TUM School of Computation, Information and Technology
Technische Universität München

Machine Learning Methods for the Simulation and Characterization of Many-Body Quantum Systems

Irene López Gutiérrez

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades einer

Doktorin der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz:

Prof. Dr. Helmut Seidel

Prüfende der Dissertation:

1. Prof. Dr. Christian Mendl
2. Prof. Giuseppe Carleo

Die Dissertation wurde am 14.02.2024 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 24.06.2024 angenommen.

Para mamá

Abstract

Quantum computing has the potential to revolutionize scientific computing by leveraging phenomena such as superposition and entanglement. Theoretically, quantum algorithms could offer significant speed-ups in tackling some NP-complete problems. However, due to the inherent noise in current quantum computers, the classical simulation of quantum systems remains crucial for their development, as it holds an important role in developing and testing new algorithms, as well as in obtaining a better understanding of such systems.

Nevertheless, the classical simulation of quantum systems is strongly limited by the memory requirements, which increase exponentially with the number of qubits due to the tensor product structure of quantum systems. An approach to mitigate this problem involves the use of the so-called Neural Network Quantum States (NNQS), i.e., neural networks used as a variational Ansatz for the wavefunction. This thesis explores NNQS for simulating the time evolution of many-body quantum systems. Specifically, we introduce an optimization method for NNQS based on gradient descent applied to a numerical integration cost function. This approach offers a less sensitive alternative compared to Stochastic Reconfiguration.

Classical simulation is also valuable as a means to study the noise present in quantum computers. Quantum process tomography (QPT), a fundamental tool in quantum information theory, is able to characterize the underlying experimental processes. However, QPT faces challenges in resource requirements, particularly for systems with a large number of qubits. In this thesis, we explore an alternative approach to QPT, utilizing optimization methods to reduce the number of measurements required. The Takens embedding theorem provides the overarching mathematical framework for this technique. This approach requires only a few known initial states in combination with time-delayed measurements to reconstruct the unitary map and corresponding Hamiltonian of the time evolution.

Acknowledgements

First and foremost I want to thank my supervisor, Christian Mendl, for his support and guidance during these past five years. I greatly admire your dedication to research and your insatiable curiosity. I have learned a lot working besides you, and not just quantum physics. The quantum group at our chair has become a welcoming and positive environment mainly because of your leadership.

I also want to thank all of the people with whom I have worked closely, both in Dresden and in Munich. Thank you in particular to Felix Dietrich, for sharing his expertise and ideas on system identification. Thank you to all of the staff at the Chair of Scientific Computing in TU Dresden, where it all started. And also to the Chair of Scientific Computing in TU Munich, where the journey ended. Thank you for the interesting discussions and the willingness to share your knowledge.

During these past few years many of my co-workers became friends, and to them I feel like a separate expression of gratitude is necessary. Thank you Santiago, for making each day enjoyable and filled with laughs. Thank you Shuo, for your kindness and your constant smile. And to Keefe, Hayden and Paul, for those intense weekends of puzzle solving.

And finally, thank you to my friends and family back in Spain: Mamá, gracias por siempre esperar lo mejor de nosotros y por empujarnos hasta conseguirlo. Gracias por haber sido siempre una fuente de cariño y fuerza. Papá, gracias por tu paciencia, por tu flexibilidad y por tu apoyo, siempre incondicional. A Miriam, porque eres, has sido y serás siempre mi modelo a seguir. Y a Pere, por 53 ± 7 años de amistad; qué decirte que no te haya dicho ya antes.

Contents

List of Figures

Acronyms

ANN	Artificial Neural Networks.
CNN	Convolutional Neural Network.
IC-POVM	informationally complete POVM.
MPS	Matrix Product States.
MUB	Mutually Unbiased Bases.
NNQS	Neural Network Quantum States.
POVM	positive operator-valued measure.
QPT	Quantum Process Tomography.
QST	Quantum State Tomography.
RBM	Restricted Boltzmann Machine.
RNN	Recurrent Neural Network.
SIC-POVM	symmetric IC-POVM.
SR	Stochastic Reconfiguration.
t-VMC	time-dependent variational Monte Carlo.
TFI	Transverse-field Ising.

1 Introduction

In modern physics, the field of quantum mechanics has stood out as one of the most profound and revolutionary discoveries of the 20th century. Quantum mechanics has not only redefined our understanding of the fundamental building blocks of nature, but has also paved the way for the development of new technologies fundamentally different from their classical counterparts. One of these technologies is that of quantum computing, which promises to solve some hard problems exponentially faster than classical computers, revolutionizing fields such as cryptography and optimization. At the heart of this quantum revolution lies a profound and complex challenge: the study of many-body quantum systems.

Quantum mechanics has proven to be extraordinarily successful in describing the behaviour of individual particles and small assemblies. However, as we venture into the realm of systems containing a large number of particles, the complexity and richness of the phenomena that emerge is staggering. The difficulties associated with understanding many-body quantum systems come from the complex interactions among the constituent particles, resulting in high correlations (i.e., entanglement). As the number of particles grows, the computational resources required to precisely describe the quantum states of these systems increase exponentially, presenting an infeasible challenge for methods reliant on exact matrix diagonalization. Consequently, understanding and predicting the properties of many-body quantum systems has become an active area of research, combining theory with state-of-the-art computational methods.

This thesis deals with two important facets of the study of these types of systems: their simulation and their characterization. In the field of quantum simulation, the goal is to model an arbitrary quantum system without experimentally constructing it exactly. Instead its behaviours are emulated through a different format, such as numerical computations or the use of a different physical system which is easier to realize and control experimentally.

Quantum computing introduces a revolutionary paradigm for addressing the simulation of many-body quantum systems. In fact, it was originally proposed by Feynman in the early 1980s, as a response to the growing realization that simulating quantum systems using classical computers was exceedingly challenging [1]. Feynman's key insight was that if nature inherently operates according to the laws of quantum mechanics, then a computer that also follows those same principles would be ideally suited for simulating them. In other words: to simulate quantum systems efficiently, one should use a quantum computer itself.

Since then, there have been many successes in the road to build a universal quantum computer. Although there is not a definite leader in the hardware race, many research groups around the globe have been able to implement intermediate-scale computers using superconducting qubits [2], ion traps [3], and neutral atoms [4] among others. These prototypes have already been utilized to prove their advantage in solving certain toy mathematical problems [5]. And, on top of that, their number of qubits increases year after year, with the current largest (self-reported) quantum computer containing over a thousand qubits [6].

Nevertheless, while it is undeniable that quantum computing holds great potential for solving many-body quantum problems, it is essential to acknowledge the challenges associated with building and operating them. Current quantum hardware is still in the nascent stages of development, with limited gate operations and high error rates. Quantum simulation could hold the key to a better understanding of the associated technologies, allowing us to refine them. Thus, it becomes necessary to utilize classical technologies as a bridge during these intermediate phases of development.

As an example of this, consider tensor networks: a technique for the simulations of quantum systems, which uses variational tensors as a basis to model the underlying quantum states [7]. By using them, it has been possible to obtain numerical approximations of even the largest computations performed on state-of-the-art quantum computers [8]. This allows us to benchmark the performance of the developed technologies, both in terms of correctness and computational cost.

The field of classical Computer Science, in fact, has also been experiencing a revolution during the past few years. The development and perfecting of the graphical processing unit (GPU), with its specialized processing power on linear algebra computations [9], has paved the way for a boom in Artificial Intelligence applications. Our day to day has been filled with mentions of recommender systems, image recognition and even natural language generation, all of which are based on the idea of machine learning –i.e. optimization based computational models designed to "learn" how to perform specific tasks [10].

This wide spread of the use of machine learning models has also had consequences in physics research. Their methods have been introduced as an efficient way of solving partial differential equations, which are central to most areas of physics [11]. Not only that, but they have been used to handle the growing amount of data generated by astrophysics and particle physics experiments [12, 13].

In this paradigm, an obvious question arises: how can we best utilize machine learning to solve the quantum many-body problem? Here, we will talk about a simulation framework related to –but less mature than– tensor networks. It is a method based on the use of Neural Networks [14] as variational Ansatz to represent quantum systems. We will first discuss in more general terms the challenges associated with the classical simulation of quantum systems, to understand how the correlations between parts of the system can lead to significant computational bottlenecks. We will then focus on the numerical challenges of neural networks themselves, and will propose an optimization algorithm meant to approximate the numerical integration of the Schrödinger time dependent equation:

$$\frac{\partial}{\partial t}|\psi(t)\rangle = \frac{-i}{\hbar}H|\psi(t)\rangle. \tag{1.1}$$

In fact, throughout the thesis we pay particularly close attention to the study of quantum dynamics, as opposed to focusing on finding steady states or groundstates. The time-evolution of quantum systems is pivotal in the realization of quantum computers, as quantum gates are, fundamentally, dynamical processes.

The second facet of the quantum many-body problem with which we are concerned in this thesis is that of system characterization. The goal of characterization is to develop a comprehensive and accurate representation of a physical system by means of measuring some of its properties. Beyond quantum technologies, this is relevant in some engineering applications as a means to automatically adjust their control parameters according to the observed behaviour [15]. In the context of quantum mechanics, however, it comes with additional challenges.

This is primarily because of two factors: the so-called "wavefunction collapse" and the aforementioned exponential growth in dimension of quantum systems. The wavefunction collapse is a concept that refers to the sudden –and seemingly instantaneous– reduction of a particle's wave function to a specific state when a measurement is made. The nature of this collapse is subject to interpretation, and different philosophical schools of thought offer different perspectives on how to understand it. However, it is a postulated axiom of quantum mechanics, and undeniably observed in experiments. This is the case of the quantum Zeno effect where a system is kept static in a specific state by applying frequent measurements to it [16].

Such a collapse introduces a significant challenge in the characterization of quantum systems. Every time a property of the state is measured, the system collapses and, in doing so, all other relevant information is destroyed. In this thesis we will provide a full description of a protocol called Quantum Tomography, used to overcome this problem. But, although through Quantum Tomography it is possible to fully characterize arbitrary systems and their dynamics, it is an extremely involved method in its experimental realization. This is because of the high-dimensionality of the systems. As the system grows, more measurements bases are necessary to accurately reconstruct the system.

Thus, in this work we will propose an alternative approach based on similar optimization techniques to those used machine learning. It is worth noting that in the classical physics realm, machine learning has already been successfully applied to the problem of system identification [17], so extending this to the field of quantum mechanics seems only natural.

2 Quantum Many-body Systems

The first two main chapters in this thesis serves as the gateway to some of the core concepts upon which the subsequent discussions will be built. The motivation behind dedicating two entire chapters to the theoretical background stems from the recognition that the subject matter of this thesis is interdisciplinary, drawing from both quantum physics and machine learning. However, those with expertise in those two areas are encouraged to jump directly to Chapter 4, where the nucleus of the thesis begins.

We have divided the theoretical background into two sections: the first one covering the most relevant quantum physics concepts –note that the reader is expected to at least have a basic understanding of quantum states and unitary operations, cf. [18, 19]–, and the second one covering the basics of gradient descent and neural networks.

Let us, thus, begin with a description of the quantum mechanical concepts that will be relevant throughout the thesis. For an extended discussion of the topics covered here, we recommend Refs. [20–22], all of which have been consulted during the writing of this thesis.

First, we will delve into the concept of entanglement, placing special emphasis on how entanglement entropy scales in quantum many-body systems (Sec. 2.1). This scaling is of the utmost importance in the context of quantum simulation, as it is the primary limiting factor of the various variational Ansatze used to represent quantum systems.

Next, we will cover the so-called Ising model which has been used as application system for the methods proposed in this work. The quantum Ising model has become a standard test case in the literature of many-body quantum systems. This is due to its simplicity and interesting properties, which can be generalized to more realistic physical systems. We will therefore present the model itself together with some of its most relevant characteristics in Sec. 2.2.

Last, but not least, we will review the concept of quantum tomography for the characterization of systems in Sec. 2.3. We will describe in detail the standard algorithms for state and process tomography. We will also briefly touch on the mathematical description of quantum processes, as some readers may be unfamiliar with it and it is necessary for the process tomography protocol.

2.1 Entanglement

Entanglement is arguably the most important and perplexing phenomenon of quantum physics. It refers to the impossibility of describing two (or more) components of a larger system as separate smaller systems. It was first predicted in a paper by Einstein, Podolski and Rosen [23] in an attempt to prove the incompleteness of quantum theory. Many years later, however, the violation of Bell's inequality [24] was verified experimentally [25–28], leading physicists to two fundamental conclusions: quantum theory is complete (at least in regards to missing hidden variables), and entanglement is a true intrinsic property of quantum systems.

But what do we mean when we say that the system components cannot be described separately? To visualize it, let us consider a bipartite system conformed by subsystems A and B. An arbitrary state, $|\psi\rangle$, of this system can be written as

$$|\psi\rangle = \sum_{j,k} C_{j,k} \, |j\rangle_A \, |k\rangle_B \,, \tag{2.1}$$

where $C_{j,k} \in \mathbb{C}$, and $\{|j\rangle_A\}$ and $\{|k\rangle_B\}$ are the orthonormal basis states for the Hilbert spaces of A and B respectively. Recall that quantum systems preserve a tensor product structure, as the Hilbert space of the larger system can be written as the tensor product of its subsystems' Hilbert spaces (that is, $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$). Separating the state $|\psi\rangle$ into two descriptions for A and B would therefore require finding a representation such that

$$|\psi\rangle = |\phi\rangle_A \otimes |\chi\rangle_B \,. \tag{2.2}$$

Going back to Eq. (2.1), we can apply a Singular Value Decomposition to C in order to arrive to the so-called Schmidt decomposition:

$$|\psi\rangle = \sum_{j} \lambda_j \, |j\rangle_A \, |j\rangle_B \,, \tag{2.3}$$

where $\lambda_j \geq 0$ are called the Schmidt coefficients and satisfy the relationship $\sum_j \lambda^2 = 1$. On top of that, $\{|j\rangle_A\}$ is a set of orthonormal states for subsystem A, and similarly for $\{|j\rangle_B\}$. Although the full proof has been omitted here, it can be read on Ref. [18]. The important point is that the state $|\psi\rangle$ will only be separable into the subsystems if there is only one non-zero λ_j. Otherwise, we say that the subsystems A and B are entangled.

Entanglement is also often understood in terms of measurement correlations between the subsystems. This can be intuitively understood from a simple inspection of Eq. (2.3).

If subsystem A is measured to be in state $|j\rangle_A$, this inevitably leads to the conclusion that $|\psi\rangle$ has collapsed to $|\psi\rangle = |j\rangle_A |j\rangle_B$. A subsequent (or, in fact, simultaneous) measure of subsystem B can only lead to $|j\rangle_B$.

For a more formal analysis, let us consider a local measurement operator M_i, where the subscript denotes the subsystem to which it is being applied. The correlation of observable M between the subsystems is given by

$$\operatorname{corr}(M_A, M_B) = \langle\psi|M_A \otimes M_B|\psi\rangle - \langle\psi|M_A|\psi\rangle \langle\psi|M_B|\psi\rangle . \tag{2.4}$$

This can only be non-zero when $|\psi\rangle \neq |\phi\rangle_A \otimes |\chi\rangle_B$. Ergo, the correlations of such a local observable must be a consequence of entanglement, or 'non-separableness'.

2.1.1 Entanglement entropy

At this point an obvious question arises: how do we quantify entanglement? When thinking of entanglement in the context of correlations it becomes apparent that certain sets of subsystems may exhibit greater entanglement than others. However, relying solely on correlations for quantifying this phenomenon poses the challenge of being dependent on the model in use. In order to compare the entanglement within different physical systems we would be required to find observables which are common to all. Although this is certainly possible, here we will focus on an alternative approach: *the von Neumann entropy.*

This concept is very closely related to that of Shannon entropy in information theory. Often denoted as H, the Shannon entropy is defined as

$$H(X) = -\sum_i p(X_i) \log(p(X_i)), \tag{2.5}$$

where $p(X_i)$ corresponds to the probability of outcome i for the random variable X. This quantity provides a measure of the average "surprise" associated with observing a particular outcome of the random variable. It reaches its maximum when all outcomes are equally likely, indicating the highest uncertainty, and has a minimum at 0 for the case when a single outcome has probability 1, a.k.a when there is no surprise in the outcome of the variable X.

The von Neumann entropy generalizes this concept for quantum systems. For a quantum system composed of N subsystems, denoted as $i = 0, \ldots, N$, the von Neumann entropy of subsystem i, characterized by its reduced density matrix ρ_i, is expressed as

$$S(\rho_i) = -\operatorname{tr}(\rho_i \log(\rho_i)). \tag{2.6}$$

Similar to the Shannon entropy, it reaches its maximum when the quantum state ρ_i is a maximally mixed state (where all possible outcomes are equally probable), and it decreases as the state becomes more pure. A completely unentangled subsystem i will be a pure state (i.e. it has complete separable description) and therefore its von Neumann entropy will be zero.

2.1.2 Area and Volume Laws

The exponential growth of a quantum system's Hilbert space is very often cited as the main challenge for quantum simulation. However, we need to be precise with our phrasing. If quantum systems showed no entanglement between its subcomponents, nothing would stop us from leveraging the tensor product structure of the Hilbert space, allowing us to simulate each of the subsystems separately. Hence, the primary limiting factor is not the exponential growth of the Hilbert space *per se*, but the "non-separableness" of its components: its entanglement.

Note that the specific value of the entanglement entropy depends on how we decide to split the quantum system. Following this insight, it becomes imperative to study in more detail how the entanglement entropy scales with the size of the considered subsystems, as, ultimately, the efficacy of various Ansatze for quantum simulation hinges on their ability to accurately reproduce such scaling behaviours.

A state is said to follow a **volume law** scaling if, for a bipartition into subsystems A and B, we have

$$S(\rho_A) \sim |A|, \tag{2.7}$$

where $|A|$ is the size of subsystem A. Typically, arbitrary quantum states are presumed to conform to this scaling pattern. This assumption is based on the established proof that quantum states randomly selected from the Hilbert space tend to be volume states with a very high probability (see Refs. [29, 30]). Hence, an optimal Ansatz for quantum simulation should, at a minimum, be able to capture an entropy scaling of this magnitude.

There is, however, an interesting class of systems for which the entanglement entropy scales slower. Consider, for example, the lattice system depicted in Fig. 2.1, and let us denote the boundary length of subsystem A as $|\partial A|$. Then, the system is said to follow an **area law** scaling if it holds that

$$S(\rho_A) \sim |\partial A|. \tag{2.8}$$

Admittedly, the states conforming to this law occupy a very small portion of the Hilbert space. Nevertheless, they are states of great physical relevance in a variety of fields, from black hole physics [31] to quantum many-body systems. Within the scope of this thesis is the case of the ground states of local, gapped Hamiltonians, which follow an area law scaling [32–34].

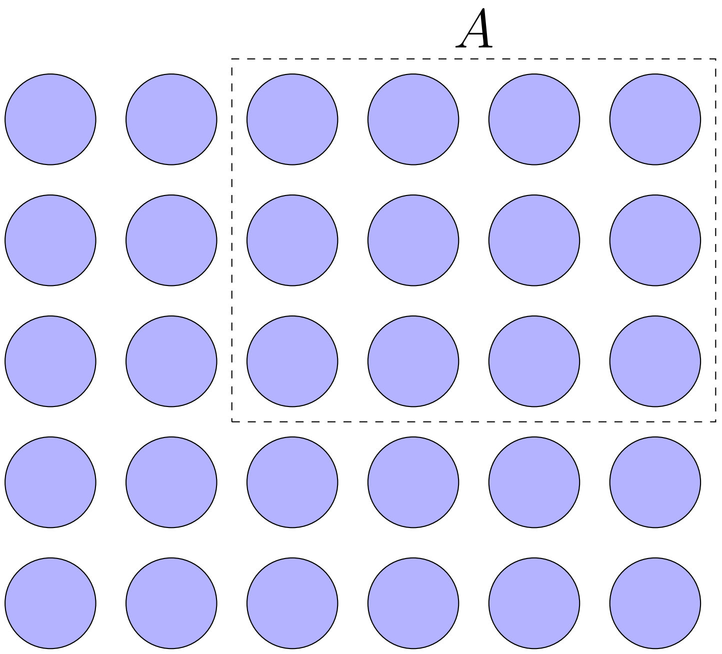


Figure 2.1: Bipartition of a lattice quantum system

In the context of entanglement scaling, we refer to the lattice site size of A as its “volume”, whereas the lattice length of its boundary is called its “area”.

2.1.3 Time-evolution

Since we are interested in the simulation of a system’s dynamics –and not just particular steady states– it also becomes necessary to study how the von Neumann entropy behaves as a function of time. Even if we found an Ansatz capable of capturing the entanglement of an initial state, it would not be guaranteed to capture the potential increase in entanglement throughout the time trajectory.

In this direction, Ref. [35] presents an analysis of the time evolution of the entanglement entropy for a partition of L spins within a 1D chain. The time trajectory considered is the result of the "quench" of a Hamiltonian $H(\lambda)$. The state is initialized at the groundstate of $H(\lambda_1)$, after which the physical parameter is tuned to a different value, λ_2, and the initial state is allowed to evolve under the modified Hamiltonian $H(\lambda_2)$. The time evolution itself is computed with two different approaches: one based on conformal field theory, and a second analytic computation for a particular solvable system. In both calculations, the same behaviour is observed. The entanglement entropy, S, grows linearly up to a time $t = L/2$ (note that the units here are such that the maximum speed of spin waves is 1). After this point, $S(t)$ asymptotically converges to a volume law behaviour. Concisely, this means that

$$S(t) \sim \begin{cases} t & \text{for } t \leq L/2 \\ L & \text{for } t > L/2. \end{cases} \tag{2.9}$$

Moreover, when considering the quench of a local, gapped Hamiltonian, we may use the fact that the groundstate is an area law state to bound the entanglement entropy of subsystem A:

$$S_A(t) - S_A(0) \leq c|\partial A|t, \tag{2.10}$$

where $c > 0$ is a constant depending on the strength of the interaction between the system components [36, 37]. Note that these behaviours have been verified for a variety of lattice models, using both analytical and numerical approaches [38–40].

This growth in entanglement presents a significant challenge for the classical simulation of quantum many-body dynamics. In Chapter 4 we will discuss an approach for the simulation of quantum time-evolution using neural networks as an Ansatz. Due to their high flexibility in architecture, neural networks have the potential to exhibit a volume law scaling behaviour, which makes them excellent candidates for achieving the simulation of systems beyond $t = L/2$.

2.2 Ising Model

As it may have become evident from the preceding discussions on entanglement, the field of quantum many-body systems frequently revolves around lattice structures. The Ising model in particular is a very popular choice of model to study, for reasons that will hopefully become apparent throughout this section. Named after physicist Ernst Ising, who introduced it in his 1925 doctoral thesis [41], the Ising model serves as a mathematical framework for investigating the behaviour of magnetic spins within a system. Its quantum version, called the Transverse-field Ising (TFI) model to distinguish it from its classical counterpart, admits several formulations.

Let us first consider a more general system: one where spins may present any orientation, and therefore their interactions are not restricted to any particular axis. This is called the Heisenberg model, and its Hamiltonian is given by

$$H = -\sum_{\langle i,j\rangle} \left(J^x_{i,j}\sigma^x_i\sigma^x_j + J^y_{i,j}\sigma^y_i\sigma^y_j + J^z_{i,j}\sigma^z_i\sigma^z_j \right) - \sum_i h_i\sigma^x_i. \tag{2.11}$$

Here, the constants $J^{\mu}_{i,j}$, with $\mu = x, y, z$, determine the interaction strength between the spins i and j. Note that the sum is over neighbouring lattice sites, denoted by $\langle i, j\rangle$. On the other hand, h_i corresponds to the strength of an external magnetic field being applied at site i. The operators σ^{μ}_i are the X, Y and Z Pauli matrices acting on site i. Without loss of generality, in the above formula we have chosen to orient the external magnetic field along the x-axis.

The TFI is a specific case of this model, where we restrict the spin interactions to an axis perpendicular to the external magnetic field. Again, without loss of generality to our analysis in this section, we may choose the interactions to occur on the z-axis, in which case the TFI Hamiltonian becomes

$$H = -\sum_{\langle i,j\rangle} J_{i,j}\sigma^z_i\sigma^z_j - \sum_i h_i\sigma^x_i. \tag{2.12}$$

Note that this still describes a disordered system, where the magnetic field and the interactions can present differences between the lattice sites. Sometimes, however, we may want to further simplify the picture and consider only uniform fields, in which case the Hamiltonian is given by

$$H = -J\sum_{\langle i,j\rangle} \sigma^z_i\sigma^z_j - h\sum_i \sigma^x_i. \tag{2.13}$$

2.2.1 Phases and critical points

One particularly interesting property of the TFI model is the fact that, despite its simplicity, it displays two distinct phases –and therefore a critical transition point between them. The phases arise as a consequence of the interplay between the external field and the spin interactions.

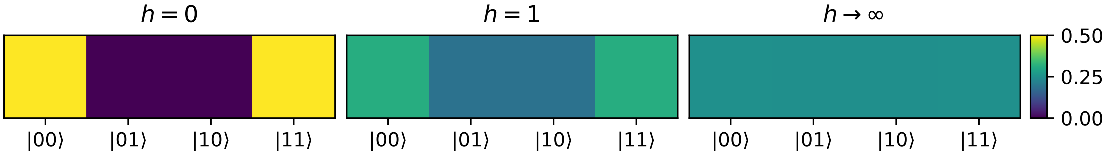


Figure 2.2: Groundstate probabilities of the uniform Ising model with two particles

Probability of measuring each of the basis states in the groundstates at extreme points of the phases: $h = 0, 1$ and ∞. Note that, given the degeneracy of these groundstates, we have opted for the maximally entangled state for the sake of simplicity.

In the absence of an external field (a.k.a $h = 0$), the spins exhibit a tendency to align themselves in the same direction. Consequently, the ground states manifest as $|\uparrow\uparrow \ldots \uparrow\rangle$, $|\downarrow\downarrow \ldots \downarrow\rangle$ or any superposition of the two. This is typically called the ferromagnetic phase.

On the other hand, as the strength of the external magnetic field overpowers the interactions between lattice sites, a.k.a $|h/J| \gg 1$, the spins will align themselves in the field axis (in the particular case of Eq. (2.13), the x-axis). The groundstate is therefore $|\rightarrow, \rightarrow \cdots \rightarrow\rangle$. This is called the paramagnetic phase. Fig. 2.2 visually illustrates these phases for a system comprising only two sites, with the coupling constant assumed to be $J = 1$.

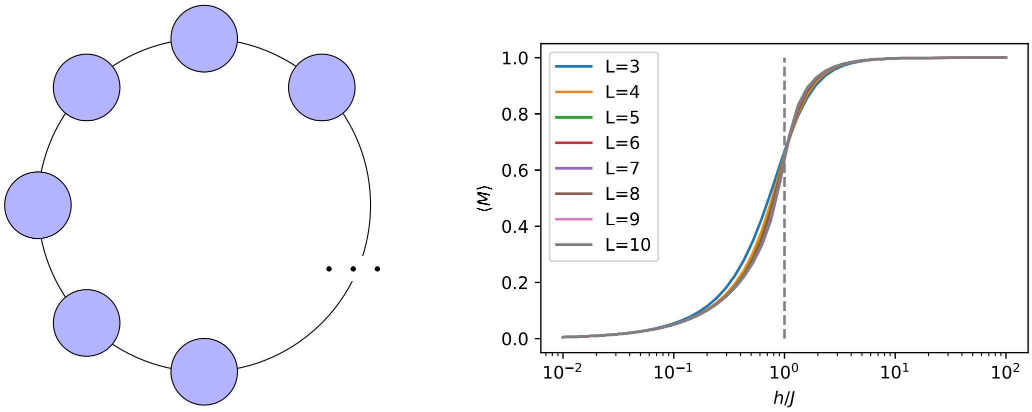
A convenient method for visualizing the phase transition, and consequently identifying the critical point, involves plotting the magnetization along a specific axis for different values of h/J. Let us consider the average magnetization along the x-axis, for which we can define the observable

$$M = \frac{1}{L} \sum_i \sigma_i^x, \tag{2.14}$$

where L is the number of sites in the system.

Fig. 2.4b shows how the expectation value of M in the groundstate changes as the external field becomes stronger relative to the inter-site couplings. The underlying models in this plot are 1D chains with periodic boundary conditions, as shown in Fig. 2.4a.

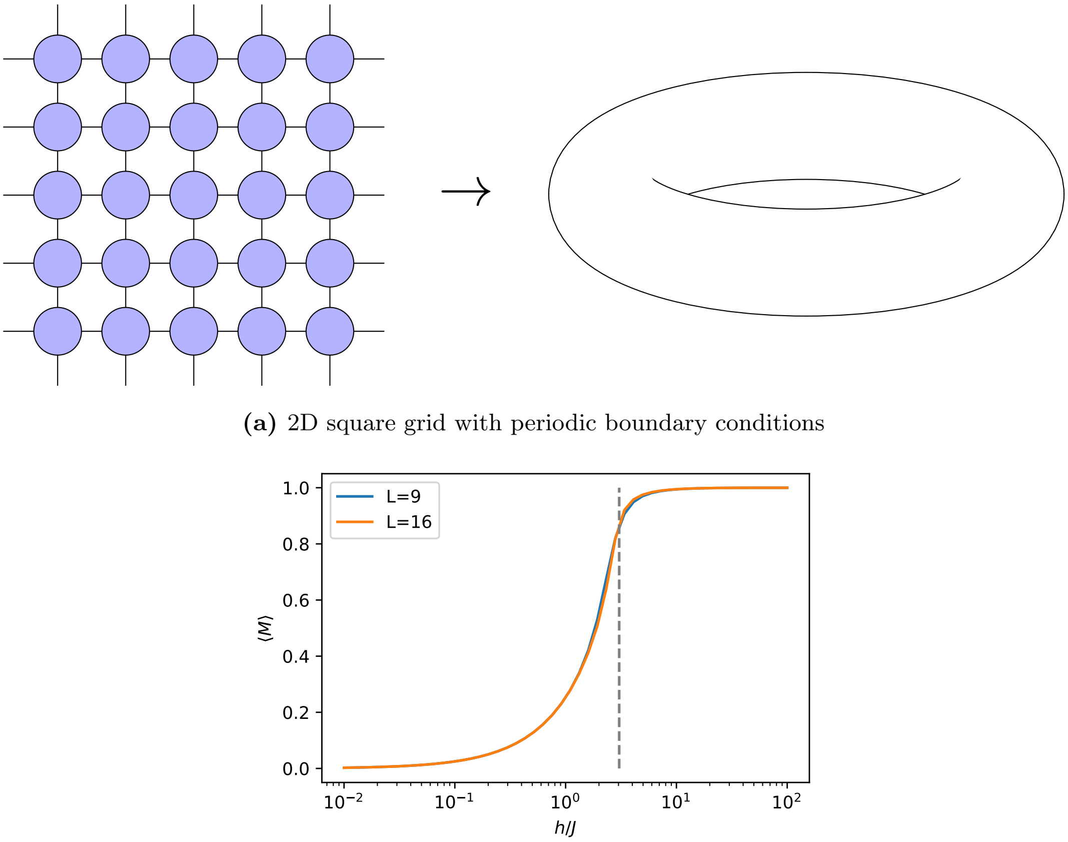
Note that the behaviour here seems very similar to that of a phase transition, from a regime where the average magnetization tends to 0, to a regime where it tends to 1. Notably, the critical point appears to be approximately at $|h/J| = 1$, in agreement with the analytical computations performed on the subject [42]. This point is independent of system size, at least for the periodic boundary conditions considered here.



(a) 1D lattice chain with periodic boundary conditions

(b) Average magnetization of an Ising chain for different field strengths

Figure 2.3: Phase transition for a uniform 1D Ising model



(a) 2D square grid with periodic boundary conditions

(b) Average magnetization of a square grid for different field strengths

Figure 2.4: Phase transition for a uniform 2D squared Ising model

For higher dimensions, however, finding the critical point becomes a harder problem, as the interactions between spins become very dependent on the shape of the underlying lattice. If we repeat our previous analysis for a 2D squared lattice, we will notice that the critical point appears to have moved slightly to higher values of $|h/J|$ (see Fig. 2.4). Studies done with Cluster Monte Carlo analysis, show the phase transition point to be at $h \sim 3.04438(2)$ for such a lattice, if we fix $J = 1$ [43].

2.2.2 Entanglement in the TFI model

As covered in Sec. 2.1, our capabilities to classically simulate a quantum many-body system are primarily limited by the amount of entanglement in the system. Let us thus investigate the behaviour of the entanglement entropy for the groundstate at both phases and at the critical point, as well as its time evolution.

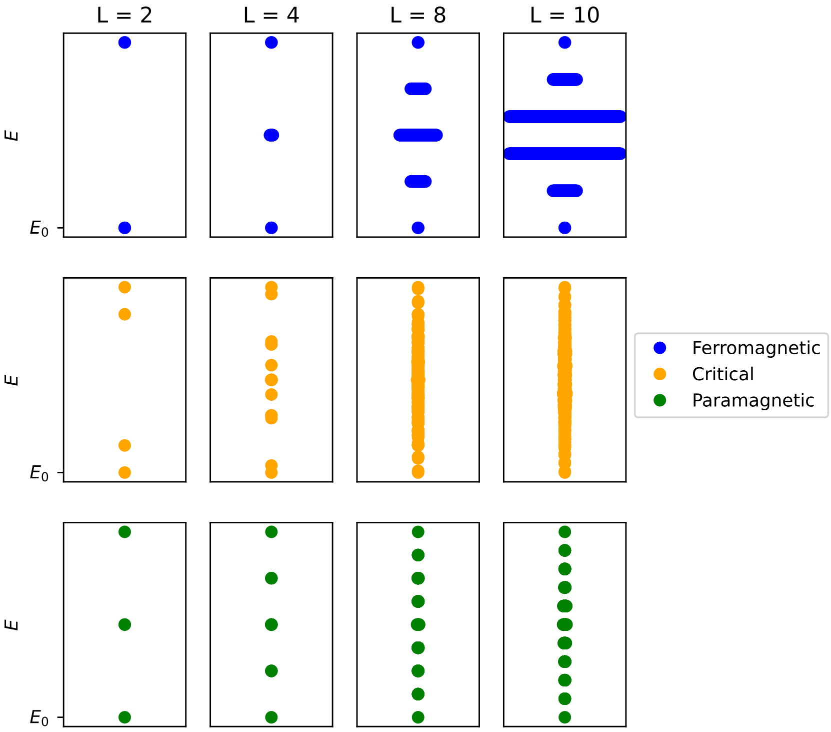
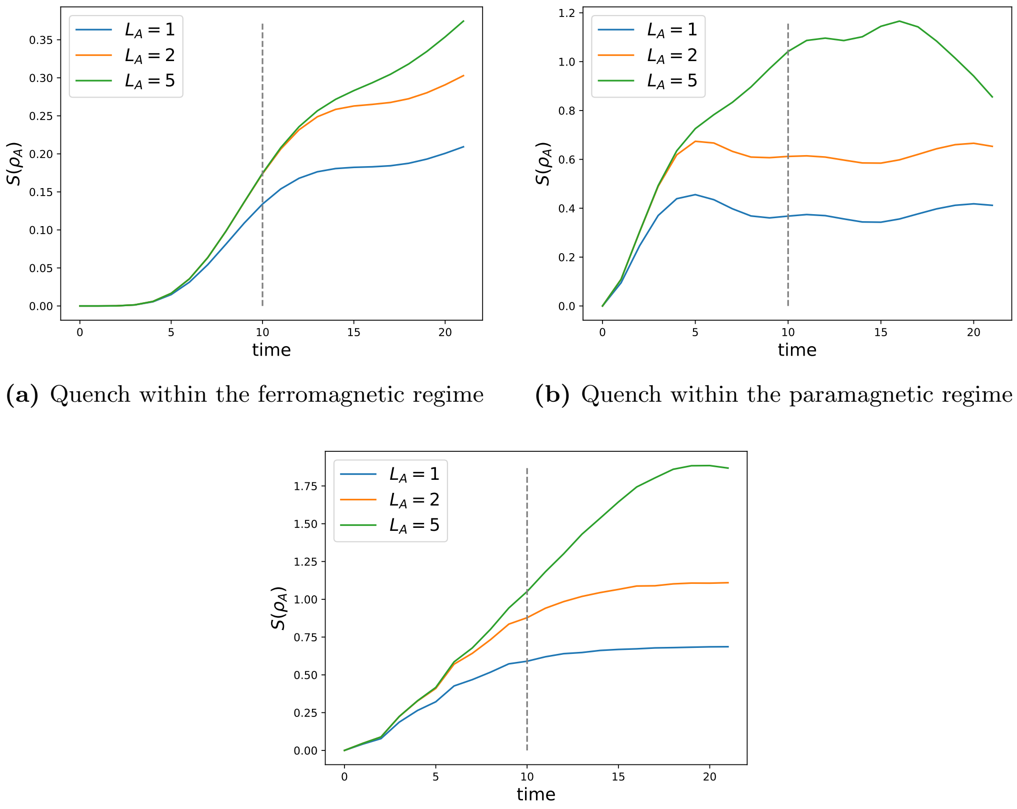


Figure 2.5: Eigenvalue spectrum for a variety of TFI Hamiltonians

This figure is meant as a visual representation of the spectral gap's asymptotic behaviour. Both in the ferromagnetic and paramagnetic regimes, there is a noticeable gap between the groundstate energy and the energy of the first excited state. At the critical point, however, the gap quickly vanishes as we increase the system's size.

Recall that the groundstates of local, gapped Hamiltonians exhibit an area law scaling. The Ising model is inherently local; so the only uncertainty lies in whether, in the thermodynamic limit $L \to \infty$, there exists a finite gap between the groundstate energy and the energy of the first excited state. It turns out that the asymptotic behaviour of this gap depends on the value of h. For 1D chains, it can be shown that the gap at the critical points vanishes as the system size increases, while the Hamiltonians in the ferromagnetic and paramagnetic phases are always gapped [21]. Specifically, the size of the gap outside of the critical point is $|2J|$. Although we will not prove this here, Fig. 2.5 should help as a an intuitive visualization of this behaviour.

The same is true for higher dimensions: as we approach the critical point –different for each grid structure– the gap in the Hamiltonian decreases. For more details on this, we refer the reader to Ref. [44] and references therein. This tells us that, outside of the critical point, the groundstate of the TFI model displays an area law scaling of the entanglement entropy.



(a) Quench within the ferromagnetic regime

(b) Quench within the paramagnetic regime

(c) Quench across the critical point

Figure 2.6: Entanglement entropy for 1D Ising chain as a function of time

Having established this, we can use Eq. (2.9) to understand how entanglement will behave as a function of time. For a Hamiltonian quench, the initial state will be an area law state. The upper bound for the entanglement entropy then increases linearly with time, until it reaches a volume law scaling. As a concrete example, consider the 1D uniform Ising chain with $L = 10$ lattice sites. When we time evolve the system across different domains and consider the entanglement entropy of subsystems of different size L_A, we observe the expected behaviour. For $t \ll L$, L_A has little impact on the entanglement. But, as time progresses, there is a noticeable difference in the entanglement increase for different subsystem sizes, which is consistent with a volume law scaling (see Fig. 2.6).

2.2.3 Applications

By this point it may already be evident why the TFI model elicits a considerable degree of theoretical interest. However, it is worth noting that the Ising model is more than a mere thought experiment. Beyond its theoretical appeal, the Ising model is a very accurate description of a variety of real world systems.

The TFI model has a straightforward application to condensed matter systems with magnetic interactions. Such is the case of ferroelectric materials [45, 46], for which the model was first introduced. Furthermore, it can be applied to a variety of magnetic molecules [47], as well as materials containing Rydberg states [48,49]. But, perhaps more relevant to this thesis are its multiple applications in the field of quantum computing.

As a first example, consider quantum annealing, a quantum computing approach utilized for solving optimization problems. In this method, the problem to be solved is mapped to a qubit grid, which then undergoes a carefully constructed time evolution in order to reach its groundstate, i.e. the solution to the optimization. The mathematical framework used to model the qubit lattice is, in fact, the TFI model [50, 51].

Some of the competing hardware proposals for a universal digital computer are also very closely connected to the Ising model. This is clearly true for ion trap computers, as their qubits are, at its core, spins interacting among them and subject to external forces [52,53]. A similar parallelism can be done for neutral and Rydberg atoms [4,54,55]. On top of this, quantum dots display behaviours that can be explained through an Ising model of their electrons [56, 57].

The striking similarities between the Ising model and many experimental set-ups have fostered a strong symbiotic relationship between theory and experiment. While theory serves as a benchmark to control and correct noise or calibration issues, experiments allow us to test our theoretical understanding of the model and push our simulation capabilities [55, 58–62]. This, combined with its simple formulation and non-trivial behaviour, has established the Ising model as the primary model for evaluating simulation and characterization methods.

2.3 Quantum Tomography

While the preceding two sections have primarily focused on the simulation of quantum many-body systems, characterizing them is equally important and poses significant challenges. It is well-established that a single copy of a quantum state cannot be entirely observed. The application of a measurement operator induces the wavefunction collapse, leading to the loss of information associated with all non-commuting operators. Therefore, the question arises: how can we be sure that our quantum devices are indeed managing the desired states and executing the intended operations? Although there exist various methods to estimate errors in experiments, Quantum Tomography stands out as the only approach providing a comprehensive account of states, their dynamics, and the types of errors occurring in the system.

In the next sections, we will describe the standard textbook approach for performing Quantum Tomography, paying particular attention to its cost in terms of resources. What will become apparent is that its demands in number of measurement bases grow exponentially with the system size, making it impractical for experiments involving a large number of qubits.

2.3.1 Quantum State Tomography

Let us thus begin by considering how to determine an unknown quantum state via Quantum State Tomography (QST). Because of the wavefunction collapse, we know that a single copy of the state, which we will denote ρ, is not sufficient. Instead, we need to be able to measure multiple times, so we assume that we have multiple copies of ρ available to us.

In QST, we aim to reconstruct the density matrix for the system by a positive operator-valued measure (POVM). This measure is a generalization of projection-valued measurements, such that its measurement values correspond to operators instead of vectors. To be more concrete, a POVM is a set of positive semi-definite Hermitian matrices $\{F_i\}$ such that

$$\sum_i F_i = I. \tag{2.15}$$

The probability of obtaining measurement outcome i is given by

$$p(i) = \mathrm{tr}(\rho F_i), \tag{2.16}$$

after which the state will have collapsed to F_i. This tells us that it is always possible to describe an arbitrary quantum state ρ by using a set of projection-operators $\{|\psi_i\rangle \langle\psi_i|\}$:

$$\rho = \sum_i \mathrm{tr}\left(\rho \,|\psi_i\rangle \langle\psi_i|\right) |\psi_i\rangle \langle\psi_i| \,. \tag{2.17}$$

Thus, utilizing an appropriately chosen set of observables we should be able to reconstruct the full state by estimating their expectation values and applying Eq. (2.17). There is a caveat, however. The density matrix ρ contains $d^2 - 1$ free parameters. Hence, in order for all entries of ρ to be uniquely determined we need at least d^2 projections in the set. This is called an informationally complete POVM (IC-POVM). In the special case where the set has exactly d^2 projection operators, it is called a minimal IC-POVM.

Although finding such a minimal IC-POVM is not a trivial task, it is worth mentioning two important examples which have been shown to be relatively resource efficient. The first one is the symmetric IC-POVM (SIC-POVM): a minimal set composed of d^2 projections which have a constant inner product between them. That is,

$$|\langle\psi_i|\psi_j\rangle|^2 = \begin{cases} 1/(1+d) & \text{if } i \neq j \\ 1 & \text{if } i = j. \end{cases} \tag{2.18}$$

Although examples of SIC-POVMs have been found for Hilbert spaces of a low number of dimensions, it is still an open problem whether they exist for an arbitrary dimension d [63].

The second example of an efficient IC-POVM are the so-called Mutually Unbiased Bases (MUB). They are closely connected to SIC-POVMs, but their formulation is slightly more general. Consider again a Hilbert space of arbitrary dimension d. Two bases in this space are said to be “mutually unbiased” if each of the vectors of one have a constant projection on the basis vectors of the other, and vice versa. If such a set of MUB exists, the number of bases in the set is at most $d + 1$.

Similarly to the previous example, the existence of such a set of bases has not been proven for arbitrary dimensions. However, its more relaxed requirements compared to SIC-POVMs have proven helpful in finding concrete sets for specific dimensions. Moreover, their existence has been proven when the number of dimensions is a power of a primer number (particularly relevant for qubit systems, for which $d = 2^N$) and they have been shown to be the optimal ensemble of projections to minimize statistical error [64]. Note that this is not as efficient in terms of required measurements, as it involves up to $d^2 + d$ projection operators.

When dealing with a qubit system it is straightforward to find such a set. For a single qubit, the set of normalized Pauli matrices ($\{\sigma_i\} = \{I/\sqrt{2}, X/\sqrt{2}, Y/\sqrt{2}, Z/\sqrt{2}\}$ for $i = 0, 1, 2, 3$) forms an orthonormal matrix basis in its Hilbert space. Therefore, their respective eigenvectors form a set of MUB. In particular, by renormalizing them we obtain the following IC-POVM:

$$\left\{ \frac{|0\rangle\langle 0|}{3}, \frac{|1\rangle\langle 1|}{3}, \frac{|+\rangle\langle +|}{3}, \frac{|-\rangle\langle -|}{3}, \frac{|1+i\rangle\langle 1+i|}{6}, \frac{|1-i\rangle\langle 1-i|}{6} \right\} \tag{2.19}$$

For systems of size N, we can extend the above by taking advantage of the tensor product structure of the Hilbert space to form their orthonormal bases:

$$\sigma_{\vec{v}} = \bigotimes_j \sigma_{v_j}, \tag{2.20}$$

where the vector $\vec{v} = (v_1, \ldots, v_{2^N})$ has all entries taken from the set 0, 1, 2, 3. As in the single-qubit scenario, the POVM would be constructed from the renormalized eigenprojections of these operators.

2.3.2 Quantum Process Tomography

Now that we know how to reconstruct an unknown static quantum state, we can aim to generalize the protocol to the characterization of the system's dynamics. This is called Quantum Process Tomography (QPT). But before that, let us recall that the dynamics of any quantum system are described by a set of Kraus operators E_k such that

$$\rho' = \mathcal{E}(\rho) = \sum_k E_k \rho E_k^\dagger \tag{2.21}$$

where ρ denotes the initial state and ρ' is the output of the process. If the process has a probability 1 of occurring, then $\sum_k E_k^\dagger E_k = I$. The Kraus operators can be decomposed into a different set of known operators, $\{B_m\}$, which form a basis such that

$$E_k = \sum_m e_{k,m} B_m \tag{2.22}$$

with $e_{k,m} \in \mathbb{C}$. The process may then be rewritten into this basis:

$$\mathcal{E}(\rho) = \sum_{m,n} B_m \rho B_n^\dagger \left(\sum_k e_{k,m} e_{n,k}^* \right) . \tag{2.23}$$

Hence, reconstructing the matrix $\chi_{m,n} = \sum_k e_{k,m} e_{n,k}^*$ would allow us to fully characterize the process.

But how do we determine χ experimentally? Consider a set of matrices $\{\rho_i\}$ which form a basis in the space of density matrices, and let us assume that we can initialize our system to be in any of the states in this set. By letting the state evolve under the process and performing QST on the output state, we could determine the coefficients $\lambda_{i,j}$ such that

$$\mathcal{E}(\rho_j) = \sum_i \lambda_{i,j} \rho_i. \tag{2.24}$$

Equating this with the expression in Eq. (2.23) yields

$$\sum_{m,n} \chi_{m,n} B_m \rho_j B_n^\dagger = \sum_i \lambda_{i,j} \rho_i. \tag{2.25}$$

where the only unknown is χ. Moreover, note that $B_m \rho_j B_n^\dagger$ can also be decomposed in the basis $\{\rho_i\}$,

$$B_m \rho_j B_n^\dagger = \sum_i \beta_{i,j}^{n,m} \rho_i, \tag{2.26}$$

which simplifies Eq. (2.25) into

$$\sum_{m,n} \chi_{m,n} \beta_{i,j}^{n,m} = \lambda_{i,j}. \tag{2.27}$$

This expression can be vectorized into a linear system, which can be solved with a variety of numerical approaches. We will not discuss these algorithms here. In turn, let us stop and consider the dimensionality of the system.

A general linear map of acting on $d \times d$ matrices will have d^4 free parameters. However, in our case χ is restricted to only those matrices which satisfy the trace preserving requirement $\sum_k E_k^\dagger E_k = \mathrm{I}$. Thus, it has $d^4 - d^2$ free parameters. In order for Eq. (2.27) to be a determined system we need to perform QST on at least $d^4 - d^2$ time-evolved basis states.

How does this translate in terms of resource cost for a qubit system? For a system of size N, the dimension of its Hilbert space is 2^N. Hence, to fully determine a process in this system we would need to compute the expectation value of $2^N + 1$ observables (using MUB) for $16^N - 4^N$ basis states. In practice that means collecting statistical data of $(4^N + 2^N)(16^N - 4^N)$ projection measurements.

It should not be surprising that such requirements make QPT completely impracticable for experiments with more that one or two qubits. In Chapter 5, an alternative approach based on time-delayed measurements will be discussed, for which only a few initial states and measurement bases are required.

3 Machine Learning methods

Having covered the most fundamental quantum physics concepts relevant to the thesis' core, in the subsequent discussion we will move on to the description of some of the computational methods used. In particular, we will focus on those methods which are more closely related to the field of machine learning. It is important to note that machine learning is a very loosely defined concept, and the reader may feel –and rightly so– that some of the methods described here fall under the broader scope of optimization techniques. This is particularly true for Sec. 3.1, covering gradient descent. However, for the sake of conciseness and due to its relevance in the field, we have chosen to include it here.

In fact Sec. 3.1 will be relevant for the two main chapters of this thesis: Chapts. 4 and 5. After introducing gradient descent, we will provide an introduction to neural networks in Sec. 3.2, which is meant to be concise yet as comprehensive as possible, starting from the most fundamental concepts and ending with Deep Learning. And finally, we will see how gradient descent can be applied efficiently in the context of neural networks, in Sec. 3.3.

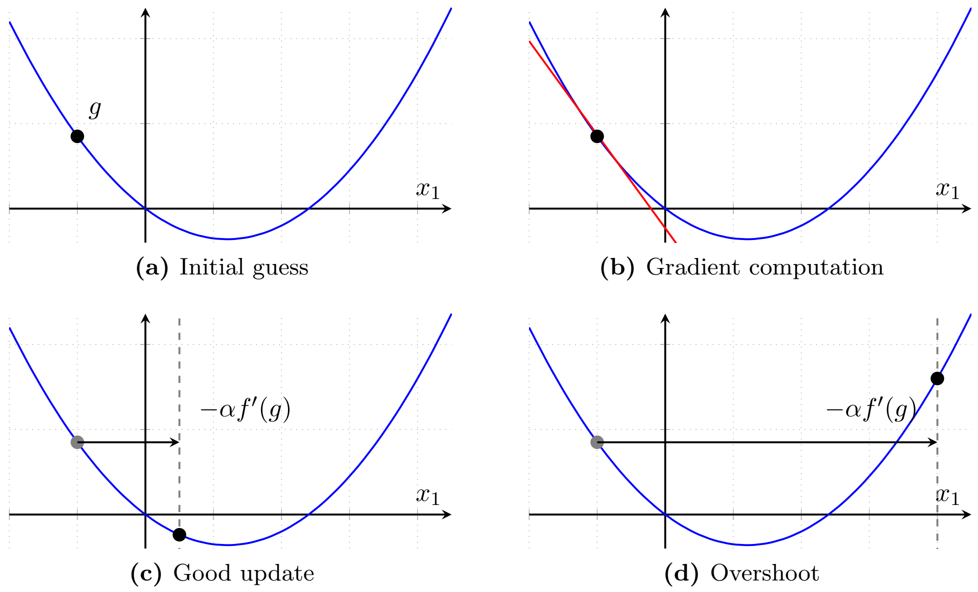
For a more detailed and extended description of the numerical methods associated with the construction and optimization of neural networks we refer the reader to Ref. [65], which has been used as a reference for this chapter.

3.1 Gradient Descent

Gradient descent is the cornerstone of computational optimization, as it forms the basis for a widely used family of algorithms designed to identify the minimum of an unknown function. Described in general terms, the process consists of iterative updates to an initial random guess, with the ultimate goal of progressively approaching the global or local minimum of the target function. To better understand how this works in practice, let us consider a function $f(\vec{x})$ with a *single, global* minimum, $\vec{x}_0$.

When the full analytical form of $f(\vec{x})$ is known, finding and identifying its extrema is possible through the use of its first and second derivatives with respect to each variable in $\vec{x}$. However, there are times when finding these derivatives may not be possible. For example, $f(\vec{x})$, may be the output of a black box computation, or even a computation so complex that it effectively acts as a black box. It could also be that $\vec{x}$ lives in a very high dimensional space, and since finding the extrema requires solving a linear system of the first derivatives with respect to each dimension, it could become impracticable at a certain point. In the field of machine learning, both are usually true.

The power of gradient descent stems from the realization that, when an approximation of the minimum is enough, the full analytical description of f is not necessary. Instead, one can start at any random point and use the local gradient to travel through the functional landscape in the direction that minimizes f. Take for example the function shown in Fig. 3.1. If we are able to estimate the slope of the tangent at our initial guess g, a.k.a $f'(g)$, we know in which direction we should update it. Furthermore, the steepness of the tangent is a good indicator of how much we should move in that direction, as the gradient naturally approaches zero as we move closer to an extremum.



(a) Initial guess

(b) Gradient computation

(c) Good update

(d) Overshoot

Figure 3.1: One dimensional cut of a gradient descent update

In practice, for a function with n parameters, this means computing the gradient vector

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right)^T \tag{3.1}$$

and updating the current point, $\vec{x}_s$, by a predefined step-size α (most commonly called "learning rate"), such that

$$\vec{x}_{s+1} = \vec{x}_s - \alpha \nabla f(\vec{x}_s). \tag{3.2}$$

Note that if we have access to a black box producing the output $f(\vec{x})$ it is always possible to approximate the gradients by the use of finite differences. However, in Sec. 3.3 we will discuss a more efficient method to do this in the context of neural networks, or more generally for complicated Ansatze.

The importance of the choice of α in the convergence behaviour of gradient descent cannot be overstated. On the one hand, if the learning rate is too small, the convergence will be slow, increasing unnecessarily the computational cost of the algorithm. On the other hand, a learning rate too large will cause the update to "overshoot" the minimum. In some cases this may even cause the next iteration step to be a worse approximation, as seen in Fig. 3.1d. To avoid this, state-of-the-art versions of gradient descent include updates to the learning rate itself, making it smaller as the algorithm progresses.

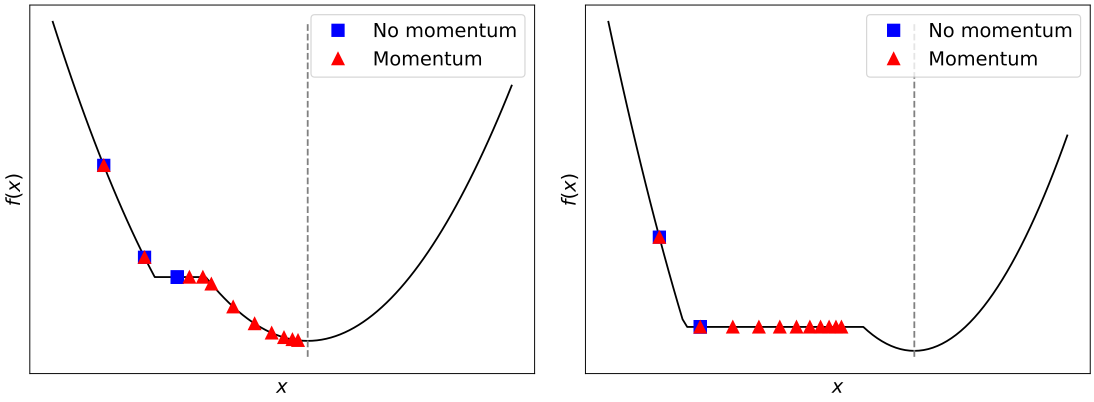
But there is an even bigger problem. In the previous example we have started with the assumption that f has a single minimum, and the function depicted in Fig. 3.1 is a strictly convex function, with no plateaus. In practice, these two assumptions are unlikely to be true. If during the optimization the iterations encounter a region of the function with gradient close to or exactly zero, the update will also be (close to) zero and the optimization will get stuck, see Eq. (3.2). Hence, the most powerful algorithms within the gradient descent family keep track, in one way or another, of the previous updates, and use them as additional information for the current iteration. The simplest way to do this is by incorporating the concept of momentum:

$$\Delta \vec{x}_s = -\alpha \nabla f(\vec{x}_s) + \gamma \Delta \vec{x}_{s-1} \tag{3.3}$$

Here, γ is another predetermined hyperparameter, which enforces the importance of previous updates on the current step. By this simple adjustment, the change in $\vec{x}$ no longer becomes zero immediately after entering a region of small gradients, and the algorithm has a chance to escape plateaus or shallow local minima.

To demonstrate this, Fig. 3.2a shows 10 iterations of a gradient descent update with and without momentum for a piecewise function with a constant plateau. In both cases we initialize the algorithm at the same point. Despite entering the plateau region, when momentum is included the algorithm manages to escape and continue its convergence towards the minimum.

Note however that this simple approach has its limits. If γ is too small, the optimizer may loose all its momentum within the region with gradient zero, in which case it also gets stuck, as seen in Fig. 3.2b. If, on the contrary, it is too large, it will exacerbate the overshoot behaviours previously discussed. In the past decade, multiple approaches have been proposed to overcome these problems. These approaches very often use second derivatives or keep track of weighted averages of the previous N iteration steps, making them more computationally expensive. This is why these types of algorithms are still an active area of research.



(a) Short plateau

(b) Long plateau

Figure 3.2: Effect of momentum on gradient descent

Ten iterations of gradient descent performed with and without momentum and starting from the same initial guess. With the right choice of hyperparameters, the algorithm with momentum is able to continue updating even in regions of gradient zero.

For the methods proposed in Chapters 4 and 5, we have used two algorithms in the gradient descent family: the Adam algorithm [66] and RMSprop [67]. Both of these methods include some kind of momentum in their updates, and have been utilized with great success in a variety of applications.

3.2 Neural Networks

Let us next consider the concept of neural networks. Loosely based on their biological counterpart, Artificial Neural Networks (ANN) are a type of variational computational model with applications mostly in the field of machine learning. The first mathematical description of a neural network dates back to 1943 [68], but the perceptron, which is closest to the basic unit of ANNs as we understand them today, was not proposed until 1958 [69]. Since then, and thanks to the technological revolution of past decades, the architectures of these networks has grown both in size and complexity, allowing researchers to use them to model extremely complicated functions and find solutions to a wide class of challenging optimization problems, ranging from game theory to language generation [70, 71].

3.2.1 The neuron

Neurons on their own perform a very simple computation, but its the combination of many of them which leads to the emergence of more complex phenomena. To understand better how this works, we should start with a detailed description of artificial neurons themselves.

Mathematically, the action of a neuron on an input vector $\vec{x}$ is the concatenation of two steps. First, a linear combination of the elements of $\vec{x}$ is computed according to some coefficients $\{w_j\}$. Then, a non-linear map σ is applied to the resulting scalar:

$$y = \sigma\left(\sum_j w_j x_j\right) . \tag{3.4}$$

Imagine, as a concrete example, that we were interested in computing the market price of a property according to some characteristics, such as number of squared meters, number of bedrooms and construction year. Those three values could be given as an input to the neuron, which would multiply them by the contribution of each characteristic to the final price. The non-linear map would then act as final adjustment for any non-linear effects perceived on the market –government regulations on maximum prices, for instance. This simple, yet effective, function approximation works for a broad spectrum of problems; including situations with non-numeric input data, as long as the data can be mapped to a specified set of token values.

Very often, we will be interested in neurons with outputs in a specified range. This is particularly true for neurons which are part of a larger network. Even if the final output is not restricted, we usually want the contribution of neurons in the middle of the network to be a value interpretable as the strength of that particular path to the final outcome. This allows us to interpret different neuronal connections in the network as logical computations which may or may not contribute to the output depending on the input. The non-linear map is therefore commonly called "activation function", its most important examples being the sigmoid function or logistic function,

$$\sigma(x) = \frac{1}{1+e^{-x}}, \tag{3.5}$$

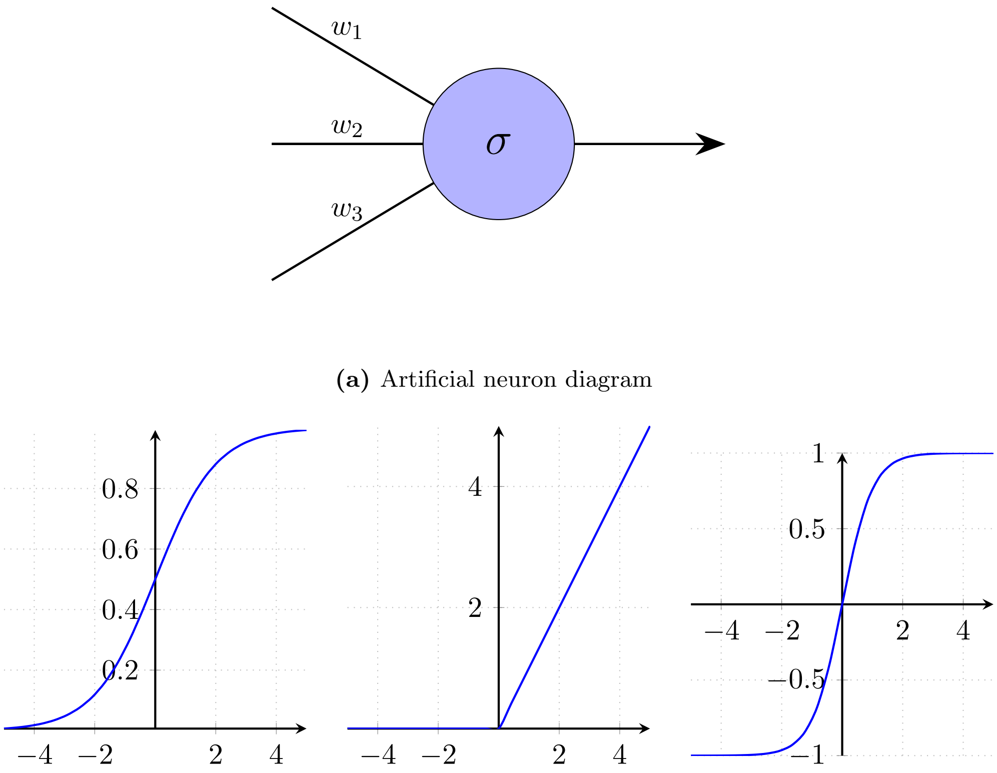
the Rectified Linear Unit,

$$\mathrm{ReLU}(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \tag{3.6}$$

and the hyperbolic tangent,

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}. \tag{3.7}$$

All of these are shown in Fig. 3.3, together with the standard diagram used to represent an artificial neuron.



(a) Artificial neuron diagram

(b) Sigmoid activation

(c) ReLU activation

(d) Tanh activation

Figure 3.3: The neuron: basic unit of Artificial Neural Networks

Note that all of these display at least partial saturation. The output of the sigmoid function, for example, can never be larger that 1 or smaller than 0, which limits the contributions of individual neurons to the network output. The same is true for the hyperbolic tangent, albeit with the values $[-1, 1]$. The Rectified Linear Unit, on the other hand, ensures that neurons can only provide positive contributions, otherwise the paths containing the neuron are effectively non-existent.

3.2.2 Shallow networks

But how are networks constructed? And how can they be used to solve optimization problems? To understand this, let us start by considering a so-called shallow network, as the one depicted in Fig. 3.4.

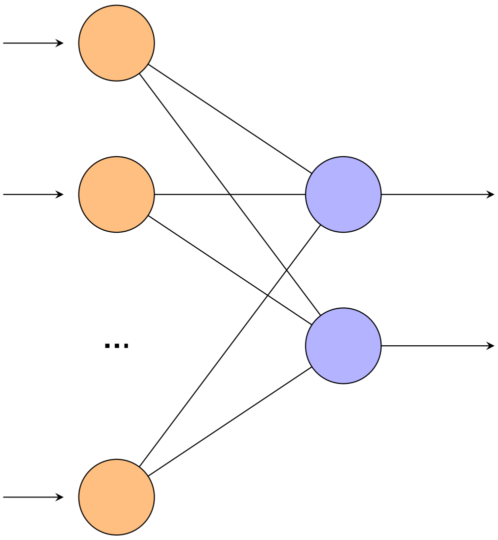


Figure 3.4: Shallow Neural Network

In feed-forward ANNs, neurons are organized into layers, each of which generates a vector of outputs which is fed into the next one as its input. Typically, networks are called “shallow” when their number of layers is less than 4, and “deep” otherwise. While the number of neurons in the intermediate layers is very flexible and mostly dependent on the computational resources of the hardware and your accuracy goals, the last layer must contain as many neurons as the desired output dimension. Let us denote the output of neuron j in layer ℓ as h_j^ℓ. Then, for an input $\vec{x}$, the outputs y_j of a two layer network are given by

$$y_j = \sigma\left(\sum_j w_j^{(2)} h_j^1\right) \tag{3.8}$$

where

$$h_j^1 = \sigma \left(\sum_k w_k^{(1)} x_k \right) . \tag{3.9}$$

Using vector notation, these two equations can be concisely written as

$$\vec{y} = \sigma \left(W^{(2)} \sigma \left(W^{(1)} \vec{x} \right) \right) , \tag{3.10}$$

where the activation function is acting elementwise. Thus, feed-forward networks can be simply implemented as a succession of matrix-vector multiplications with non-linearities between them.

Now that we have an expression for the function represented by the network, we can describe the common approach for solving an optimization problem. The matrices $W^{(\ell)}$ containing the neurons' weights are the variational parameters which we aim to optimize in order to find a solution for our problem. The output of the network is fed into a function which quantifies how well the model is performing, called the cost function, $C(\vec{y})$. The lower its value (or *loss*) the better the solution. Then, an iterative algorithm is applied to minimize the loss: the gradient with respect to each of the network parameters is computed, and the matrices $W^{(\ell)}$ are updated according to the preferred method.

Note that in the field of machine learning we want the networks to be able to generalize to unseen data. However, if our only focus is minimizing the cost function C, we run the risk of "memorizing" the training data and producing nonsensical results for new input data. This is because any noise in the training data will be interpreted as a feature of the data itself, and the network will be fitted to it. As an example of this, see Fig. 3.5. This phenomenon is called overfitting, and it is something worth considering when dealing with any kind of stochastic data, as we will in this thesis. There are many approaches one could take in order to avoid this problem, but here we will only discuss the two more relevant ones: early-stopping and regularization by the inclusion of a penalization term.

In the early-stopping approach, the input data is split into two batches, one which will be used for training, and one used for validation. During the optimization, the value of the cost function is computed for both batches, but only the training data is used in the parameters update. When the validation loss starts to increase, the optimization is interrupted, as it is clear sign that the network has started to fit to the noise present in the training batch. The second approach, on the other hand, consists on including a new term in the cost function which is not directly related to its performance, but instead serves as a way to penalize extreme values of the parameters. This helps to

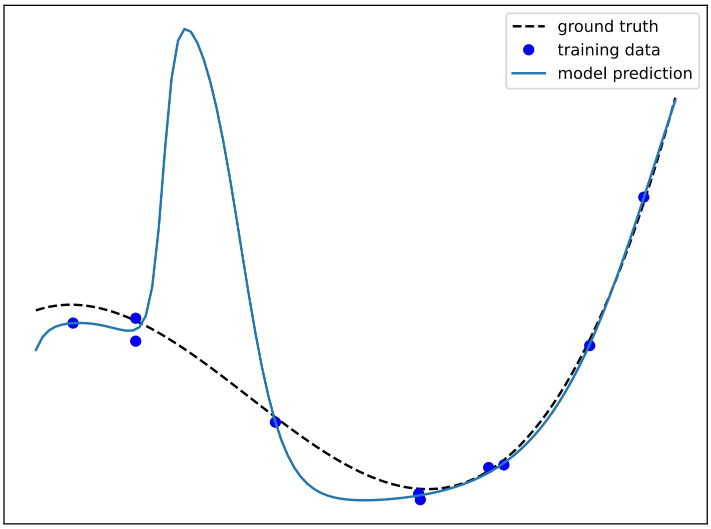


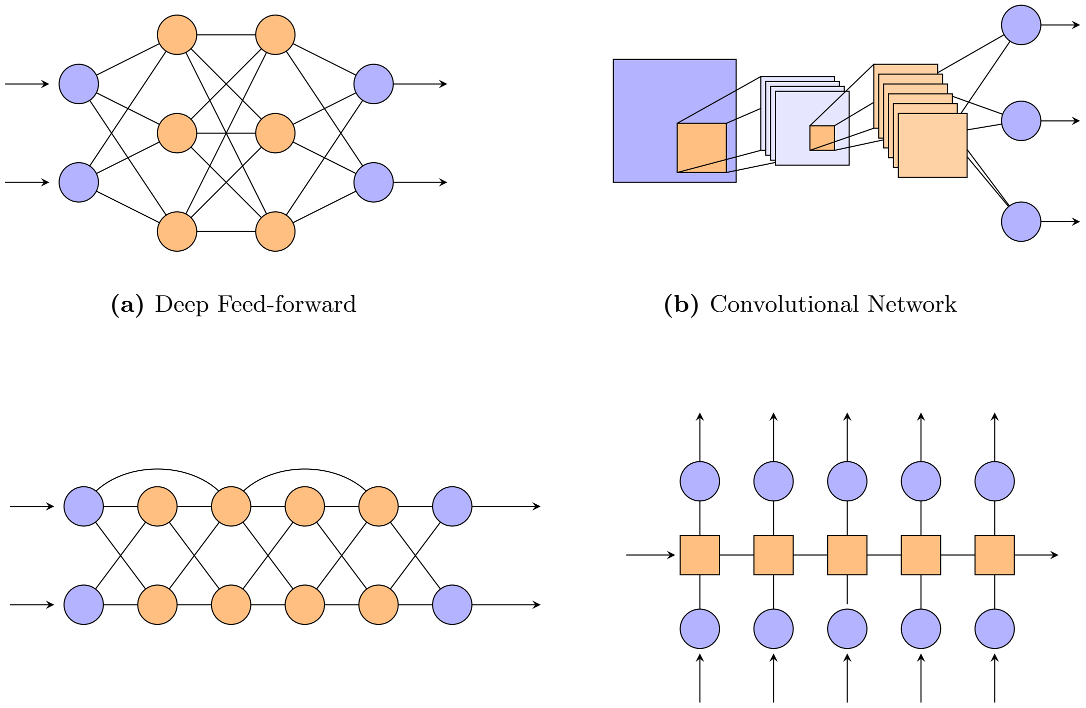
Figure 3.5: Example of an optimization overfitting to noisy data

avoid extreme predictions (as the one seen in Fig. 3.5) which are likely to be stochastic artefacts.

3.2.3 A deep network example: The Convolutional Neural Network

It turns out that wide shallow networks –that is, networks with few layers but many neurons– are already universal function approximators [72]. Nevertheless, due to limited computational resources, it is not realistic to infinitely increase the width of the network layers. This is where the architectural flexibility of neural networks comes into play.

By building networks with more intricate connections between the neurons, researchers are able to represent more complex function that we would with a shallow network with an equivalent number of neurons [73]. This improvement may come from an extension as simple as increasing the number or layers, although very often it also involves the utilization of specialized layers or neurons, designed with a particular application problem in mind. There are many examples of this, some of which can be seen on Fig. 3.6, but in this discussion we will focus on the so-called Convolutional Neural Network (CNN), as it is used in Chapter 4 and it has particular properties that may make it a good candidate in the representation of many-body quantum states.



(a) Deep Feed-forward

(b) Convolutional Network

(c) Deep Residual Network

(d) Recurrent Neural Network

Figure 3.6: Prominent examples of Deep Neural Networks

CNNs are a type of feed-forward network originally designed for the classification of images. They are based on the idea of looking for patterns that may repeat across or within inputs. In the past, people used to select the patterns manually, but thanks to advancements in the optimization techniques and hardware used to train neural networks, nowadays the patterns are fully optimized from a random initial guess. This is done via the use of *convolutional filters*.

Recall that in our previous description of ANN, we assigned different weights to each neuron-to-neuron connection. In Eq. (3.10), for instance, the parameter matrices W^ℓ are of size $n \times m$ where n and m are the number of neurons in layers $\ell - 1$ and ℓ respectively. However, if we are looking for patterns in the inputs, it may be a good idea to reuse some of the weights across several connections. That is exactly what convolutional filters do.

The filters are grids, usually squared, of variational parameters. These parameters are applied by sliding the grid across the input and multiplying them entry by entry. To be more concrete, consider an input X of size $N \times N$ to which we want to apply a $n \times n$

filter composed of weights $w_{i,j}$. Then, the outputs of the operation would be

$$y_{k,l} = \sigma\left(\sum_{i=1}^{n}\sum_{j=1}^{n} w_{i,j} X_{k+i,l+j}\right). \tag{3.11}$$

Fig. 3.7 visually depicts this same operation for a small filter and system size. Note that, even though in this equation we assume that the filter moves one step at a time, in practice one may choose different step sizes depending on the desired output size.

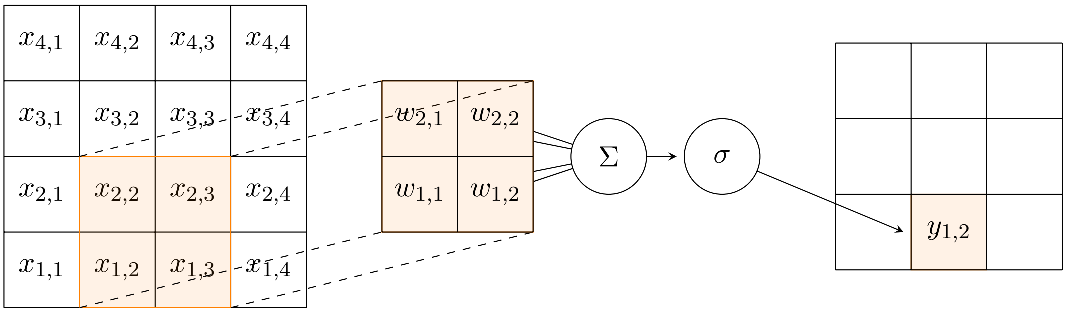


Figure 3.7: Breakdown of a convolutional filter

On top of that, in the above description we have been intentionally vague about what happens at the boundaries of the input. There are many approaches to tackle the boundaries, such as padding the input with zeros or other constant values, using periodic boundary conditions, or simply stopping the filter from sliding further as it covers a boundary entry. In the context of quantum many-body systems simulation, this flexibility allows us to adapt the filter operation depending on the boundary conditions of the system itself.

Not only that, but the translational equivariance of the convolution operation seems ideally suited for many quantum lattice systems, which are also translational invariant. In fact, this is precisely the reason why CNNs have become so successful and popular for tasks related to image recognition and generation [74].

Since it is very unlikely that a single feature would be enough to accurately model any system, CNNs are usually composed by layers of stacked filters of the same size. And, although not always necessary, it also sometimes useful to include some plain feed-forward layers are the end of the network, to aid in the post-processing of the extracted features. Such a structure of filters acting on the output from the previous ones naturally leads to a hierarchical distribution of the optimized features: the more high-level ones being closer to the input and the more low-level ones closer to the output.

To further motivate the use of CNNs in the context of many-body quantum systems, let us explore two particularly relevant works on the classification of experimental data from condensed matter systems. In Ref. [75] a CNN was used to classify snapshot images from two different ultracold atoms systems. The two systems undergo a phase transition across a critical point. However, there is no physical model to determine the phase solely from a simple image, and humans are also no capable of distinguishing between them, especially for images taken in close proximity to the phase transition. By training the CNN with images at extreme points of the two phases, the researchers were then able to classify images closer to the phase transition. This allowed them to identify narrow ranges in which the critical points exist, narrower, in fact, than the ranges obtained by other commonly employed statistical models.

In Ref. [76], on the other hand, researchers used the classification power of CNN to compare two competing theoretical models for the phases in the Fermi-Hubbard model, which for simplicity we will denote as theories A and B here. The network is trained with simulated data generated with each of the theories, and is then used to classify images into three classes: experimental data, simulated data from theory A and simulated data from theory B. The network is not able to accurately distinguish between experimental and simulated data, which implies that it should be able to classify the images of the experiment as corresponding to one of the two theories. Despite not achieving perfect accuracy, the primary source of confusion was observed between the classes "experimental" and "theory A", which suggests that theory A it better at capturing the real behaviour of the system.

These studies are proof that, in principle, machine learning models are able to detect patterns and nuances in quantum systems that are critical for understanding their behaviour. In Chapter 4 we will try to apply these capabilities to the *simulation* of quantum systems, which, due to the reasons explained in Sec. 2.1 proves to be a more challenging task.

3.3 Backpropagation

But before that, it is critical to revisit how ANNs are trained via gradient descent. Up until this point we have glossed over the central step in gradient descent methods –the computation of the gradients themselves–, and we have assumed that there exists a computationally efficient way of computing them for any network, regardless of its complexity.

As previously mentioned, finite differences methods are indeed a valid approach to compute the gradient for any black box f. At its core, this approach consists of perturbing the parameter for which we want to compute the gradient, w, by a small amount h and approximating the slope of the tangent by expressions such as

$$\left.\frac{\partial f}{\partial w}\right|_{w_0} \approx \frac{f(w_0 - h) - f(w_0 + h)}{2h}. \tag{3.12}$$

These methods require at least two applications of f for each parameter in the network. Taking into account that networks usually contain thousands, or even millions, of variational weights, it becomes necessary to find an alternative approach to avoid this substantial computational cost.

In 1986, a highly efficient method named "backpropagation" was proposed [77]. This technique, which we will describe in detail in this section, leverages the chain rule two reduce the computational cost of gradient calculations up to the cost of a single pass of the network. As a reminder, for a function $f(y(x))$ the chain rule allows us to decompose the derivative of f with respect to x as

$$\frac{df}{dx} = \frac{df}{dy} \cdot \frac{dy}{dx}. \tag{3.13}$$

Given that neural networks are nothing more than outputs of functions being fed as inputs into other functions, the chain rule becomes immediately applicable. For instance, consider the output of single neuron,

$$y = \sigma(z) = \sigma\left(\sum_j w_j x_j\right), \tag{3.14}$$

which is being fed to the cost function C. Then, the gradient of the cost with respect to each of the neurons weights w_j is

$$\frac{\partial C}{\partial w_j} = \frac{\partial C}{\partial y} \cdot \frac{\partial y}{\partial z} \cdot \frac{\partial z}{\partial w_j} = \frac{\partial C}{\partial y} \cdot \frac{\partial y}{\partial z} \cdot x_j. \tag{3.15}$$

Note that the analytical expression for all these partial derivatives is known as soon as the network is constructed and the cost function defined. The strategy is therefore to implement "forward" and "backward" passes of each operation of the network. The forward pass is used to compute the output of the network, while the backward passes correspond to the derivatives of that particular step with respect to the previous one. Fig. 3.8 illustrates this for a fully connected shallow network. Keep in mind that, to compute the gradient of the cost with respect to a specific weight, it is necessary to add the gradients computed across all paths containing that weight. In Fig. 3.8 specifically, we show all the paths containing the weights of the second neuron in the first hidden layer.

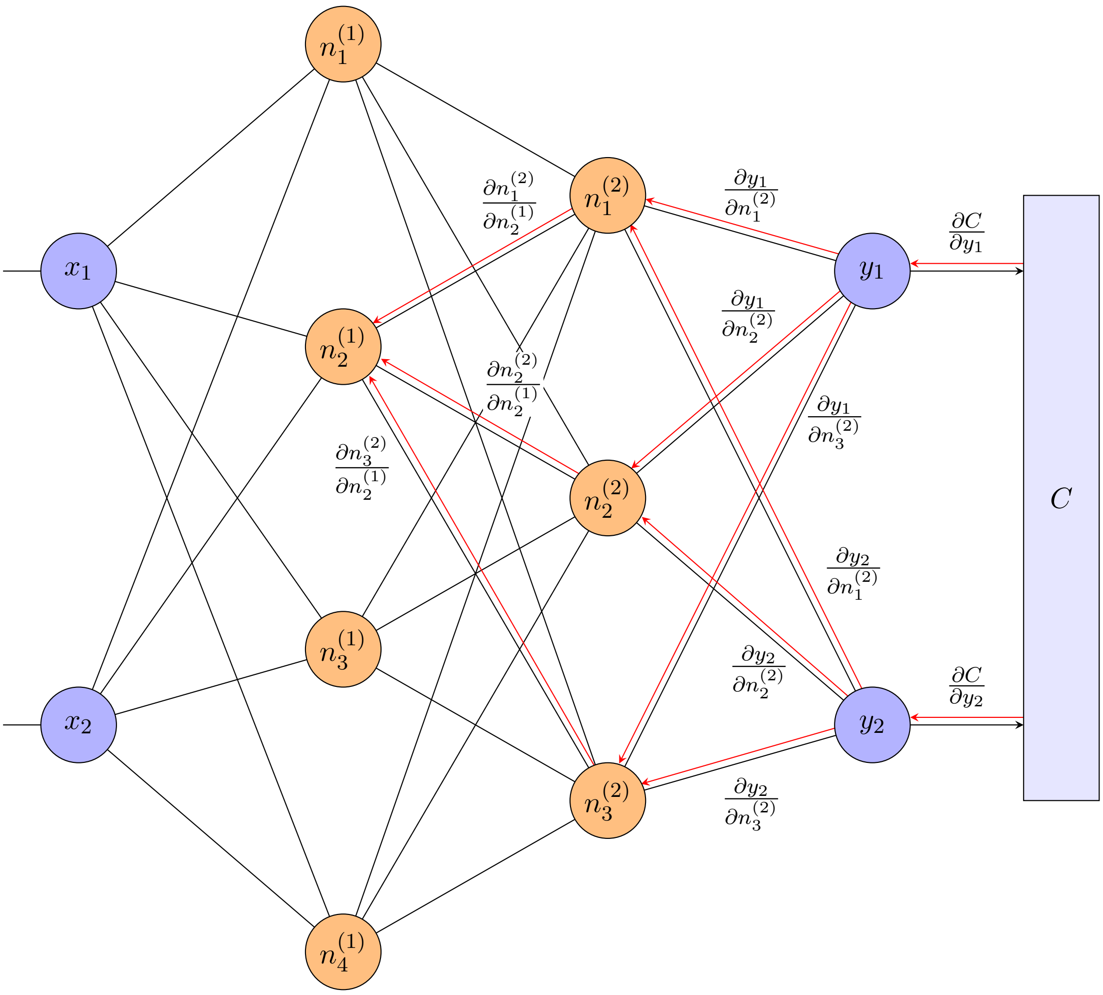

$n_1^{(1)}$
x_1
$n_2^{(1)}$
$n_3^{(1)}$
x_2
$n_4^{(1)}$
$\frac{\partial n_1^{(2)}}{\partial n_2^{(1)}}$
$n_1^{(2)}$
$\frac{\partial n_2^{(2)}}{\partial n_2^{(1)}}$
$\frac{\partial n_3^{(2)}}{\partial n_2^{(1)}}$
$n_2^{(2)}$
$n_3^{(2)}$
$\frac{\partial y_1}{\partial n_1^{(2)}}$
$\frac{\partial y_1}{\partial n_2^{(2)}}$
$\frac{\partial y_1}{\partial n_3^{(2)}}$
$\frac{\partial y_2}{\partial n_1^{(2)}}$
$\frac{\partial y_2}{\partial n_2^{(2)}}$
$\frac{\partial y_2}{\partial n_3^{(2)}}$
y_1
y_2
$\frac{\partial C}{\partial y_1}$
$\frac{\partial C}{\partial y_2}$
C

Figure 3.8: Backpropagation method applied to a fully connected network

These operations can also be written as a combination of matrix-vector multiplications. For example, for a feed-forward fully connected network, the gradient of the cost function with respect to the weights in the last layer, $w^{\ell}_{j,k}$, is

$$\frac{\partial C}{\partial w^{\ell}_{j,k}} = \sum_j \frac{\partial C}{\partial y_j} \cdot \frac{\partial y_j}{\partial w_{j,k}}. \tag{3.16}$$

But we know that the output of each neuron is given by

$$y_k = \sigma(z_k) = \sigma\left(\sum_l w^{l,k} n^{\ell-1}_l\right), \tag{3.17}$$

where n^{ℓ}_l is the output of the l-th neuron in the previous layer. Thus, we can write

$$\frac{\partial y_k}{\partial w^{\ell}_{l,k}} = \frac{\partial y_k}{\partial z_k} \cdot n^{\ell-1}_l. \tag{3.18}$$

And denoting the vectors of derivatives as $\frac{\partial f}{\partial x} = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots\right)$, this gives us an expression for row l of the weight matrix

$$\frac{\partial C}{\partial w^{\ell}_{l,i}} = \frac{\partial C}{\partial y_i} \cdot \frac{\partial y_i}{\partial z_i} \cdot n^{\ell-1} \tag{3.19}$$

and for the full matrix

$$\frac{\partial C}{\partial W^{\ell}} = \frac{\partial C}{\partial y} \cdot \left(\frac{\partial y_1}{\partial z_1}, \frac{\partial y_2}{\partial z_2}, \dots\right) \times n^{\ell-1}. \tag{3.20}$$

To avoid computing n^{ℓ} every time we need the gradients, the output of each layer is stored in a cache so that it can be quickly retrieved at each backward pass of the network. This means that the time cost of computing the gradients with respect to all parameters is of the same order as the forward pass, at the simple expense of a very minor overhead in terms of required memory.

To finalize this section, let us mention briefly that current machine learning packages have an automatic differentiation protocol already implemented, which allows the user to only implement the forward pass of the network. These automatic differentiation techniques are based on the fact that all computations can be reduced to combinations of products and sums, for which the derivative can straightforwardly computed. Their reliability for functions involving complex numbers, however, depends on whether they are holomorphic. This is something that will be addressed in the next chapter.

4 Neural Network Quantum States

The following chapter is dedicated to the classical simulation of many-body quantum systems. It is worth restating that the exponential growth in computational resources required for the simulation of such systems may present an insurmountable bottleneck, and that the possible viable alternative would be the use of quantum computers. Nonetheless, quantum computers are still in the early stages of development and have limitations in terms of error rates and qubit count and connectivity. Pushing the limits of classical simulations can therefore bridge the gap between current quantum capabilities and the potential future power of fully functional quantum computers. The results presented here were published in Ref. [78].

4.1 State of the Art

Neural Network Quantum States (NNQS) were first proposed in 2017 by Carleo and Troyer [79]. The idea behind them is to exploit the high flexibility in neural network architectures in order to accurately represent a many-body quantum state. Even if the Hilbert space of a quantum system grows exponentially with system size, it may still be possible to construct an appropriate parametrized Ansatz which captures the physically relevant subspace without such an exponential growth.

One particular example which has been studied in depth is the Restricted Boltzmann Machine (RBM) architecture. Numerical experiments using such an Ansatz have been shown to achieve great precisions, comparable or even improving on the results achieved with tensor networks [79]. It has been shown that the RBM architecture is equivalent to an approximation of the perturbation series [80]. This explains the high accuracy of the RBM Ansatz when simulating the ground-state of a variety of systems, and suggests the existence of a larger family of Ansatze with similar properties. Furthermore, fully-connected RBM states have the potential to display volume-law entanglement [81], implying that RBMs can effectively represent highly-correlated quantum states.

Expanding on the success of RBMs as an efficient parametrization, many works have combined them with tensor network architectures, such as Pair-Product States [82], Generalized Transfer Matrix States [83], and Correlator Product States [84]. RBMs themselves have also been shown to be equivalent to a non-local version of Matrix Product States (MPS) (the so-called String-Bond States) [85]. In addition to that, the use of deep RBMs [86] and limiting the node edges in a carefully constructed way [87] can improve the accuracy of the Ansatz for specific systems. All of these results bring to light the potential of neural network architectures in the simulation of quantum systems.

But the RBM may not be the only promising architecture. Deep learning architectures have also been shown to efficiently represent highly-entangled quantum states [88–91]. Specifically, CNN and Recurrent Neural Network (RNN) architectures support a volume-law scaling polynomially more efficient than RBMs [92]. In this direction, RNN have been employed to successfully calculate ground-state energies, correlation functions, and entanglement entropies in several spin models in 1D and 2D [93, 94]. And, when compared to more popular tensor network Ansatze, they seem to be able to represent wavefunctions with a 'bond dimension' orders of magnitude lower than that of MPS [95].

Unsurprisingly, there are still many challenges in the field of NNQS. These types of Ansatze seem to perform best with physics-informed architectures, and struggle to achieve state-of-the-art accuracies around critical points [96]. That is the case of the maximally frustrated regime for the spin-1/2 $J_1 - J_2$ Heisenberg model, where the NNQS present difficulties in learning the underlying sign structure [97–100].

More relevant to this thesis is the challenge of real-time evolution. Finding a good parametrization of the states in the time trajectory poses a significant challenge. Luckily for us, however, it has been shown that the manifold of the states time-evolved under an arbitrary Hamiltonian occupies an exponentially small volume in the Hilbert Space [101] (see also [102] for an analysis of the dimensionality of the quantum state manifold). This suggests that an efficient parametrization is still possible, and the question then becomes how to best represent that manifold.

Established tensor network methods have already attempted to tackle the simulation of time-evolution under arbitrary Hamiltonians [103–107]. However, the increase in entanglement with time leads to an exponential increase in bond dimension, effectively limiting their applicability to short time intervals [35, 108].

Given that neural networks are universal function approximators [72, 109–111], they represent an excellent candidate to parametrize the time evolution manifold. Up to this point, research has predominantly focused on applying them within the context of quenches of the spin-1/2 Ising and Heisenberg models in one and two dimensions, with varying levels of success [96, 112–114].

One of the key problems seems to be an exponential growth in the number of parameters necessary to accurately represent the quantum state as a function of time. Interestingly, this growth rate was found to be relatively unaffected by variations in network architecture across a wide spectrum of design choices: shallow and deep networks, different filter sizes, dilated and normal convolutions, and the presence or absence of short-cut connections [115]. But it is worth noting that, due to the infinite possibilities of neural network architectures, this is still an active area of research.

A second important challenge within the context of real time evolution is that of numerical stability. Typically, NNQS are time propagated using time-dependent variational Monte Carlo (t-VMC) methods. The stochastic nature of these methods becomes an unavoidable source of error, which is not only accumulated through time, but also amplified due to the ill-conditioned equations of motion used for the optimization [89, 116].

In the next sections we will explain how to apply NNQS to simulate the time-evolution under a Hamiltonian. In particular, we will focus on the Ising Hamiltonian, and we will explore the source of numerical instabilities in the time trajectories before suggesting an alternative method of optimization to avoid this problem.

4.2 Groundstate optimization

Before diving into time-evolution, it is useful to first consider the problem of groundstate optimization. Variational methods typically rely on the existence of a parametrized function to be minimized. In the case of groundstate simulation, one can simply aim to minimize the energy of the Ansatz. Given an Ansatz

$$|\psi(\boldsymbol{\theta})\rangle = |\psi_{\boldsymbol{\theta}}\rangle\,, \tag{4.1}$$

where $\boldsymbol{\theta}$ is the vector of variational parameters, one can find the energy of the state by computing

$$E = \frac{\langle\psi_{\boldsymbol{\theta}}|H|\psi_{\boldsymbol{\theta}}\rangle}{\langle\psi_{\boldsymbol{\theta}}|\psi_{\boldsymbol{\theta}}\rangle}. \tag{4.2}$$

If the Ansatz $|\psi_{\boldsymbol{\theta}}\rangle$ has been chosen appropriately, minimizing Eq. (4.2) will yield a good approximation of the groundstate of the Hamiltonian H.

It is important to note that for large systems sizes it may not be feasible to store in memory a full representation of H. However, for local Hamiltonians, the analytic computation of $H\,|\psi_{\boldsymbol{\theta}}\rangle$ is still possible for basis states, allowing for the estimation of Eq. (4.2) through sampling methods, such as the Metropolis-Hastings algorithm [117].

But how is Eq. (4.2) minimized in practice? A commonly employed strategy is that of *imaginary time evolution.* This strategy is based on the fact that, as long as $|\psi(0)\rangle$ has some overlap with the groundstate of H,

$$|\psi(\tau)\rangle \propto e^{-\tau H} |\psi(0)\rangle \tag{4.3}$$

will converge to that state as $\tau \to \infty$. Note that this equation is equivalent to the solution of the time-dependent Schödinger equation under an imaginary time $i \cdot t$, hence the name.

Such an operation is not guaranteed to preserve the state in the space of the variational Ansatz, making it necessary to project the state back to the variational manifold after each step in the imaginary time evolution. This, in turn, will interfere with the convergence to the groundstate. However, provided a good selection of the Ansatz, the resulting steady state will be a good approximation of the groundstate.

4.2.1 Stochastic Reconfiguration

In the field of NNQS,a very commonly employed way of doing this is via the so-called Stochastic Reconfiguration (SR) method [79, 87], a type of Dirac-Frenkel variational method [118, 119].

Assuming that the Ansazt depends on τ through the variational parameters, $\boldsymbol{\theta}(\tau)$, each step in the imaginary time evolution is given by

$$|\psi_{\boldsymbol{\theta}(\tau+d\tau)}\rangle = e^{-d\tau H} |\psi_{\boldsymbol{\theta}(\tau)}\rangle \,. \tag{4.4}$$

Expanding both sides of the equation up to first order in $d\tau$, reads

$$|\psi_{\boldsymbol{\theta}(\tau+d\tau)}\rangle \simeq |\psi_{\boldsymbol{\theta}(\tau)}\rangle + d\tau \sum_k \dot{\boldsymbol{\theta}}_k(\tau) \left[|\partial_{\boldsymbol{\theta}_k} \psi_{\boldsymbol{\theta}(\tau)}\rangle - |\psi_{\boldsymbol{\theta}(\tau)}\rangle \langle \psi_{\boldsymbol{\theta}(\tau)} \mid \partial_{\boldsymbol{\theta}_k} \psi_{\boldsymbol{\theta}(\tau)}\rangle \right] \tag{4.5}$$

and

$$e^{-d\tau H} |\psi_{\boldsymbol{\theta}(\tau)}\rangle \simeq |\psi_{\boldsymbol{\theta}(\tau)}\rangle - d\tau H |\psi_{\boldsymbol{\theta}(\tau)}\rangle \,. \tag{4.6}$$

As already mentioned, we will need to project each step back to the variational manifold. The tangent space of the manifold at point $\boldsymbol{\theta}$ is spanned by the basis vectors

$$|\partial_{\boldsymbol{\theta}_j} \psi_{\boldsymbol{\theta}}\rangle - |\psi_{\boldsymbol{\theta}}\rangle \langle \psi_{\boldsymbol{\theta}} \mid \partial_{\boldsymbol{\theta}_j} \psi_{\boldsymbol{\theta}}\rangle \,. \tag{4.7}$$

Therefore, taking the product of both sides through the left and equating them yields

$$\sum_k \dot{\boldsymbol{\theta}}_k(\tau)\left[\langle \partial_{\boldsymbol{\theta}_j}\psi_{\boldsymbol{\theta}} \mid \partial_{\boldsymbol{\theta}_k}\psi_{\boldsymbol{\theta}}\rangle - \langle \partial_{\boldsymbol{\theta}_j}\psi_{\boldsymbol{\theta}} \mid \psi_{\boldsymbol{\theta}}\rangle \langle \psi_{\boldsymbol{\theta}} \mid \partial_{\boldsymbol{\theta}_k}\psi_{\boldsymbol{\theta}}\rangle\right] = \\ - \langle \partial_{\boldsymbol{\theta}_j}\psi_{\boldsymbol{\theta}}| H |\psi_{\boldsymbol{\theta}}\rangle + \langle \psi_{\boldsymbol{\theta}}| H |\psi_{\boldsymbol{\theta}}\rangle \langle \psi_{\boldsymbol{\theta}} \mid \partial_{\boldsymbol{\theta}_j}\psi_{\boldsymbol{\theta}}\rangle . \tag{4.8}$$

Note that

$$S_{j,k} = \langle \partial_{\boldsymbol{\theta}_j}\psi_{\boldsymbol{\theta}} \mid \partial_{\boldsymbol{\theta}_k}\psi_{\boldsymbol{\theta}}\rangle - \langle \partial_{\boldsymbol{\theta}_j}\psi_{\boldsymbol{\theta}} \mid \psi_{\boldsymbol{\theta}}\rangle \langle \psi_{\boldsymbol{\theta}} \mid \partial_{\boldsymbol{\theta}_k}\psi_{\boldsymbol{\theta}}\rangle \tag{4.9}$$

is the covariance matrix of the current point in the monifold, whereas

$$\mathbf{F}_j = \langle \partial_{\boldsymbol{\theta}_j}\psi_{\boldsymbol{\theta}}| H |\psi_{\boldsymbol{\theta}}\rangle - \langle \psi_{\boldsymbol{\theta}}| H |\psi_{\boldsymbol{\theta}}\rangle \langle \psi_{\boldsymbol{\theta}} \mid \partial_{\boldsymbol{\theta}_j}\psi_{\boldsymbol{\theta}}\rangle \tag{4.10}$$

effectively behaves as a force vector. We can thus rewrite Eq. (4.8) in vector notation as

$$S\dot{\boldsymbol{\theta}} = -\mathbf{F}. \tag{4.11}$$

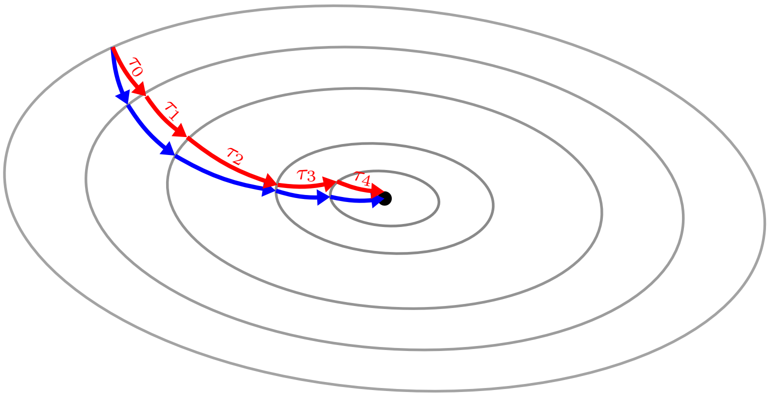


Figure 4.1: Trajectory of a possible imaginary time evolution

The black dot at the center represents the groundstate. The blue trajectory corresponds to the exact trajectory, whereas the red arrows represent each step with added stochastic error. Despite the imperfect individual steps, the convergence behaviour to a steady state is preserved.

At this point, an attentive reader may have noticed a problem with the above equation. As previously mentioned, a complete representation of H is not possible for large systems. Instead, we make use of the local nature of most (physically relevant) Hamiltonians to compute a stochastic approximation of Eq. (4.11). This, in turn, results in the covariance matrix, S, being non-invertible. And, although finding a solution is still possible through the use of a pseudo-inverse or Krylov subspace methods, it is important to keep in mind that these solutions will be approximations.

Fortunately, within the context of groundstate optimization, as the trajectory steadily converges to an equilibrium state, the stochastic errors arising during each step in the imaginary time evolution tend to self-correct when combined with those from previous steps. This self-correction is a consequence of the fundamental nature of the problem, which is a true minimization. Given that the evolution ultimately converges to a specific location within the Hilbert space, minor errors incurred throughout the process become irrelevant. Fig. 4.1 serves as an illustrative visual representation of this phenomenon.

4.3 Real time evolution

The previous derivation for the imaginary time evolution can be easily adapted to real time, as the only difference is the imaginary number in the exponential of Eq. (4.4). That is, each time step is given by

$$|\psi_{\boldsymbol{\theta}(\tau+d\tau)}\rangle = e^{-id\tau H}\,|\psi_{\boldsymbol{\theta}(\tau)}\rangle\,. \tag{4.12}$$

which leads to the linear system to be solved

$$S\dot{\boldsymbol{\theta}} = -i\mathbf{F}. \tag{4.13}$$

In contrast to what was explained for the groundstate optimization, real-time evolution deals with the dynamic behaviour of a quantum system over time. This requires tracking the system's behaviour at each moment in time, and any errors introduced accumulate and affect the accuracy of the simulation. Similarly to how chaotic classical systems are extremely sensitive to initial conditions, even minor inaccuracies at each time step can be amplified throughout the system's evolution and lead to significant deviations from the true behaviour.

We must therefore play closer attention to the stochastic error arising from the sampling to calculate S. One thing to be noted is that the singular values of S span several orders of magnitude –many of them being close to zero–, with no clear jumps between their magnitudes. When calculating the pseudo-inverse one must choose the threshold below which singular values will be set to zero. Without any clear structure from S to inform this choice, one often relies on educated guesses based on the numerical results.

In the course of our experiments, we observed a high degree of sensitivity in the time trajectory concerning our selection of the threshold. This poses a significant challenge,

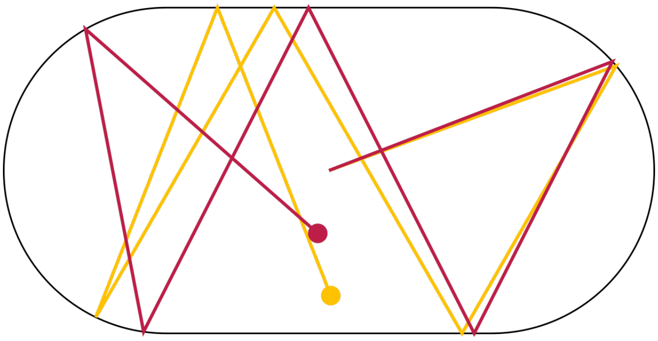


Figure 4.2: Billiards in a Bunimovich stadium

Example of a classical system where a small deviation at a particular point leads to a completely different time trajectory. [1]

as it implies that it is not possible to find a (close-to-)optimal solution for the ill-defined system in Eq. (4.13) without comparing each step to the ground truth. A ground truth which, evidently, remains inaccessible to us for large systems.

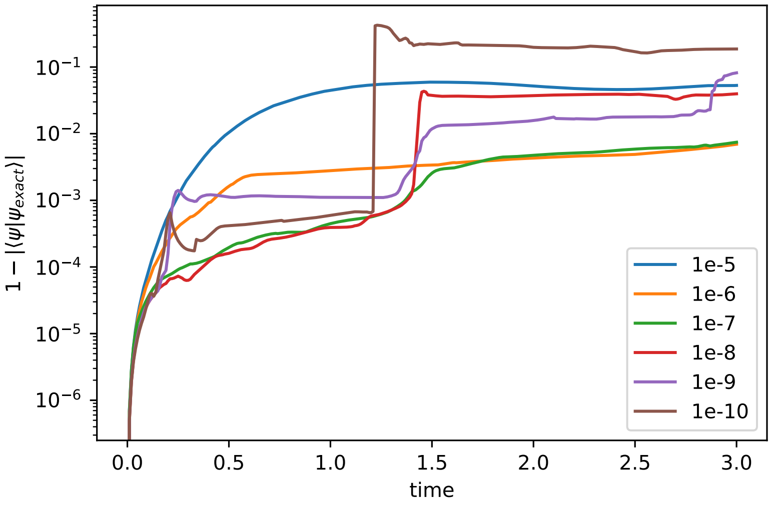


Figure 4.3: Time evolution error for different pseudo-inverse thresholds.

Fig. 4.3 is a comparison of the accuracies obtained using SR with different thresholds for the pseudo-inverse for a particular system. There are two interesting behaviours to note from this graph:

[1] Image from Wikipedia Commons. Created by Jakob Scholbach under a Creative Commons Attribution-Share Alike 3.0 Unported licence.

1. The optimal trajectory (which improves the accuracy from the worst by more than two orders of magnitude at points) is not the same at each time-step.

2. There are particular points in time where the error increases dramatically over few time-steps –see for example the 1e-10 trajectory at the ~ 1.25 timestamp. Although it is hard to trace back where this error comes from, it appears like it could be a negative interaction between the unavoidable stochastic error from the sampling and the inaccuracies introduced by trying to solve the ill-defined linear system.

But using the pseudo-inverse is not the only way to solve such a linear equation. A popular class of methods to solve these type of systems is that of Krylov subspace methods: a class of iterative numerical techniques used to solve large linear systems of equations. In fact, they are often used precisely when the matrix is not invertible or when it is computationally expensive or impractical to compute its inverse [120].

The convergence of Krylov subspace methods is based on the properties of the matrix and the choice of method. For non-invertible matrices, the methods aim to find a solution in the space that is compatible with the system, even if the matrix does not have a unique solution. The methods work by minimizing the error within the Krylov subspace. Convergence to the optimal solution is therefore not guaranteed, and they often require some preconditioners chosen to make the coefficient matrix more well-conditioned.

In Fig. 4.4, we present our numerical findings for the evaluation of the accuracy of various Krylov subspace methods in comparison to the pseudo-inverse. The outcomes indicate that the Krylov subspace methods, as depicted, fail to attain the optimal solution.

4.3.1 Gradient descent method

In this thesis, we propose a different approach, which fits more directly to the paradigm of neural network training: for each time step Δt, we optimize the network parameters to minimize the error

$$\left\|\psi[\theta_{n+1}] - \Phi^{\Delta t}\left(\psi[\theta_n]\right)\right\| \tag{4.14}$$

with respect to θ_{n+1}, where $\Phi^{\Delta t}$ is the discrete flow of a numerical ODE method applied to the Schrödinger equation. Note that, this way, each time step becomes a minimization problem. Although the stochastic error will continue to accumulate over time, our objective is to minimize its impact by implicitly incorporating a convergence point at

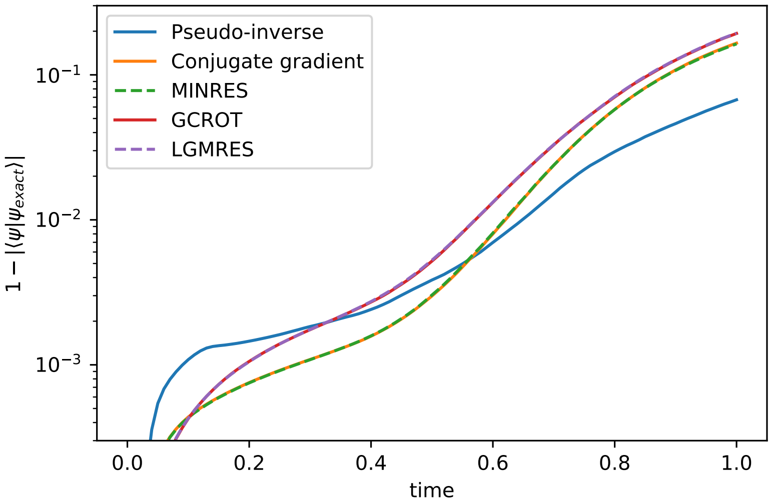


Figure 4.4: Performance comparison between Krylov subspace methods and the pseudo-inverse.

each step. The aim is that, in the absence of the instabilities caused by the pseudo-inverse, the error will not experience disproportionate amplification during later stages of the time evolution.

We will use the implicit midpoint method here. For an ODE $y'(t) = f(t, y(t))$ and a time-step Δt, this method is defined by

$$y_{n+1} = y_n + \Delta t f\left(t_n + \frac{\Delta t}{2}, \frac{1}{2}\left(y_n + y_{n+1}\right)\right). \tag{4.15}$$

In the specific case of the Schrödinger equation, this leads to (cf. the Cayley transform)

$$\psi[\theta_{n+1}] \approx \psi[\theta_n] - i\Delta t H\left(\frac{\psi[\theta_{n+1}] + \psi[\theta_n]}{2}\right). \tag{4.16}$$

The implicit midpoint method has two important favourable properties: firstly, it preserves the symplectic form of Hamiltonian dynamics [121], and secondly, it does not contain intermediate quantities that would complicate the network optimization. Additionally, in Sec. A.2 we include a more detailed analysis of the error contribution from this integration method, demonstrating that the residuals will not substantially exacerbate the error as a consequence of time-evolution approach.

In the case of larger systems, where sampling becomes necessary, we minimize the following cost function for a single midpoint rule time step:

$$C(\theta_{n+1}) = \sum_{j=1}^{N} \left| \Big(\big(I + \tfrac{i\Delta t}{2} H\big)\, \psi[\theta_{n+1}] - \big(I - \tfrac{i\Delta t}{2} H\big)\, \psi[\theta_n] \Big) \big(\sigma^{(j)}\big) \right|^2 , \tag{4.17}$$

with the $\sigma^{(j)}$, $j = 1, \ldots, N$ a batch of input configurations, and the network parameters at the current time point, θ_n, regarded as fixed. To be specific, we focus on spin variables as our input, and we represent the system's size as L, signifying the dimension of the quantum Hilbert space (corresponding to the number of potential spin configurations, which is 2^L). The cost function can be expressed in the form of a least squares representation as follows:

$$C(\theta) = \| A\psi[\theta] - b \|^2 , \tag{4.18}$$

with $A = \mathbb{C}^{N \times 2^L}$ the (sparse) submatrix of $I + \frac{i\Delta t}{2} H$ containing the rows corresponding to the spin configurations $\sigma^{(j)}$, and the vector $b \in \mathbb{C}^N$ with entries

$$b_j = \big(\big(I - \tfrac{i\Delta t}{2} H\big)\, \psi[\theta_n] \big) \big(\sigma^{(j)}\big). \tag{4.19}$$

Usually, cost functions only contain real values, but here we need complex values due to the nature of the problem. Hence, computing the gradients of the loss with respect to the parameters requires a more careful approach than the commonly employed automatic differentiation tools. Assuming that $\psi[\theta]$ is a holomorphic function of the parameters θ and following the derivation in appendix A.1 leads to

$$\frac{\partial C(\theta)}{\partial \theta_\ell} = \left\langle A\psi[\theta] - b \middle| A \frac{\partial \psi[\theta]}{\partial \theta_\ell} \right\rangle . \tag{4.20}$$

Note that since $C(\theta)$ is not holomorphic, the partial derivative on the left side of this expression is a Wirtinger derivative. It's worth mentioning that, while we have chosen to address both the real and imaginary components of the problem simultaneously, it remains a valid approach to compute the gradient independently for these two components. In such cases, the Wirtinger formalism is not a requisite tool.

4.3.2 Application to an Ising spin model

In this section, we apply the NNQS Ansatze to the transverse-field Ising model, which is explained in more detail in Sec. 2.2. As a reminder, its Hamiltonian for general lattice dimension is given by

$$H_{\mathrm{TFI}} = -J \sum_{\langle i,j \rangle} \sigma_i^z \sigma_j^z - h \sum_i \sigma_i^x, \tag{4.21}$$

where h is the external magnetic field and J is the coupling constant. Such a system undergoes a phase transition from the ferromagnetic to the paramagnetic regime at $h = J$ in one dimension [42] and at $h/J = 3.04438(2)$ in two dimensions [43]. In the following, we set the coupling constant J to 1.

4.3.2.1 1D chain with a Restricted Boltzmann machine

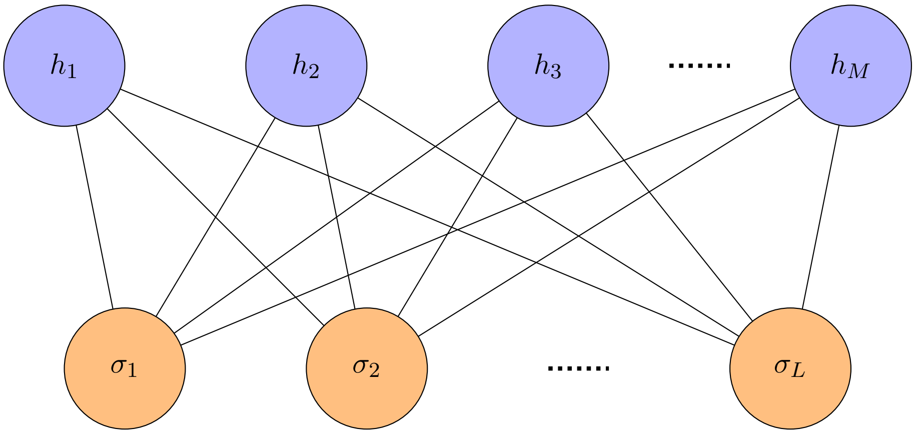


Figure 4.5: Restricted Boltzmann Machine architecture

We will start by using the RBM, given its proven success and popularity in the field. As visualized in Fig. 4.5, a RBM consists of two layers of neurons, referred to as the 'visible' and 'hidden' layers, which are connected with one another but have no intra-layer connections. The input of the visible layer is a specific spin configuration σ.

From this architecture one obtains a variational ansatz for the wavefunction of the system

$$\psi(\sigma) = \sum_{\{h_i\}} \mathrm{e}^{\sum_j a_j \sigma_j + \sum_i b_i h_i + \sum_{i,j} w_{ij} h_i \sigma_j}, \tag{4.22}$$

where a_j, b_i and w_{ij} are the network parameters, and h_i are the auxiliary spin variables which only take the values ±1. Due to the sumation over h_i in Eq. (4.22), it is possible to trace out the hidden spin variables to obtain:

$$\psi(\sigma) = \mathrm{e}^{\sum_j a_j \sigma_j} \prod_i 2\cosh\Big(b_i + \sum_j w_{ij} \sigma_j\Big). \tag{4.23}$$

We will apply it to a one dimensional chain of length $L = 20$, which allows us to calculate the exact time evolution as a reference. For the training we use the Adam optimizer, a popular gradient-based optimization algorithm used in machine learning and deep learning for training neural networks. The name "Adam" is derived from "adaptive moment estimation". It combines ideas from two other optimization techniques, RMSprop and Momentum, and is known for its effectiveness in a wide range of deep learning tasks making it often the default choice for many neural network models. We choose the recommended hyperparameters from Ref. [66].

Fig. 4.6 shows a comparison between the error obtained with stochastic reconfiguration and the error of our method. The time evolution labeled SR was computed with a pseudo-inverse threshold of 10^{-10}, while for SR(2) a threshold of 10^{-9} was used. The importance and sensitivity of this threshold is reflected in the difference of one order of magnitude in the final error between those two time evolution.

We also plot the contribution to the error by the gradient-based midpoint method. Even without a finely tuned optimization, our method yields comparable results to SR with an optimal pseudo-inverse cut-off. The increased error in comparison to exact midpoint integration suggests a limitation in expressibility arising from the RBM ansatz, notwithstanding the inherent difficulty in tracing back the origin of these errors.

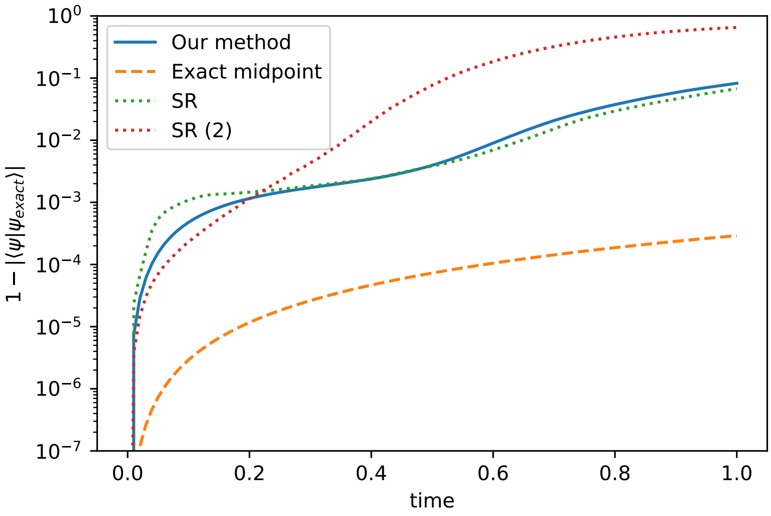


Figure 4.6: Stochastic Reconfiguration vs a gradient based method

Since we are interested in physically realistic states, the initial state was found by performing a Hamiltonian quench with respect to the field strength h. We first optimize the network parameters to represent the ground state for $h = 1.5$ and then change h to 0.75 for the real time evolution. As a measure of the accuracy of the ground state, the deviation from the exact energy is $|(E[\theta] - E_0)/E_0| \approx 0.051$, where E_0 is the exact energy found by diagonalising the Hamiltonian and $E[\theta]$ is the energy of the quantum neural network state.

Note that such a quench represents a phase transition accross the critical point. As mentioned in the introduction of this chapter, these kind of transition points still represent a challenge for NNQS [96], which could also help explain the difference in accuracy that both methods show compared to the exact numerical integration.

Our implementation comprises custom C code for the neural network architectures and optimization components, while we use a Python interface for scheduling and the evaluation of runs. The training was conducted on a single CPU with 64 GB of memory, leveraging multi-threading across 28 cores. Within this configuration, we observe that a single optimization step for the RBM network, for 20 lattice sites and 50,000 samples, takes approximately 3.6 seconds.

Fig. 4.6 directly compares SR with the midpoint optimization using the same number of time steps. However, the implicit midpoint integration method is a second-order method, which suggests the possibility of achieving a similar final error using a larger time step size for the real time evolution. To obtain a full picture of how the two methods compare, it is therefore important to consider the effects of varying the time step size.

To that effect, we apply the midpoint optimization to simulate a quench from $h = 1.5$ to $h = 0.75$ for a one-dimensional system with 15 lattice sites. As shown in Fig. 4.7 the overlap error resulting from the optimization shows very little variation as one decreases Δt, while there are big differences for the exact midpoint integrations. These results further point towards the idea that most of the error originates from the expressiveness of the architecture, and not the integration method (note that this is again a quench accross the critical point).

The small variations that are seen between time step sizes can be explained by the fact that the same learning rate and number of optimization parameters are being used in each simulation. To elaborate on this: for a larger Δt one can expect a larger update in the network parameters, and a longer optimization may be necessary (i.e. larger learning rate or more optimization steps). In practice, the small improvement may not be worth the computational expense of decreasing Δt nor increasing the number of optimization steps.

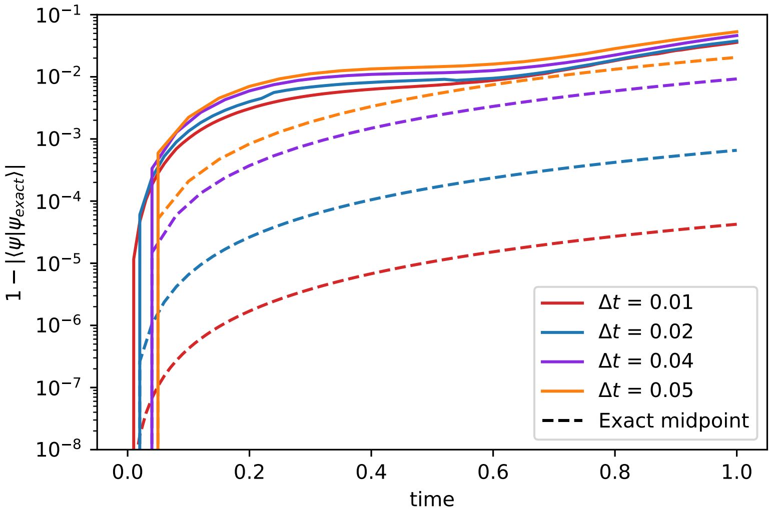


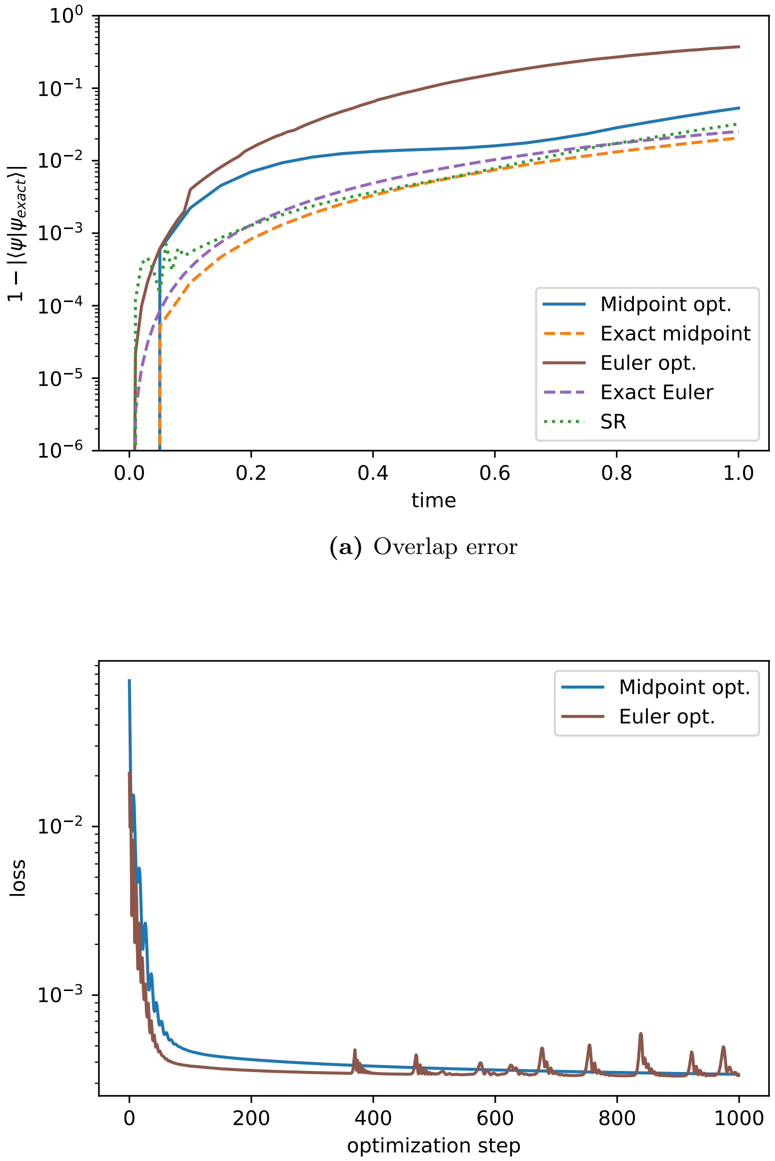
Figure 4.7: Time step size effect on the overlap error

Overlap error achieved with different time steps for the same system under the same optimization hyperparameters (RBM with 60 hidden units, same learning rate and same number of optimization steps).

Next we compare these results with those obtained using an Euler step for the optimization. In a similar fashion to Eq. (4.17) we choose as cost function

$$C(\theta_{n+1}) = \sum_{j=1}^{N} \left| \left(\psi[\theta_{n+1}] - i\Delta t H \ \psi[\theta_n]\right) \left(\sigma^{(j)}\right) \right|^2 . \tag{4.24}$$

Interestingly, we find that the results obtained are worse than those obtained with the implicit midpoint rule (see Fig. 4.8a), despite the similar errors of the exact Euler and midpoint integrations for the chosen Δt sizes. Taking a closer look at the optimization landscape shows that the loss exhibits a smoother convergence with the midpoint optimization, as shown in Fig. 4.8b. Implementing an early stop mechanism may be beneficial for the Euler optimization, although we recommend sticking to the midpoint method for its aforementioned benefits.



(a) Overlap error

(b) Example of loss convergence at $t = 0.5$

Figure 4.8: Comparison of different integration methods and time steps

We compare SR and the Euler method with $\Delta t = 0.01$ and the midpoint method with $\Delta t = 0.05$. In (b) the loss is normalized with respect to the number of samples and time steps.

4.3.2.2 2D grid with a Convolutional Neural Network

To demonstrate the flexibility of our method we change the neural network architecture and consider the time evolution governed by the two-dimensional Ising model on a $L \times L$ lattice with periodic boundary conditions, setting $L = 3$ in the following. As an ansatz, we directly use the output of a convolutional neural network,

$$\psi(\sigma) = \mathrm{CNN}(\sigma) \tag{4.25}$$

as shown in Fig. 4.9. Specifically, the network architecture consists of a convolutional layer with five complex-valued 2×2 filters and periodic boundary conditions, an intermediate dense layer with 10 neurons, and a single output. This correspond to 496 complex parameters in total.

For the activation functions, we found the polynomial functions described in Ref. [89] to achieve errors several orders of magnitude smaller compared to other more commonly used activation functions, such as ReLUs or sigmoid functions. Specifically, we use

$$\sigma(z) = \frac{1}{2}z^2 - \frac{1}{12}z^4 + \frac{1}{45}z^6 \tag{4.26}$$

for the convolutional layer, and

$$\sigma(z) = z - \frac{1}{3}z^3 + \frac{2}{15}z^5 \tag{4.27}$$

for subsequent layers.

We perform three different Hamiltonian quenches, from “infinite” h to $h = 2h_c$, h_c and $h_c/10$, where h_c is the critical point at which the system undergoes a phase transition. Here “infinite” h is equivalent to retaining only the second term in the Hamiltonian (4.21), such that the corresponding ground state is the paramagnetic state $\psi_0 = \prod_{j=1}^{L} |+\rangle_j$ with $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. We are able to precisely represent ψ_0 using our network ansatz, such that the relative energy error is on the order of 10^{-11}.

The top row in Fig. 4.10 shows the overlap error with respect to the exact wavefunction as a function of time, as well as the error resulting from a plain midpoint integration. Each time step was optimized using the same learning rate and number of iterations, but a more careful optimization could further lower the error. The bottom row of Fig. 4.10 shows the evolution of the transverse magnetization, based on our CNN ansatz and the numerically exact curve as reference. In agreement with Ref. [96], we find quenches to the critical point to be the hardest to optimize.

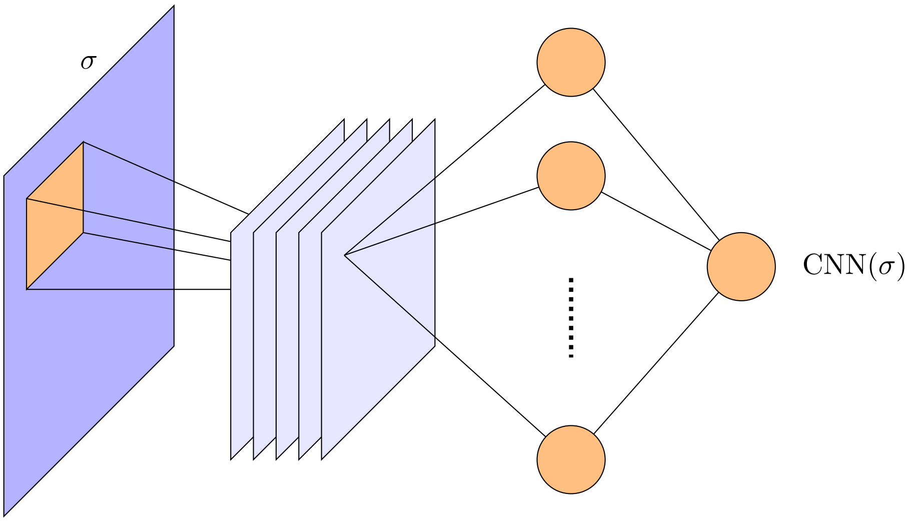


Figure 4.9: Schematic diagram of the Convolutional Neural Network used

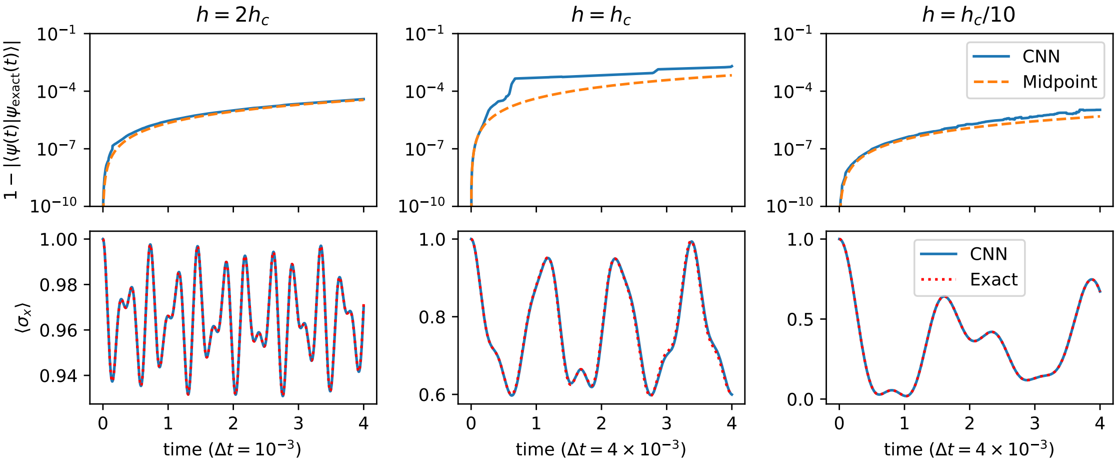


Figure 4.10: Performance of midpoint method for a 2D Ising model

Overlap error (top row) and transverse magnetization (bottom row) of the real time evolution governed by the Ising Hamiltonian (4.21) on a 3×3 lattice after a quench of h. Each column corresponds to a different value of h after the quench, starting from "infinite" h. We used 500 uniformly drawn samples for each individual optimization step of our method (CNN).

5 Quantum Process Tomography via time-delayed measurements

Next, we will focus on the characterization of many-body quantum systems via QPT. QPT aims to reconstruct the mathematical representation, often a matrix or an operator, that describes how a quantum system transforms its initial state into a final state. This is particularly relevant in the field of quantum information and quantum computing, as it plays a crucial role in the development and verification of quantum algorithms and quantum devices [122–128]. By characterizing the quantum processes involved, researchers can identify and address issues such as decoherence, noise, and other errors in quantum computations.

However, the implementation of the textbook algorithm for QPT in laboratory settings poses a great challenge due to the exponential increases in the needed resources. Such an algorithm requires the preparation of a 'complete set' of N^2 initial states –where N denotes the Hilbert space dimension– and to measure the process outputs with respect to a similarly 'complete' set of observables (see Sec. 2.3). Both of those sets increase exponentially as we add qubits to the system. On top of that, the intrinsically noisy nature of real-life experiments complicates the picture, as the state preparation and measurement steps may not be completely reliable.

In this chapter, we will propose an optimization based method based on time-delayed measurement, which aims to reduce the number of resources needed to perform QPT. These results were published in Ref. [129]

5.1 State of the art

A common technique used to bypass the limitations of standard QPT is that of randomized benchmarking [130–133]. In this protocol, random quantum gate sequences are generated such that, if the sequences were implemented without errors, they would be equivalent to an identity operator. Calculating the fidelity of multiple runs of sequences of different length, allows for the estimation of the expected value of the sequence fidelity. This, in turn, provides a calibration reference for experimental data.

The simplicity of the approach has rendered randomized benchmarking as one of the most popular choices of experimentalists to benchmark their devices [134–139]. Let us emphasize, however, that this protocol is not a method to perform QPT. While it offers an estimate of gate fidelity, randomized benchmarking cannot reconstruct a completely unknown gate, nor can it shed any light on the types of errors occurring in the circuit.

A similar thing occurs with Gate Set Tomography [140–142]. This protocol is a calibration-free approach to process tomography. By 'calibration-free' we mean that it does not rely on the common QPT assumption that the initial states and the measurement operations are well known, and, as a consequence, it provides an estimation of the process which includes the errors (and their relationships) of the preparation and measurement stages. In experimental settings, this is many times all that is needed, so Gate Set Tomography has been extensively and successfully utilized for the calibration and benchmarking of quantum devices [143–148]. But, as with randomized benchmarking, this approach has limitations when the focus is on the comprehensive characterization of the dynamics alone.

However, if we are restricting our analysis to situations with only quantum almost-unitary gates, the resource-expensiveness of the standard QPT algorithm can be decreased. Specifically, the reduced number of free parameters for reconstruction enables a corresponding reduction in the number of observables needed, decreasing from $\mathcal{O}(N^4)$ to $\mathcal{O}(N^2)$ [149,150]. It is important to recall that, here, N represents the Hilbert space dimension, so this still corresponds to an exponential increase with system size.

More drastically, it has been shown that the minimal set of initial states required for the characterization of a unitary consists of only *two* states, regardless of the size of the system [151]. This has two important caveats. First of all, it only allows for the reconstruction of the unitary part of a process, so the error sources cannot be completely identified. And secondly, one of the states of the minimal set is a totally mixed thermal state, of impractical use during experiments.

Building on the idea of utilizing information of the underlying process to reduce the experimental requirements, the method of compressed sensing was proposed [152–154]. The underpinning theory of this protocol comes from the field of classical signal processing [155], and relies on the assumption that the dynamical process we want to identify is (nearly) sparse. This, for example, holds for unitary operators, which are maximally sparse in their eigenbasis. Compressed sensing allows for an exponential reduction of the required measurements –although at the cost of an exponentially expensive classical post-processing step. For this reason, it has become a very popular approach to conduct QPT [156–160].

An important remark is that the main step of compressed sensing involves a least squares minimization. In a time where machine learning and optimization techniques are constantly refined and even surpassed by more complex methods, it is not surprising that, more recently, a variety of numerical optimization-based techniques have been proposed. An example of this can be found in Ref. [161], where the authors employ an efficient representation of a Choi matrix using a tensor network. They combine this representation with an unsupervised machine learning algorithm to learn the optimal parameters for the tensor network. The high model flexibility of gradient descent has allowed for similar works to be done with a variety of Ansatze, from neural networks to simple parametrizations of the Kraus operators [162–164].

All of the methods mentioned up to this point do not make explicit use of the time trajectories themselves. The strategies that leverage the dynamics as a resource to reduce the number of measurement basis needed are few and far between (but see [165–169]). And the theoretical foundation for such approaches –as well as the specific bounds of required resources– has not been laid.

5.2 Takens' theorem

Here we introduce a strategy for conducting QPT, one that relies on the concept of time-delay embedding. The underpinning theory for this embedding technique draws its roots from the theorems of Takens and Ruelle [170, 171], which are, in turn, based on the embedding theorems of Whitney [172]. Before delving into the details of our specific approach it is therefore useful to stop and consider in detail what we can learn from the theorems of Takens [171] in combination with the results from Packard et al. [173] and Aeyels [174].

Let $k \geq d \in \mathbb{N}$, and $\mathcal{M} \subset \mathbb{R}^k$ be a d-dimensional, compact, smooth, connected, oriented manifold with Riemannian metric g induced by the embedding in k-dimensional Euclidean space. We can define a dynamical system through its state space (here, the manifold $\mathcal{M}$) and a diffeomorphism $\phi : \mathcal{M} \to \mathcal{M}$. Note that ϕ can simply be interpreted as the time-evolution mapping of the state space. The last ingredient necessary for the theorem is the concept of observability. That is, for each state of the manifold we assume that we are able to measure a property y. Then, one version of Takens theorems states:

Theorem 1. ***Generic delay embeddings*** *For pairs (ϕ, y), $\phi : \mathcal{M} \to \mathcal{M}$ a smooth diffeomorphism and $y : \mathcal{M} \to \mathbb{R}$ a smooth function, it is a generic property that the map*

$\Phi_{(\phi,y)} : \mathcal{M} \to \mathbb{R}^{2d+1}$, defined by

$$\Phi_{(\phi,y)}(x) = \Big(y(x), y(\phi(x)), \ldots, y(\underbrace{\phi \circ \cdots \circ \phi}_{2d\ times}(x))\Big) \tag{5.1}$$

is an embedding of $\mathcal{M}$; here, "smooth" means at least C^2.

Genericity in the above is understood as "an open and dense set of pairs (ϕ, y)" in the C^2 function space. Open and dense sets can have measure zero, so Sauer et al. [175] later refined this result significantly by introducing the concept of prevalence (a "probability one" analogue in infinite dimensional spaces). A similar result can be obtained even when considering stochastic systems [176].

It is important to note that the use of Takens theorem for system dynamics reconstruction is not new. In fact, it has been fundamental to several system identification methods for general dynamical systems [177]. Recently, it has also been used in combination with machine learning techniques [178–181], with a focus on partial differential equations [182, 183] or classical Hamiltonian dynamics [184, 185]. The reconstruction of unitary dynamics had also been considered [186, 187], and these works have even been extended for dissipative systems [188–190]

Our goal here is to apply this theorem in the context of quantum processes, and to consider the challenges associated with quantum Hilbert spaces. Once again, let N denote the quantum Hilbert space dimension. Then, in our particular case, $\mathcal{M}$ is the special unitary group $\mathrm{SU}(N) \subset \mathbb{C}^{N\times N}$ when identifying $\mathbb{C} \simeq \mathbb{R}^2$. In particular, $\mathcal{M}$ (with underlying field $\mathbb{R}$) has dimension $d = N^2 - 1$. In physical terms, the elements of $\mathrm{SU}(N)$ are the unitary time evolution matrices.

Note that such manifold satisfies the compact requirement due to the property

$$\det(U) = 1. \tag{5.2}$$

Furthermore, by definition, any Lie group is smooth and can be shown to be orientable [191]. The connected requirement, on the other hand, is more restrictive than allowed by the results presented later in this chapter. However, this setting is sufficient for our presentation.

At the core of our approach lies the fundamental concept of "embedding" the manifold of unitary matrices, which characterizes the dynamics of a quantum system, into the space of measurement trajectories. We work with the assumption that the Hamiltonian H is traceless, as adding to H a term which is a multiple of the identity only leads to a global phase factor in the time evolution, which is unobservable in subsequent measurements as envisioned here. We also fix a time step Δt, which we absorb into H without loss of generality, and therefore set U to

$$U = \mathrm{e}^{-iH} \,. \tag{5.3}$$

From the perspective of the manifold of unitary time-evolution operators, the diffeomorphism ϕ consists of a scaling by a power $\gamma \in \mathbb{R}$ ($\gamma > 0$, $\gamma \neq 1$), as this is equivalent to scaling the Hamiltonian by a coefficient, which can be interpreted as scaling of the absorbed time step:

$$\phi(U) = U^{\gamma} = \mathrm{e}^{-i\gamma H} \,. \tag{5.4}$$

Regarding the observability characteristic y, for quantum states this is naturally defined by the concept of observable hermitian matrices M. By fixing a randomly chosen initial quantum state $\psi \in \mathbb{C}^N$, we can define y to be the corresponding expectation value of M:

$$y : \mathcal{M} \to \mathbb{R}, \quad y(U) = \langle\psi|\, U^{\dagger} M U \,|\psi\rangle \,. \tag{5.5}$$

Putting all these ingredients together we can now apply Takens theorem, which tells us that the embedding of our dynamical process $\Phi_{(\phi,y)}$ in Eq. (5.1) is given by

$$\begin{aligned}\Phi_{(\phi,y)}(U) = \Big(&\langle\psi|\,\mathrm{e}^{iH}\, M\, \mathrm{e}^{-iH}\,|\psi\rangle \,, \langle\psi|\,\mathrm{e}^{i\gamma H}\, M\, \mathrm{e}^{-i\gamma H}\,|\psi\rangle \,, \\ &\ldots, \langle\psi|\,\mathrm{e}^{i\gamma^{2d} H}\, M\, \mathrm{e}^{-i\gamma^{2d} H}\,|\psi\rangle \Big),\end{aligned} \tag{5.6}$$

physically corresponding to measurements at time points $t_q = \gamma^q$ for $q = 0, \ldots, 2d$.

The main point here is that, since such an embedding exists, there is a possibility of identifying the time step matrix U, and consequently the Hamiltonian, based on a single measurement time trajectory, under the assumption that the initial quantum state is known. A possible way of thinking about this is as finding the inverse of a function for which we know the result. As caveat, the relation $U = \mathrm{e}^{-iH\Delta t}$ determines the eigenvalues of H only up to multiples of $2\pi/\Delta t$. But more worryingly, due to the "collapsible" nature of quantum measurements, the map $\Phi_{(\phi,y)}$ might not be one-to-one, in the sense that two different unitary matrices give rise to the same measurement trajectory.

In the next section we will discuss this problem in more detail for single-qubit systems, which allows us to shed some light on how these issues can be avoided in practice: either by additional assumptions on the structure of H or by extra measurements to resolve the ambiguity.

5.3 Algorithm for single-qubit system

For our measurement definition in Eq. (5.5) we have assumed that it is feasible to reliably prepare a single (or when indicated several) known initial state(s) $\psi \in \mathbb{C}^N$ in an experiment. Although there is no such thing as a faultless quantum experiment, the high-fidelity preparation of states has been demonstrated amongst a variety of hardware platforms (see for example [192–194]). We will therefore uphold this assumption throughout the remainder of the chapter.

Besides that, our methods do not impose any requirements on the initial state. Thus, unless otherwise specified, ψ is chosen at random in the following algorithms and numerical simulations. More concretely, the entries of ψ before normalization are independent and identically distributed (i.i.d.) random numbers sampled from the standard complex normal distribution.

In this section, we will consider the reconstruction of a single-qubit Hamiltonian, as the simplest toy-model case that we can tackle. Although such a system may not be the most interesting from an experimenters perspective, it allows us to derive a concrete algorithm for the reconstruction of the Hamiltonian, which in turn provides some insight on the problems that may arise for higher dimensions.

The Bloch sphere picture [18] provides a geometric framework to work with single-qubit quantum states and operations. The Bloch vector $\vec{r} \in \mathbb{R}^3$ associated with $\psi \in \mathbb{C}^2$ is defined via the relation

$$|\psi\rangle \langle\psi| = \frac{1}{2}(I + \vec{r} \cdot \vec{\sigma}). \tag{5.7}$$

Since we are dealing with a single isolated qubit –a.k.a a pure state–, $\vec{r}$ is a unit vector. We can parametrize any traceless single-qubit Hamiltonian as

$$H = \vec{h} \cdot \vec{\sigma} \tag{5.8}$$

with $\vec{h} \in \mathbb{R}^3$. Recall that we are ignoring the global phase factors of quantum states due to their lack of relevance for our measurement framework. It follows that the corresponding time evolution operator is the rotation

$$U(t) = \mathrm{e}^{-iHt} = \cos(\omega t/2)I - i\sin(\omega t/2)(\vec{v} \cdot \vec{\sigma}) \in \mathrm{SU}(2), \tag{5.9}$$

when representing $\vec{h} = \omega\vec{v}/2$ by a unit vector $\vec{v} \in \mathbb{R}^3$ and $\omega \in \mathbb{R}$. The geometrical interpretation of this operator on the Bloch sphere is that of a classical rotation about

$\vec{v}$, as illustrated in Fig. 5.1 and described by Rodrigues' rotation formula ($\theta = \omega t$):

$$\mathsf{U}_{\theta,\vec{v}}\,\vec{r} = \cos(\theta)\vec{r} + \sin(\theta)(\vec{v}\times\vec{r}) + (1-\cos(\theta))(\vec{v}\cdot\vec{r})\,\vec{v}. \tag{5.10}$$

$\mathsf{U}_{\theta,\vec{v}} \in \mathrm{SO}(3)$ is a rotation matrix (parametrized by θ and $\vec{v}$) applied to $\vec{r}$. Although having previously applied Takens' theorem to $U = U(\Delta t)$, we can similarly work with $\mathsf{U} = \mathsf{U}_{\omega\Delta t,\vec{v}}$ and its matrix powers.

The time trajectory effected by the rotation applied to $\vec{r}$ results in a circle on the surface of the Bloch sphere (see Fig. 5.1). Full knowledge of such circular trajectory and its dynamics would allow us to determine $\vec{v}$, and the angular frequency ω. By construction we know one the initial point of the circle, $\vec{r}$, but we only have access to the expectation value of an observable M for $t > 0$. Such expectation values can be interpreted geometrically as the projection of the trajectory onto a specified measurement vector. In the following, we denote the "measurement direction" by $\vec{m} \in \mathbb{R}^3$, i.e., M is parametrized as $M = \vec{m}\cdot\vec{\sigma}$.

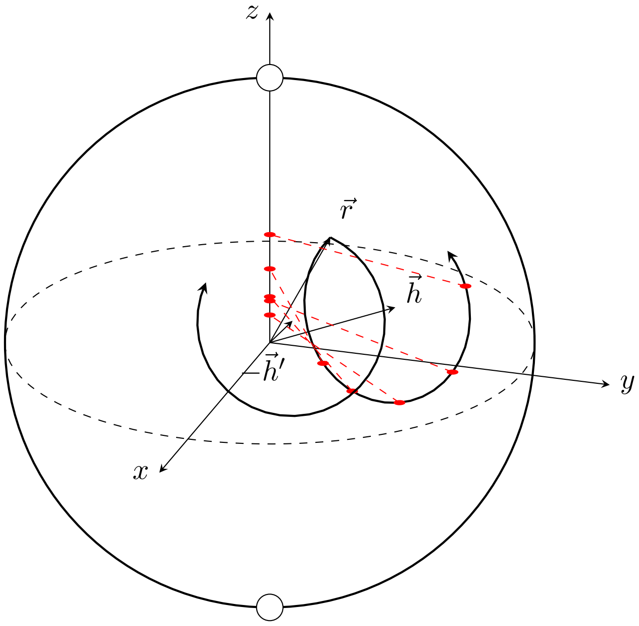


Figure 5.1: Single qubit time evolution on a Bloch sphere

The black arrows represent the time trajectory of a qubit starting at $\vec{r}$ under a Hamiltonian defined by the vector $\vec{h}$. The red dots represent the measurement time points and their respective projections on the z-axis (i.e. the expectation value of the observable Z). Note how a symmetric trajectory shares the same projections.

Algorithm 1 enables the recovery of ω and $\vec{v}$ using the projections of the time trajectory on $\vec{m}$. In broad terms, this algorithm involves initially reconstructing ω based on the time dependence, and use the found frequency to determine the unit vector $\vec{v}$.

Algorithm 1 Reconstruction of a single-qubit Hamiltonian (measurement direction $\vec{m} \in \mathbb{R}^3$ with $\vec{m} \nparallel \vec{r}$)

```
   Input: Measurement averages $y_q = \vec{m} \cdot (\mathsf{U}_{\omega t_q, \vec{v}}\, \vec{r})$ with $t_q = \Delta t\, \gamma^q$ for $q = 0, \dots, 2d$
   Output: Hamiltonian parameters $\omega$ and $\vec{v}$.
1: Find $\omega$ based on the dependency on $\omega t$ in Eq. (5.10):
       $\omega = \operatorname*{argmin}_{\omega > 0} \min_{a,b,c} \sum_{q=0}^{2d} |y_q - a\cos(\omega t_q - b) - c|^2$
2: Set $\tilde{y}_q = y_q - \cos(\omega t_q)(\vec{m} \cdot \vec{r})$ for $q = 0, \dots, 2d$ (subtract term which is independent
   of $\vec{v}$)
3: Find $\alpha_1 = \vec{m} \cdot (\vec{v} \times \vec{r})$ and $\kappa = (\vec{v} \cdot \vec{r})(\vec{m} \cdot \vec{v})$ via a least squares fit:
       $\alpha_1, \kappa = \operatorname*{argmin}_{\alpha_1, \kappa} \sum_{q=0}^{2d} |\tilde{y}_q - \sin(\omega t_q)\alpha_1 - (1 - \cos(\omega t_q))\kappa|^2$
4: Represent $\vec{v}$ with respect to the orthonormal basis
       $\left\{ \vec{u}_1 = \frac{\vec{r} \times \vec{m}}{\|\vec{r} \times \vec{m}\|}, \vec{u}_2 = \frac{\vec{r} + \vec{m}}{\|\vec{r} + \vec{m}\|}, \vec{u}_3 = \frac{\vec{r} - \vec{m}}{\|\vec{r} - \vec{m}\|} \right\} :$
       $\vec{v} = \alpha_1 \vec{u}_1 + \alpha_2 \vec{u}_2 + \alpha_3 \vec{u}_3,$
   with to-be determined coefficients $\alpha_2, \alpha_3 \in \mathbb{R}$. Using that $\vec{u}_1$, $\vec{u}_2$ and $\vec{u}_3$ are eigen-
   vectors of $K = \frac{1}{2}(\vec{m} \otimes \vec{r} + \vec{r} \otimes \vec{m})$ with respective eigenvalues 0, $\lambda_+ = \frac{1}{2}(1 + \vec{m} \cdot \vec{r})$
   and $\lambda_- = -\frac{1}{2}(1 - \vec{m} \cdot \vec{r})$, it follows that
       $\kappa = \vec{v} \cdot K\vec{v} = \lambda_+ \alpha_2^2 + \lambda_- \alpha_3^2.$
   Together with the normalization condition
       $\alpha_1^2 + \alpha_2^2 + \alpha_3^2 = 1,$
   this leads to $\alpha_2^2 = \kappa - (1 - \alpha_1^2)\lambda_-$ and $\alpha_3^2 = 1 - \alpha_1^2 - \alpha_2^2$. Decide between the four
   possible signs of $\alpha_2, \alpha_3$ using a-priori information of $H$ or one additional measurement
   in a different basis.
```

It is important to note that the non-linear optimization in the first step of the algorithm might get trapped in a local minimum, which could be resolved by restarting the optimization. Fortunately, when neglecting uncertainties associated with the measurements, a correct solution is indicated by a zero residual both in steps 1 and 3. We assume that a range of realistic frequencies ω is known beforehand, such that the Nyquist condition (given the non-uniform sampling points t_q) holds [195].

5.3.1 An aside on ambiguities

As mentioned in the previous section, the solution to this problem is not unique. A closer inspection of the algorithm showcases clearly where the problem lies: there are four possible combinations for the signs of the coefficients α_2 and α_3. Fig. 5.1 visualizes how two rotations can generate the same projection onto the z-axis, with $\vec{h}'$ resulting from $(\alpha_2, \alpha_3) \to -(\alpha_2, \alpha_3)$. The two equations for α_2 and α_3 are shown in Fig. 5.2. (In the example, $\alpha_1 = -0.2051$, and hence the radius of the norm constraint circle is close to 1.)

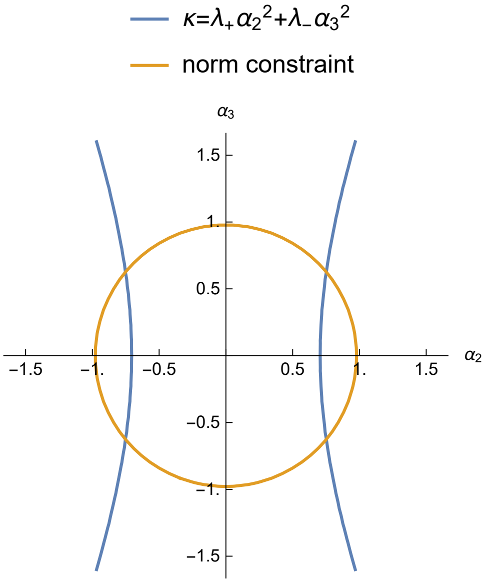


Figure 5.2: System of equations to obtain the vector representation of the Hamiltonian The points where the lines meet are the possible solutions for α_2 and α_3.

In order to decide between these distinct solutions, one may use several approaches. The simplest way to resolve the ambiguity is to include one further measurement in a different basis. In most cases, a projection to a different vector will be distinct for the different trajectories, which allows to determine which of the four solutions corresponds to the additional measurement. However, there are some edge cases where this is not true. As an example, consider the two time trajectories depicted in Fig. 5.1. Any measurement vector lying on the plane defined by $\vec{r}$ and the z-axis will result in the same expectation values for both Hamiltonians. Although identifying and avoiding these problematic measurement vectors is feasible for a single-qubit system, it becomes harder as the number of dimensions increases, so an alternative method may be more suitable.

A second approach to distinguish between the solutions could be to consider the time trajectories for two different initial states. Under the same Hamiltonian, two states with Bloch vectors $\vec{r}$ and $\vec{r'}$ will display different dynamical evolutions. Using Alg. 1 one finds four possible rotation vectors for each. The common solution will correspond to the desired Hamiltonian. It is not guaranteed, however, that the two will have a single common solution. Going back to the example shown in Fig. 5.1, if $\vec{r'}$ also lies in the plane defined by $\vec{r}$ and the z-axis, they will have at least two common solutions for $\vec{h}$. Ref. [151] might shed some light on the optimal selection of the initial states to avoid this problem.

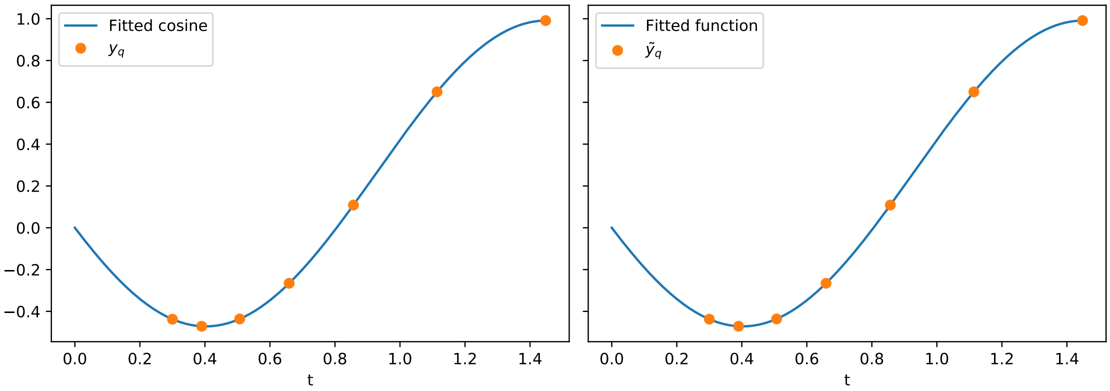
Note, however, that the exceptions to the above two approaches are edge cases, unlikely to occur when both the initial state(s) and measurement basis (or bases) are chosen randomly. And the probability further decreases when the two approaches are combined. However, one may be tempted to get rid of the randomness and select all the involved vectors from a simplified set of unit vectors, such as vectors on a specified plane. Such a setting will inevitably lead to unresolvable ambiguities.

At this point it may be valuable to reiterate that the above picture becomes much more obscure when dealing with systems with more than one qubit, as the reflections to obtain symmetrical solutions are harder to visualize. If an experimenter only has access to a simplified set of measurement bases and initial states, the final ambiguity in solutions may only be resolved by incorporating previous knowledge of the Hamiltonian into the model.

5.3.2 Single qubit numerical experiments

With Algorithm 1 we are able to reconstruct a single-qubit Hamiltonian up to numerical precision. Fig. 5.3 shows the results obtained for the ω and least squares optimizations. Following the Takens framework, we use $2d+1$ time points, where $d = 3$ is the number of Hamiltonian parameters to be reconstructed. Specifically, we have used the time points $t_q = 0.3 \times (1.3)^q$ for $q = 0, \dots, 6$ here. With this, we are able to find ω, α_1 and κ up to an error of $\sim 10^{-15}$. Thus 7 measurements, plus a further measurement in a different basis to distinguish between the possible solutions due to symmetry, are sufficient for the reconstruction.

This tells us that, in the ideal setting with no noise at the stage of state preparation nor the stage of observation, using time-measurements as described in Sec. 5.2 allows us to exactly characterize the dynamical process. But such a setting is not realistic. Even when maintaining the assumption of a high fidelity initial state, the measurement process itself may be a source of error.



(a) Step 1 from Alg. 1

(b) Step 3 from Alg. 1

Figure 5.3: Single-qubit Hamiltonian reconstruction in a noiseless experiment

We apply Algorithm 1 using Z-basis measurements, for the time evolution shown in Fig. 5.1. (a) shows the cosine fit to find ω, while (b) shows the fit for the function containing α_1 and κ. In this specific case, $y_q = \bar{y}_q$.

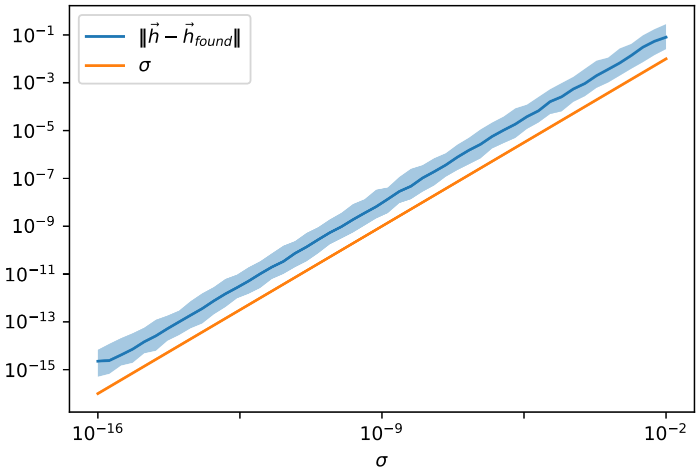


Figure 5.4: Sample sensitivity in single-qubit Hamiltonian reconstruction

The solid line shows the median, while the lighter shade shows the 10% quantiles.

On the one hand, the expectation value is by definition only an estimation obtained from a number of samples. In the real life case with finite samples, stochastic fluctuations are unavoidable. One can model such source of noise by replacing the expectation values from Eq.(5.5) by

$$y'(U) = y(U) + \mathcal{N}(0, \sigma^2) \tag{5.11}$$

where the standard deviation, σ quantifies the uncertainty stemming from the limited number of samples (the higher the sample number, the lower the σ). Figure 5.4 shows that the reconstruction error is related linearly to σ. This is to be expected, as the arguments of the cost functions minimized in Alg. 1 are related linearly to the obtained measurements.

On the other hand, and perhaps more importantly, measurements are also subject to environmental noise and technical limitations [196, 197]. Consider for example the case of superconducting qubits, where the qubit readout involves the frequency measurement of an electromagnetic wave [198,199]. Multiple issues could affect the readout here, from environmental photons entering the measurement device, to the inherent uncertainty in pulse width and energy [200, 201].

The analysis done to produce Fig. 5.4 fails to take into account two important types of errors: systematic errors which shift all measurements in the same direction (stemming, for example, from a problem in calibration), and non-linear errors. Systematic errors can be taken into account by introducing a shift in the mean of the measurement,

$$y'(U) = y(U) + \mu. \tag{5.12}$$

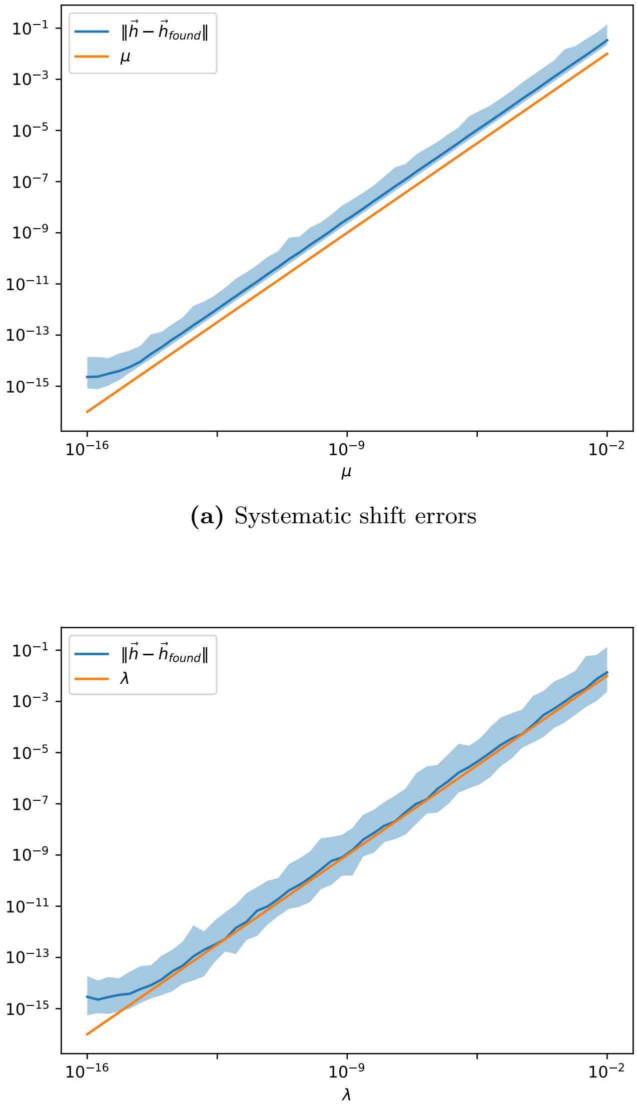
Fig. 5.5a shows how the error in $\vec{h}$ behaves as a function of the shift μ. Unsurprisingly, we also observe a linear relationship here.

Non-linear errors, on the other hand, are more dependent on the hardware set-up itself. In an attempt to keep our results as general as possible, we opt for applying a simple non-linear modification to the measurement results themselves

$$y'(U) = \operatorname{sgn}\left(y(U)\right) \left|y(U)\right|^{1+\mathcal{N}(\lambda,1)}, \tag{5.13}$$

where we have taken the measurement sign out of the exponential to ensure $y'(U) \in \mathbb{R}$.

On Fig. 5.5b we can see that the Hamiltonian error also presents a linear dependency on such a λ. This leads us to conclude that our QPT approach is resistant to at least some non-linear sources of error. Notice, however, that these previous two models only consider a consistent source of systematic or non-linear error. When each sample in the experiment may be subject to different noise sources, the estimated accuracy of the Hamiltonian reconstruction should be regarded with skepticism. Thus, that a more complex simulation might be necessary to ensure that the method is not sensitive to hardware-specific sources of errors [200, 202, 203].



(a) Systematic shift errors

(b) Non-linear errors

Figure 5.5: Sensitivity to different types of errors in single-qubit Hamiltonian reconstruction The solid line shows the median, while the lighter shade shows the 10% quantiles.

5.4 Relaxation as a numerical strategy for multiple qubit optimization

The next natural step after considering in detail the single-qubit reconstruction algorithm would be to extend it to multiple qubit systems. Unfortunately for us, however, Algorithm 1 does not extend trivially to larger dimensions, and we are forced to look for an alternative optimization strategy.

As a straightforward approach for determining a unitary time step matrix U which matches the measurement averages, one could start from the natural parametrization $U = \mathrm{e}^{-iH}$ (with the time step already absorbed into H), and then optimize the Hermitian matrix H directly. However, this method encounters the difficulty of a complicated optimization landscape with many local minima, in particular when using gradient descent-based approaches.

To visualize this, let us consider again a Hamiltonian as in Eq.(5.8), where the coefficients to the Pauli matrices have been drawn from a standard normal distribution. Let us also fix the time-step length to $\Delta t = 0.1$. We can then model our time evolution operator as

$$U = e^{-i\Delta t \vec{h}\cdot\vec{\sigma}}, \tag{5.14}$$

where the entries of $\vec{h}$ are our variational parameters. The manifold for the Takens embedding has dimension $d = 3$, and hence, $2d + 1 = 7$ time points should be used. However, as discussed in Sec. 5.3.1, a single measurement basis will lead to to multiple symmetric solutions. For this reason, we combine measurements in the Pauli-X and Pauli-Z bases and reduce the number of time points to 4 to compensate for the additional source of data.

The parameters can be optimized directly by minimising

$$\mathcal{L} = \sum_{q=1}^{4} \left\| y_q^{(Z)} - \langle\psi|(U^q)^\dagger Z U^q|\psi\rangle \right\|^2 + \sum_{q=1}^{4} \left\| y_q^{(X)} - \langle\psi|(U^q)^\dagger X U^q|\psi\rangle \right\|^2, \tag{5.15}$$

where $y_q^{(M)}$ denotes the measurement at time point q using the measurement basis of operator M. We utilize the Adam optimizer as described in Ref. [66], training for up to 12000 epochs (note that considering we are dealing with only 8 data points this is not as computationally intensive as it would appear on first sight).

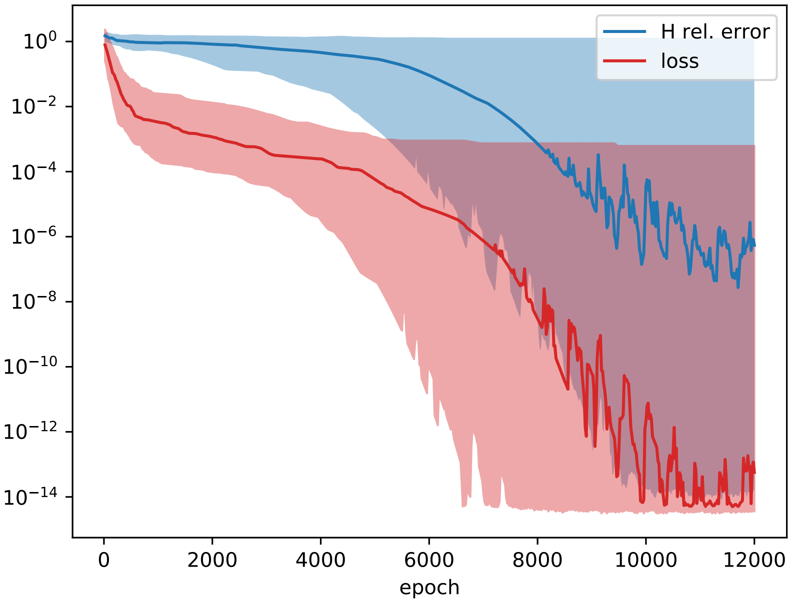


Figure 5.6: Direct optimization of H via gradient descent methods

Solid lines show the median, and lighter shades are the 25% quantiles.

We repeat the optimization multiple times for several random Hamiltonians in order to obtain statistics of how the loss and relative error behave in a general scenario. Fig. 5.6 shows the median and confidence interval throughout the optimization. Despite the fact that both the loss and Hamiltonian error decrease on average, we can deduce from the wide confidence bound that there is a non-negligible number of cases where the optimizer gets stuck in local minima.

To mitigate the considerable variation observed across different runs, we suggest an alternative numerical approach designed for generic (dense) few-qubit Hamiltonians. As discussed in Sect. 5.3, for single qubits a transformation $|\psi'\rangle = U\,|\psi\rangle$ by a unitary matrix $U \in \mathrm{SU}(2)$ can be geometrically interpreted as a spatial rotation in three dimensions on the Bloch sphere: $\vec{r}' = \mathsf{U}r$, with $\mathsf{U} \in \mathrm{SO}(3)$ given by Rodrigues' formula (5.10). It is possible to obtain U from U via

$$\mathsf{U}_{\alpha,\beta} = \frac{1}{2}\,\mathrm{tr}\big[\sigma^\alpha U \sigma^\beta U^\dagger\big] \tag{5.16}$$

for $\alpha, \beta = 1, 2, 3$, where we have used Eq. (5.7) together with the orthogonality relation of the Pauli matrices:

$$\frac{1}{2}\,\mathrm{tr}[\sigma^\alpha \sigma^\beta] = \delta_{\alpha\beta}. \tag{5.17}$$

The main advantage of using the Bloch picture in this setting is that measurements as described in Eq. (5.5) depend linearly on the Bloch vector $\vec{r}$, and hence on U, as

$$\langle\psi'|M|\psi'\rangle = \vec{m}\cdot\vec{r}' = \vec{m}\cdot(\mathsf{U}\vec{r}). \tag{5.18}$$

Note that $\vec{m}$ can always be normalized.

This enables us to divide the problem into two stages: initially optimizing the search for a U that aligns with measurement data, which is hopefully a simpler task than directly optimizing for U; followed by computing U using the optimized U and Eq. (5.16).

Such a model, in contrast to Alg.1, can be generalized to larger systems via tensor products of Pauli strings. For instance, Eq. (5.16) can be extended for the n-qubit case to $\mathsf{U}\in \mathrm{SO}(4^n-1)$ with entries

$$\mathsf{U}_{\alpha,\beta} = \frac{1}{2^n}\operatorname{tr}\left[(\sigma^{\alpha_1}\otimes\cdots\otimes\sigma^{\alpha_n})U(\sigma^{\beta_1}\otimes\cdots\otimes\sigma^{\beta_n})U^\dagger\right], \tag{5.19}$$

where $U\in \mathrm{SU}(2^n)$, and α and β are index tuples from the set $\{0,1,2,3\}^{\otimes n}\backslash\{(0,\dots,0)\}$, using the convention that σ^0 is the 2×2 identity matrix. Note that we are able to exclude the tuple $(0,\dots,0)$ from the set of indices, as the identity matrix is always mapped to itself by unitary conjugation.

After switching to the described Bloch representation, we "relax" the condition that an involved (real) matrix U is orthogonal, by allowing for any real matrix, but a penalisation term of the form

$$\mathcal{L}_{\text{orth}} = \left\|\mathsf{U}\mathsf{U}^T - I\right\|_{\mathsf{F}}^2 \tag{5.20}$$

to the overall cost function, where $\|\cdot\|_{\mathsf{F}}$ denotes the Frobenius norm. By doing so, we avoid parametrising U in a way that enforces strict orthogonality. Despite the fact that this, in principle, extends the optimization landscape to the larger space of all real matrices, we observe that the a better behaviour throughout the optimization, as the optimizer seems to encounter less local minima than in the previous one.

By choosing $\gamma = 2$ in Eq. (5.4), the time steps $t_q = \gamma^q$ are integers, and we can generate corresponding powers of U by defining $\mathsf{U}_0 = \mathsf{U}$, $\mathsf{U}_q = \mathsf{U}_{q-1}^2$ for $q = 1,\dots,2d$, such that

$$\mathrm{e}^{-i\gamma^q H} = \mathsf{U}^{\gamma^q} = \mathsf{U}_q \quad \text{for } q = 0,\dots,2d. \tag{5.21}$$

As a further "relaxation", we treat the U_q's as separate matrices. The relation between them is only set via an additional penalization term:

$$\mathcal{L}_{\text{steps}} = \sum_q \left\| \mathsf{U}_q - \mathsf{U}_{q-1}^2 \right\|_{\mathsf{F}}^2. \tag{5.22}$$

To test this approach we repeat the experiments described previously for Fig. 5.6, this time with the overall cost function

$$\mathcal{L} = \sum_{q,M} \left\| y_q^{(M)} - \vec{m} \cdot (\mathsf{U}\vec{r}) \right\|^2 + \mathcal{L}_{\text{orth}} + \mathcal{L}_{\text{steps}}. \tag{5.23}$$

Fig. 5.7 shows the median and confidence interval with this relaxation strategy. When compared to the previous results, it is clear that in this new optimization landscape the optimizer provides more consistent results.

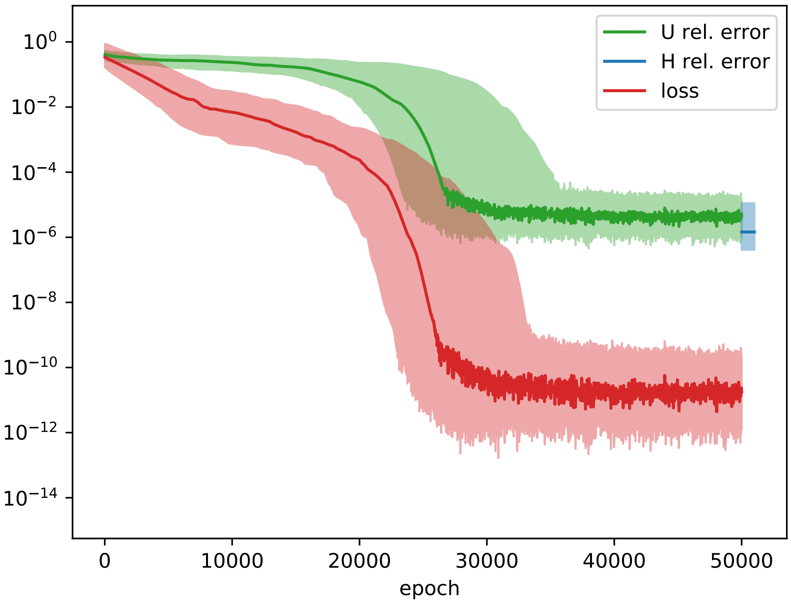


Figure 5.7: Single-qubit Hamiltonian reconstruction with the relaxation method Solid lines show the median, and lighter shades correspond to the 25% quantiles.

5.4.1 Two qubit case

Relaxing the unitary condition, however, comes with a drawback. Specifying an element of $\mathrm{SO}(4^n - 1)$ involves more free parameters than for a unitary matrix from $\mathrm{SU}(2^n)$ in the case where $n \geq 2$. This means that the optimization might find an orthogonal U

which reproduces the measurement data, but cannot be mapped to a $U \in \mathrm{SU}(2^n)$ via (5.19).

To illustrate how this problem can be avoided, let us focus on the case of two qubits. Every $U \in \mathrm{SU}(4)$ can be decomposed as [204, 205]

$$U = \big(u^{\mathrm{a}} \otimes u^{\mathrm{b}}\big) R \big(u^{\mathrm{c}} \otimes u^{\mathrm{d}}\big) \tag{5.24}$$

where R is the so-called "entanglement" gate

$$R = \mathrm{e}^{-\frac{i}{2}(\theta_1\, \sigma^1 \otimes \sigma^1 + \theta_2\, \sigma^2 \otimes \sigma^2 + \theta_3\, \sigma^3 \otimes \sigma^3)}, \quad \theta_1, \theta_2, \theta_3 \in \mathbb{R} \tag{5.25}$$

and $u^{\mathrm{a}}, u^{\mathrm{b}}, u^{\mathrm{c}}, u^{\mathrm{d}} \in \mathrm{SU}(2)$ are single qubit unitaries. These single qubit gates can be handled, as described before, by orthogonal rotation matrices $\mathsf{u}^{\mathrm{a}}, \mathsf{u}^{\mathrm{b}}, \mathsf{u}^{\mathrm{c}}, \mathsf{u}^{\mathrm{d}} \in \mathrm{SO}(3)$. In order to find the Bloch representation of R, we make use of Eq. (5.19) (for more details see Ref. [206, 207]). With this approach the parameters $\theta_1, \theta_2, \theta_3$ will only appear in the matrix as inputs to the trigonometric terms $\cos(\theta_j)$ and $\sin(\theta_j)$ for $j = 1, 2, 3$. All of these ingredients put together give us a Bloch picture representation of the decomposition in Eq. (5.24) in terms of to-be found orthogonal matrices.

To all of these orthogonal matrices we will apply the same relaxation conditions, in terms of unitarity and relation between the time steps, as described on the previous section. On top of that, we introduce one further relaxation to the trigonometric terms in R: instead of the entries $(\cos(\theta_j), \sin(\theta_j))$ we use (c_j, s_j), where $c_j, s_j \in \mathbb{R}$. We impose the condition $c_j^2 + s_j^2 = 1$ by introducing another penalty term in the overall cost function:

$$\mathcal{L}_\theta = \sum_{j=1}^{3} |c_j^2 + s_j^2 - 1|^2. \tag{5.26}$$

This is done for the exact same reason as the previous two relaxations: the optimization landscape appeared to be better behaved than when imposing stricter constraints.

Fig. 5.8 shows the results we obtain when trying to reconstruct an arbitrary two-qubit Hamiltonian using this method. We again note that the Hamiltonian relative error and the loss tend to decrease throughout the optimization. However, in this case the relaxation method does not provide such a dramatic improvement in terms of the confidence interval that the optimal solution can be found for any random Hamiltonian. Specifically, it is noteworthy that, despite the consistent reduction of loss to values below 10^{-5}, the error in U does not consistently exhibit a corresponding decrease. An explanation could be that U, and consequently H, are not uniquely determined.

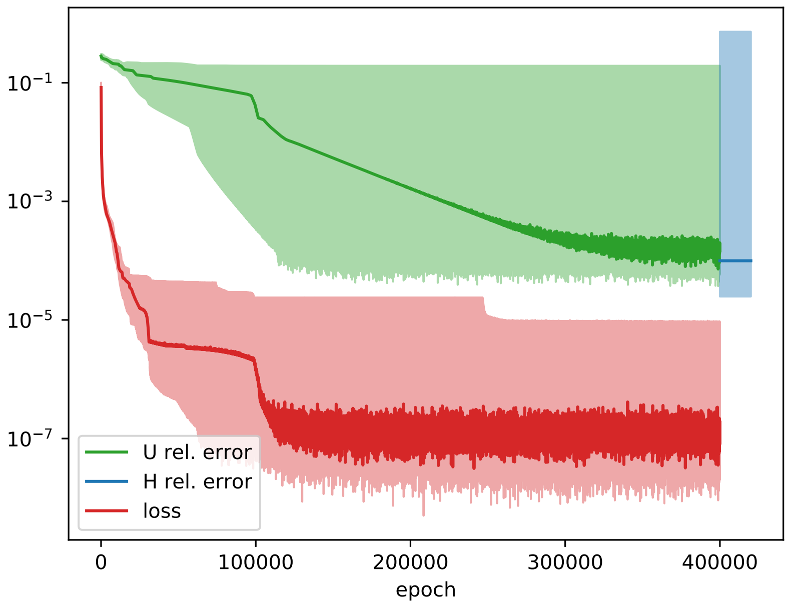


Figure 5.8: Two-qubit Hamiltonian reconstruction with the relaxation method

Solid lines show the median, and lighter shades correspond to the 25% quantiles.

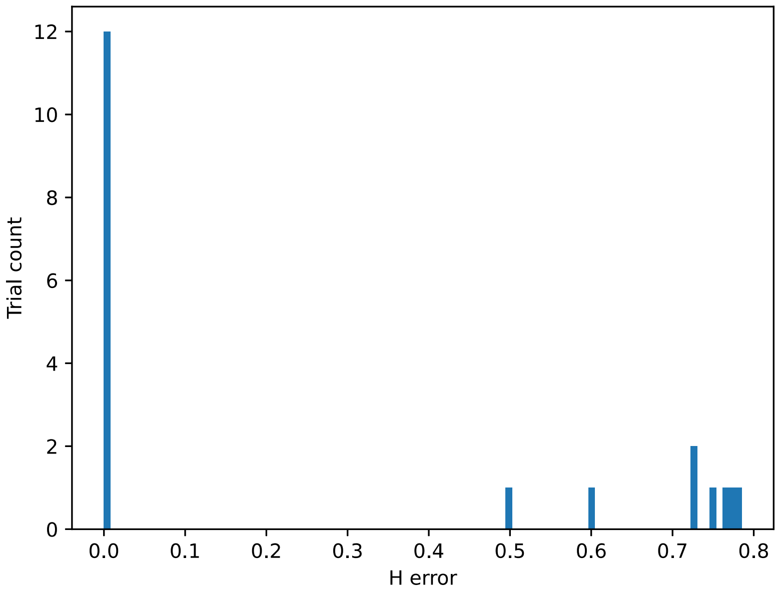


Figure 5.9: Histogram of optimization performance

The x-axis is the relative error in H at the end of the optimization. Despite the majority of the optimization trials converging to the optimal solution, there seem to be more than one minima in the optimization landscape.

In Fig.5.9, a histogram depicts the final error in the reconstructed Hamiltonian. While the majority of trials converge to the correct solution, there are several instances where the optimization becomes stuck at other extrema. We expect this issue to worsen as the Hilbert space of the underlying system increases, and for this reason we do not believe this method to be suitable for unstructured many-body systems.

5.5 Partial (subsystem) measurements

On a more positive note, the time-delayed measurements framework provides an compelling possibility: the characterization of a Hamiltonian based only on measurements restricted to a subsystem. This can be noted from the fact that our application of the Takens theorem imposes no restrictions on the observable characteristic. It is therefore possible for the measurement operator in Eq. (5.5) to be decomposable as a tensor product of an identity and smaller measurement operator.

As a simple toy-model to verify the applicability of the method, consider the following: we have a dynamical two-qubit system behaving under an unknown time evolution operator which we aim to characterize. One of the qubits is available for measurement, while the other is inaccessible to us. Since we will suffer from the ambiguities described in Sec. 5.3.1, and we have zero knowledge of the underlying Hamiltonian, we need to resolve the ambiguity by using at least two initial states and measurement bases. We assume that we are able to prepare the desired states with a high-fidelity, and that we can switch between measurement bases via the gate C. Fig. 5.10 illustrates this setting.

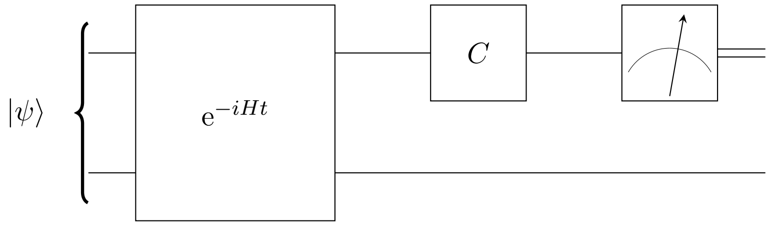


Figure 5.10: Toy-model for the time-delayed partial measurement scenario

For the unambiguous reconstruction of H, we assume that one can prepare with high-fidelity two different initial states and can select a measurement basis via the gate C.

For the numerical experiment shown in Figs. 5.11 and 5.12 below, we construct a random Hamiltonian by drawing standard normal-distributed coefficients of Pauli strings, which in sum form the Hamiltonian, H. This general Ansatz for the (traceless) H has 15 real parameters. So according to Takens theorem 31 time-points would be enough for the reconstruction. However, we heuristically found that 12 time-delayed measurements for each measurement basis are necessary for the optimization to reliably converge.

The measurements are performed at the 12 first time points generated by selecting $\Delta t = \frac{1}{5}$ and $\gamma = 1.15$. The specific observables are the X, Y and Z gates applied solely on one of the two qubits. Fig. 5.11 shows exemplary measurement time trajectories.

The loss function is then mean squared error between the model prediction for the measurement averages and the ground-truth values. Note that the goal of this experiment is simply to demonstrate the feasibility of this method for a scenario with an inaccessible subsystem. For this reason, we keep the model and optimizer simple, and, to avoid getting trapped in the local minima observed in the previous section, we use the best out of 10 attempts (in terms of the loss function) with different random starting points.

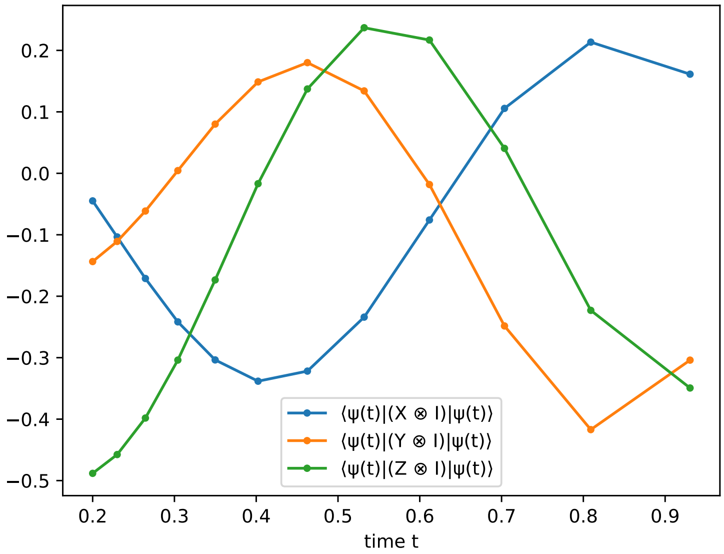


Figure 5.11: Exemplary measurement time trajectories

The measurements correspond to the expectation values of the operators $\sigma^x \otimes I$, $\sigma^y \otimes I$ and $\sigma^z \otimes I$. Two random initial states and their respective trajectories are used as input to the reconstruction algorithm.

Fig. 5.12 shows the reconstruction error and loss function (median and 25% quantiles based on 20 random realizations of the overall set-up). One observes that a reliable and precise reconstruction of the Hamiltonian (even up to numerical rounding errors) is possible in principle, when neglecting inaccuracies of the measurement process.

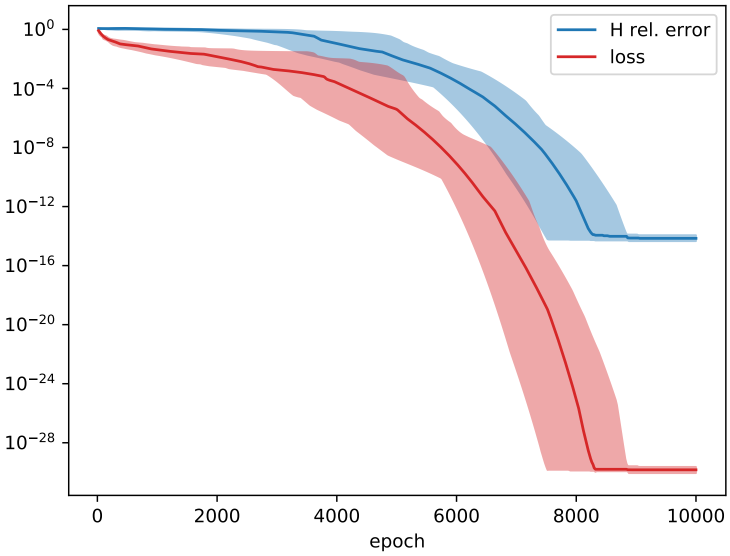


Figure 5.12: Reconstruction error for a partially inaccessible system

The evolution of the relative error and loss throughout the optimization plotted here based on 20 random realizations. The solid lines are the median, whereas the shaded areas correspond to the 25% quantiles.

5.6 Lattice systems

Until now, our focus has consistently been on random, arbitrary Hamiltonians. Nevertheless, many experimental set-ups exhibit some form of underlying structure. For instance, in the field of quantum computing, qubits are frequently organized in regular grids [208–212] or 1D chains [213]. Hence, we shall now turn our attention to the task of characterizing a structured Hamiltonian.

The application of Takens theorem here appears to be particularly fitting for two main reasons. First of all, systems with a fixed underlying structure present less free parameters compared to arbitrary systems of the same size. The embedding described in Theorem 1 is dependent on the free parameters of H rather than the dimension of the Hilbert space. In certain situations, as illustrated in the scenarios presented next, this dependency could result in an exponential improvement in terms of the required resources compared to conventional QPT. The second advantage is related to the ambiguity problem presented in Sec. 5.3.1. Although we are not guaranteed to completely resolve the issue, imposing assumptions on the physical nature of the Hamiltonian will certainly help in the optimizer convergence to the ground-truth.

In order to put the method to the test, we first consider the case of an Ising Hamiltonian (see Sec. 2.2) with *uniform* parameters (not depending on the lattice site), i.e.,

$$H = -\sum_{\langle j,\ell \rangle} J\, \sigma_j^z \sigma_\ell^z - \sum_j \vec{h} \cdot \vec{\sigma}_j \tag{5.27}$$

with $J \in \mathbb{R}$ and $\vec{h} \in \mathbb{R}^3$. Thus the task consists of reconstructing 4 real numbers, irrespective of system size.

For the following numerical experiment we consider a 3×4 lattice with periodic boundary conditions, and the initial state is a single wavefunction with complex random entries (independently normally distributed). The parameters are set to $J = 1$ and $\vec{h} = (0.5, -0.8, 1.1)$. We compute the time-evolved quantum state $\psi(t)$ via the KrylovKit Julia package [214], and use the squared entries of $\psi(t)$ as reference Born measurement averages.

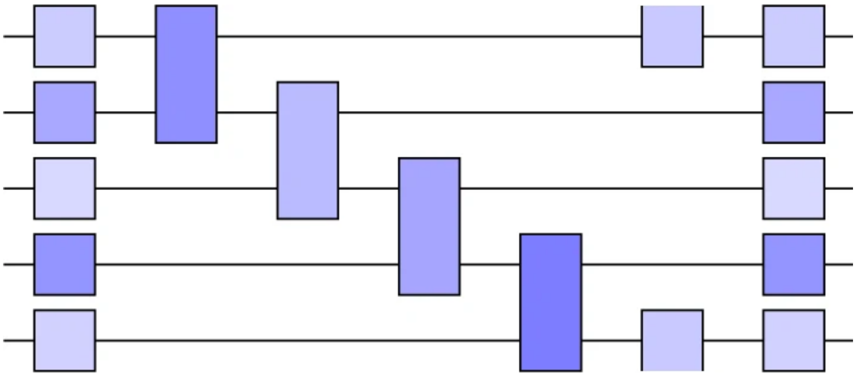


Figure 5.13: Variational circuit used as a model for a lattice system

Single time step of the time evolution approximated via a Strang splitting into interaction and local field terms. For simplicity, it is illustrated here for a one-dimensional lattice with $n = 5$ sites.

As a variational model for our optimization, we make use of the regular structure of quantum circuits and construct a circuit Ansatz which benefits from the second order accuracy of Strang splitting [215, 216]. A single time step is therefore approximated via the circuit shown in Fig. 5.13. Note that all single- and two-qubit gates are identical, as they share their parameters $\tilde{J}$ and $\tilde{\vec{h}}$. In particular, the single-qubit gates are the rotations

$$\exp\left(-i\Delta t \tilde{\vec{h}} \cdot \vec{\sigma}_j / 2\right), \tag{5.28}$$

where σ_j is the operator σ being applied to qubit j, and the two qubit gates correspond to the interaction

$$\exp\left(-i\Delta t J \sigma_j^z \sigma_{j+1}^z\right). \tag{5.29}$$

For simplicity, we use three uniform time points Δt, $2\Delta t$, $3\Delta t$ with $\Delta t = 0.2$ (instead of time points $t_q = \gamma^q \Delta t$), such that the quantum circuit for realizing a single time step can be reused. This, naturally, implies that we are not strictly adhering to Theorem 1. Nevertheless, in this particular scenario, we found that the requirements of Takens' theorem are more stringent than what is necessary in practice.

As a cost function for the optimization, we quantify the 'distance' between the exact Born measurement probabilities, $p(t) = |\psi(t)|^2$, and the probabilities $\tilde{p}(t)$ resulting from the trotterized circuit Ansatz, via the Kullback-Leibler divergence:

$$D_{\mathrm{KL}}\left(p(t) \parallel \tilde{p}(t)\right) = \sum_j p_j(t) \log \frac{p_j(t)}{\tilde{p}_j(t)}. \tag{5.30}$$

Fig. 5.14 shows the relative errors of the Hamiltonian parameters throughout the optimization, as well as the evolution of the loss function from Eq. (5.30). We have used the RMSProp optimizer [67] with a learning rate of 0.005. As, usual the solid lines correspond to the medians over 100 trials for random initial states, whereas the lighter shaded areas correspond to the 25% quantiles.

The dashed horizontal line represents the loss function evaluated at the ground truth values of J and $\vec{h}$, showing a non-zero value due to the Strang splitting approximation of a time step. Notably, the optimization achieves an even smaller loss value starting around training epoch 70. However, we interpret this as an artifact of the Strang splitting approximation –observe that around this epoch, the relative error of $\vec{h}$ slightly increases again. From this plot we can conclude that the reachable relative error is around 0.02, and higher accuracy would likely require a smaller time step to reduce the Strang splitting error.

Next, we consider an extension to the Hamiltonian in Eq. (5.27) by introducing disorder, i.e. both the external field and the interaction terms depend on the lattice site of application:

$$H = -\sum_{\langle j,\ell \rangle} J_{j,\ell}\, \sigma_j^z \sigma_\ell^z - \sum_j h_j \cdot \sigma_j^x \tag{5.31}$$

with i.i.d. random coefficients $J_{j,\ell}$ and h_j. For the numerical experiments, $J_{j,\ell}$ is uniformly distributed in the interval $[0.8, 1.2]$, and h_j is obtained from a normal distribution, $0.5\mathcal{N}(0,1)$. As the variational Ansatz we continue to use the circuit structure shown

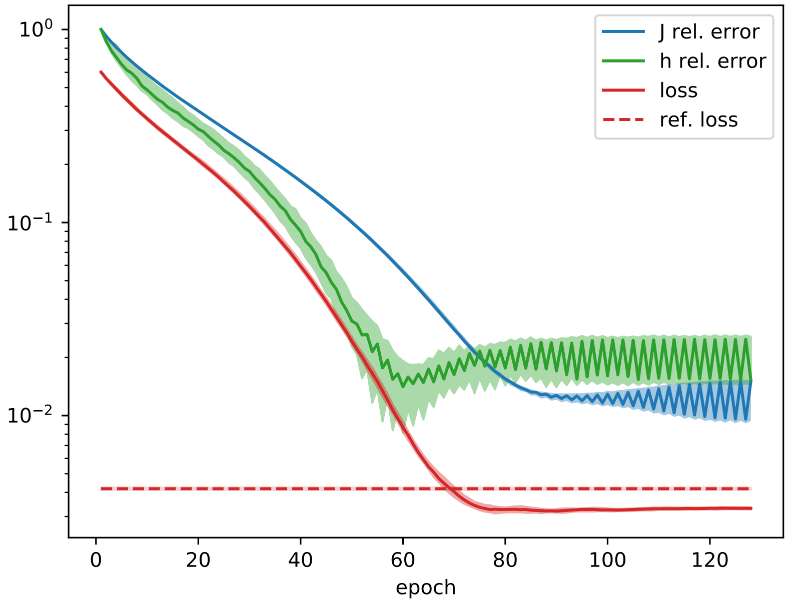


Figure 5.14: Reconstruction error for a uniform lattice system

The solid lines correspond to the median and the shaded areas to the 25% quantiles. The reference loss was obtained by inputting the ground truth parameters into the cost function, thereby capturing the error arising from the time-step approximation of the Ansatz.

in Fig. 5.13, but note that this time each gate is different and therefore the number of parameters grows linearly with system size.

As for the uniform lattice experiment, we use the three uniform time points Δt, $2\Delta t$, $3\Delta t$ with $\Delta t = 0.2$, and Eq. (5.30) as our cost function. We evaluate 100 realizations of the experiment, with a set of coefficients and random initial state drawn independently for each trial. The median and 25% quantiles of the errors are shown in Fig. 5.15. Note that we are plotting the maximum over the relative errors of $J_{j,\ell}$ for all j, ℓ that we observed throughout the random trials, and likewise for h_j. Once again, the reference loss function (the dashed line) is the KL divergence evaluated at the ground truth coefficients. This time the ground truth loss fluctuates due to the different random coefficients at each trial.

Similarly to the previous case, it is observed that the loss attains, and in fact exceeds, the reference ground truth value. This implies that the errors observed for the optimized parameters are optimal for the chosen time-step size. Achieving higher accuracies would require improvements in the time-step approximation used for the Ansatz.

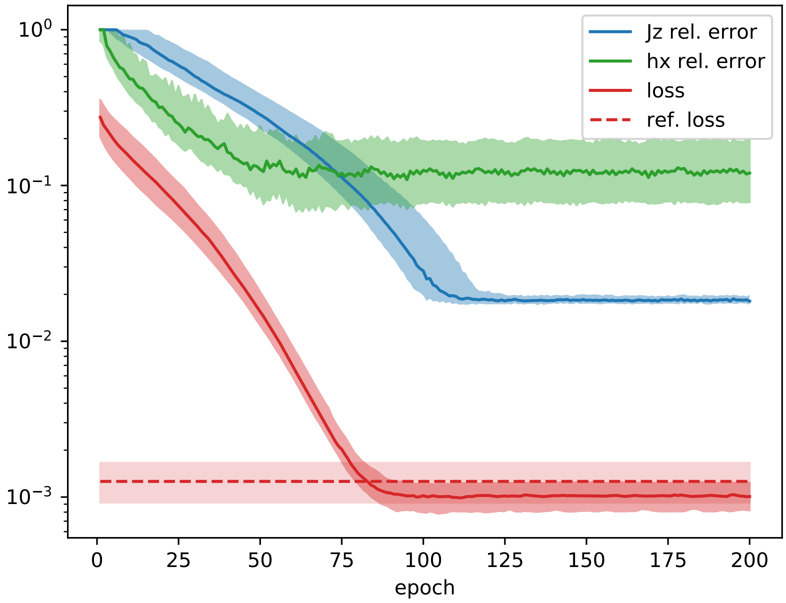


Figure 5.15: Reconstruction error for a disordered lattice system

The solid lines correspond to the median and the shaded areas to the 25% quantiles. The ground truth parameters where sampled randomly for each trial. This is also the reason why the baseline loss presents fluctuations.

6 Conclusion & Outlook

It is undeniable that the classical simulation and characterization of quantum systems is a very challenging task. It is also true that breakthroughs in quantum technologies could end up making the classical simulation of quantum systems redundant. However, in the current paradigm of noisy intermediate-scale quantum computers, we must aim to push the limits of the classical methods as a way to complement and benchmark the current technologies.

Throughout this thesis, we have explored different topics spanning quantum physics and machine learning. We have established the relevant theoretical background for both fields, describing the key concepts used in the proposed methods. On the side of quantum mechanics we have defined entanglement and entanglement entropy, and have considered the challenges posed by the exponential growth of quantum systems. In contrast, on the machine learning front we have explored the role of gradient descent in computational optimization, and we have described how neural networks are constructed and optimized. We have then discussed how all of this can be applied for the numerical approximation of quantum systems.

In Chapter 4 we have seen how NNQS can be used as variational Ansatze for quantum systems. Due to the extreme flexibility of their architectures, they hold great potential to be able to capture the entanglement scaling of arbitrary quantum states. We have demonstrated that it is possible, in principle, to use them for the simulation of quantum real-time evolution. Our proposed optimization method based on gradient descent avoids the numerical instabilities associated to SR, which helps mitigate the sudden decreases in accuracy observed previously.

Although NNQS are not, as of right now, at the level of tensor networks in terms of efficiency, it is important to keep in mind that they are still in the early stages of theoretical study. A better understanding of the relevant physical region of the Hilbert space, as well as of the geometrical manifolds represented by the networks, could guide the designs towards for efficient choices. On top of that, there is a lot to be learned from previous breakthroughs in the fields of tensor networks and machine learning.

For instance, consider the renormalization techniques utilized in tensor networks, currently missing in the optimization of NNQS. These techniques are based on truncating the weights matrices and are one of the most important factor in obtaining efficient and accurate representations of the underlying states. These techniques are not directly applicable to neural networks due to the introduction of non-linear maps, but it is still worth considering if something can be learned from them. Could connections between neurons be truncated in a systematic way to achieve similar results? Could the optimization benefit from a similar approximation based on SVD applied to the weights' update matrix? These questions remain open and warrant further exploration. Moreover, while in this work we have optimized the network parameters without any constraints, introducing symmetries or a specific structure –particularly to the CNN filters– has the potential to improve the observed accuracies.

On the machine learning front, on the other hand, two techniques have been fundamental in the processing of high dimensional and time series data: respectively, transfer learning and transformers [217]. Transfer learning refers to the reutilization of previously trained weights for related or extended application problems. In the context of lattice quantum systems, this technique could prove very useful as a means to reduce the computational overhead. Specifically, CNN filters could be reused to represent models whose only difference is the system size. In contrast, transformers are a different type of network model. They have become the go-to model for machine learning task involving time series, replacing and surpassing the performance of previously used CNN and RNN models. However, they were only introduced as quantum Ansatze very recently [218] and their full potential is still to be realized.

With respect to the characterization of quantum processes, in this thesis we began by describing in detail the standard QPT protocol and, in Chapter 5 we proposed an optimization based method with the aim of reducing the number of required measurement operators. Our work demonstrates the feasibility of using measurements at different time points to reconstruct the time evolution operator. The approach we presented has two main benefits: Firstly, it requires only a few initial state and reduces the number of different measurement bases needed. Although this does not translate to an exponential decrease in the number of measurements, it does reduce the overall experimental cost, particularly when maintaining the time evolution of the system is easier than implementing a wide variety of projection measurements. Secondly, the method presents great flexibility in terms of the chosen measurement operators, which leads to interesting scenarios, such as being able to reconstruct the full Hamiltonian of system by measuring only part of it.

A natural extension to the proposed QPT method is to the characterization of non-unitary processes, a.k.a dissipation and noise processes governed by a Lindblad equation. This would introduce the additional challenge of a limited time window for extracting information and a larger number of free parameters. However, the results from Sec. 5.5 point towards an attractive alternative: mapping the noisy processed to a unitary evolution with an unobserved environment.

However, it is important to keep in mind the limitations of the method. For systems of more than one qubit, the problem of multiple Hamiltonians describing the same measurement trajectories has not been resolved. Although the relaxation method in Sec. 5.4 works as a tool to smoothen the optimization landscape, it is still not possible to guarantee the convergence of the optimization to the ground truth solution. Thus, a more careful analysis of the symmetries in the Hilbert space with respect to the measurement operators is needed. In this direction, the combination of the proposed method with the idea of MUB and SIC-POVMs could be helpful.

On top of that, a more careful analysis of the protocol's sensitivity with respect to inaccuracies and noise in the measurement data is needed. A modified version of Takens theorem still applies in this situation, but it is not clear how well the numerical optimization would behave. Perturbation or gradient analysis of the algorithm could help clarify this for a general scenario.

Bibliography

[1] R. P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6-7):467–488, jun 1982. `doi:10.1007/bf02650179`.

[2] H.-L. Huang, D. Wu, D. Fan, and X. Zhu. Superconducting quantum computing: a review. *Science China Information Sciences*, 63(8), July 2020. `doi:10.1007/s11432-020-2881-9`.

[3] C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage. Trapped-ion quantum computing: Progress and challenges. *Applied Physics Reviews*, 6(2), May 2019. `doi:10.1063/1.5088164`.

[4] L. Henriet, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Reymond, and C. Jurczak. Quantum computing with neutral atoms. *Quantum*, 4:327, September 2020. `doi:10.22331/q-2020-09-21-327`.

[5] F. Arute et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, October 2019. `doi:10.1038/s41586-019-1666-5`.

[6] D. Castelvecchi. Ibm releases first-ever 1,000-qubit quantum chip. *Nature*, December 2023. `doi:10.1038/d41586-023-03854-1`.

[7] R. Orús. Tensor networks for complex quantum systems. *Nature Reviews Physics*, 1(9):538–550, August 2019. `doi:10.1038/s42254-019-0086-7`.

[8] C. Guo, Y. Zhao, and H.-L. Huang. Verifying random quantum circuits with arbitrary geometry using tensor network states algorithm. *Physical Review Letters*, 126(7):070502, February 2021. `doi:10.1103/physrevlett.126.070502`.

[9] J. Owens, M. Houston, D. Luebke, S. Green, J. Stone, and J. Phillips. Gpu computing. *Proceedings of the IEEE*, 96(5):879–899, May 2008. `doi:10.1109/jproc.2008.917757`.

[10] P. P. Shinde and S. Shah. A review of machine learning and deep learning applications. In *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. IEEE, August 2018. doi:10.1109/iccubea.2018.8697857.

[11] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, May 2021. doi:10.1038/s42254-021-00314-5.

[12] M. Garofalo, A. Botta, and G. Ventre. Astrophysics and big data: Challenges, methods, and tools. *Proceedings of the International Astronomical Union*, 12(S325):345–348, October 2016. doi:10.1017/s1743921316012813.

[13] G. Karagiorgi, G. Kasieczka, S. Kravitz, B. Nachman, and D. Shih. Machine learning in the search for new fundamental physics. *Nature Reviews Physics*, 4(6):399–412, May 2022. doi:10.1038/s42254-022-00455-1.

[14] D. R. Vivas, J. Madroñero, V. Bucheli, L. O. Gómez, and J. H. Reina. Neural-network quantum states: A systematic review, 2022. doi:10.48550/ARXIV.2204.12966.

[15] L. Ljung. Perspectives on system identification. *Annual Reviews in Control*, 34(1):1–12, April 2010. doi:10.1016/j.arcontrol.2009.12.001.

[16] W. M. Itano, D. J. Heinzen, J. J. Bollinger, and D. J. Wineland. Quantum zeno effect. *Physical Review A*, 41(5):2295–2300, March 1990. doi:10.1103/physreva.41.2295.

[17] A. Chiuso and G. Pillonetto. System identification: A machine learning perspective. *Annual Review of Control, Robotics, and Autonomous Systems*, 2(1):281–304, May 2019. doi:10.1146/annurev-control-053018-023744.

[18] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition.* Cambridge University Press, June 2012. doi:10.1017/cbo9780511976667.

[19] P. Kaye, R. Laflamme, and M. Mosca. *An introduction to quantum computing.* Oxford University Press, Oxford [u.a.], repr edition, 2010.

[20] J. Eisert, M. Cramer, and M. B. Plenio. Colloquium: Area laws for the entanglement entropy. *Reviews of Modern Physics*, 82(1):277–306, February 2010. doi:10.1103/revmodphys.82.277.

[21] G. B. Mbeng, A. Russomanno, and G. E. Santoro. The quantum ising chain for beginners, 2020. doi:10.48550/ARXIV.2009.09208.

[22] G. Mauro D'Ariano, M. G. Paris, and M. F. Sacchi. *Quantum Tomography*, volume 128, pages 205–308. Elsevier, 2003. doi:10.1016/s1076-5670(03)80065-4.

[23] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Physical Review*, 47(10):777–780, May 1935. doi:10.1103/physrev.47.777.

[24] J. S. Bell. On the einstein podolsky rosen paradox. *Physics Physique Fizika*, 1(3):195–200, November 1964. doi:10.1103/physicsphysiquefizika.1.195.

[25] S. J. Freedman and J. F. Clauser. Experimental test of local hidden-variable theories. *Physical Review Letters*, 28(14):938–941, April 1972. doi:10.1103/physrevlett.28.938.

[26] W. Tittel, J. Brendel, B. Gisin, T. Herzog, H. Zbinden, and N. Gisin. Experimental demonstration of quantum correlations over more than 10 km. *Physical Review A*, 57(5):3229–3232, May 1998. doi:10.1103/physreva.57.3229.

[27] G. Weihs, T. Jennewein, C. Simon, H. Weinfurter, and A. Zeilinger. Violation of bell's inequality under strict einstein locality conditions. *Physical Review Letters*, 81(23):5039–5043, December 1998. doi:10.1103/physrevlett.81.5039.

[28] S. Gröblacher, T. Paterek, R. Kaltenbaek, C. Brukner, M. Zukowski, M. Aspelmeyer, and A. Zeilinger. An experimental test of non-local realism. *Nature*, 446(7138):871–875, April 2007. doi:10.1038/nature05677.

[29] D. N. Page. Average entropy of a subsystem. *Physical Review Letters*, 71(9):1291–1294, August 1993. doi:10.1103/physrevlett.71.1291.

[30] S. K. Foong and S. Kanno. Proof of page's conjecture on the average entropy of a subsystem. *Physical Review Letters*, 72(8):1148–1151, February 1994. doi:10.1103/physrevlett.72.1148.

[31] L. Bombelli, R. K. Koul, J. Lee, and R. D. Sorkin. Quantum source of entropy for black holes. *Physical Review D*, 34(2):373–383, July 1986. doi:10.1103/physrevd.34.373.

[32] M. B. Hastings. An area law for one-dimensional quantum systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(08):P08024–P08024, August 2007. `doi:10.1088/1742-5468/2007/08/p08024`.

[33] M. B. Plenio, J. Eisert, J. Dreißig, and M. Cramer. Entropy, entanglement, and area: Analytical results for harmonic lattice systems. *Physical Review Letters*, 94(6):060503, February 2005. `doi:10.1103/physrevlett.94.060503`.

[34] M. Cramer and J. Eisert. Correlations, spectral gap and entanglement in harmonic quantum systems on generic lattices. *New Journal of Physics*, 8(5):71–71, May 2006. `doi:10.1088/1367-2630/8/5/071`.

[35] P. Calabrese and J. Cardy. Evolution of entanglement entropy in one-dimensional systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(04):P04010, apr 2005. `doi:10.1088/1742-5468/2005/04/p04010`.

[36] J. Eisert and T. J. Osborne. General entanglement scaling laws from time evolution. *Physical Review Letters*, 97(15):150404, October 2006. `doi:10.1103/physrevlett.97.150404`.

[37] S. Bravyi, M. B. Hastings, and F. Verstraete. Lieb-robinson bounds and the generation of correlations and topological quantum order. *Physical Review Letters*, 97(5):050401, July 2006. `doi:10.1103/physrevlett.97.050401`.

[38] M. Fagotti and P. Calabrese. Entanglement entropy of two disjoint blocks inxychains. *Journal of Statistical Mechanics: Theory and Experiment*, 2010(04):P04016, April 2010. `doi:10.1088/1742-5468/2010/04/p04016`.

[39] L. Cincio, J. Dziarmaga, M. M. Rams, and W. H. Zurek. Entropy of entanglement and correlations induced by a quench: Dynamics of a quantum phase transition in the quantum ising model. *Physical Review A*, 75(5):052321, May 2007. `doi:10.1103/physreva.75.052321`.

[40] N. Schuch, M. M. Wolf, K. G. H. Vollbrecht, and J. I. Cirac. On entropy growth and the hardness of simulating time evolution. *New Journal of Physics*, 10(3):033032, March 2008. `doi:10.1088/1367-2630/10/3/033032`.

[41] E. Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, 31(1):253–258, February 1925. `doi:10.1007/bf02980577`.

[42] S. Suzuki, J. ichi Inoue, and B. K. Chakrabarti. *Quantum Ising Phases and Transitions in Transverse Ising Models*. Springer Berlin Heidelberg, 2013. `doi:10.1007/978-3-642-33039-1`.

[43] H. W. J. Blöte and Y. Deng. Cluster monte carlo simulation of the transverse ising model. *Physical Review E*, 66(6):066110, dec 2002. `doi:10.1103/physreve.66.066110`.

[44] M. Schmitt, M. M. Rams, J. Dziarmaga, M. Heyl, and W. H. Zurek. Quantum phase transition dynamics in the two-dimensional transverse-field ising model. *Science Advances*, 8(37), September 2022. `doi:10.1126/sciadv.abl6850`.

[45] F. Y. Wu. Critical behavior of hydrogen-bonded ferroelectrics. *Physical Review Letters*, 24(26):1476–1478, June 1970. `doi:10.1103/physrevlett.24.1476`.

[46] R. Blinc, B. Žekš, J. F. Sampaio, A. S. T. Pires, and F. C. S. Barreto. Ising model in a transverse tunneling field and proton-lattice interaction in h-bonded ferroelectrics. *Physical Review B*, 20(5):1991–2001, September 1979. `doi:10.1103/physrevb.20.1991`.

[47] W. P. Wolf. The ising model and real magnetic materials. *Brazilian Journal of Physics*, 30(4):794–810, December 2000. `doi:10.1590/s0103-97332000000400030`.

[48] V. Borish, O. Marković, J. Hines, S. Rajagopal, and M. Schleier-Smith. Transverse-field ising dynamics in a rydberg-dressed atomic gas. *Physical Review Letters*, 124(6):063601, February 2020. `doi:10.1103/physrevlett.124.063601`.

[49] A. W. Carr and M. Saffman. Preparation of entangled and antiferromagnetic states by dissipative rydberg pumping. *Physical Review Letters*, 111(3):033607, July 2013. `doi:10.1103/physrevlett.111.033607`.

[50] T. Kadowaki and H. Nishimori. Quantum annealing in the transverse ising model. *Physical Review E*, 58(5):5355–5363, November 1998. `doi:10.1103/physreve.58.5355`.

[51] G. E. Santoro, R. Martonak, E. Tosatti, and R. Car. Theory of quantum annealing of an ising spin glass. *Science*, 295(5564):2427–2430, March 2002. `doi:10.1126/science.1068774`.

[52] J. I. Cirac and P. Zoller. A scalable quantum computer with ions in an array of microtraps. *Nature*, 404(6778):579–581, April 2000. `doi:10.1038/35007021`.

[53] W. Wu, T. Zhang, and P.-X. Chen. Quantum computing and simulation with trapped ions: On the path to the future. *Fundamental Research*, 1(2):213–216, March 2021. `doi:10.1016/j.fmre.2020.12.004`.

[54] A. Browaeys and T. Lahaye. Many-body physics with individually controlled rydberg atoms. *Nature Physics*, 16(2):132–142, January 2020. `doi:10.1038/s41567-019-0733-z`.

[55] P. Schauss. Quantum simulation of transverse ising models with rydberg atoms. *Quantum Science and Technology*, 3(2):023001, January 2018. `doi:10.1088/2058-9565/aa9c59`.

[56] D. S. Lyubshin, A. U. Sharafutdinov, and I. S. Burmistrov. Statistics of spin fluctuations in quantum dots with ising exchange. *Physical Review B*, 89(20):201304, May 2014. `doi:10.1103/physrevb.89.201304`.

[57] B. Nissan-Cohen, Y. Gefen, M. N. Kiselev, and I. V. Lerner. Interplay of charge and spin in quantum dots: The ising case. *Physical Review B*, 84(7):075307, August 2011. `doi:10.1103/physrevb.84.075307`.

[58] N. Bingham, S. Rooke, J. Park, A. Simon, W. Zhu, X. Zhang, J. Batley, J. Watts, C. Leighton, K. Dahmen, and P. Schiffer. Experimental realization of the 1d random field ising model. *Physical Review Letters*, 127(20):207203, November 2021. `doi:10.1103/physrevlett.127.207203`.

[59] J. L. Dunn, C. Stahl, A. J. Macdonald, K. Liu, Y. Reshitnyk, W. Sim, and R. W. Hill. Testing the transverse field ising model in lihof4 using capacitive dilatometry. *Physical Review B*, 86(9):094428, September 2012. `doi:10.1103/physrevb.86.094428`.

[60] R. Islam, E. Edwards, K. Kim, S. Korenblit, C. Noh, H. Carmichael, G.-D. Lin, L.-M. Duan, C.-C. Joseph Wang, J. Freericks, and C. Monroe. Onset of a quantum phase transition with a trapped ion quantum simulator. *Nature Communications*, 2(1), July 2011. `doi:10.1038/ncomms1374`.

[61] P. Scholl et al. Quantum simulation of 2d antiferromagnets with hundreds of rydberg atoms. *Nature*, 595(7866):233–238, July 2021. `doi:10.1038/s41586-021-03585-1`.

[62] A. Kinross, M. Fu, T. Munsie, H. Dabkowska, G. Luke, S. Sachdev, and T. Imai. Evolution of quantum fluctuations near the quantum critical point of the transverse

field ising chain system conb2o6. *Physical Review X*, 4(3):031008, July 2014. doi:10.1103/physrevx.4.031008.

[63] J. M. Renes, R. Blume-Kohout, A. J. Scott, and C. M. Caves. Symmetric informationally complete quantum measurements. *Journal of Mathematical Physics*, 45(6):2171–2180, May 2004. doi:10.1063/1.1737053.

[64] W. K. Wootters and B. D. Fields. Optimal state-determination by mutually unbiased measurements. *Annals of Physics*, 191(2):363–381, May 1989. doi:10.1016/0003-4916(89)90322-9.

[65] M. A. Nielsen. *Neural networks and deep learning*. Determination press San Francisco, CA, USA, 2015.

[66] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *3rd International Conference for Learning Representations, San Diego*, 2015. arXiv:1412.6980v9.

[67] G. Hinton, N. Srivastava, and K. Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012.

[68] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, December 1943. doi:10.1007/bf02478259.

[69] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958. doi:10.1037/h0042519.

[70] D. Silver et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016. doi:10.1038/nature16961.

[71] OpenAI. Gpt-4 technical report, 2023. doi:10.48550/ARXIV.2303.08774.

[72] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, dec 1989. doi:10.1007/bf02551274.

[73] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. doi:10.1038/nature14539.

[74] O. S. Kayhan and J. C. v. Gemert. On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[75] B. S. Rem, N. Käming, M. Tarnowski, L. Asteria, N. Fläschner, C. Becker, K. Sengstock, and C. Weitenberg. Identifying quantum phase transitions using artificial neural networks on experimental data. *Nature Physics*, 15(9):917–920, July 2019. `doi:10.1038/s41567-019-0554-0`.

[76] A. Bohrdt, C. S. Chiu, G. Ji, M. Xu, D. Greif, M. Greiner, E. Demler, F. Grusdt, and M. Knap. Classifying snapshots of the doped hubbard model with machine learning. *Nature Physics*, 15(9):921–924, July 2019. `doi:10.1038/s41567-019-0565-x`.

[77] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1986. `doi:10.1038/323533a0`.

[78] I. López Gutiérrez and C. B. Mendl. Real time evolution with neural-network quantum states. *Quantum*, 6:627, January 2022. `doi:10.22331/q-2022-01-20-627`.

[79] G. Carleo and M. Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017. URL: `https://www.science.org/doi/abs/10.1126/science.aag2302`, `doi:10.1126/science.aag2302`.

[80] A. Borin and D. A. Abanin. Approximating power of machine-learning ansatz for quantum many-body states. *Physical Review B*, 101(19):195141, may 2020. `doi:10.1103/physrevb.101.195141`.

[81] D.-L. Deng, X. Li, and S. D. Sarma. Quantum entanglement in neural network states. *Physical Review X*, 7(2):021021, may 2017. `doi:10.1103/physrevx.7.021021`.

[82] Y. Nomura, A. S. Darmawan, Y. Yamaji, and M. Imada. Restricted boltzmann machine learning for solving strongly correlated quantum systems. *Physical Review B*, 96(20):205152, nov 2017. `doi:10.1103/physrevb.96.205152`.

[83] L. Pastori, R. Kaubruegger, and J. C. Budich. Generalized transfer matrix states from artificial neural networks. *Physical Review B*, 99(16):165123, apr 2019. `doi:10.1103/physrevb.99.165123`.

[84] S. R. Clark. Unifying neural-network quantum states and correlator product states via tensor networks. *Journal of Physics A: Mathematical and Theoretical*, 51(13):135301, feb 2018. `doi:10.1088/1751-8121/aaaaf2`.

[85] I. Glasser, N. Pancotti, M. August, I. D. Rodriguez, and J. I. Cirac. Neural-network quantum states, string-bond states, and chiral topological states. *Physical Review X*, 8(1):011006, jan 2018. `doi:10.1103/physrevx.8.011006`.

[86] X. Gao and L.-M. Duan. Efficient representation of quantum many-body states with deep neural networks. *Nature Communications*, 8(1), sep 2017. `doi:10.1038/s41467-017-00705-2`.

[87] R. Kaubruegger, L. Pastori, and J. C. Budich. Chiral topological phases from artificial neural networks. *Physical Review B*, 97(19):195136, may 2018. `doi:10.1103/physrevb.97.195136`.

[88] D. Pfau, J. S. Spencer, A. G. D. G. Matthews, and W. M. C. Foulkes. Ab initio solution of the many-electron schrödinger equation with deep neural networks. *Physical Review Research*, 2(3):033429, sep 2020. `doi:10.1103/physrevresearch.2.033429`.

[89] M. Schmitt and M. Heyl. Quantum many-body dynamics in two dimensions with artificial neural networks. *Physical Review Letters*, 125(10):100503, sep 2020. `doi:10.1103/physrevlett.125.100503`.

[90] D. Luo, Z. Chen, K. Hu, Z. Zhao, V. M. Hur, and B. K. Clark. Gauge invariant and anyonic symmetric autoregressive neural networks for quantum lattice models. 2021. `doi:10.48550/ARXIV.2101.07243`.

[91] D. Luo and B. K. Clark. Backflow transformations via neural networks for quantum many-body wave functions. *Physical Review Letters*, 122(22):226401, jun 2019. `doi:10.1103/physrevlett.122.226401`.

[92] Y. Levine, O. Sharir, N. Cohen, and A. Shashua. Quantum entanglement in deep learning architectures. *Physical Review Letters*, 122(6):065301, feb 2019. `doi:10.1103/physrevlett.122.065301`.

[93] M. Hibat-Allah, M. Ganahl, L. E. Hayward, R. G. Melko, and J. Carrasquilla. Recurrent neural network wave functions. *Physical Review Research*, 2(2):023358, jun 2020. `doi:10.1103/physrevresearch.2.023358`.

[94] M. Hibat-Allah, R. G. Melko, and J. Carrasquilla. Supplementing recurrent neural network wave functions with symmetry and annealing to improve accuracy. *Machine Learning and the Physical Sciences, NeurIPS*, 2021. `doi:10.48550/ARXIV.2207.14314`.

[95] D. Wu, R. Rossi, F. Vicentini, and G. Carleo. From tensor-network quantum states to tensorial recurrent neural networks. *Physical Review Research*, 5(3):l032001, jul 2023. `doi:10.1103/physrevresearch.5.l032001`.

[96] S. Czischek, M. Gärttner, and T. Gasenzer. Quenches near ising quantum criticality as a challenge for artificial neural networks. *Physical Review B*, 98(2):024311, jul 2018. `doi:10.1103/physrevb.98.024311`.

[97] K. Choo, T. Neupert, and G. Carleo. Two-dimensional frustrated j1-j2 model studied with neural network quantum states. *Physical Review B*, 100(12):125124, sep 2019. `doi:10.1103/physrevb.100.125124`.

[98] T. Westerhout, N. Astrakhantsev, K. S. Tikhonov, M. I. Katsnelson, and A. A. Bagrov. Generalization properties of neural network approximations to frustrated magnet ground states. *Nature Communications*, 11(1), mar 2020. `doi:10.1038/s41467-020-15402-w`.

[99] A. Szabó and C. Castelnovo. Neural network wave functions and the sign problem. *Physical Review Research*, 2(3):033075, jul 2020. `doi:10.1103/physrevresearch.2.033075`.

[100] J. Thibaut, T. Roscilde, and F. Mezzacapo. Long-range entangled-plaquette states for critical and frustrated quantum systems on a lattice. *Physical Review B*, 100(15):155148, oct 2019. `doi:10.1103/physrevb.100.155148`.

[101] D. Poulin, A. Qarry, R. Somma, and F. Verstraete. Quantum simulation of time-dependent hamiltonians and the convenient illusion of hilbert space. *Physical Review Letters*, 106(17):170501, apr 2011. `doi:10.1103/physrevlett.106.170501`.

[102] Z. Huang and A. V. Balatsky. Complexity and geometry of quantum state manifolds, 2017. `doi:10.48550/ARXIV.1711.10471`.

[103] A. J. Daley, C. Kollath, U. Schollwöck, and G. Vidal. Time-dependent density-matrix renormalization-group using adaptive effective hilbert spaces. *Journal of Statistical Mechanics: Theory and Experiment*, 2004(04):P04005, apr 2004. `doi:10.1088/1742-5468/2004/04/p04005`.

[104] U. Schollwöck. The density-matrix renormalization group. *Reviews of Modern Physics*, 77(1):259–315, apr 2005. `doi:10.1103/revmodphys.77.259`.

[105] 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`.

[106] G. Vidal. Efficient simulation of one-dimensional quantum many-body systems. *Physical Review Letters*, 93(4):040502, jul 2004. `doi:10.1103/physrevlett.93.040502`.

[107] S. R. White and A. E. Feiguin. Real-time evolution using the density matrix renormalization group. *Physical Review Letters*, 93(7):076401, aug 2004. `doi:10.1103/physrevlett.93.076401`.

[108] V. Alba and P. Calabrese. Entanglement and thermodynamics after a quantum quench in integrable systems. *Proceedings of the National Academy of Sciences*, 114(30):7947–7951, jul 2017. `doi:10.1073/pnas.1703516114`.

[109] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–356, 1989.

[110] D.-X. Zhou. Universality of deep convolutional neural networks. *Applied and Computational Harmonic Analysis*, 48(2):787–794, mar 2020. `doi:10.1016/j.acha.2019.06.004`.

[111] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, jan 1993. `doi:10.1016/s0893-6080(05)80131-5`.

[112] G. Fabiani and J. Mentink. Investigating ultrafast quantum magnetism with machine learning. *SciPost Physics*, 7(1), jul 2019. `doi:10.21468/scipostphys.7.1.004`.

[113] G. Fabiani and J. H. Mentink. Ultrafast dynamics of entanglement in heisenberg antiferromagnets, 2019. `doi:10.48550/ARXIV.1912.10845`.

[114] G. Fabiani, M. Bouman, and J. Mentink. Supermagnonic propagation in two-dimensional antiferromagnets. *Physical Review Letters*, 127(9):097202, aug 2021. `doi:10.1103/physrevlett.127.097202`.

[115] S.-H. Lin and F. Pollmann. Scaling of neural-network quantum states for time evolution. *physica status solidi (b)*, 259(5), jan 2022. doi:10.1002/pssb.202100172.

[116] D. Hofmann, G. Fabiani, J. Mentink, G. Carleo, and M. Sentef. Role of stochastic noise and generalization error in the time propagation of neural-network quantum states. *SciPost Physics*, 12(5), may 2022. doi:10.21468/scipostphys.12.5.165.

[117] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, apr 1970. doi:10.1093/biomet/57.1.97.

[118] J. Broeckhove, L. Lathouwers, E. Kesteloot, and P. V. Leuven. On the equivalence of time-dependent variational principles. *Chemical Physics Letters*, 149(5-6):547–550, sep 1988. doi:10.1016/0009-2614(88)80380-4.

[119] A. Raab. On the dirac–frenkel/McLachlan variational principle. *Chemical Physics Letters*, 319(5-6):674–678, mar 2000. doi:10.1016/s0009-2614(00)00200-1.

[120] J. Liesen and Z. Strakos. *Krylov Subspace Methods*. Numerical Mathematics and Scientific Computation. Oxford University Press, 2012.

[121] E. Hairer, G. Wanner, and C. Lubich. *Geometric Numerical Integration*. Springer-Verlag, 2006. doi:10.1007/3-540-30666-8.

[122] A. M. Childs, I. L. Chuang, and D. W. Leung. Realization of quantum process tomography in nmr. *Physical Review A*, 64(1):012314, June 2001. doi:10.1103/physreva.64.012314.

[123] J. L. O'Brien, G. J. Pryde, A. Gilchrist, D. F. V. James, N. K. Langford, T. C. Ralph, and A. G. White. Quantum process tomography of a controlled-not gate. *Physical Review Letters*, 93(8):080502, August 2004. doi:10.1103/physrevlett.93.080502.

[124] R. C. Bialczak et al. Quantum process tomography of a universal entangling gate implemented with josephson phase qubits. *Nature Physics*, 6(6):409–413, April 2010. doi:10.1038/nphys1639.

[125] Y. S. Weinstein, T. F. Havel, J. Emerson, N. Boulant, M. Saraceno, S. Lloyd, and D. G. Cory. Quantum process tomography of the quantum fourier transform. *The Journal of Chemical Physics*, 121(13):6117–6133, September 2004. doi:10.1063/1.1785151.

[126] H. Kampermann and W. S. Veeman. Characterization of quantum algorithms by quantum process tomography using quadrupolar spins in solid-state nuclear magnetic resonance. *The Journal of Chemical Physics*, 122(21), June 2005. doi:10.1063/1.1904595.

[127] T. Jullien, P. Roulleau, B. Roche, A. Cavanna, Y. Jin, and D. C. Glattli. Quantum tomography of an electron. *Nature*, 514(7524):603–607, October 2014. doi:10.1038/nature13821.

[128] R. Bisognin et al. Quantum tomography of electrical currents. *Nature Communications*, 10(1), July 2019. doi:10.1038/s41467-019-11369-5.

[129] I. López Gutiérrez, F. Dietrich, and C. B. Mendl. Quantum process tomography of unitary maps from time-delayed measurements. *Quantum Information Processing*, 22(6), June 2023. doi:10.1007/s11128-023-04008-y.

[130] J. Emerson, R. Alicki, and K. Życzkowski. Scalable noise estimation with random unitary operators. *Journal of Optics B: Quantum and Semiclassical Optics*, 7(10):S347–S352, September 2005. doi:10.1088/1464-4266/7/10/021.

[131] E. Knill, D. Leibfried, R. Reichle, J. Britton, R. B. Blakestad, J. D. Jost, C. Langer, R. Ozeri, S. Seidelin, and D. J. Wineland. Randomized benchmarking of quantum gates. *Physical Review A*, 77(1):012307, January 2008. doi:10.1103/physreva.77.012307.

[132] E. Magesan, J. M. Gambetta, and J. Emerson. Scalable and robust randomized benchmarking of quantum processes. *Physical Review Letters*, 106(18):180504, May 2011. doi:10.1103/physrevlett.106.180504.

[133] J. P. Gaebler, A. M. Meier, T. R. Tan, R. Bowler, Y. Lin, D. Hanneke, J. D. Jost, J. P. Home, E. Knill, D. Leibfried, and D. J. Wineland. Randomized benchmarking of multiqubit gates. *Physical Review Letters*, 108(26):260503, June 2012. doi:10.1103/physrevlett.108.260503.

[134] J. M. Chow et al. Randomized benchmarking and process tomography for gate errors in a solid-state qubit. *Physical Review Letters*, 102(9):090502, March 2009. doi:10.1103/physrevlett.102.090502.

[135] S. Olmschenk, R. Chicireanu, K. D. Nelson, and J. V. Porto. Randomized benchmarking of atomic qubits in an optical lattice. *New Journal of Physics*, 12(11):113007, November 2010. doi:10.1088/1367-2630/12/11/113007.

[136] A. Smith, B. E. Anderson, H. Sosa-Martinez, C. A. Riofrío, I. H. Deutsch, and P. S. Jessen. Quantum control in the cs 6s1/2 ground manifold using radio-frequency and microwave magnetic fields. *Physical Review Letters*, 111(17):170502, October 2013. `doi:10.1103/physrevlett.111.170502`.

[137] K. R. Brown, A. C. Wilson, Y. Colombe, C. Ospelkaus, A. M. Meier, E. Knill, D. Leibfried, and D. J. Wineland. Single-qubit-gate error below 10-4 in a trapped ion. *Physical Review A*, 84(3):030303, September 2011. `doi:10.1103/physreva.84.030303`.

[138] R. Barends et al. Superconducting quantum circuits at the surface code threshold for fault tolerance. *Nature*, 508(7497):500–503, April 2014. `doi:10.1038/nature13171`.

[139] J. T. Muhonen et al. Quantifying the quantum gate fidelity of single-atom spin qubits in silicon by randomized benchmarking. *Journal of Physics: Condensed Matter*, 27(15):154205, March 2015. `doi:10.1088/0953-8984/27/15/154205`.

[140] E. Nielsen, J. K. Gamble, K. Rudinger, T. Scholten, K. Young, and R. Blume-Kohout. Gate set tomography. *Quantum*, 5:557, October 2021. `doi:10.22331/q-2021-10-05-557`.

[141] S. T. Merkel, J. M. Gambetta, J. A. Smolin, S. Poletto, A. D. Córcoles, B. R. Johnson, C. A. Ryan, and M. Steffen. Self-consistent quantum process tomography. *Physical Review A*, 87(6):062119, June 2013. `doi:10.1103/physreva.87.062119`.

[142] R. Blume-Kohout, J. K. Gamble, E. Nielsen, J. Mizrahi, J. D. Sterk, and P. Maunz. Robust, self-consistent, closed-form tomography of quantum logic gates on a trapped ion qubit, 2013. `doi:10.48550/ARXIV.1310.4492`.

[143] D. Kim et al. Microwave-driven coherent operation of a semiconductor quantum dot charge qubit. *Nature Nanotechnology*, 10(3):243–247, February 2015. `doi:10.1038/nnano.2014.336`.

[144] R. Blume-Kohout, J. K. Gamble, E. Nielsen, K. Rudinger, J. Mizrahi, K. Fortier, and P. Maunz. Demonstration of qubit operations below a rigorous fault tolerance threshold with gate set tomography. *Nature Communications*, 8(1), February 2017. `doi:10.1038/ncomms14485`.

[145] J. P. Dehollain et al. Optimization of a solid-state electron spin qubit using gate set tomography. *New Journal of Physics*, 18(10):103018, October 2016. `doi:10.1088/1367-2630/18/10/103018`.

[146] T. Proctor, M. Revelle, E. Nielsen, K. Rudinger, D. Lobser, P. Maunz, R. Blume-Kohout, and K. Young. Detecting and tracking drift in quantum information processors. *Nature Communications*, 11(1), October 2020. `doi:10.1038/s41467-020-19074-4`.

[147] M. K. Joshi, A. Elben, B. Vermersch, T. Brydges, C. Maier, P. Zoller, R. Blatt, and C. F. Roos. Quantum information scrambling in a trapped-ion quantum simulator with tunable range interactions. *Physical Review Letters*, 124(24):240505, June 2020. `doi:10.1103/physrevlett.124.240505`.

[148] S. Mavadia, C. L. Edmunds, C. Hempel, H. Ball, F. Roy, T. M. Stace, and M. J. Biercuk. Experimental quantum verification in the presence of temporally correlated noise. *npj Quantum Information*, 4(1), February 2018. `doi:10.1038/s41534-017-0052-0`.

[149] G. Gutoski and N. Johnston. Process tomography for unitary quantum channels. *Journal of Mathematical Physics*, 55(3), March 2014. `doi:10.1063/1.4867625`.

[150] C. H. Baldwin, A. Kalev, and I. H. Deutsch. Quantum process tomography of unitary and near-unitary maps. *Physical Review A*, 90(1):012110, July 2014. `doi:10.1103/physreva.90.012110`.

[151] D. M. Reich, G. Gualdi, and C. P. Koch. Minimum number of input states required for quantum gate characterization. *Physical Review A*, 88(4):042309, October 2013. `doi:10.1103/physreva.88.042309`.

[152] R. L. Kosut. Quantum process tomography via l1-norm minimization, 2008. `doi:10.48550/ARXIV.0812.4323`.

[153] A. Shabani, R. L. Kosut, M. Mohseni, H. Rabitz, M. A. Broome, M. P. Almeida, A. Fedrizzi, and A. G. White. Efficient measurement of quantum dynamics via compressive sensing. *Physical Review Letters*, 106(10):100401, March 2011. `doi:10.1103/physrevlett.106.100401`.

[154] A. Kalev, R. L. Kosut, and I. H. Deutsch. Quantum tomography protocols with positivity are compressed sensing protocols. *npj Quantum Information*, 1(1), December 2015. `doi:10.1038/npjqi.2015.18`.

[155] G. Kutyniok and Y. C. Eldar, editors. *Compressed sensing.* Cambridge University Press, Cambridge, 2012.

[156] A. Gaikwad, Arvind, and K. Dorai. Efficient experimental characterization of quantum processes via compressed sensing on an nmr quantum processor. *Quantum Information Processing*, 21(12), November 2022. `doi:10.1007/s11128-022-03695-3`.

[157] Y. S. Teo, G. I. Struchalin, E. V. Kovlakov, D. Ahn, H. Jeong, S. S. Straupe, S. P. Kulik, G. Leuchs, and L. L. Sánchez-Soto. Objective compressive quantum process tomography. *Physical Review A*, 101(2):022334, February 2020. `doi:10.1103/physreva.101.022334`.

[158] S. T. Flammia, D. Gross, Y.-K. Liu, and J. Eisert. Quantum tomography via compressed sensing: error bounds, sample complexity and efficient estimators. *New Journal of Physics*, 14(9):095022, September 2012. `doi:10.1088/1367-2630/14/9/095022`.

[159] A. Steffens, C. A. Riofrío, W. McCutcheon, I. Roth, B. A. Bell, A. McMillan, M. S. Tame, J. G. Rarity, and J. Eisert. Experimentally exploring compressed sensing quantum tomography. *Quantum Science and Technology*, 2(2):025005, May 2017. `doi:10.1088/2058-9565/aa6ae2`.

[160] A. Smith, C. A. Riofrío, B. E. Anderson, H. Sosa-Martinez, I. H. Deutsch, and P. S. Jessen. Quantum state tomography by continuous measurement and compressed sensing. *Physical Review A*, 87(3):030102, March 2013. `doi:10.1103/physreva.87.030102`.

[161] G. Torlai, C. J. Wood, A. Acharya, G. Carleo, J. Carrasquilla, and L. Aolita. Quantum process tomography with unsupervised learning and tensor networks. *Nature Communications*, 14(1), May 2023. `doi:10.1038/s41467-023-38332-9`.

[162] A. M. Palmieri, E. Kovlakov, F. Bianchi, D. Yudin, S. Straupe, J. D. Biamonte, and S. Kulik. Experimental neural network enhanced quantum tomography. *npj Quantum Information*, 6(1), February 2020. `doi:10.1038/s41534-020-0248-6`.

[163] S. Ahmed, F. Quijandría, and A. F. Kockum. Gradient-descent quantum process tomography by learning kraus operators. *Physical Review Letters*, 130(15):150402, April 2023. `doi:10.1103/physrevlett.130.150402`.

[164] S. Xue, Y. Liu, Y. Wang, P. Zhu, C. Guo, and J. Wu. Variational quantum process tomography of unitaries. *Physical Review A*, 105(3):032427, March 2022. `doi:10.1103/physreva.105.032427`.

[165] J. Zhang and M. Sarovar. Quantum hamiltonian identification from measurement time traces. *Physical Review Letters*, 113(8):080401, August 2014. `doi:10.1103/physrevlett.113.080401`.

[166] J. Zhang and M. Sarovar. Identification of open quantum systems from observable time traces. *Physical Review A*, 91(5):052121, May 2015. `doi:10.1103/physreva.91.052121`.

[167] L. Che, C. Wei, Y. Huang, D. Zhao, S. Xue, X. Nie, J. Li, D. Lu, and T. Xin. Learning quantum hamiltonians from single-qubit measurements. *Physical Review Research*, 3(2):023246, June 2021. `doi:10.1103/physrevresearch.3.023246`.

[168] D. Zhao, C. Wei, S. Xue, Y. Huang, X. Nie, J. Li, D. Ruan, D. Lu, T. Xin, and G. Long. Characterizing quantum simulations with quantum tomography on a spin quantum simulator. *Physical Review A*, 103(5):052403, May 2021. `doi:10.1103/physreva.103.052403`.

[169] E. Flurin, L. Martin, S. Hacohen-Gourgy, and I. Siddiqi. Using a recurrent neural network to reconstruct quantum dynamics of a superconducting qubit from physical observations. *Physical Review X*, 10(1):011006, January 2020. `doi:10.1103/physrevx.10.011006`.

[170] D. Ruelle and F. Takens. On the nature of turbulence. *Communications in Mathematical Physics*, 20(3):167–192, sep 1971. `doi:10.1007/bf01646553`.

[171] F. Takens. Detecting strange attractors in turbulence. In *Lecture Notes in Mathematics*, pages 366–381. Springer Berlin Heidelberg, 1981. `doi:10.1007/bfb0091924`.

[172] H. Whitney. Differentiable manifolds. *The Annals of Mathematics*, 37(3):645, jul 1936. `doi:10.2307/1968482`.

[173] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw. Geometry from a time series. *Physical Review Letters*, 45(9):712–716, sep 1980. `doi:10.1103/physrevlett.45.712`.

[174] D. Aeyels. Generic observability of differentiable systems. *SIAM Journal on Control and Optimization*, 19(5):595–603, sep 1981. `doi:10.1137/0319037`.

[175] T. Sauer, J. A. Yorke, and M. Casdagli. Embedology. *Journal of Statistical Physics*, 65(3-4):579–616, nov 1991. doi:10.1007/bf01053745.

[176] J. Stark, D. Broomhead, M. Davies, and J. Huke. Takens embedding theorems for forced and stochastic systems. *Nonlinear Analysis: Theory, Methods & Applications*, 30(8):5303–5314, dec 1997. doi:10.1016/s0362-546x(96)00149-6.

[177] S. Makridakis. Time series prediction: Forecasting the future and understanding the past. *International Journal of Forecasting*, 10(3):463–466, 1994.

[178] T. Berry, J. R. Cressman, Z. Gregurić-Ferenček, and T. Sauer. Time-scale separation from diffusion-mapped delay coordinates. *SIAM Journal on Applied Dynamical Systems*, 12(2):618–649, January 2013. doi:10.1137/12088183x.

[179] F. Dietrich, G. Köster, and H.-J. Bungartz. Numerical model construction with closed observables. *SIAM Journal on Applied Dynamical Systems*, 15(4):2078–2108, January 2016. doi:10.1137/15m1043613.

[180] F. Dietrich, M. Kooshkbaghi, E. M. Bollt, and I. G. Kevrekidis. Manifold learning for organizing unstructured sets of process observations. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(4), April 2020. doi:10.1063/1.5133725.

[181] D. Giannakis. Delay-coordinate maps, coherence, and approximate spectra of evolution operators. *Research in the Mathematical Sciences*, 8(1), January 2021. doi:10.1007/s40687-020-00239-y.

[182] F. P. Kemeth et al. An emergent space for distributed data with hidden internal order through manifold learning. *IEEE Access*, 6:77402–77413, 2018. doi:10.1109/access.2018.2882777.

[183] F. P. Kemeth, T. Bertalan, T. Thiem, F. Dietrich, S. J. Moon, C. R. Laing, and I. G. Kevrekidis. Learning emergent pdes in a learned emergent space, 2020. doi:10.48550/ARXIV.2012.12738.

[184] T. Bertalan, F. Dietrich, I. Mezić, and I. G. Kevrekidis. On learning hamiltonian systems from data. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(12), December 2019. doi:10.1063/1.5128231.

[185] S. Greydanus, M. Dzamba, and J. Yosinski. Hamiltonian neural networks. *Advances in neural information processing systems*, 32, 2019.

[186] B. O. Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, May 1931. `doi:10.1073/pnas.17.5.315`.

[187] B. O. Koopman and J. v. Neumann. Dynamical systems of continuous spectra. *Proceedings of the National Academy of Sciences*, 18(3):255–263, March 1932. `doi:10.1073/pnas.18.3.255`.

[188] I. Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1–3):309–325, August 2005. `doi:10.1007/s11071-005-2824-x`.

[189] I. Mezić. Spectrum of the koopman operator, spectral expansions in functional spaces, and state-space geometry. *Journal of Nonlinear Science*, 30(5):2091–2145, December 2019. `doi:10.1007/s00332-019-09598-5`.

[190] F. Dietrich, T. N. Thiem, and I. G. Kevrekidis. On the koopman operator of algorithms. *SIAM Journal on Applied Dynamical Systems*, 19(2):860–885, January 2020. `doi:10.1137/19m1277059`.

[191] B. C. Hall. An elementary introduction to groups and representations. 2000. `doi:10.48550/ARXIV.MATH-PH/0005032`.

[192] Y. Kim et al. High-fidelity three-qubit itoffoli gate for fixed-frequency superconducting qubits. *Nature Physics*, 18(7):783–788, May 2022. `doi:10.1038/s41567-022-01590-3`.

[193] C. R. Clark et al. High-fidelity bell-state preparation with ca+40 optical qubits. *Physical Review Letters*, 127(13):130505, September 2021. `doi:10.1103/physrevlett.127.130505`.

[194] M. Atatüre, J. Dreiser, A. Badolato, A. Högele, K. Karrai, and A. Imamoglu. Quantum-dot spin-state preparation with near-unity fidelity. *Science*, 312(5773):551–553, April 2006. `doi:10.1126/science.1126074`.

[195] F. Marvasti, editor. *Nonuniform Sampling: Theory and Practice.* Information Technology: Transmission, Processing and Storage (PSTE). Springer, 2001.

[196] H. F. Hofmann. Information and noise in quantum measurement. *Physical Review A*, 62(2):022103, July 2000. `doi:10.1103/physreva.62.022103`.

[197] A. A. Clerk, M. H. Devoret, S. M. Girvin, F. Marquardt, and R. J. Schoelkopf. Introduction to quantum noise, measurement, and amplification. *Reviews of Modern Physics*, 82(2):1155–1208, April 2010. `doi:10.1103/revmodphys.82.1155`.

[198] R. D. Delaney, M. D. Urmey, S. Mittal, B. M. Brubaker, J. M. Kindem, P. S. Burns, C. A. Regal, and K. W. Lehnert. Superconducting-qubit readout via low-backaction electro-optic transduction. *Nature*, 606(7914):489–493, June 2022. `doi:10.1038/s41586-022-04720-2`.

[199] D. I. Schuster et al. Resolving photon number states in a superconducting circuit. *Nature*, 445(7127):515–518, February 2007. `doi:10.1038/nature05461`.

[200] H. Y. Wong, P. Dhillon, K. M. Beck, and Y. J. Rosen. A simulation methodology for superconducting qubit readout fidelity. *Solid-State Electronics*, 201:108582, March 2023. `doi:10.1016/j.sse.2022.108582`.

[201] J. Janesick and J. Tower. Particle and photon detection: Counting and energy measurement. *Sensors*, 16(5):688, May 2016. `doi:10.3390/s16050688`.

[202] S. J. van Enk and R. Blume-Kohout. When quantum tomography goes wrong: drift of quantum sources and other errors. *New Journal of Physics*, 15(2):025024, February 2013. `doi:10.1088/1367-2630/15/2/025024`.

[203] N. K. Langford. Errors in quantum tomography: diagnosing systematic versus statistical errors. *New Journal of Physics*, 15(3):035003, March 2013. `doi:10.1088/1367-2630/15/3/035003`.

[204] B. Kraus and J. I. Cirac. Optimal creation of entanglement using a two-qubit gate. *Physical Review A*, 63(6):062309, May 2001. `doi:10.1103/physreva.63.062309`.

[205] J. Zhang, J. Vala, S. Sastry, and K. B. Whaley. Geometric theory of nonlocal two-qubit operations. *Physical Review A*, 67(4):042313, April 2003. `doi:10.1103/physreva.67.042313`.

[206] O. Gamel. Entangled bloch spheres: Bloch matrix and two-qubit state space. *Physical Review A*, 93(6):062320, June 2016. `doi:10.1103/physreva.93.062320`.

[207] Q. Huang and C. B. Mendl. Classical simulation of quantum circuits using a multiqubit bloch vector representation of density matrices. *Physical Review A*, 105(2):022409, February 2022. `doi:10.1103/physreva.105.022409`.

[208] A. Córcoles, E. Magesan, S. J. Srinivasan, A. W. Cross, M. Steffen, J. M. Gambetta, and J. M. Chow. Demonstration of a quantum error detection code using a square lattice of four superconducting qubits. *Nature Communications*, 6(1), April 2015. doi:10.1038/ncomms7979.

[209] S. Schmidt and J. Koch. Circuit qed lattices: Towards quantum simulation with superconducting circuits. *Annalen der Physik*, 525(6):395–412, April 2013. doi:10.1002/andp.201200261.

[210] G. Springholz, V. Holy, M. Pinczolits, and G. Bauer. Self-organized growth of three- dimensional quantum-dot crystals with fcc-like stacking and a tunable lattice constant. *Science*, 282(5389):734–737, October 1998. doi:10.1126/science.282.5389.734.

[211] G. Markovich, C. P. Collier, S. E. Henrichs, F. Remacle, R. D. Levine, and J. R. Heath. Architectonic quantum dot solids. *Accounts of Chemical Research*, 32(5):415–423, January 1999. doi:10.1021/ar980039x.

[212] P. Jessen and I. Deutsch. *Optical Lattices*, pages 95–138. Elsevier, 1996. doi:10.1016/s1049-250x(08)60099-3.

[213] H. Haffner, C. Roos, and R. Blatt. Quantum computing with trapped ions. *Physics Reports*, 469(4):155–203, December 2008. doi:10.1016/j.physrep.2008.09.003.

[214] A. Montoison and D. Orban. Krylov.jl: A julia basket of hand-picked krylov methods. *Journal of Open Source Software*, 8(89):5187, September 2023. doi:10.21105/joss.05187.

[215] G. Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5(3):506–517, September 1968. doi:10.1137/0705041.

[216] D. Li, C. Quan, and J. Xu. Stability and convergence of strang splitting. part i: Scalar allen-cahn equation. *Journal of Computational Physics*, 458:111087, June 2022. doi:10.1016/j.jcp.2022.111087.

[217] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2017. doi:10.48550/ARXIV.1706.03762.

[218] Y.-H. Zhang and M. Di Ventra. Transformer quantum state: A multipurpose model for quantum many-body problems. *Physical Review B*, 107(7):075147, February 2023. doi:10.1103/physrevb.107.075147.

[219] V. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999. doi:10.1109/72.788640.

[220] E. Freitag and R. Busam. *Complex Analysis*. Springer-Verlag, 2005. doi:10.1007/3-540-30823-7.

[221] E. M. Stein. *Complex analysis*, volume 2. Princeton University Press, Princeton, 2003.

[222] A. Hirose. *Complex-Valued Neural Networks*. Springer Berlin Heidelberg, 2012. doi:10.1007/978-3-642-27632-3.

[223] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal. Deep complex networks. In *International Conference on Learning Representations*, 2018. URL: https://openreview.net/forum?id=H1T2hmZAb.

[224] M. Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. URL: https://www.tensorflow.org/.

[225] A. Paszke et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

A Appendix

A.1 Wirtinger formalism

As mentioned in the main text, the evaluation of the cost function we define in Eq. (4.18) requires operations with complex numbers. Our objective is to minimize this function, or, to be precise, the expected prediction error for data not used during training [219]. In order to use a gradient descent method for its optimization, we thus need to apply complex analysis (cf. [220, 221]).

We utilize the Wirtinger formalism to calculate the gradients of the cost function with respect to the complex-valued parameters. While not extensively adopted in the realm of artificial neural networks, this formalism has found application in certain contexts, as seen in, for example, [222, 223] and related literature. The *Wirtinger* or *Dolbeault* operators are formally defined as follows:

$$\frac{\partial}{\partial z} := \frac{1}{2}\left(\frac{\partial}{\partial x} - i\frac{\partial}{\partial y}\right), \quad \frac{\partial}{\partial z^*} := \frac{1}{2}\left(\frac{\partial}{\partial x} + i\frac{\partial}{\partial y}\right) \tag{A.1}$$

where z is a complex variable, and x and y are its real and imaginary parts respectively. Here, the partial derivative operators act on (real-) differentiable (identifying $\mathbb{C} \simeq \mathbb{R}^2$) functions $f : U \to \mathbb{C}$ (with $U \subset \mathbb{C}$ some open subset of $\mathbb{C}$), which are not required to be holomorphic.

Nevertheless, consider the case where

$$f(z) = f(x + iy) = u(x, y) + iv(x, y) \tag{A.2}$$

is indeed holomorphic. For a complex function to be differentiable at point (x_0, y_0) it must satisfy the Cauchy-Riemann equations

$$\left.\frac{\partial u}{\partial x}\right|_{x_0,y_0} = \left.\frac{\partial v}{\partial y}\right|_{x_0,y_0}, \qquad \left.\frac{\partial u}{\partial y}\right|_{x_0,y_0} = -\left.\frac{\partial v}{\partial x}\right|_{x_0,y_0}. \tag{A.3}$$

This implies that, in this case, the Wirtinger derivative $\partial/\partial z$ is equal to the complex derivative of f, whereas the conjugated Wirtinger derivative vanishes:

$$\frac{\partial f(z)}{\partial z} = f'(z), \quad \frac{\partial f(z)}{\partial z^*} = 0 \quad \forall z \in U, \quad f \text{ holomorphic} \tag{A.4}$$

Also, complex conjugating the second identity leads to $\partial f^*(z)/\partial z = 0$. Thus, if one chooses a holomorphic Ansatz ψ, computing the derivative of ψ with respect to its parameters can be done via commonly used methods in machine learning. Note that popular machine learning packages, such as Tensorflow [224] and Pythorch [225], currently support (at least some) complex valued operations, including differentiation of holomorphic functions.

Unfortunately for us, the cost function considered in Eq. (4.18) is not holomorphic, which can be directly seen from the fact that its output is a real number. For functions of this kind, $f : U \to \mathbb{R}$, the partial derivatives with respect to x and y can be obtained from the Wirtinger derivative via

$$\frac{\partial f}{\partial x} = 2\,\mathrm{Re}\left(\frac{\partial f}{\partial z}\right), \quad \frac{\partial f}{\partial y} = -2\,\mathrm{Im}\left(\frac{\partial f}{\partial z}\right). \tag{A.5}$$

Furthermore, with a straightforward calculation one can confirm that the chain rule becomes:

$$\frac{\partial}{\partial z}(g \circ f) = \left(\frac{\partial g}{\partial w} \circ f\right) \cdot \frac{\partial f}{\partial z} + \left(\frac{\partial g}{\partial w^*} \circ f\right) \cdot \frac{\partial f^*}{\partial z}. \tag{A.6}$$

This, in turn, can be naturally generalized to functions of several variables. For a vector $z \in \mathbb{C}^n$, we use the notation

$$\nabla_z^{\mathrm{W}} = \left(\frac{\partial}{\partial z_1}, \dots, \frac{\partial}{\partial z_n}\right)^T \tag{A.7}$$

for the Wirtinger nabla operator.

Consider the map $f_\theta : U \subset \mathbb{C}^n \to \mathbb{C}^m$, where $\theta \in \mathbb{C}^p$ represents the complex-valued parameters of an artificial neural network, and the network has an input dimension of n and an output dimension of m. Then our goal is to minimize:

$$\min_\theta C(\theta), \quad C(\theta) = \sum_{j=1}^{N} c\Big(f_\theta\big(x^{(j)}\big), y^{(j)}\Big) \tag{A.8}$$

with $c : \mathbb{C}^m \times \mathbb{C}^k \to \mathbb{R}$ depending on the network output and training labels $y^{(j)} \in \mathbb{C}^k$. Here $(x^{(j)}, y^{(j)})_{j=1,\dots,N}$ a given sequence of training samples. We assume that f_θ is holomorphic as function of the parameters θ (as we can always choose our Ansatz ψ to be). Applying the chain rule (A.6) and using the previous result that $\nabla_\theta^{\mathrm{W}} f_\theta^* = 0$ leads us to

$$\nabla_\theta^{\mathrm{W}} C(\theta) = \sum_{j=1}^{N} \sum_{k=1}^{m} \frac{\partial c\big(f_\theta\big(x^{(j)}\big), y^{(j)}\big)}{\partial f_{\theta,k}\big(x^{(j)}\big)} \, \nabla_\theta^{\mathrm{W}} f_{\theta,k}\big(x^{(j)}\big) \tag{A.9}$$

with the subscript k denoting the k-th output component of f_θ. In this equation, the gradient with respect to the real and imaginary parts of θ is obtained via Eq. (A.5).

As basic example, and relevant to the cost function used in the main text, the Wirtinger derivative of the quadratic cost (for $a, y \in \mathbb{C}^m$)

$$c(a, y) = \|a - y\|^2 = \sum_{j=1}^{m} |a_j - y_j|^2 \tag{A.10}$$

reads

$$\frac{\partial c(a, y)}{\partial a_j} = (a_j - y_j)^*. \tag{A.11}$$

Combined with the chain rule, we can conclude that the gradients of Eq. (4.18) with respect to the optimization parameters are

$$\frac{\partial C}{\partial \theta_\ell} = \frac{\partial C}{\partial (A\psi)} \frac{\partial (A\psi)}{\partial \psi} \frac{\partial \psi}{\partial \theta_\ell} = \left\langle A\psi - b \middle| A \frac{\partial \psi}{\partial \theta_\ell} \right\rangle. \tag{A.12}$$

A.2 Error analysis

In order to proof the feasibility of this method to avoid the numerical amplification of stochastic errors, we need to carefully consider how the errors at one time-step propagate through the numerical integration. To this goal, we differentiate between three different wave functions: the exact one, $\psi(t)$; the one obtained by the exact midpoint time-evolution, $\psi_\Delta(t)$; and the one represented by the network, $\psi_N(t)$.

The error of the network Ansatz at time t will be given by

$$\varepsilon(t) = \psi(t) - \psi_N(t). \tag{A.13}$$

This, in turn, can be split into the error due to the the midpoint method and the error due to the network optimization. That is, the total error is

$$\varepsilon(t) = \varepsilon_\Delta(t) + \varepsilon_N(t) \tag{A.14}$$

where the numerical integration error is given by

$$\varepsilon_\Delta(t) = \psi(t) - \psi_\Delta(t) \tag{A.15}$$

and the error strictly coming from the network optimization is

$$\varepsilon_N(t) = \psi_\Delta(t) - \psi_N(t). \tag{A.16}$$

Using the triangle inequality we can set an upper bound to the absolute error:

$$|\varepsilon(t)| \leq |\varepsilon_\Delta(t)| + |\varepsilon_N(t)| \tag{A.17}$$

As indicated by Eq. (4.16), to determine the state at the subsequent time step using the midpoint rule, one needs to solve the following linear matrix equation:

$$A\psi_\Delta(t_{n+1}) - B\psi_\Delta(t_n) = 0 \tag{A.18}$$

where

$$A = I + \frac{i\Delta t}{2}H \tag{A.19}$$

and

$$B = I - \frac{i\Delta t}{2}H. \tag{A.20}$$

As a second-order Runge-Kutta method, its global error is of the order of $O(\Delta t^2)$. However, this linear system is not solved exactly, but instead the parameters in $\psi_N(t_{n+1})$ are optimized to minimize Eq. (A.18). This means that in reality, we are solving the equation

$$A\psi_N(t_{n+1}) - B\psi_N(t_n) = r^{(n+1)}, \tag{A.21}$$

where $r^{(n+1)}$ is the residual obtained at the end of the optimization in time-step $n+1$.

To see how this residual and the network error interact, we introduce $\varepsilon_N(t_n)$ into Eq. (A.21) by using its previous definition, and we obtain

$$A(\psi_\Delta(t_{n+1}) - \varepsilon_N(t_{n+1})) - B(\psi_\Delta(t_n) - \varepsilon_N(t_n)) = r^{(n+1)}. \tag{A.22}$$

Since the exact midpoint rule must be 0 (see Eq. (A.18)) this simplifies to

$$\varepsilon_N(t_{n+1}) = A^{-1}\left(B\varepsilon_N(t_n) - r^{(n+1)}\right). \tag{A.23}$$

Similar to our approach for the midpoint error, we have now established a recursive relation for the network error at the next time step. Subsequently, through a proof by induction, one can demonstrate that

$$\varepsilon_N(t_n) = -B^{-1} \sum_{m=1}^{n} A^{-m} B^m r^{(n+1-m)}, \tag{A.24}$$

for all $n > 0$. Observe that the application of $A^{-1}B$ corresponds to a single time step in the midpoint time evolution. Consequently, the residual from each time step is incorporated into the error and undergoes unitary time evolution –thus preserving it's norm. We can then conclude that the residual does not significantly amplify the error as an artefact of the integration technique.

A.3 Geometrical perspective

Recall from the description of SR in the main text that, from a geometrical perspective, the architecture of the neural network defines the manifold $\mathcal{M}$ to which the time-evolution is restrained. In SR the parameters are updated by projecting the true derivative at the current time onto $\mathcal{M}$ (see Fig. A.1)

On the other hand, the gradient-based minimization proposed in this thesis is already restricted to $\mathcal{M}$. What happens in this case is that the parameters converge to the point of minimum distance to the time-step obtained from the integration flow Φ. Note that this point must necessarily be the projection of $\Phi(t)$ onto the manifold. Both optimization approaches are therefore equivalent from a geometrical perspective, and their differences reside simply in their numerical stability and computational complexity –SR being the more computationally efficient one and the gradient-based optimization being the more stable approach.

A.4 Filter weights visualization

For completeness, in Fig. A.2 we present the five complex-valued convolution layer filter weights for the time evolution of the 2D system at three time points, for the quench to $h = 2h_c$. It appears that the amplitudes vary rather slowly as compared to the complex arguments (phases), although it is not clear why this is or how this effect could be leveraged.

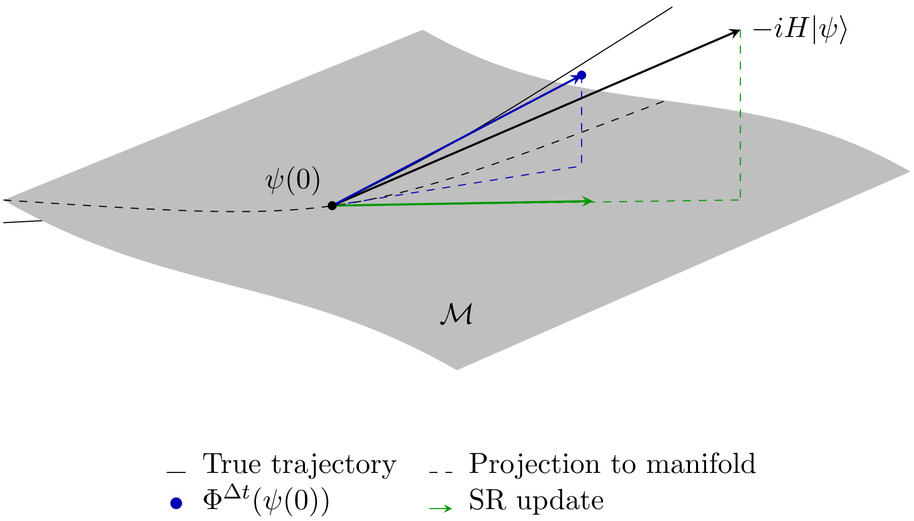
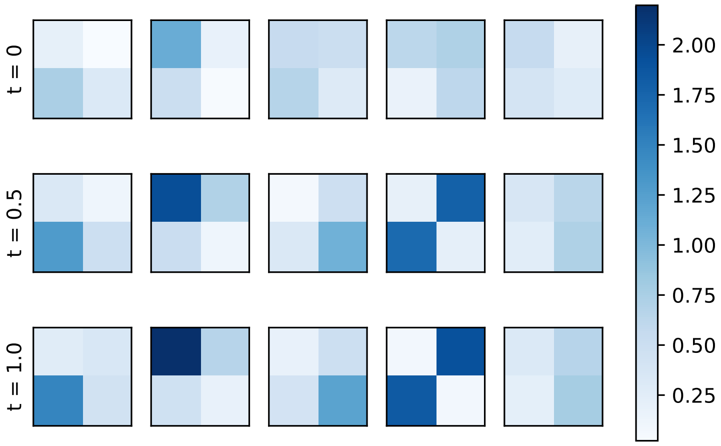
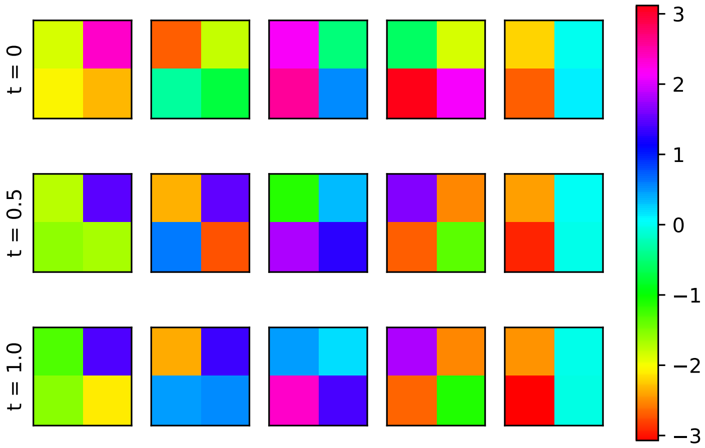


Figure A.1: Geometric interpretation of the optimization updates

Comparison from a geometrical point of view of the update achieved by SR and a gradient descent based method from a numerical integration flow $\Phi^{\Delta t}$



(a) Amplitudes of filter weights



(b) Arguments of filter weights

Figure A.2: CNN filters

(a) Amplitudes and (b) arguments of the complex weights for each filter of the CNN in Fig. 4.9, for the $h = 2h_c$ simulation shown in Fig. 4.10. The rows corresponds to different time points.