf{c})$,
  - set $\mathbf{h} = \sigma(\mathbf{v} \cdot W + \mathbf{b})$,
  - define $W_- = \mathbf{v}\mathbf{h}^{\mathrm{T}}$.
- average over mini batch,

- set W to $W + \lambda(W_+ - W_-)$.

In the last line I introduced the learning rate λ. $\mathbf{vh}^{\mathrm{T}}$ denotes the outer product, i.e. the correlation matrix between visible and hidden layer. It is important to set the hidden units to the binary values before sampling the visible layer, as the hidden layer has the role of an informational bottleneck. When the correlation is calculated, the hidden units can be set to their probability of being one to reduce sampling noise [Hinton, 2012].

The derivation of the learning for the biases $\mathbf{a}, \mathbf{b}$ is analogue to the weights and given by:

$$\begin{aligned} \mathbf{b} &\leftarrow \mathbf{b} + \lambda\left(\langle\mathbf{h}\rangle_{P^0} - \langle\mathbf{h}\rangle_{P^1}\right), \\ \mathbf{c} &\leftarrow \mathbf{c} + \lambda\left(\langle\mathbf{v}\rangle_{P^0} - \langle\mathbf{v}\rangle_{P^1}\right). \end{aligned} \tag{5.2.3}$$

Together, they are the Contrastive Divergence (CD) learning algorithm.

5.3 Error Measures

With CD we have a stochastic gradient descent method at hand that is able to sample the gradients very efficiently in order to minimize a distance function between the network probability distribution and the target probability distribution. How does the chosen distance function relate to other standard distant measures? In the following section we will briefly describe the major aspects of applied error functions.

5.3.1 Kullback-Leibler Divergence and Fidelity

The Kullback-Leibler Divergence has already been introduced in Equation 5.2.1. As a difference between data distribution P_{data} and model probability distribution P_{RBM} it is written:

$$D_{\text{KL}}(P_{\text{data}} \| P_{\text{RBM}}) = \sum_v P_{\text{data}}(v) \log\left(\frac{P_{\text{data}}(v)}{P_{\text{RBM}}(v)}\right). \tag{5.3.1}$$

The Fidelity between two probability distributions P_{data} and P_{RBM} is defined to be

$$F(P_{\text{data}})\|P_{\text{RBM}}) = \left(\sum_v \sqrt{P_{\text{data}}(v) \cdot P_{\text{RBM}}(v)}\right)^2 . \tag{5.3.2}$$

To evaluate these quantities it is necessary to know the whole probability distributions of the training data and of the network. This is exponentially expensive and thus only applicable for small system sizes.

5.3.2 Reconstruction Error

The reconstruction error is defined to be the expectation value of the squared distance between the data vector and a reconstruction from one Gibbs-step

$$D_{\text{recon.}} = \langle \|v - \langle v \rangle_{P^1}\|^2 \rangle_{P^0}. \tag{5.3.3}$$

This error measure has the advantages that it is an expectation value, thus it can approximately calculated from a mini-batch, and that the first reconstruction is very easy to get. But there is the subtlety that a small mixing rate from the Gibbs-step also leads to a small reconstruction error. Mixing rate is the inverse of the Markov mixing time and describes how fast the system approaches its thermal equilibrium. A small mixing rate is induced by large absolute values of the weights. The weights change the most from their small initial value during the first few epochs of training, which correlates with the major decrease of the reconstruction error. An epoch the the phase of training in which the whole training data is used exactly once. Training requires several epochs. A causal statement between change of the weights and decrease of the reconstruction error cannot easily be made. Thus the reconstruction error should be viewed as a sanity check only and large increases are a marker for something going wrong.

5.3.3 Log-Likelihood and Pseudo-Log-Likelihood

Equation 5.0.2 defines a likelihood function if it is viewed as a function of the network parameters θ given a visible sample $\mathbf{v}$

$$\mathcal{L}_\theta(\mathbf{v}) = \frac{1}{Z}\sum_{\mathbf{h}} \exp(E_\theta(\mathbf{v},\mathbf{h})), \tag{5.3.4}$$

where the network parameters are $\theta = W, a, b$ and the network energy as a function of the network parameters for a given hidden and visible sample $E_\theta(\mathbf{v},\mathbf{h}) = \mathbf{v}\cdot W \cdot \mathbf{h} + \mathbf{c}\cdot\mathbf{v} + \mathbf{b}\cdot\mathbf{h}$. Taking the logarithm defines the log-likelihood. Log-likelihood is numerically more stable and still a distant measure because the logarithm is a strictly monotonic function

$$l_\theta(\mathbf{v})) = \log(\mathcal{L}_\theta(\mathbf{v})). \tag{5.3.5}$$

A pseudo-likelihood is defined to be an approximation in the sense that it neglects the conditional dependence of v_i on v_j for all i, j. For an RBM the approximation is exact, because the layers do not have intra-layer connections. Thus the pseudo-likelihood can be decomposed as a product, the pseudo-log-likelihood as a sum over Log-likelihoods of one visible v_i conditioned on all others $v_{\setminus i}$:

$$\log(\mathcal{L}_\theta(\mathbf{v})) = \sum_i l_\theta(v_i|v_{\setminus i}). \tag{5.3.6}$$

Knowing the conditional probability,

$$P(v_i|v_{\setminus i}) = \frac{P(v)}{P(v) + P(v, v_i \to 1 - v_i)}, \tag{5.3.7}$$

we can make a probabilistic ansatz

$$\begin{aligned} g &= N \cdot \log\left(P(v_i|v_{\setminus i})\right), \\ i &\sim U(0, N), \end{aligned} \tag{5.3.8}$$

where i follows a uniform distribution. Then the expectation value of g is the pseudo-log-likelihood

$$\mathbb{E}[g] = \sum_i l_\theta(v_i|v_{\setminus i}). \tag{5.3.9}$$

Altogether, writing $\tilde{\mathbf{v}}_i$ for $\mathbf{v}$ with $v_i \to (1 - v_i)$,

$$\begin{aligned} l_\theta(\mathbf{v}) &\approx N \cdot \log\left(\frac{P(\mathbf{v})}{P(\mathbf{v}) + P(\tilde{\mathbf{v}}_i)}\right) \\ &= N \cdot \log\left(\frac{\exp(F(\mathbf{v}))}{\exp(F(\mathbf{v})) + \exp(F(\tilde{\mathbf{v}}_i))}\right) \\ &= N \cdot \log(\sigma\left(F(\tilde{\mathbf{v}}_i) - F(\mathbf{v})\right)), \end{aligned} \tag{5.3.10}$$

where σ is the sigmoid function, as introduced in equation 5.1.2 and $F(v)$ is the free energy. The pseudo-log-likelihood measures approximately how close the thermal distribution of the RBM is to the data distribution. The pseudo-log-likelihood is easy to compute and gives a good overview over the learning progress, i.e. convergence, but it is difficult to develop an intuition for the absolute numbers.

5.3.4 Hyperparameters

Choosing suitable hyperparameters is a real issue in ML and requires some experience with the system. In this section we present the isolated effects of each hyperparameter, they can have combined effects which are much more involved to study. In this section we use relative descriptions like 'small value'. In Section 5.5 we present the quantitative results of hyperparameters which we found suitable from our experience.

The *number of hidden units* is a defining quantity for the expressiveness of the model, as the hidden layer is the informational bottle neck. The hidden units are binary and each one encodes one bit of information about the correlations. Increasing the number of hidden units also increases the number of model parameters and is expected to improve the learning result but slowing down the learning.

The *initial value of connecting weights* W are Gaussian random values with zero mean and 0.01 variance, the biases $\mathbf{a}$ and $\mathbf{b}$ as constant 0.1. Those values were recommended by [Hinton, 2002]. The weights need to be small but non-zero in order to have non-zero probability for the hidden units to be one and to explicitly break symmetry.

When choosing a large *learning rate* λ, we expect very fast convergence in all of the above error functions with a larger error. The CD algorithm might get stuck in local minima of the parameter space of the model which are far away from optimal or overshoot minima which are close to optimal.

Large *number of training samples* (10^6 for N=2) exhibit successful learning after very few epochs.

The *number of epochs* is how often the network is trained with the training data. The learning success measured by the error function is largest for the first epochs and expected to flatten to smaller and smaller training improvements. Where these two phases are strongly depends on the learning rate and the amount of training data.

The *size of a mini batch* influences the learning process. Small batches lead to fast decrease of the error functions but large oscillations as the averaged gradients might point into a wrong direction. Small mini batches lead to larger calculation times per epoch. That is due to the implementation of batch-wise updates. Small batches tend to run into local minima or even worse do not reflect the topology of the parameter space of the model and lead to wrong results. For large mini batches the gradient might become very small and learning is slower.

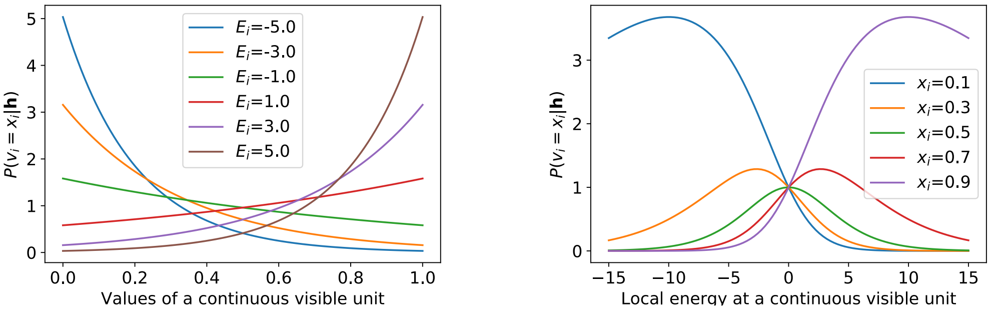
Small *sample sizes from the RBM* (10^3 for number of visible units $N = 2$) lead to large deviations of the sampled distribution from the analytic probability distribution of the RBM calculated from the network parameters. Taking many samples from the RBM via Gibbs sampling leads to small deviations from the analytic probabilities.

5.4 Continuous States

For the sake of completeness, we are also interested in the question of how to handle continuous local degrees of freedom. In the following, we will propose that continuous visible units can be treated very similarly to binary ones by following three different approaches: Investigating the expectation value of conditional probabilities, referring to existing thermal models and a limiting case. If we allow the visible entries to take continuous values between 0 and 1, Equation 5.1.2 does not hold any more for the conditional probability but instead we have to write for the conditional probability density of one visible unit v_i to take the value x_i, given the hidden units $\mathbf{h}$:

$$\begin{aligned} P(v_i = x_i|\mathbf{h}) &= \frac{\exp(x_i W_{ij} h_j + b_i h_i + x_i c_i)}{\int_0^1 \mathrm{d}v \exp(E(\mathbf{v}, \mathbf{h}))} \\ &= \frac{\exp(x_i(W_{ij}h_j + c_i))}{\frac{1}{W_{ij}h_j + a_i}\left[\exp(x_i(W_{ij}h_j + c_i))\right]_{x_i=0}^{x_i=1}} \\ &= \frac{E_i \exp(x_i E_i)}{\exp(E_i) - 1}. \end{aligned} \tag{5.4.1}$$

It is plotted in Figure 5.3. We see that for x close to its boundary the conditional probability density diverges for absolute large E_i.



(a) (b)

Figure 5.3: Conditional probability density in case of continuous visible units (see Equation 5.4.1) as a function of the values x_i the unit can take (a) and as a function of the local energy $E_i = W_{ij}h_j + c_i$ in (b). It diverges for x_i close to zero and one.

The expectation value of value of the visible unit under the conditional probability $P(v_i = x_i|\mathbf{h})$ as a function of the local energy $E_i = W_{ij}h_j + c_i$ is given by

$$\mathbb{E}_{P(v_i=x_i|\mathbf{h})}[x_i] = \frac{E_i}{\exp(E_i) - 1} \int_0^1 \mathrm{d}x_i x_i \exp(x_i \cdot E_i) = \frac{-1}{E_i} + \frac{1}{1 - \exp(-E_i)}. \tag{5.4.2}$$

The expectation value as a function of the energy E_i at site i is plotted in Figure 5.4. The expectation value has a similar shape to the sigmoid function $\sigma(E_i)$ and

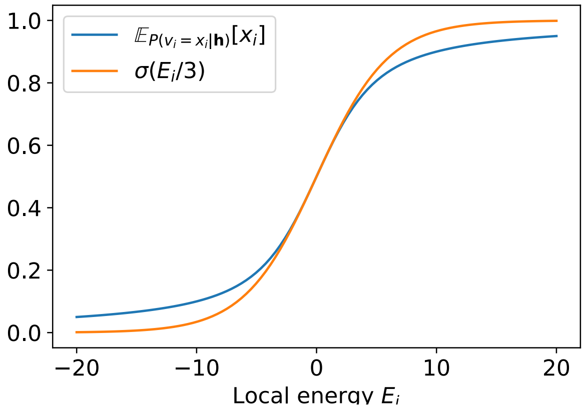


Figure 5.4: Expectation value of the conditional probability in case of continuous visible units (see Equation 5.4.2) and sigmoid function. Qualitatively similar, the latter is used to approximate the first.

approximates it well for small energies with a relative factor of 3 in steepness. This justifies the standard assumption that for continuous variables (if normalized to values $\in (0, 1)$) one can just use the value of the sigmoid function instead of stochastic binary units [Hinton, 2002].

Another implementation for continuous states is used by [Chen and Murray, 2002], originally introduced by [Movellan, 1998]. The stochasticity of the so called Diffusion Network is rooted in a Langevin equation which describes diffusion processes in classical physics - thus its name. These processes are characterized by a deterministic (or classical) and a thermal (or quantum) probabilistic part. For the probabilistic part, again the value of the sigmoid function is directly used as a continuous visible variable. For the stochastic impact there are different approaches. The first one is to hope for enough statistics resulting from the update procedure of the binary hidden units via Gibbs sampling [Chu et al., 2018]. The second is to add a Gaussian noise term to the visible units [Freund and Haussler, 1991].

Furthermore, [Freund and Haussler, 1991] point out that the case of continuous visible units supplemented with a Gaussian term can be approximated by the case of binary visible units and vice versa, both in representational power and the learning algorithm in the limit of small weights. This limiting case, in which the sigmoid is linear, motivates the linear model for the visible units used in [Chen and Murray, 2002].

5.5 Results of Stationary RBM Representation

In this section, we want to give a quantitative understanding of the learning process by presenting the used hyperparameters (Section 5.3.4) and some of the error measures (Section 5.3) of the corresponding learning process. Therefore we study three examples, two spin states and one example of continuous states.

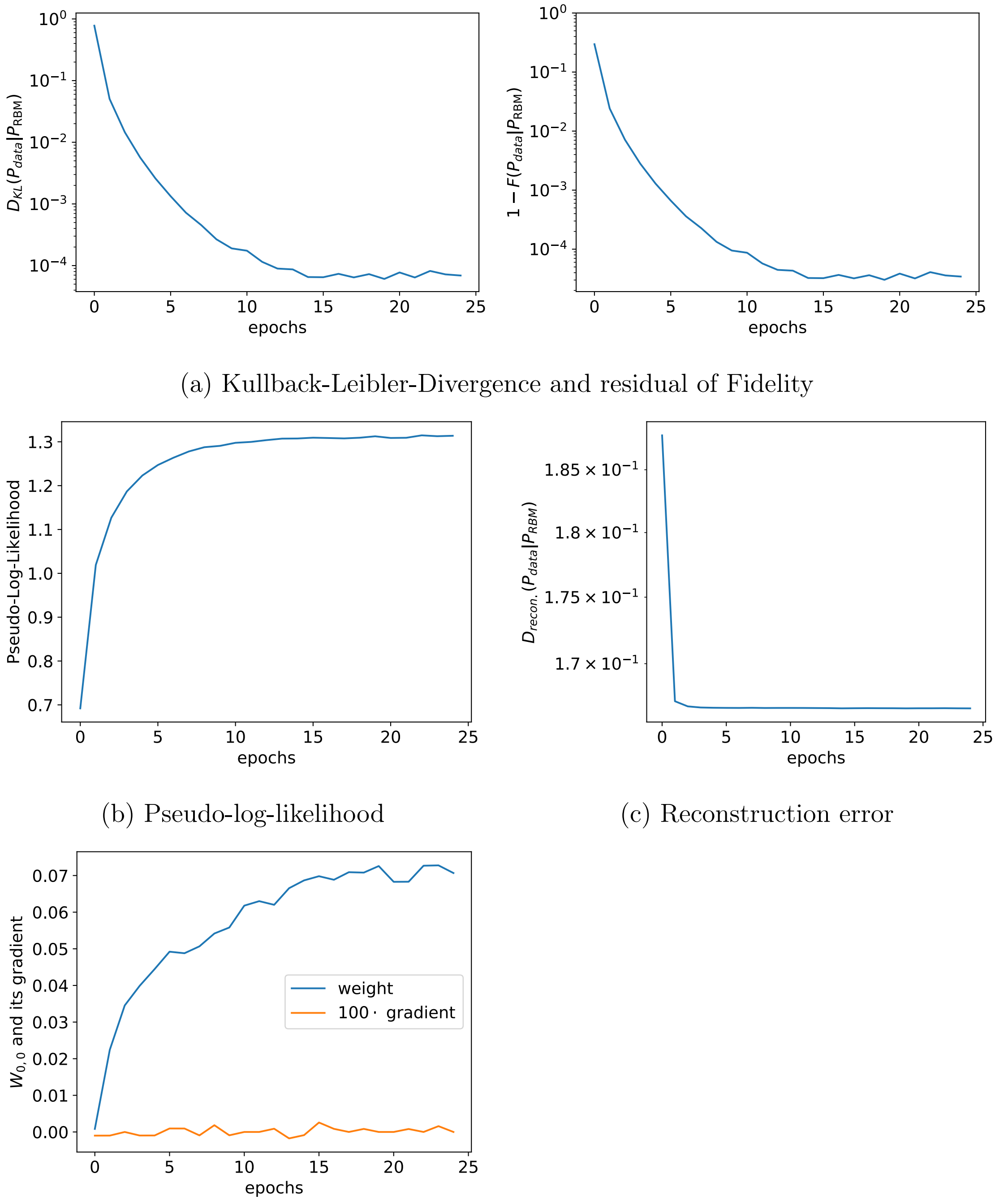
5.5.1 Discrete States

One-Hot encoding

Given a density matrix or a state, one can easily get the probability distribution (Equation 4.2.1). We use a Metropolis-Hastings method (compare Subsection 5.1.2) to generate samples from the distribution. For our spin $\frac{1}{2}$ systems, we chose the number of local operators to be four, so a sample of one spin can equivalently be written as an integer $1, \ldots 4$ or as a so called one-hot $(1,0,0,0), (0,1,0,0), (0,0,1,0), (0,0,0,1)$ which can be represented in the RBM as four binary visible units and treated analogously to standard binary units.

Ground State of TFI Hamiltonian

The ground state for small system sizes, e.g. $N = 2$ can be calculated by diagonalization of the Hamiltonian for different field strengths h_f. An example of $h_f \to \infty$ is given in Section A.1. For $N = 2$, $h_f = 100$, the tetrahedral POVM, 4 hidden units, learning rate $\lambda = 10^{-4}$, $5 \cdot 10^5$ training samples describing P^a_{data}, 25 epochs and mini-batch size of 20, we trained the RBM with CD. These numbers of the hyperparameters are the result of heuristics, our experience and suggestions in [Hinton, 2012]. They might be optimized further but reliably lead to convergence of the error functions, as can be seen in Figure 5.5.



(a) Kullback-Leibler-Divergence and residual of Fidelity

(b) Pseudo-log-likelihood

(c) Reconstruction error

(d) Weight $W_{0,0}$ and its gradient

Figure 5.5: Error functions of training the ground state of TFI for $N = 2$ and $h_f = 100$ with hyperparameters given in the main text (Section 5.5.1). Different error functions suggest convergence after different number of epochs. Reconstruction error (c) after five, pseudo-log-likelihood (b) after ten, Kullback-Leibler divergence and fidelity (a) after 15 epochs, the weight (d) only settles in after 20 epochs.

We observe that all error functions show convergence but the plateau sets in after different number of epochs (see Figure 5.5). The reconstruction error decreases the most, when the weights take absolute large values which is linked to the change of the mixing rate, thus reconstruction error is not reliable. Kullback-Leibler divergence and fidelity (Equation 5.3.2) show similar behavior and are most reliable as they take the whole probability distributions into account but are therefore computationally exponentially expensive for larger systems. Pseudo-log-likelihood gives a qualitative approximation and only scales linearly with the number of network parameters and the size of mini-batches. The RBM represents the ground state of the TFI Hamiltonian with high accuracy, the residual of one minus the fidelity is smaller than 10^{-4}.

GHZ State

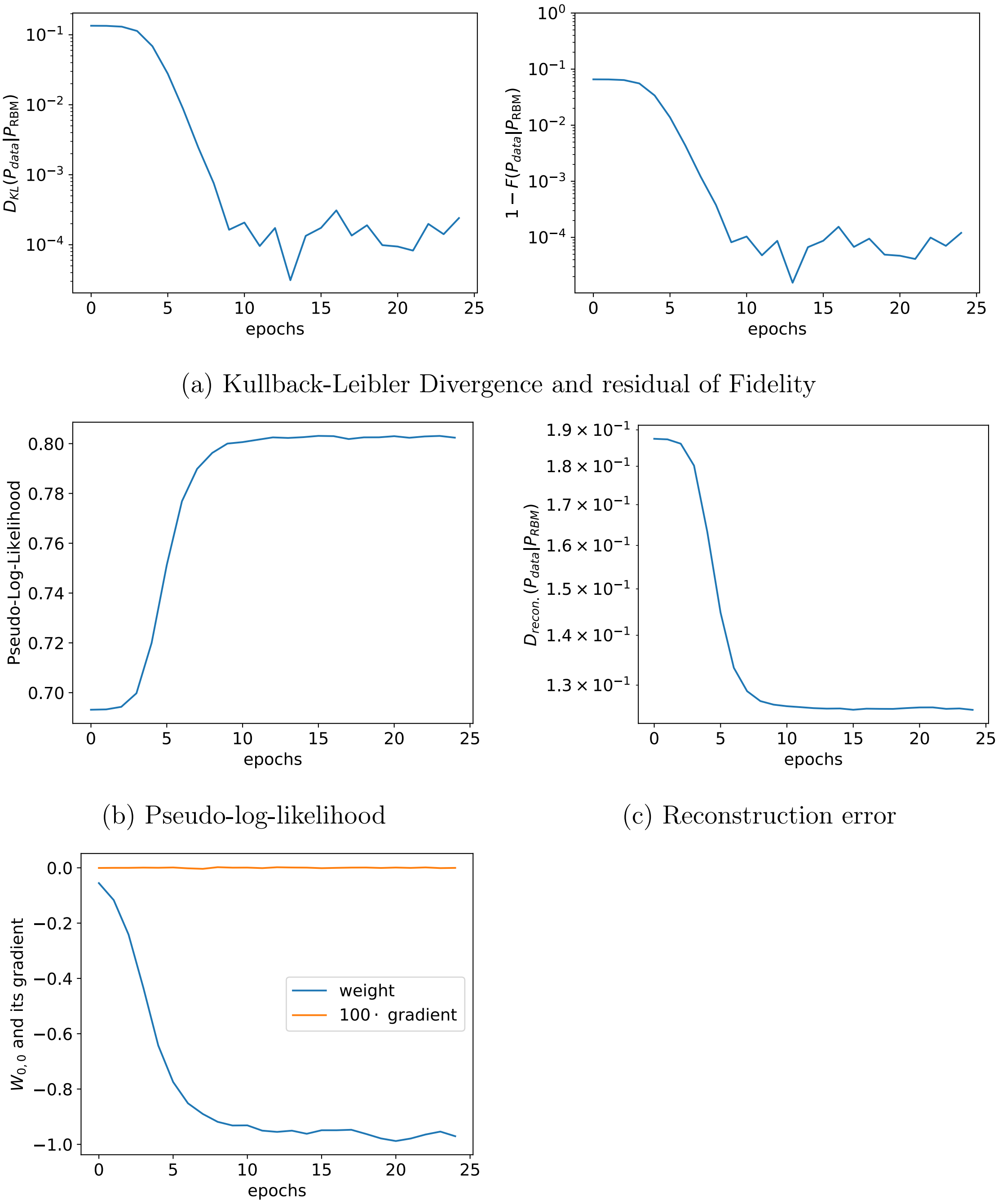
The GHZ state is named after Greenberger, Horne and Zeilinger and is highly non-classical in the sense that bipartitions are not separable. Written as a wave function it is defined as

$$|\Psi_{GHZ}\rangle = \frac{1}{\sqrt{2}} \left(|\uparrow\rangle^{\otimes N} + |\downarrow\rangle^{\otimes N} \right), \tag{5.5.1}$$

where N is the system size. It describes an equal superposition of all states being in the state $|\uparrow\rangle$ and all states being in the state $|\downarrow\rangle$. If we chose the z-basis, the eigenvectors of σ_z as local basis, and $N = 2$, the density matrix is written as

$$\rho_{GHZ} = \frac{1}{2} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} . \tag{5.5.2}$$

We train it with the same hyperparameters as above, this time we add a momentum method for the updates of the network parameters, where 0.8 of the last update gets added again, which increases stability and lowers the effect of local minima. We show the resulting error functions in Figure 5.6.



(a) Kullback-Leibler Divergence and residual of Fidelity

(b) Pseudo-log-likelihood

(c) Reconstruction error

(d) Weight $W_{0,0}$ and its gradient

Figure 5.6: Error functions of training the GHZ state for $N = 2$ with hyperparameters given in the main text. Different error functions suggest convergence after ten epochs. Kullback-Leibler divergence and fidelity (a), pseudo-log-likelihood (b), Reconstruction error (c) and the weight $W_{0,0}$ (d).

We observe a small plateau at the beginning of learning, which indicates that the gradient is very small at the beginning. The initial values might sit a close to a saddle point of the parameter space for the GHZ state. All error functions change rapidly until ten epochs, when convergence sets in. One minus the fidelity and Kullback-Leibler divergence oscillate around 10^{-4} and do not improve further. The highly entangled GHZ state can be represented by the RBM.

5.5.2 Continuous States

As an example for continuous visible units we take measurement data from a quantum many-body experiment with ultra-cold Bosons [Kunkel et al., 2019], also performed in Heidelberg. In a Bose-Einstein condensate of an elongated atomic cloud of ^{87}Rb the short time dynamics leads to an entangled many-body state [Kunkel et al., 2019]. We take the relative occupation numbers, i.e. the occupation number divided by the number of atoms in the cloud, of $F = 1$ hyperfine manifold $n_{1,+1}, n_{1,-1}$ and $F = 2$ manifold $n_{2,-2}, n_{2,+2}$, whose difference are the spin in x-direction S_x and the quadrupole moment in yz-direction Q_{yz} respectively (Figure 5.7). The squeezing, the object of interest for the experiment, is given by the variance of the data cloud in Figure 5.7 depending on the angle of the projection. This is shown in Figure 5.8. Training the network on the relative occupation numbers, it has to extract the information about the squeezing on its own.

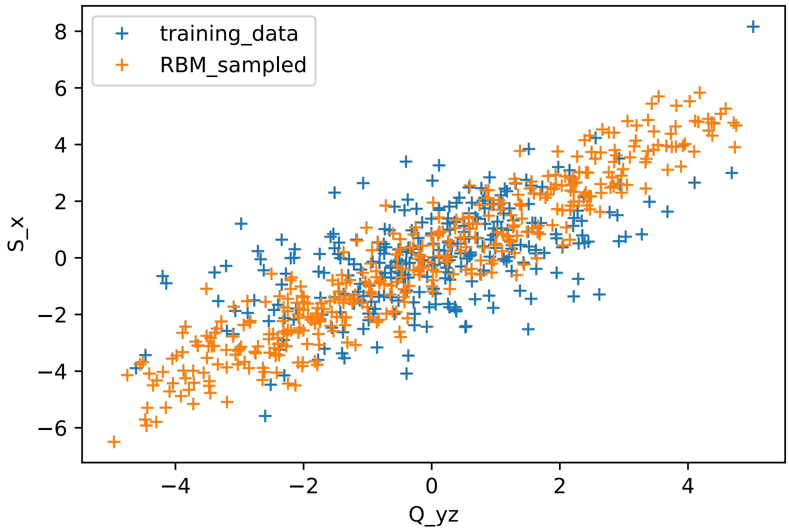


Figure 5.7: Experimental data from [Kunkel et al., 2019] and samples from RBM. Occupation numbers of hyperfine levels in ultracold ^{87}Rb carry information about magnetization in x-direction S_x and the quadrupole moment Q_{yz}. There is spin squeezing, i.e. an elongation along the diagonal, indicating quantum entanglement. The RBM is able to capture this quantum feature from learning data of continuous occupation numbers. Optimal training parameters are given in the main text.

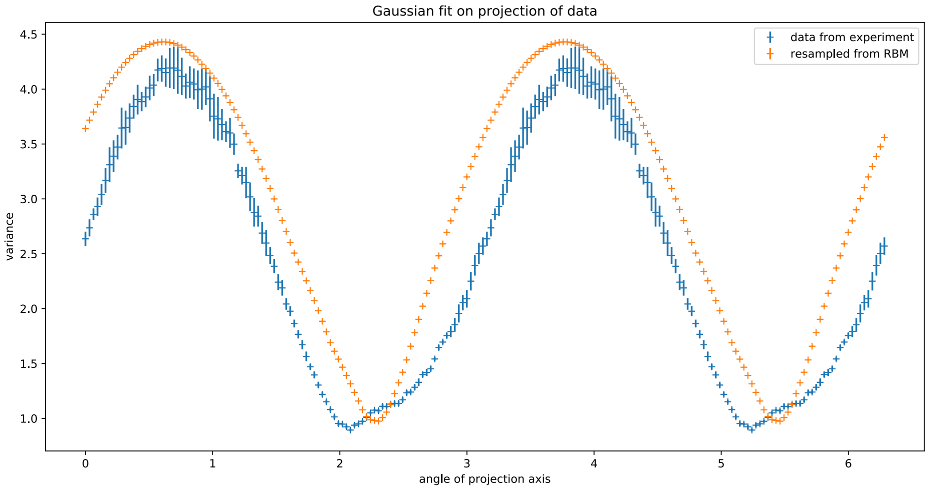


Figure 5.8: Spin Squeezing: Variance from Gaussian fit of projection over angle from Figure 5.7. The RBM with continuous visible units is able to capture the quantum nature of trapped ions

We implemented the RBM with continuous visible units by taking the value of the sigmoid function as the value of the visible units with the following hyperparameters: 4 visible units representing $n_{1,+1}, n_{1,-1}, n_{2,-2}, n_{2,+2}$, 30 hidden units, learning rate $\lambda = 10^{-4}$, batch size of 10, 200 epochs and trained the RBM on the occupation numbers of 284 measurements. One can conclude that taking the value of the sigmoid as the output for continuous visible units is justified in this case. The squeezing is indeed captured by the network but the variance is slightly overestimated.

6 Time Evolution - POVM Equation of Motion

Now, we bring together the POVM description of states representable by RBM, seen in the previous Section 5.5, with time evolution known from standard quantum mechanics and derive the POVM equation of motion (e.o.m.). This can approximately be captured and solved by RBMs (Section 8). Therefore we start with unitary time evolution in standard quantum mechanics. In the Heisenberg picture, time evolution of operators $O(t)$ is given by a unitary transformation

$$O(t+\mathrm{d}t) = e^{-iH\cdot\mathrm{d}t} O(t) e^{iH\cdot\mathrm{d}t}, \tag{6.0.1}$$

with H the Hamiltonian of the system. This can be equivalently written in the continuous time limit $\mathrm{d}t \to 0$ and up to first order in $\mathrm{d}t$

$$\frac{\mathrm{d}O(t)}{\mathrm{d}t} = i\,[H,O]\,, \tag{6.0.2}$$

where $[.,.]$ denotes the standard commutator of two operators.

The exact time evolution $\rho(t)$ of the initial density matrix ρ_0 under a Hamiltonian H is written

$$\rho(t) = \exp(-iHt)\cdot\rho_0\cdot\exp(iHt), \tag{6.0.3}$$

where exp() is the matrix-exponential and can be calculated in the following sense. If v are the eigenvalues of H and w the matrix of their eigenvectors, then $\exp(H) = w\cdot\mathrm{diag}(\exp(v))\cdot w^{\dagger}$. This procedure is called exact diagonalization.

From Equation 6.0.2 we can derive the time evolution of the POVM probability distribution $P(a)$ by explicitly plugging in the representation defined in Section 4. To make clear Einstein sum convention, we also write P^a equivalently to $P(a)$:

$$\begin{aligned}\frac{\mathrm{d}P^{a'''}}{\mathrm{d}t} &= \frac{\mathrm{d}\rho_{ij}}{\mathrm{d}t}M_{ji}^{a'''}\\ &= i\left[H,\rho\right]_{ij}M_{ji}^{a'''}\\ &= iH_{ik}\rho_{kj}M_{ji}^{a'''} - i\rho_{ik}H_{kj}M_{ji}^{a'''}\\ &= iQ_H^{a'}M_{ik}^{a'}P^aT_{aa''}^{-1}M_{kj}^{a''}M_{ji}^{a'''} - iP^aT_{aa'}^{-1}M_{ik}^{a'}Q_H^{a''}M_{kj}^{a''}M_{ji}^{a'''}\\ &= iQ_H^{a'}K^{a'a''a'''}P^aT_{aa''}^{-1} - iQ_H^{a''}K^{a'a''a'''}P^aT_{aa'}^{-1}\\ &= iP^aT_{aa'}^{-1}\left(K^{a'a'''a''} - K^{a'a''a'''}\right)Q_H^{a''}.\end{aligned} \tag{6.0.4}$$

In the first line we take the time derivative of the POVM probability defined in Equation 4.2.1 and use Einstein sum convention for the trace. From the first to the second line, we plugged in the time evolution (Equation 6.0.2) of the density matrix ρ. In the following we used the POVM basis for the operators and defined the trace over three M-matrices

$$K^{aa'a''} \equiv M_{ij}^{a}M_{jk}^{a'}M_{ki}^{a''} \tag{6.0.5}$$

and used that it is cyclic to reorder the indices so that the structure of the POVM equation of motion can be captured on first sight. The time derivative of the probability distribution $P(a)$ is a linear transformation of $P(a)$ and the matrix in between contains an antisymmetric ingredient in form of $K^{a'a'''a''} - K^{a'a''a'''}$. To make the structure even more apparent, we repeat our calculations with its graphical representation introduced in [Carrasquilla et al., 2019b] in the Appendix C.1

In principle, one could also start from other equations of motion like the Lindblad master equation and insert the POVM basis to capture the time evolution of open systems. The aim of this thesis is just a prove of principle, so we stick to the unitary case.

7 Exact Solution

To check consistency of our developed theory of time evolution for the probability distribution, let us solve small systems exactly and compare to exact diagonalization. By writing Equation 6.0.4 as a matrix multiplication

$$\frac{\mathrm{d}P^a(t)}{\mathrm{d}t} = P^{a'}(t) \cdot R^{a',a}, \tag{7.0.1}$$

we define the time evolution matrix

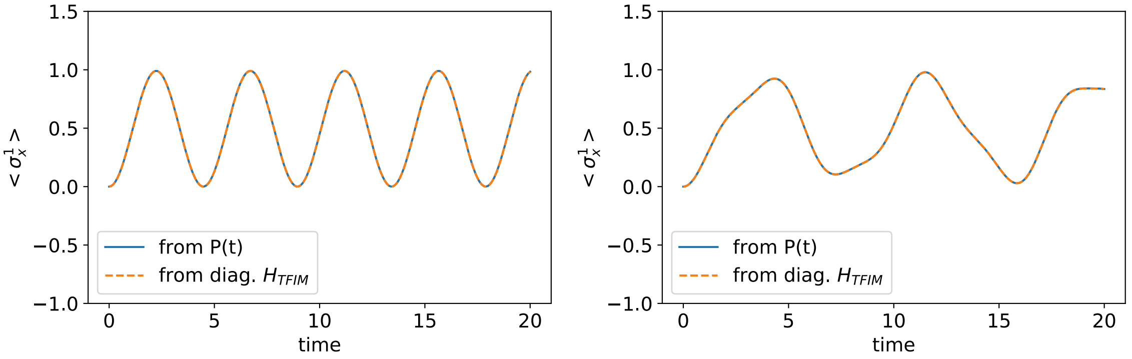
$$R^{a,a'''} = iT^{-1}_{aa'} \left(K^{a'a''a'''} - K^{a'a'''a''} \right) Q^{a''}_H. \tag{7.0.2}$$

If N is the system size, its size is $4^N, 4^N$. The measurement outcomes of P^a are sorted that way, that the last index is the one changing first. As an example for $N = 2$: $a = ((0,0), \dots (0,3), (1,0), \dots (1,3), \dots (3,3))$. $R^{a,a'}$ is a real matrix whose columns and rows add up to zero. For the time evolution to first order in the infinitesimal time step $\mathrm{d}t$, one can also write

$$\begin{aligned} P^a(t + \mathrm{d}t) &= P^a(t) + \mathrm{d}t \frac{\mathrm{d}P^a(t)}{\mathrm{d}t} \\ &= (\mathbb{I}_{a,a'} + \mathrm{d}t R^{a,a'}) P^{a'}. \end{aligned} \tag{7.0.3}$$

The expression $\mathbb{I}_{a,a'} + \mathrm{d}t R^{a,a'}$ is called pseudo-stochastic matrix [Carrasquilla et al., 2019a]. Its columns add up to one, thus it preserves the norm of a probability, but its entries can be negative and positivity is not preserved in general, thus the prefix 'pseudo'. But one can choose $\mathrm{d}t$ small enough, such that $\|\mathbb{I}_{a,a'} + \mathrm{d}t R^{a,a'}\| \geq 0$ in the operator norm.

As a consistency check, we solve this differential equation (Equation 7.0.1) ana-



(a) N=3

(b) N=4

Figure 7.1: Comparison of exact diagonalization and exact time evolution of P(a) (Equation 7.0.4) for GHZ(N), with $N = 3, 4$ under H_{TFI} with magnetic field $h_f = 1.1$. For both cases $N = 3$ and $N = 4$ the two curves lie exactly on top of each other. The mapping is exact.

lytically by the matrix exponential

$$P^a(t) = \exp\Big(tR^{a,a'}\Big) \cdot P^a(t=0). \tag{7.0.4}$$

In the POVM basis, Equation 4.2.4 induces the following coefficient for the TFI Hamiltonian

$$Q_H = -\sum_i Q_z^{a_i} Q_z^{a_{i+1}} - h_f \sum_i Q_x^{a_i}, \tag{7.0.5}$$

which enters the time evolution through the time evolution matrix $R^{a,a'}$. With the GHZ-state as initial conditions for $P^a(t=0)$, this gives the results shown in Figure 7.1a for system size of three and in figure 7.1b for system size of four spins. We see that the developed theory is exact so far.

8 POVM Time Evolution with RBM

In this section, we will make a proof of principle that the quantum many-body time evolution can be solved solely with a neural network based on probability distributions.

We propose an approach for solving the Equation 6.0.4 which is similar to the Euler integration method but based on training the network and drawing samples from it. We iteratively train the RBM on the probability distribution at a given time t and use the network to efficiently draw samples from $P(t) + \mathrm{d}t\frac{\mathrm{d}P(t)}{\mathrm{d}t}$ which can be used to train the network parameters for $P(t + \mathrm{d}t)$. The sampling from the 'future distribution' $P(t + \mathrm{d}t) = P(t) + \mathrm{d}t\frac{\mathrm{d}P(t)}{\mathrm{d}t}$ is done via a Metropolis-Hastings algorithm. The acceptance of the new sample $\tilde{a}$ coming from sample a is given by

$$A(\tilde{a}, a) = \frac{P^{\tilde{a}}(t) + \mathrm{d}t\frac{\mathrm{d}P^{\tilde{a}}(t)}{\mathrm{d}t}}{P^{a}(t) + \mathrm{d}t\frac{\mathrm{d}P^{a}(t)}{\mathrm{d}t}} \tag{8.0.1}$$

where $P^a(t)$ is either known and stored from the previous time step or can easily be calculated from the network parameters (Equation 5.0.2). Before we present our results, we need to answer some questions concerning representational power and how we chose suitable hyperparameters.

8.1 Integration Step Size

An important aspect of the integration scheme with the RBM is the integration step size $\mathrm{d}t$. As this integration scheme is simply a version of Euler integration, it is expected to be unstable in the sense that the norm is not conserved. So, for $N = 2$, we integrate the equation $P^a(t + \mathrm{d}t) = P^a(t) + \mathrm{d}t \cdot \frac{\mathrm{d}P^a(t)}{\mathrm{d}t}$ with two different step sizes. The resulting spin expectation values can be seen in Figure 8.1.

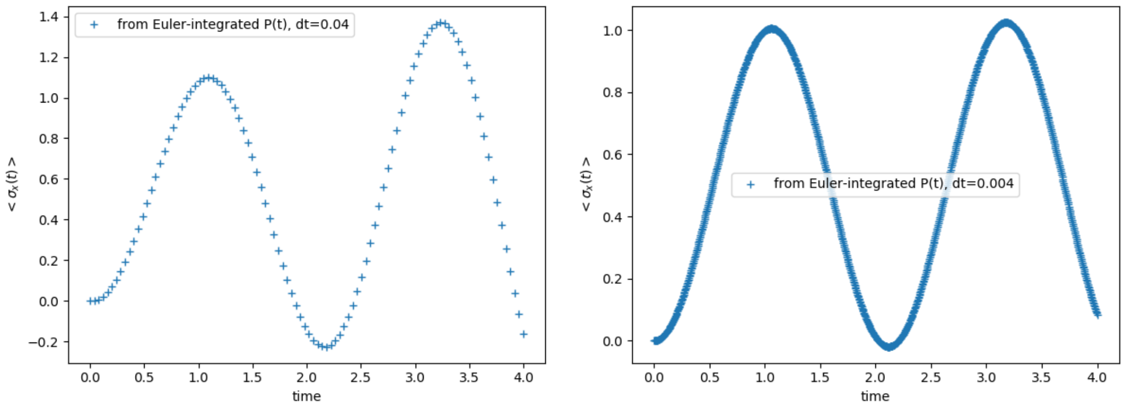


Figure 8.1: Numerically integrated $P^a(t+\mathrm{d}t) = P^a(t) + \mathrm{d}t \cdot \frac{\mathrm{d}P^a(t)}{\mathrm{d}t}$ with Euler for two different $\mathrm{d}t$. For $\mathrm{d}t$ much larger than 10^{-3} the integration is unstable.

Indeed it is clear from the figure that the spin expectation value increases unphysically. That is the spin expectation value should not be out of the interval $[-1, 1]$ for a physical spin, but it is after short times if $\mathrm{d}t$ is chosen to be of the order 10^{-2}.

8.2 Representational Power

An important question on the way is, if the RBM is able to learn any probability distribution that is generated during time evolution. To answer this question, we solve the time evolution equation of the density matrix via exact diagonalization (Equation 6.0.3). For discrete points in time t, the exactly evolved density matrix $\rho(t)$ gets mapped on the POVM probability, on which we train the RBM. From both, the exactly solved density matrix and the network parameters, we calculate the spin expectation value and plot it in Figure 8.2. The network parameters enter in Equation 4.2.7 through the probability distribution $P(a)$, which is $P_{RBM}(a)$ in this case (with $a = \mathbf{v}$ in Equation 5.0.8). The initial state is the ground state of TFI with magnetic field $h_f = 100$, quenched to $h_f = 1.1$ for the dynamics. The hyperparameters of training are: the number of hidden units $m_{hid} = 4$, number of epochs is 10, size of mini-batch is 10, learning rate $\lambda = 10^{-4}$ and the number of training samples is $5 \cdot 10^5$. The deviations are small and seem to be of statistical nature. One can see that the training of the RBM is successful for any probability distribution generated by time evolution.

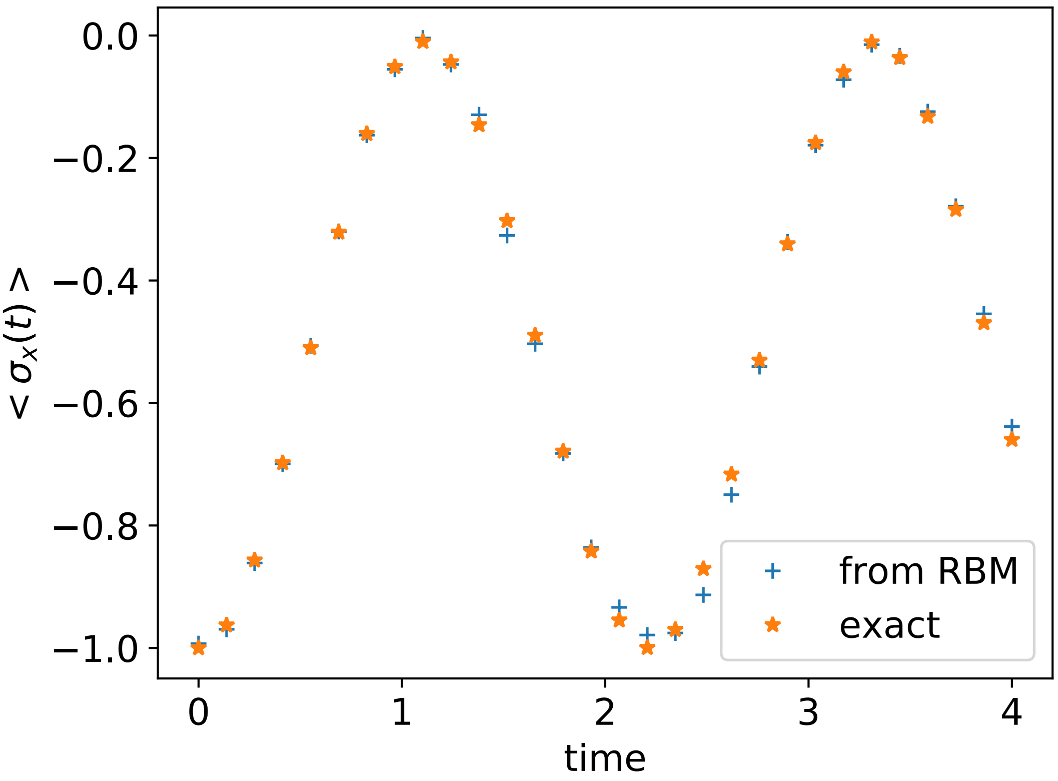


Figure 8.2: Time Evolution of expectation value of spin in x-direction from exact diagonalization (Equation 6.0.3) '*' and afterwards training the RBM to represent the state '+' at several points in time.

8.3 Reduction of Statistical Errors

We want to reduce the statistical error of learning and sampling such that the RBM is able to represent small changes in the probability distribution when integrating the POVM e.o.m. with the RBM over a small time step dt. Therefore we investigate learning and sampling behavior for different sample sizes, as this hyperparameter reduces statistical errors.

The question one should consider is, if the Metropolis sampling from the 'future distribution' accurately represents the underlying probability distribution. To investigate how the sampling noise can be reduced, we vary the sample size in the range from 10^2 to 10^6 logarithmically, train the network on a ground state of the TFI Hamiltonian with transverse field $h_f = 100$ with following hyperparameters: four hidden units, a learning rate of $\lambda = 10^{-4}$, 10 epochs with mini batches of size 10. We plot the first five measurement outcomes of the exact probability calculated directly from the density matrix of the ground state ('*'), the probability distribution calculated from the sample set as a histogram with error bars from Metropolis Sampling ('+') and the probability distribution calculated from the network parameters of the trained RBM ('x') in Figure 8.3. We see that the probabilities calculated from the network parameters only converge for the last two crosses, corresponding to sample sizes of order $> 10^5$. We also varied other parameters a little and always found the same behavior in sample size.

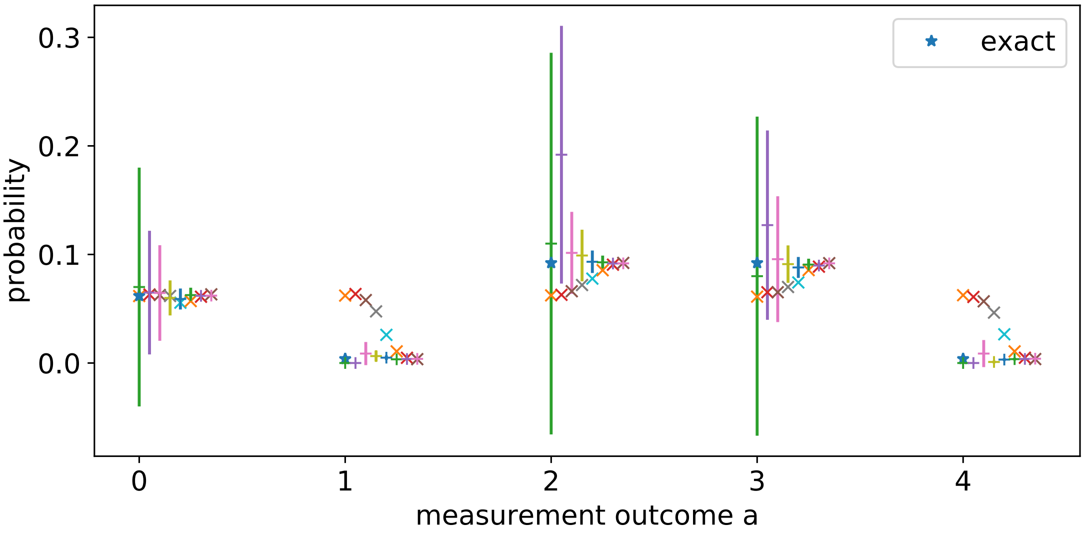


Figure 8.3: Convergence of sampling from the exact probability distributions ('+') and learning result ('x') to the exact probability distribution ('*'). First five measurement outcomes of the exact probability calculated directly from the density matrix of the ground state ('*'), the probability distribution calculated from the sample set as a histogram with error bars from Metropolis sampling ('+') and the probability distribution calculated from the network parameters of the trained RBM ('x') for logarithmically varied sample size in the range from 10^2 to 10^6. The different colors are different sample sizes, increasing from left to the right for each measurement outcome. Error bars of probability approximated by Metropolis sampling decrease rapidly and are negligible for sample sizes larger than 10^4, the learned probabilities only converge for sample sizes larger than 10^5.

In this context, one should ask if the sampling and learning noise is already decreased that much that the impact on the probability distribution of an infinitesimal time step is larger than the one of the noise. Therefore we investigate the stability of the RBM integration for d$t = 0$. We train the RBM on the ground state (see Equation A.1) of the TFI Hamiltonian with transverse field $h_f = 100$ with 10^6 samples and draw the same amount of samples again, just to train it on those data. This procedure is expected to be unstable in the sense that there is no force bringing the probability distribution back to the physical state once sampling noise has driven it away. In Figure 8.4, we see the first five entries of the probability distribution for each of the 10 repetitions of sampling and learning. The probabilities do not change much compared to the sampling noise for smaller sample size in Figure 8.3.

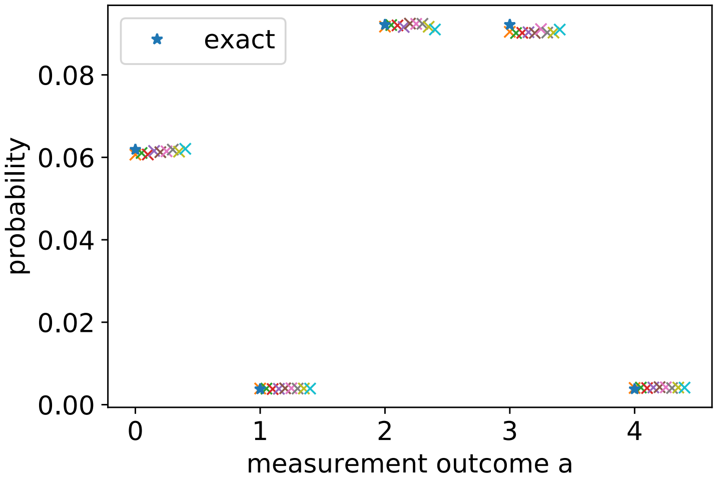
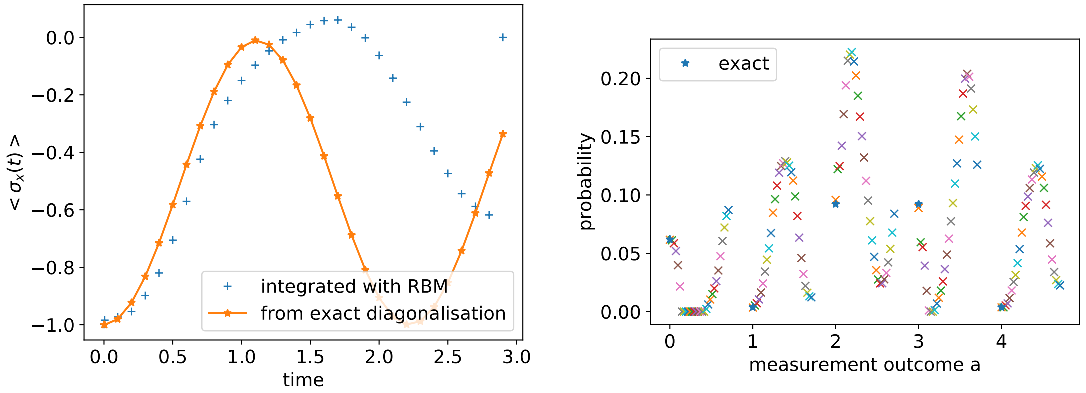


Figure 8.4: Stability check of repetitively learning and sampling the RBM using Metropolis Hastings Equation 8.0.1 with d$t = 0$. The first five entries of the probability distribution of the ground state of the TFI Hamiltonian with transverse field $h_f = 100$ for 10 times learning and sampling. Sample size is 10^6, 4 hidden units, 10 epochs, mini batch size of 10, learning rate of 10^{-4}. Each learning result is encoded in a different color from left to right.

The repetitive learning and sampling is robust against noise and information loss for a sample size of 10^6 for a system size of $N = 2$. There is the conflict, that the RBM needs a large dt so that the physical change of the probability distribution is larger than the sampling noise, but then Euler integration becomes unstable. To reduce sampling and learning noise, the sampling size needs to be very large, which immensely slows down the learning process, that needs to be repeated for every small time step dt.

8.4 Results

After these preliminary considerations, we can put the Euler-integration scheme with the RBM together with $\mathrm{d}t = 0.1$ and sampling size of $2 \cdot 10^6$. In Figure 8.5, one can compare the result of the RBM with the solution from exact diagonalization.



(a) Expectation value of magnetization in x-direction at site 1.

(b) First five entries of the probability distribution $P^a_{RBM}(t)$ as a function of time.

Figure 8.5: Time evolution by RBM integrator as described in the main text and exact diagonalization. $N = 2$, initial state is ground state of TFI Hamiltonian with $h_f = 100$ quenched to $h_f = 1.1$. Sample size is $2 \cdot 10^6$ and $\mathrm{d}t = 0.1$.

We train the RBM with the samples from the 'future distribution' in ten epochs, as it is expected that the weights only change very little and starting point is already very close to the target distribution. We investigate how far off the integrator gets, if smaller sample sizes are used. That way we can approximate the minimum number and the scaling of the statistical errors. The analogous results to sample size of $2 \cdot 10^6$ for samples sized of 10^4, 10^5 and 10^6 are shown in the Appendix D. We observe that only for a sample size of $2 \cdot 10^6$ the expectation value of the spin in x-direction is approximately correct for short times. The time evolution for the TFI ground state of $h_f = 100$ quenched to $h_f = 1.1$ for system size $N = 2$ is captured by the RBM integrator for short times.

In our implementation for $N = 2$, we used the time evolution matrix $R^{a,a'}$, defined in Equation 7.0.1, for the Metropolis sampling in Equation 8.0.1. The problem with this is the multiplication $R^{a,a'} P^{a'}(t)$ has exponentially many terms with system size. To solve this, there are multiple options. One can find approximations in the correlation length to formulate local approximations. In the Appendix C.2, we investigate how local approximations might be formulated in the POVM setting.

9 Conclusion and Outlook

A machine learning graph (RBM) has been used to represent the POVM representation of many-body quantum states. From there, we derived a POVM equation of motion which can be solved by the RBM alone. For the RBM representation of a steady state we showed that besides discrete systems, RBMs are also able to represent entanglement features of quantum many-body systems with continuous degrees of freedom by continuous visible units. For the POVM equation of motion, we showed exact correspondence to the quantum mechanical equivalent and found a sampling scheme for the RBM that integrates it statistically. We found out that in order to reduce statistical sampling and learning errors, an enormous amount of sampling data is needed. In the case of system size $N = 2$, a sample set of 10^6 was sufficient to obtain good agreement to the solution from exact diagonalization for short times. The developed NQS solver suffers from accumulating errors with time and precision is limited. In our application, the NQS solver is eventually unstable and the computational cost is much larger than exact diagonalization even for small system sizes. By using other models than the RBM, precision and stability might improve.

We can conclude, that RBMs are a powerful tool to represent probability distributions. The POVM setting can be used to represent any quantum state as a positive probability distribution that can be learned by the RBM. RBMs are even able to learn distributions with finite continuous values. The time evolution in the POVM setting is possible, but at this very rudimentary stage of development it has no advantage over exact diagonalization. However, it has the potential to be improved so that it scales sub-exponentially in system size and appropriate approximations might be implemented.

Future works might profit from the sub-exponential scaling of network parameters to represent NQS to successfully implement a scalable NQS integrator based on real numbers. This would enable a powerful tool to simulate relevant many-body systems for experiments, like open systems.

Appendix

A Appendix for Chapter 3

A.1 Ground State of TFI

In the limit of $h_f \to \infty$ the ground state for $N = 2$ written in the z-basis is

$$\rho_{groundstate}(h_f \to \infty) = \frac{1}{4} \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1, \end{bmatrix} \tag{A.1}$$

as can be calculated for large h_f by calculating the eigenvectors of the Hamiltonian and observe convergence to the above state. It is dominated by the alignment along the external field in x-direction.

A.2 KL-Divergence and Mutual Information as Entanglement Measures

Having a proper probability distribution $P(a)$ at hand after applying POVM gives us the possibility to investigate the system from an information theoretical perspective. The POVM description preserves factorizability, i.e. the probability distribution of a product state is a product distribution over statistically independent sets of variables $P(a) = \prod_i P(a_i)$. Thus the KL-Divergence between a probability distribution and the product of its marginals might be a measure of separability:

$$D_{KL}(P(a)|| \prod_i P(a_i)) = \sum_a P(a) \log\left(\frac{P(a)}{\prod_i P(a_i)} \right) \tag{A.1}$$

It is also called Mutual Information (MI). Note that it is basis dependent and a minimization over different bases gives a basis-independent measure for separability. This optimization over POVM bases might be an analogy to Schmidt-decomposition in the MPS framework. MI describes how much two subsystems are correlated, i.e. how much information they share. If we split the system into subsystems A and B, the marginalized probability distributions are then $P_A(a)$ and $P_B(a)$. MI might be a good measure for entanglement entropy if the basis is optimized in the way mentioned above.

B Appendix for Chapter 5

B.1 Derivation of Contrastive Divergence Learning Algorithm

In the following derivation of the learning algorithm, we follow [Hinton, 2002], the introduction of Contrastive Divergence. Just to underline the fact that each visible unit is independent of the others, we introduce the following notation of a product probability

$$\begin{aligned}
P^0(v) &= \frac{1}{Z}\sum_{\mathbf{h}} \exp(E(\mathbf{v},\mathbf{h})) \\
&= \frac{1}{Z}\sum_{h} \exp\Bigg(\sum_{i,j}(v_i W_{ij} h_j + h_j b_j + a_i v_i)\Bigg) \\
&= \frac{\sum_h \exp\Big(\sum_{i,j} v_i W_{ij} h_j + h_j b_j + a_i v_i\Big)}{\sum_{\tilde{v}}\sum_h \exp\Big(\sum_{i,j} \tilde{v}_i W_{ij} h_j + h_j b_j + a_i \tilde{v}_i\Big)} \\
&= \frac{\prod_i \sum_h \exp\Big(v_i \left(\sum_j W_{ij} h_j + a_i\right)\Big) \exp\Big(\sum_j h_j b_j\Big)}{\sum_{\tilde{v}_i}\prod_i \sum_h \exp\Big(\tilde{v}_i \left(\sum_j W_{ij} h_j + a_i\right)\Big) \exp\Big(\sum_j h_j b_j\Big)} \\
&\equiv \frac{\prod_i f_{i,\theta}(v)}{\sum_{\tilde{v}}\prod_i f_{i,\theta}(\tilde{v})}.
\end{aligned} \tag{B.1}$$

The KL-divergence between the probability of all data vectors and their equilibrium distribution can be written as

$$\begin{aligned}
D_{KL}(P^0 \parallel P^\infty) &= \sum_v P^0(v)\log\big(P^0(v)\big) - \sum_v P^0(v)\log(P^\infty(v)) \\
&= -\langle \log\big(P^0\big)\rangle_{P^0} - \langle \log(P^\infty)\rangle_{P^0}
\end{aligned} \tag{B.2}$$

where angled brackets denote an expectation value following the probability in the subscript. The expectation value of the logarithmic data distribution is not dependent on the network parameters. The variation of one network parameter θ_m yields

$$\begin{aligned}
-\frac{\partial}{\partial\theta_m}D(P^0 \parallel P^\infty) &= \frac{\partial}{\partial\theta_m}\left\langle \log(P^\infty)\right\rangle_{P^0} \\
&= \frac{\partial}{\partial\theta_m}\sum_v P^0(v)\log(P^\infty(v)) \\
&= \frac{\partial}{\partial\theta_m}\sum_v P^0(v)\log\left(\frac{\prod_i f_{i,\theta}(v)}{\sum_{\tilde{v}}\prod_i f_{i,\theta}(\tilde{v})}\right) \\
&= \sum_v P^0(v)\frac{\partial}{\partial\theta_m}\log\left(\prod_i f_{i,\theta}(v)\right) - \sum_v P^0(v)\frac{\partial}{\partial\theta_m}\log\left(\sum_{\tilde{v}}\prod_i f_{i,\theta}(\tilde{v})\right) \\
&= \sum_v P^0(v)\frac{\partial}{\partial\theta_m}\sum_i \log(f_{i,\theta}(v)) - \frac{\frac{\partial}{\partial\theta_m}\sum_{\tilde{v}}\prod_i f_{i,\theta}(\tilde{v})}{\sum_{\tilde{v}}\prod_i f_{i,\theta}(\tilde{v})} \\
&= \sum_v P^0(v)\frac{\partial}{\partial\theta_m}\log(f_{\theta_m}(v)) - \frac{1}{Z}\sum_{\tilde{v}}\frac{\partial}{\partial\theta_m}f_{\theta_m}(\tilde{v})\prod_{i\neq m} f_{i,\theta}(\tilde{v}) \\
&= \left\langle \frac{\partial}{\partial\theta_m}\log(f_{\theta_m}(v))\right\rangle_{P^0} - \sum_{\tilde{v}}\frac{\prod_i f_{i,\theta}(\tilde{v})}{Z}\frac{\frac{\partial}{\partial\theta_m}f_{\theta_m}(\tilde{v})}{f_{\theta_m}(\tilde{v})} \\
&= \left\langle \frac{\partial \log f_{\theta_m}}{\partial\theta}\right\rangle_{P^0} - \left\langle \frac{\partial \log f_{\theta_m}}{\partial\theta}\right\rangle_{P^\infty}.
\end{aligned} \tag{B.3}$$

This is essentially the difference of expectation values of the same object but evaluated at different probabilities. The first term is the expectation value over the learning data whereas the second term is the expectation value over the equilibrium distribution, resulting from the parameter dependence of the partition sum. The second term is computationally costly to evaluate since in principle one has to evolve the Markov chain of Gibbs reconstruction until convergence appears. Instead, one constructs a different quantity as distance measure, the difference of KL-divergences, such that the term resulting from the partition sum cancels out:

$$\begin{aligned}
&\frac{\partial}{\partial \theta_m} \left| D(P^0 \parallel P_\theta^\infty) - D(P_\theta^1 \parallel P_\theta^\infty) \right| \\
&= \left| \left\langle \frac{\partial \log(f_{\theta_m})}{\partial \theta_m} \right\rangle_{P^0} - \left\langle \frac{\partial \log(f_{\theta_m})}{\partial \theta_m} \right\rangle_{P_\theta^1} + \frac{\partial P_\theta^1}{\partial \theta_m} \frac{\partial D(P_\theta^1 \parallel P_\theta^\infty)}{\partial P_\theta^1} \right| \\
&\sim \left| \left\langle \frac{\partial \log(f_{\theta_m})}{\partial \theta_m} \right\rangle_{P^0} - \left\langle \frac{\partial \log(f_{\theta_m})}{\partial \theta_m} \right\rangle_{P_\theta^1} \right|,
\end{aligned} \tag{B.4}$$

where the last term in the second line results from the weight-dependence of the first reconstruction P_θ^1. Empirical evidence suggests that it can be dropped and learning is still successful [Hinton, 2002]. The interpretation of the above equation is: if the thermal distribution of the model P_θ^∞ parametrizes the data distribution P^0 well, one Gibbs sampling step samples from that same distribution. Evaluating the terms inside the expectation value brackets yields:

$$\begin{aligned}
\frac{\partial}{\partial W_{ij}} \log(f_W) &= \frac{\frac{\partial}{\partial W_{ij}} f_W}{f_W} \\
&= \frac{\sum_h v_i h_j \exp(v_i W_{ij} h_j + v_i a_i + h_j b_j)}{\sum_h \exp(v_i W_{ij} h_j + v_i a_i + h_j b_j)} \\
&= \frac{v_i \exp(v_i W_{ij} + b_j)}{\exp(v_i W_{ij} + b_j) + 1} \\
&= v_i \sigma(v_i W_{ij} + b_j) \\
&= v_i P(h_j | v)
\end{aligned} \tag{B.5}$$

We can now write Equation **??** for $\theta_m = W_{ij}$ as

$$\begin{aligned}
&\frac{\partial}{\partial W_{ij}} \left| D(P^0 \parallel P_\theta^\infty) - D(P_\theta^1 \parallel P_\theta^\infty) \right| \\
&= \left| \langle v_i h_j \rangle_{P^0} - \langle v_i h_j \rangle_{P_\theta^1} \right|,
\end{aligned} \tag{B.6}$$

where P_θ^1 is the probability distribution of the network after one updating step of Gibbs sampling.

C Appendix for Chapter 6

C.1 Graphical Representation and Calculation of POVM Equation of Motion

Let us define the following graphical representations for the needed objects in Equation 6.0.4. We also remind the definitions in the Section 4. The Figures C.1, C.2 and C.3 show the graphical representation for the probability distribution $P(a)$, density matrix ρ_{ij} and for an expectation value of a product operator $< O_1 O_2 \ldots >$.

As an example Hamiltonian, we choose the TFI model defined in Equation 2.0.2. In the POVM basis, Equation 4.2.4 induces the following coefficients:

$$Q_H = -\sum_i (Q_z^{a_i} \otimes Q_z^{a_{i+1}} + h_f \cdot Q_x^{a_i}) \tag{C.1.1}$$

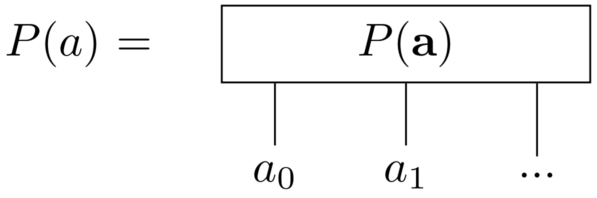


Figure C.1: Graphical representation of probability 4.2.1. $P(a) = \mathrm{Tr}\big(M^{(a)} \cdot \rho\big)$. $M^{(a)}$ is the POVM measurement set and ρ the density matrix. $P(a)$ contains all information about the state, $M^{(a)}$ is just an exact mapping to real positive numbers.

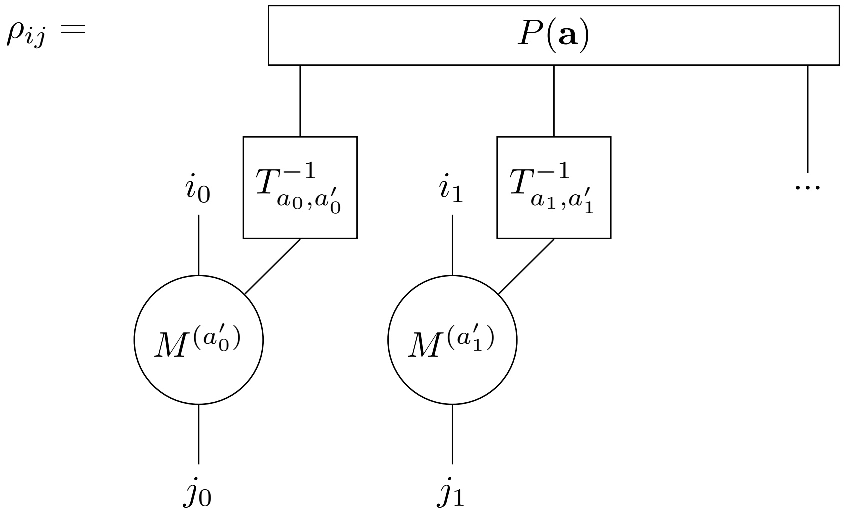


Figure C.2: Graphical representation of density matrix as it is defined in Equation 4.2.3 as a function of the probability $P^{(a)}$. $T^{-1}_{a,a'}$ is the overlap matrix 4.1.10 and M^{a}_{ij} is the POVM basis 4.1.3. Lines connecting two boxes are tensor contractions.

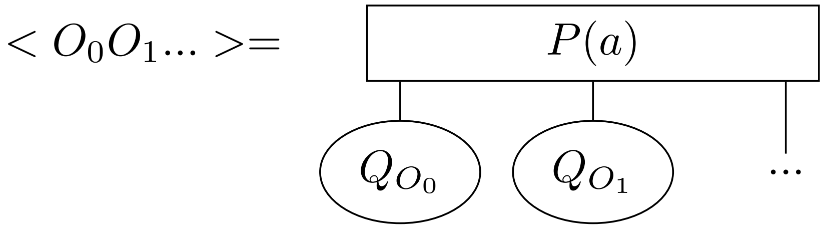


Figure C.3: Graphical representation of expectation values of product operator like in Equation C.1.2.

Thus, its expectation value is given by

$$\begin{aligned}
< H > &= P^{\mathbf{a}} \cdot Q_H^{\mathbf{a}} \\
&= -\sum_i P^{a_i,a_{i+1}} \cdot ((Q_z^{a_i} \otimes Q_z^{a_{i+1}} + h_f \cdot Q_x^{a_i}))
\end{aligned} \tag{C.1.2}$$

where we defined the marginalized probability by summing $P(a)$ over all other sites than i and $i+1$. For the energy expectation value of the TFIM, only two point correlations contribute. We define

$$P^{a_j} = \sum_{a_k} P^{a_j,a_k} \tag{C.1.3}$$

for unequal j, k. For $i, j = 1, 2$ we can write P^{a_j,a_k} as

$$\begin{aligned}
P^{a_1,a_2} &= \mathrm{Tr}((M^{a_1} \otimes M^{a_2})\rho) \\
&= \mathrm{Tr}\big((M^{a_1} \otimes M^{a_2})(\rho_1 \otimes \rho_2 + \rho_{1,2}^{connected})\big) \\
&= \mathrm{Tr}(M^{a_1}\rho_1) \cdot \mathrm{Tr}(M^{a_2}\rho_2) + \mathrm{Tr}\big((M^{a_1} \otimes M^{a_2})\rho_{1,2}^{connected}\big) \\
&= P^{a_1}P^{a_2} + P_{connected}^{a_1,a_2}.
\end{aligned} \tag{C.1.4}$$

We used the notation for reduced density matrices, where the density matrix of two sites is $\rho_{1,2}$, the reduced density matrix on site 1 is $\rho_1 = \mathrm{Tr}_2(\rho_{1,2})$ and analogue for site 2. Furthermore the connected density matrix is defined by the difference of the full density matrix and the outer product of its reduced density matrices $\rho_{1,2}^{connected} = \rho_{1,2} - \rho_1 \otimes \rho_2$. In particular, we see that Equation C.1.3 is still true as $\sum_{a_2} P_{connected}^{a_1,a_2} = 0$. The marginalized probability that contains two sites P^{a_j,a_k} and thus also captures two-point correlations, we will later call second order.

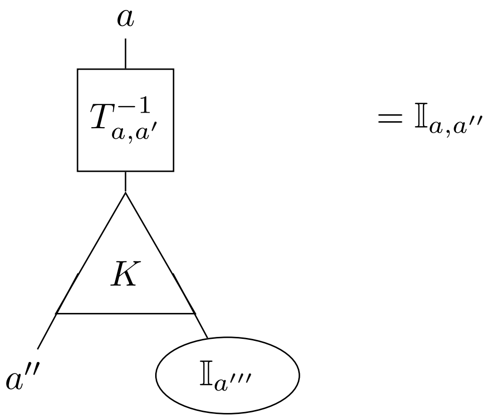


Figure C.4: Contraction identity given in Equation C.1.5. $T^{-1}_{a,a'}$ is the overlap matrix 4.1.10 and K is the trace over three M matrices defined in Equation 6.0.5. The identity with one index has shape 4 and is summed over, the right-hand side is a four dimensional Kronecker delta, the identity in POVM space.

Another import equality that will be needed in the following is:

$$\begin{aligned}\sum_{a''} T^{-1}_{a,a'} K^{a',a'',a'''} &= \sum_{a''} \operatorname{Tr}\Big(M^a M^{a'}\Big)^{-1} \operatorname{Tr}\Big(M^{a'} M^{a''} M^{a'''}\Big) \\ &= \operatorname{Tr}\Big(M^a M^{a'}\Big)^{-1} \operatorname{Tr}\left(M^{a'''} M^{a'} \sum_{a''} M^{a''}\right) \\ &= T^{-1}_{a,a'} T_{a'''a'} = \mathbb{I}_{a,a'''}\end{aligned} \tag{C.1.5}$$

$$\sum_{a'''} T^{-1}_{a,a'} K^{a',a'',a'''} = \mathbb{I}_{a,a''}$$

which follows directly from the norm condition Equation 4.1.3 and the symmetry of $T^{-1}_{a,a'}$. Its graphical representation is shown in Figure C.4.

The graphical representation of the above Equation is shown in Figure C.4. This has the consequence shown in Figure C.5. Especially, the triangle representing the tensor K vanishes, if the open index is summed over.

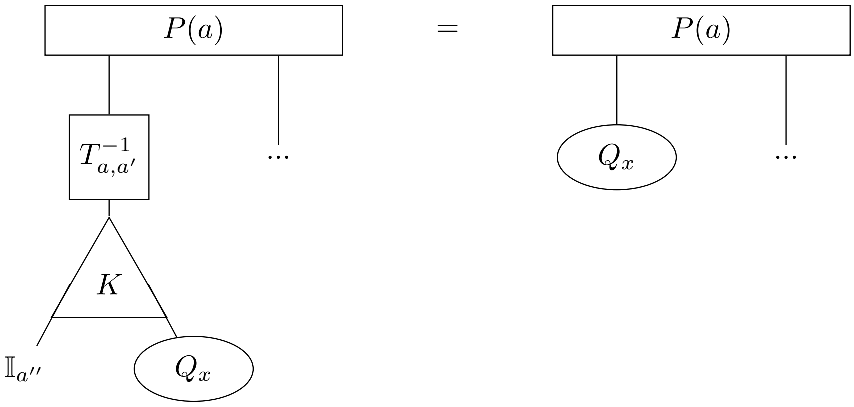


Figure C.5: Sum over open index a'' of a contribution of Equation 6.0.4 represented in Figure C.6. From the property of K (Equation 6.0.5) follows Figure C.4. When summing over the open index a'', the antisymmetric structure vanishes.

Altogether, Equation 6.0.4 can be represented as shown in Figure C.6.

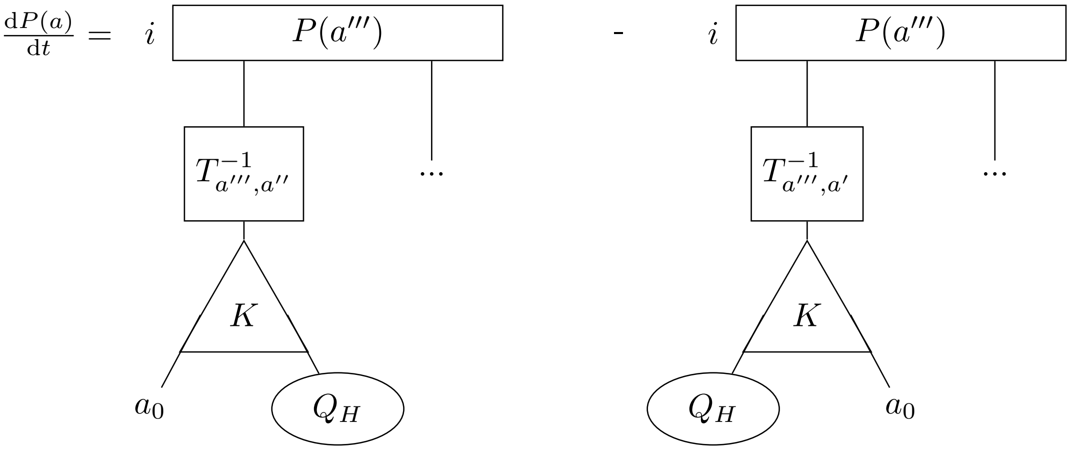


Figure C.6: Exact time evolution under a Hamiltonian of the probability $P(a)$ representing a density matrix, derived in Equation 6.0.4. $T^{-1}_{a''',a''}$ is the overlap matrix, K the trace over three POMVs and Q_H the coefficient of the Hamiltonian in the POVM basis.

We recognize the structure of the commutator and note that the tensor K is the only antisymmetric object. If we sum out all open indices, no matter what the operator coefficient Q_H is, the whole expression will vanish identically.

C.2 Local Hierarchy

The goal is to get rid of the exponential scaling with system size while only losing as little information as possible. The TFI model exhibits an inherent approximation scheme as there are only nearest-neighbour interactions. If one is interested in observables that only contain few-site correlations or even only local observables at site i, considering subsystems will probably be a good approximation. Therefore we sum over sites in Equation 6.0.4. Reminding that the index a is a multi-index for each site $a = \{a_1, a_2, \dots a_N\}$, we define the order of $P^{a_1, a_2, \cdots a_N}$ by the number of indices it carries and sum over all other indices of P^a in the POVM e.o.m..

In the following, we will illustrate what we mean by 'orders of $P(a)$' by applying the developed formalism to P^{a_1}, the probability distribution describing the site 1. We will see that it generates a hierarchy of local equations in a very natural way that couple to the two neighbors in the spin chain. P^{a_1} is first order of $P(a)$ because it is local and contains no correlations. It is the subsystem containing only the first site with all the information about the rest of the chain summed out. But as we can see (Equation C.2.1) its time evolution couples to the second order in $P(a)$ via its neighboring site 2 and N. One can say, that the order of P^a is the number of indices that it carries. We show their evolution equations in Equation C.2.2 and C.2.3.

We use a symbolic notation for the tensor $K^{a'a''a'''} - K^{a'a'''a''}$ and write it as K, knowing that the tensor contraction in the first index is with $T^{-1}_{a,a'}$ and the second and third are antisymmetrized in the given way. The contractions with the probability distribution at different sites are separated by a tensor product '$\otimes$', where the notation might be misleading. If there are two or more K in one term, one has to take the tensor product first and then antisymmetrize afterwards! Then the first order POVM equation of motion for the TFI model reads:

$$\begin{aligned}\frac{\mathrm{d}P^1}{\mathrm{d}t} = i(&P^{1,2} \cdot (T^{-1}KQ_z \otimes T^{-1}KQ_z) \\ &+ P^{1,N} \cdot (T^{-1}KQ_z \otimes T^{-1}KQ_z) \\ &+ P^1 \cdot (T^{-1}KQ_x)).\end{aligned} \tag{C.2.1}$$

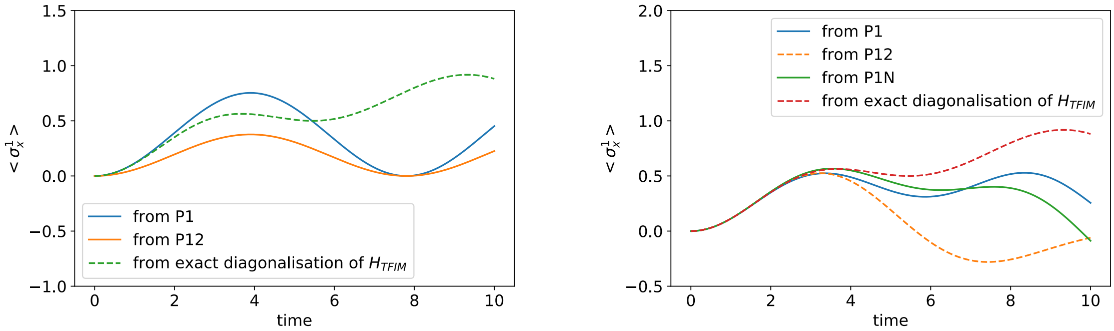
We recognize the terms from the Hamiltonian, the first two are the nearest-neighbour interaction of the first site to the second and last as we defined periodic boundary conditions. The third term is the interaction with the external magnetic field h_f, where we absorbed h_f into the coefficient Q_x. With this same notation, one can formulate the second order POVM equation of motion for the terms entering the first order for the first site, namely the second order with the two neighbors 2 and N:

$$\begin{aligned}\frac{\mathrm{d}P^{12}}{\mathrm{d}t} = i(&P^{1,2} \cdot (T^{-1}KQ_z \otimes T^{-1}KQ_z) \\ &+ P^{1,2,3} \cdot (\mathbb{I} \otimes T^{-1}KQ_z \otimes Q_z) \\ &+ P^{1,2,N} \cdot (T^{-1}KQ_z \otimes \mathbb{I} \otimes Q_z) \\ &+ P^{12} \cdot (T^{-1}KQ_x \otimes \mathbb{I}) \\ &+ P^{12} \cdot (\mathbb{I} \otimes T^{-1}KQ_x)),\end{aligned} \tag{C.2.2}$$

$$\begin{aligned}\frac{\mathrm{d}P^{1N}}{\mathrm{d}t} = i(&P^{1,N} \cdot (T^{-1}KQ_z \otimes T^{-1}KQ_z) \\ &+ P^{1,N-1,N} \cdot (\mathbb{I} \otimes Q_z \otimes T^{-1}KQ_z) \\ &+ P^{1,2,N} \cdot (T^{-1}KQ_z \otimes \mathbb{I} \otimes Q_z) \\ &+ P^{1N} \cdot (T^{-1}KQ_x \otimes \mathbb{I}) \\ &+ P^{1N} \cdot (\mathbb{I} \otimes T^{-1}KQ_x)).\end{aligned} \tag{C.2.3}$$

The identity $\mathbb{I}$ is four-dimensional and is contracted with the dimension of the probability distribution according to the order of terms in brackets. The hierarchy of equations can be continued up to the system size N, recovering the exact POVM e.o.m. Equation 7.0.1. One can still calculate local expectation values from all orders of the probability distribution, i.e. in the following example for site 1 and 2:

$$\begin{aligned}\langle \sigma_1^x \rangle &= \sum_a P^{a_1,a_2}(Q_x^{a_1} \otimes \mathbf{I}^{a_2}) \\ &= \sum_a P^{a_1} Q_x^{a_1}.\end{aligned} \tag{C.2.4}$$



(a) Time evolution of $\langle\sigma_x^1\rangle$ and its approximation to 2nd order truncated with a constant

(b) Time evolution of $\langle\sigma_x^1\rangle$ and its approximation to 3rd order truncated with a constant

Figure C.7: Time evolution of $\langle\sigma_x^1\rangle$ and its approximation to 2nd order (Equations C.2.1, C.2.2 and C.2.3) and 3rd order truncated with a constant. $\langle\sigma_x^1\rangle$ is calculated according to Equation C.2.4 and from exact diagonalization of Equation 6.0.3. The deviations of the approximate solution from the exact ones increase with time.

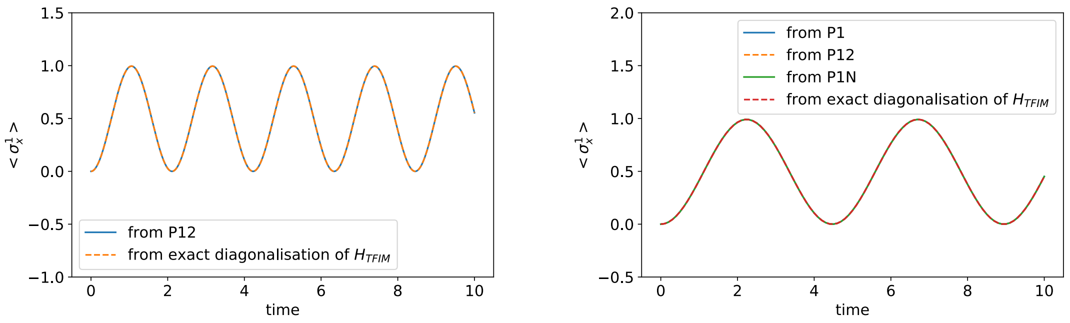
For an exact solution, the above equation is strictly true, for an approximation we will use the deviation as a marker of how trustworthy the approximation is.

C.2.1 Truncating With a Constant

We define the truncation of the hierarchy with a constant to neglect the time dependence of probability distributions with more than a given number of indices. When truncating with a constant at a given order, let us say two, we just set the third order of P^a to its initial value with no time dependence at all. That is, only the time dependence of local probability distributions and of the probability distributions that contain two-point correlations are considered. The rest gets neglected.

Implementing this set of coupled differential equations and extracting the expectation value $\langle\sigma_x^1\rangle$ yields the evolution shown in Figure C.7a. The result of implementing the third order approximation, i.e. including the time evolution for P^{123} and P^{12N} but setting P^{1234}, P^{123N} and P^{12N-1N} constant, is shown in Figure C.7b. For both cases the GHZ-state for initial conditions and the TFI model with magnetic field $h_f = 1.1$ and system size $N = 6$ was chosen.

The deviations of the approximate solution from the exact ones increase with



(a) 2nd order truncated with a constant $N = 2$

(b) 3rd order truncated with a constant $N = 3$

Figure C.8: Time evolution of $\langle \sigma_x^1 \rangle$ and its approximation to 2nd and 3rd order truncated with a constant, which becomes exact for $N = 2, 3$ respectively.

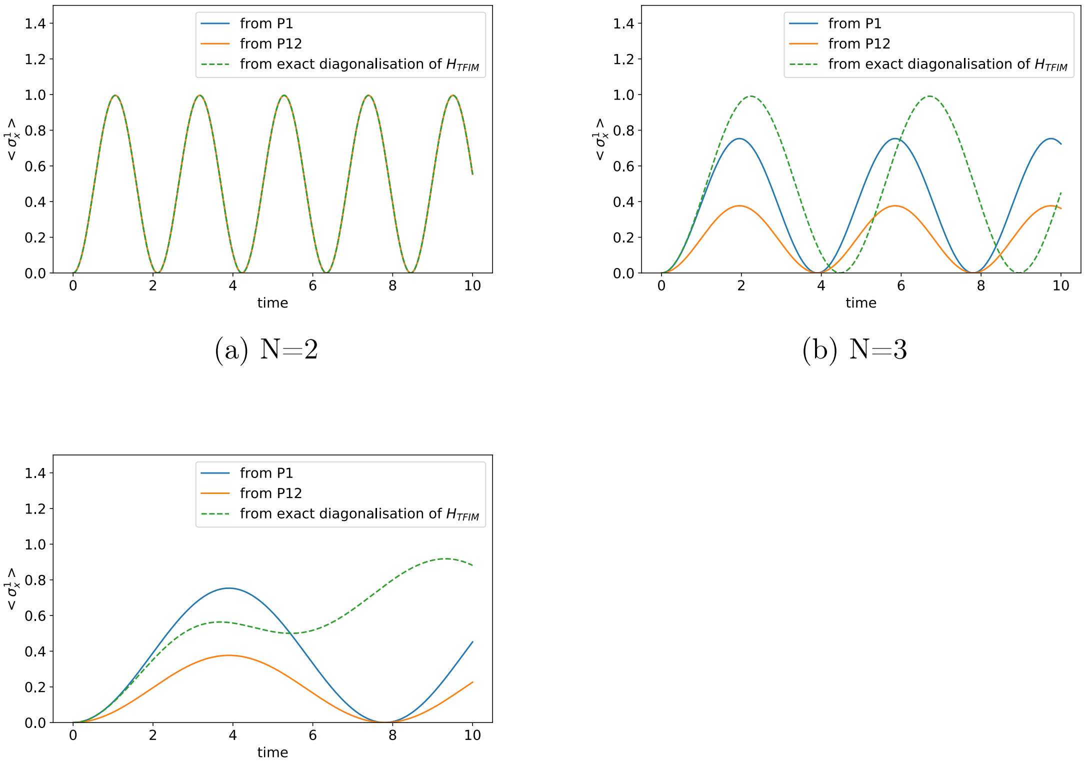
time. The exact solution is composed of at least two different oscillations, the approximation to second order only captures one oscillation. The approximation to third order shows qualitatively similar behavior to the exact solution but deviations increase after the first local maximum. Clearly, the deviation from the exact solution of the time evolution scales with time, which limits this approximation scheme to short time dynamics.

In the limit of small system sizes, i.e. for $N = 2, 3$ the respective approximations become exact. In Figure C.8 one can check that the equations lead to the correct evolution in the case of no approximation.

C.2.2 Truncating With Mean Field

In the mean field ansatz to a given order k, all correlation functions higher than k are approximated by their disconnected part to order k. For example, if only first order was considered, we would approximate the probability containing two sites by its product of local probabilities $P^{12} \approx P^1 P^2$.

We chose the GHZ-state for initial conditions and the TFI model with magnetic field $h_f = 1.1$ and system sizes $N = 2, 3, 6$. As we can see in Figure C.9 this approximation scheme to second order shows larger deviations for larger N. For $N = 6$, the approximation with mean field for the same system shows the same behavior as the approximation with a constant. Both approximation schemes neglect long distance correlations in the spin chain which are expected to build up over time in the TFI model [Czischek et al., 2018b]. Both approximations are only valid for short enough times.



(a) N=2

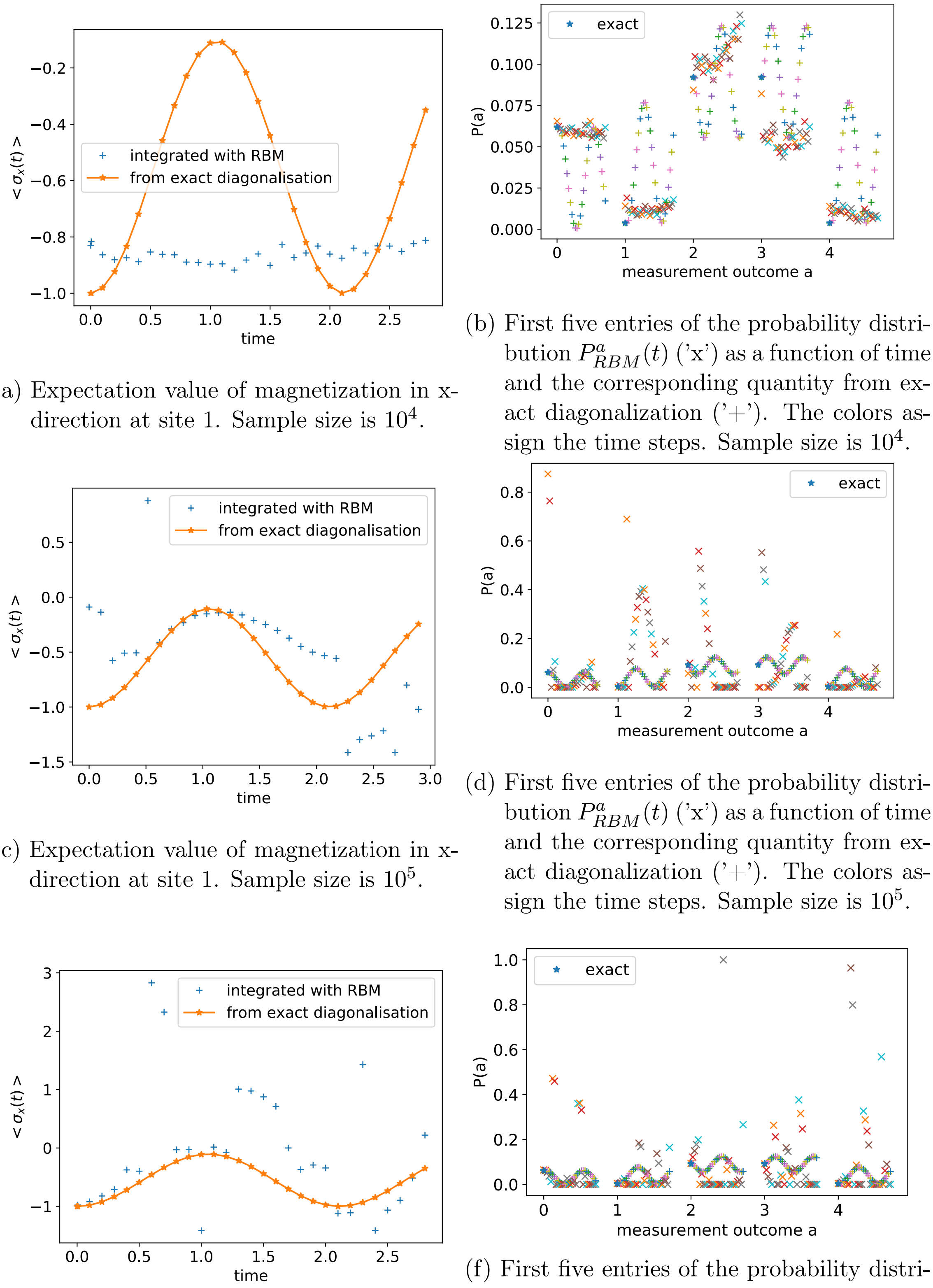
(b) N=3

(c) N=6

Figure C.9: Truncation of the local hierarchy with the mean field ansatz to second order described in the main text. We plot the magnetization in x-direction $\langle \sigma_x^1 \rangle$ at site one as a function of time with the two methods shown in Equation C.2.4 and exact diagonalization. Second order is exact for $N = 2$, deviations increase with system size. $N = 6$ compares to Figure C.7a, where the same system with truncation with a constant is considered.

D Appendix for Chapter 8

As described in the main text in Section 8.4, we investigate the effect of statistical errors on the NQS solver by choosing small sample sizes. In Figure D.1, we plot the results of the same NQS solver as in Figure 8.5, but now with sample size of 10^4, 10^5 and 10^6. The pluses '+' are the probabilities calculated from the density matrix after exact diagonalization, they are the exact solution. We observe that deviations are large and random for all three cases of sample size. This confirms our considerations regarding the sample size.



(a) Expectation value of magnetization in x-direction at site 1. Sample size is 10^4.

(b) First five entries of the probability distribution $P^a_{RBM}(t)$ ('x') as a function of time and the corresponding quantity from exact diagonalization ('+'). The colors assign the time steps. Sample size is 10^4.

(c) Expectation value of magnetization in x-direction at site 1. Sample size is 10^5.

(d) First five entries of the probability distribution $P^a_{RBM}(t)$ ('x') as a function of time and the corresponding quantity from exact diagonalization ('+'). The colors assign the time steps. Sample size is 10^5.

(e) Expectation value of magnetization in x-direction at site 1. Sample size is 10^6.

(f) First five entries of the probability distribution $P^a_{RBM}(t)$ ('x') as a function of time and the corresponding quantity from exact diagonalization ('+'). The colors assign the time steps. Sample size is 10^6.

Figure D.1: Time evolution by RBM integrator as described in the main text and exact diagonalization with sample size of 10^4 in the upper panels and 10^5 in the lower ones. $N = 2$, initial state is ground state of TFI Hamiltonian with $h_f = 100$ quenched to $h_f = 1.1$.

E Lists

E.1 List of Figures

F Bibliography

C. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006. URL https://www.microsoft.com/en-us/research/publication/pattern-recognition-machine-learning/.

D. C. Brody and L. P. Hughston. Information content for quantum states. *Journal of Mathematical Physics*, 41(5):2586–2592, 2000. doi: 10.1063/1.533260.

G. Carleo and M. Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017. doi: 10.1126/science.aag2302.

J. Carrasquilla, D. Luo, F. Pérez, A. Milsted, B. K. Clark, M. Volkovs, and L. Aolita. Probabilistic simulation of quantum circuits with the transformer, 2019a. URL https://arxiv.org/abs/1912.11052v1.

J. Carrasquilla, G. Torlai, R. G. Melko, and L. Aolita. Reconstructing quantum states with generative models. *Nature Machine Intelligence*, 1(3):155–161, 2019b. doi: 10.1038/s42256-019-0028-1.

H. Chen and A. F. Murray. A continuous restricted boltzmann machine with a hardware-amenable learning algorithm. In *Proceedings of the International Conference on Artificial Neural Networks*, ICANN '02, page 358–363, Berlin, Heidelberg, 2002. Springer-Verlag. doi: 10.5555/646259.684468.

J. Chu, H. Wang, H. Meng, P. Jin, and T. Li. Restricted boltzmann machines with gaussian visible units guided by pairwise constraints. *IEEE transactions on cybernetics*, 49:4321–4334, 2018. doi: 10.1109/TCYB.2018.2863601.

S. Czischek, M. Gärttner, and T. Gasenzer. Quenches near ising quantum criticality as a challenge for artificial neural networks. *Phys. Rev. B*, 98:024311, 2018a. doi: 10.1103/PhysRevB.98.024311.

S. Czischek, M. Gärttner, M. Oberthaler, M. Kastner, and T. Gasenzer. Quenches near criticality of the quantum ising chain-power and limitations of the discrete truncated wigner approximation. *Quantum Sci. Technol.,*, 4:014006, 2018b. doi: 10.1088/2058-9565/aae3f7.

Y. Freund and D. Haussler. Unsupervised learning of distributions of binary vectors using 2-layer networks. In *Proceedings of the 4th International Conference on Neural Information Processing Systems*, NIPS'91, page 912–919, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc. doi: 10.5555/2986916.2987028.

A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, and D.B. Rubin. *Bayesian data analysis, third edition*. Columbia University, New York, 2013. URL http://www.stat.columbia.edu/~gelman/book/.

S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, 1984. doi: 10.1109/TPAMI.1984.4767596.

G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. doi: 10.1126/science.1127647.

G.E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002. doi: 10.1162/089976602760128018.

G.E. Hinton. *A Practical Guide to Training Restricted Boltzmann Machines*, pages 599–619. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. doi: 10.1007/978-3-642-35289-8_32.

P. Kunkel, M. Prüfer, S. Lannig, R. Rosa-Medina, A. Bonnin, M. Gärttner, H. Strobel, and M. Oberthaler. Simultaneous readout of noncommuting collective spin observables beyond the standard quantum limit. *Phys. Rev. Lett.*, 123:063603, 2019. doi: 10.1103/PhysRevLett.123.063603.

J. R. Movellan. A learning theorem for networks at detailed stochastic equilibrium. *Neural Computation*, 10(5):1157–1178, 1998. doi: 10.1162/089976698300017395.

U. Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96–192, Jan 2011. doi: 10.1016/j.aop.2010.09.012.

P. Smolensky. *Information Processing in Dynamical Systems: Foundations of Harmony Theory*, page 194–281. MIT Press, Cambridge, MA, USA, 1986. doi: 10.5555/104279.104290.

M. Troyer and U.-J. Wiese. Computational complexity and fundamental limitations to fermionic quantum monte carlo simulations. *Physical Review Letters*, 94(17), 2005. doi: 10.1103/physrevlett.94.170201.

G Acknowledgement

I would like to thank my supervisor Thomas Gasenzer who gave me the opportunity to work in his group and follow my interest in machine learning and quantum many-body systems. Thanks to Martin Gärttner for his interest in and support of my work, including many fruitful discussions and literature advice. Thanks to Stefanie Czischek for an open door and very helpful feedback. Thanks for inspirational discussions to Markus Schmitt, Markus Oberthaler, Adrian Braemer, and many others.

Erklärung:

Ich versichere, dass ich diese Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Felix Behrens

Heidelberg, den 18.04.2020 ....Felix Behrens ....