Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Machine-Learning Enhanced Studies of Quantum Many-Body Systems using Experimental and Numerical Data

Dissertation zur Erlangung des Doktorgrades
an der Fakultät für Mathematik, Informatik und Naturwissenschaften
Fachbereich Physik der Universität Hamburg

vorgelegt von

Niklas Käming
aus Hamburg

Hamburg
2024

Gutachter

Gutachter der Dissertation:	Prof. Dr. Klaus Sengstock Prof. Dr. Ludwig Mathey
Zusammensetzung der Prüfungskomission:	Prof. Dr. Klaus Sengstock Prof. Dr. Ludwig Mathey Prof. Dr. Roman Schnabel Prof. Dr. Maceij Lewenstein Prof. Dr. Dieter Jaksch
Vorsitzender der Prüfungskommission:	Prof. Dr. Roman Schnabel
Datum der Disputation:	21.02.2024
Vorsitzender Fach-Promotionsausschuss PHYSIK:	Prof. Dr. Markus Drescher
Leiter des Fachbereichs PHYSIK:	Prof. Dr. Wolfgang J. Parak
Dekan der Fakultät MIN:	Prof. Dr.-Ing. Norbert Ritter

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Hilfsmittel und Quellen benutzt zu haben.

Ort, Datum	Niklas Käming

Abstract

Machine learning has evolved from a niche topic to a subject that strongly influences all of our lives in recent years. In addition to paradigm-shifting developments in machine learning with large language models like ChatGPT, there are also advancements in the field of quantum physics using similar technologies. This work provides insights into the applications of machine learning within the realm of many-body quantum physics.

Utilizing ultracold quantum gases, which exhibit interesting properties due to their macroscopic quantum states, groundbreaking investigations into quantum mechanical effects have been conducted in recent decades. Exploring various topological phases is just one of many subfields. The Haldane model, describing topological insulators that could play a crucial role in developing modern materials with technologically interesting properties, is one of the currently studied models with the experimental platform of cold quantum gases. As traditional methods reach their limits in detecting individual phases, the use of machine learning to support the recognition of such phase transitions will be a vital part of the investigations of such phases in the future. In this thesis, the applications of machine learning methods applied to such problems will be discussed.

It is demonstrated that with the help of unsupervised machine learning, it is possible to distinguish the individual topological phases of the Haldane phase diagram based on experimental data. Methods for dimensionality reduction, such as deep autoencoders, are employed to gain insights into the underlying structure of the experimental data. Additionally, the data is cleaned using generative neural networks to make it suitable for further analysis.

Neural quantum states enable the representation of a significantly larger spectrum of the state space of quantum many-body systems even on classical computers. Through specific measurement techniques, it is possible to experimentally measure and fully store these states. Tomography for neural quantum states is currently accessible only for a limited number of systems and is expanded in the last part of this thesis by a proposed new protocol based on quench dynamics.

Zusammenfassung

Maschinelles Lernen hat sich in den letzten Jahren von einem Nischenthema zu einem Thema entwickelt, welches das Leben von uns allen stark beeinflusst. Neben den paradigmenwechselnden Entwicklungen auf dem Gebiet des maschinellen Lernens bei großen Sprachmodellen wie ChatGPT, gibt es auch im Bereich der Quantenphysik Fortschirtte beim Einsatz ebensolcher Technologien. Diese Arbeit gibt einen Einblick in die Anwendungsmöglichkeiten von maschinellem Lernen innerhalb des Kosmos der Vielteilchenquantenphysik.

Unter Zuhilfenahme von ultrkalten Quantengasen, die durch ihren makroskopischen Quantenzustand interessante Eigenschaften bieten, konnten in den letzten Jahrzehnten einige wegweisende Untersuchungen zu quantenmechanischen Effekten durchgeführt werden. Die Untersuchung verschiedener topologischer Phasen stellt dabei nur eines von vielen Teilgebieten da. Das Haldane Modell welches topologische Isolatoren beschreibt, welche eine entscheidene Rolle bei der Entwicklung moderner Materialien mit technologisch interessanten Eigenschaften beitragen könnten, ist dabei eines der aktuell untersuchten Modell mit dieser experimentellen Plattform. Klassische Methoden kommen bei der Detektion der einzelnen Phasenübergänge jedoch an ihre Grenzen und der Einsatz von maschinellem Lernen zur Unterstützung der Erkennung eben solcher Phasenübergänge ist der wesentliche Teil der Untersuchungen in dieser Arbeit.

In dieser Dissertation wird gezeigt, dass es mit Hilfe von nicht angeleitetem maschinellem Lernen möglich ist, die einzelnen topologischen Phasen des Haldane Phasendiagramms auf der Basis von experimentallen Daten zu unterscheiden. Dabei werden Methoden zur Dimensionsreduzierung, wie tiefe Autoencoder genutzt, um Aufschluss über die zugrundeliegende Struktur der experimentellen Daten zu erhalten. Die Daten werden zudem durch generative neuronale Netzwerke so bereinigt, dass sie für die weitere Analyse in Frage kommen.

Durch neuronale Quantenzustände ist es möglich ein deutlich größere Spektrum des Zustandsraumes von Quantenvielteilchensysteme auch in klassischen Computern darzustellen. Durch spezielle Messverfahren ist es möglich diese Zustände durch Experimente zu vermessen und vollständig zu speichern. Die Tomographie für neuronale Quantenzustände ist derzeit nur für eine beschränkte Zahl an Systemen zugänglich und wird in dieser Arbeit um eine auf Quenchdynamik basierende Methode erweitert.

Publications

- 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". In: *Nature Physics* 15.9 (2019), pp. 917–920. ISSN: 1745-2473. DOI: 10.1038/s41567-019-0554-0

- N. Käming, A. Dawid, K. Kottmann, M. Lewenstein, K. Sengstock, A. Dauphin, and C. Weitenberg. "Unsupervised machine learning of topological phase transitions from experimental data". In: *Machine Learning: Science and Technology* 2.3 (2021), p. 035037. DOI: 10.1088/2632-2153/abffe7

The publication from 2019 about supervised phase detection was published before the work on this thesis started. Nevertheless, it inspired and influenced most parts of this thesis.

The full implementation details of the second publication are available here:

- N. Käming, A. Dawid, K. Kottmann, M. Lewenstein, K. Sengstock, A. Dauphin, and C. Weitenberg. "Code and data for Unsupervised machine learning of topological phase transitions from experimental data". In: *Zenodo* (2021). DOI: 10.5281/zenodo.4700379

Contents

CHAPTER 1

Introduction

Machine learning (ML), a dynamic and rapidly evolving field, has witnessed unprecedented growth in recent years, transforming the landscape of technology and reshaping the way we approach complex problems. As we delve into the latest developments in ML, it becomes evident that the discipline is not merely a tool but a driving force behind innovations that touch every aspect of our lives. With the introduction of ChatGPT [1] or in general large language models, it becomes clear that ML will revolutionize the way we live, work, think and make new scientific discoveries.

The advent of deep learning, fueled by advances in neural network architectures, the availability of massive datasets and computational power, has propelled the field to new heights. Models such as transformers, with their ability to capture intricate patterns and dependencies in data, have become instrumental in natural language processing, computer vision, and beyond. ML models no longer only assist humans, they also outperform them in certain fields, like playing complex games [2] by teaching themselves [3, 4]. They also speed up scientific and technological discovery by developing new mathematical algorithms [5], developing new software [6] or solving problems like protein folding much more efficient [7, 8] leading to a boost in structural biology and drug design [9].

The success of ML has recently also achieved remarkable results and led to promising applications in quantum physics [10–13]. Most promising examples are the representation of quantum many-body states [14], efficient state tomography [15–17], the optimization of experimental protocols and setups [18–21] and the detection of phase boundaries and different phases of matter from numerical [22–35] and experimental data. Experimental data, normally suffering from imperfections such as noise, limited measurement apparatus restricting the accessible observables, or finite temperature, let machine learning reveal its true potential. Examples include scanning tunneling microscopy [36, 37], neutron scattering data from spin ice systems [38], vortex detection in Bose-Einstein condensate (BEC) [39] and momentum-space and real-space images of cold quantum gases [40–45]. Other perspectives of machine learning applications are to identify novel phases in exotic regimes [23, 42] or the effective modeling of theoretical concepts [46]. Recently it also became useful to reconstruct images

from quantum gas microscopes [47]. Some of the concepts have already been analogously transformed into the quantum computing regime, making use of the computing paradigms in quantum computing [48–51]. One prominent example is quantum anomaly detection [52].

In the realm of quantum physics, the study of many-body systems stands as a captivating frontier that challenges our understanding of matter and its fundamental interactions. The exponentially growing Hilbert space however limits the theoretical study of such systems and motivates the development of a playground of controllable simulations mimicking their rich phenomenon [53, 54]. Over time multiple platforms, namely cold quantum gases [55, 56], trapped ions [57], and photonic systems [58] have been developed addressing different problems.

The advent of experimental techniques enabling the creation and manipulation of degenerate bosonic [59, 60] and fermionic [61] ultracold quantum gases has opened a new avenue for investigating the rich tapestry of quantum many-body physics [55, 62]. This interdisciplinary field lies at the intersection of atomic, molecular, and optical physics, offering a unique platform to explore emergent phenomena arising from the collective behavior of a large number of interacting quantum particles. Realized through techniques such as laser cooling and evaporative cooling, cold quantum gases provide an exquisite playground for probing the intricate quantum dynamics of many-body systems. Typically composed of ultracold atoms, this experimental platform offers an unprecedented level of control. Modulating the laser intensities or using Feshbach resonances [63] the tunneling and interaction strength can be tuned in a wide range allowing simulation of a variety of different Hamiltonians. The resulting highly versatile experimental setups enable researchers to investigate a diverse array of quantum phases and transitions, paving the way for a deeper understanding of fundamental quantum phenomena.

Since the beginning of the field, a whole zoo of different lattice geometries was engineered like cubic lattices allowing the first observations of the phase transition between a bosonic superfluid [64] and fermionic metallic [65, 66] phase to a Mott insulating phase, and hexagonal lattices allowing the direct simulation of magnetism in its classical and quantum form [67, 68] and the realization of artificial graphene [69, 70]. Of current interest are nonseparable and bipartite lattices such as triangular lattices [67, 71], honeycomb lattices [69, 71–73], checkerboard lattices [74], Lieb lattices [75], Kagome lattices [76] or quasicrystal structures [77]. This variety of accessible lattices and the fact that optical lattices are defect-free allows the realization of a variety of different Hamiltonians with high controllability and accessibility.

The development of single-site resolution imaging using quantum gas microscopes [78, 79] which gives access to more quantum mechanical observables to study the underlying systems and leads to numerous new observations [80] allow access to new observables like on-site occupation. Recently this approach has been further developed by using quantum gas mag-

nification by matter wave optics [81] allowing access to below single site resolution without experimental challenging quantum gas microscopy.

One of the central themes in the field of many-body physics is the study of quantum phase transitions, where a many-body system undergoes a drastic change in its ground state as a control parameter, such as interaction strength, is varied. Understanding these transitions and characterizing the associated critical phenomena is crucial for unraveling the underlying quantum dynamics and is central for the broader field of condensed matter physics. One of the most interesting sub-fields is the analysis and understanding of topological phases of matter and its importance was underlined by the Nobel Prize in 2016. The topological properties of materials are not only technologically relevant [82] but also explain phenomena in modern physics like the Hall effects [83, 84] and topological insulators [85, 86]. Hamiltonians featuring topological bands have been realized in quantum gas experiments with optical lattices and topological properties have been successfully extracted [40, 72, 87–90]. However, most of the techniques suffer from technical difficulties making them unfeasible for some regimes or very data intensive. This requires an alternative ansatz to detect and identify different phases of matter.

In this thesis, I will combine the two areas of machine learning and cold quantum gases. I will demonstrate a variety of machine-learning methods to reconstruct the complete Haldane phase diagram [91] and its topological phases identified by the Chern number from experimental data [40, 43]. All technical advantages in the field of cold quantum gases are challenging the theoretical predictions of different exotic phases of matter. For experimentalists, it is appealing to search for new phases of matter guided by experimental data and provide helpful insights into the regimes that modern theoretical predictions do not cover. This can be accelerated by ML enhanced data analysis which I will prove useful in this work.

1.1 | Thesis Outline

In this thesis, I will develop different machine-learning methods to identify and analyze experimental cold quantum gas and numerical data. I will explain all the ingredients needed to use supervised and unsupervised techniques to solve the challenging task of quantum phase detection. For the first time, the complete Haldane phase diagram is reconstructed from experimental data without prior knowledge of the underlying phases. Furthermore, I will develop a novel neural network-based tomography ansatz for many-body quantum systems.

Chapter 2: Machine Learning This chapter is an introduction to the building blocks of machine learning. I will introduce the building blocks and learning algorithms for deep learning models as they might be new for readers from the quantum gas community. Furthermore, I will introduce concepts to study the loss landscape of Neural networks (NNs).

Chapter 3: Detection of Phase Transitions Here I will develop the algorithms to detect quantum phase transitions from different types of data. I will introduce supervised and unsupervised techniques like principle component analysis (PCA), anomaly detection and similarity analysis and discuss the requirements and applications of the different methods.

Chapter 4: Haldane model The main part of the thesis covers the supervised and unsupervised reconstruction of the Haldane model from experimental data. I will introduce the Haldane model, how it is realized experimentally, and how the different machine learning algorithms compare to each other and traditional methods to measure the Chern number. The contents of this chapter are published in [40] and [43]

Chapter 5: State Tomography In the last chapter, I will introduce a novel ansatz for quantum state tomography from random quench dynamics with Neural quantum state (NQS). I will introduce the theoretical foundation needed to realize this novel ansatz and discuss possible applications.

1.2 | Other Projects

During the time working on this thesis, an important other toolchain was developed which is technical and hence not part of this thesis, however, consumed a lot of my time. To successfully work with experimental data it is of course important to first collect it. The former framework is not suitable for ML applications. To allow faster access and data availability a new measurement framework for the laboratories is being developed, named common analysis framework for cold atoms (CAFCA). CAFCA features expandability and a common data layout which is accessible efficiently not only from classical analysis scripts but for most machine learning applications as well. It features support for S3 storage solutions, much faster and more scalable than all other solutions used so far and is future-proof as it is the industry standard for storing large datasets. Furthermore, an arbitrary amount of sensor data can be added due to its expandability. To fulfill the standards in scientific publishing we also take explicit care of how the data is represented. Therefore we use the HDF5 file standard and merge all measurement data in one file to keep it self-explanatory. The framework will be published in the future allowing cross-institutional standards for cold quantum experiments.

CHAPTER 2

Machine Learning

In this chapter, I will give a broad overview of the field of artificial intelligence (AI) and the methods used in this thesis. This introduction is however not complete and can not cover all the details of how to create good ML models. Those who want to improve their background knowledge can get more information in good textbooks about the topic like [92]. More tailored introductions for physicists exist like [93]. To get an overview of what current research topics in the context of ML applied to physics I refer to the lecture notes [12] or these reviews [10, 94].

In the first section 2.1, I will try to zoom in from the broad field of artificial intelligence (AI) to deep learning (DL) and explain the differences, advantages, and disadvantages of the different areas of AI. Afterward, the focus is ML. In section 2.2 I will introduce the two different tasks ML tries to solve while in section 2.3 I will discuss the different types of learning. The next steps described in 2.4 include learning algorithms and discussing their ability to generalize 2.5. A special kind of learning algorithm is introduced in section 2.6 called deep neural network (DNN) including a special subgroup called autoencoder (AE). The study of the loss landscape of deep neural networks (DNNs) reveals some interesting and helpful insights, thus I will discuss some aspects of loss landscape analysis in section 2.7.

2.1 | Learning Machines

With the advent of computers in the second half of the 20th century, the dream of making computers think like humans came up. The scientific discoveries in the field of neurobiology however make it seem quite complicated to replicate our brain. Nevertheless, along the path to the ultimate AI some algorithms have been developed to solve a small part of problems even better than humans. With the release of large language models [1] and large datasets, it is nowadays also possible to mimic the complex structure of language and have conversations with AI systems. Even today it is quite obvious that humans and machines have different

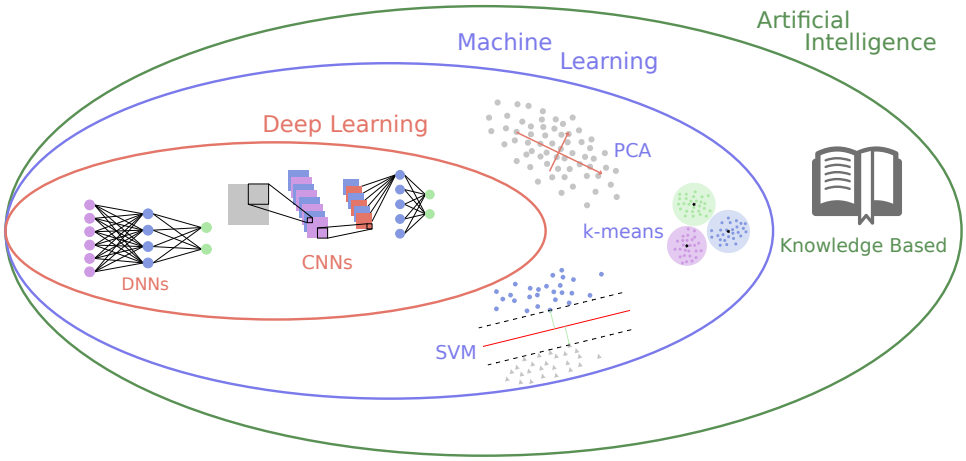

Figure 2.1: Venn diagram of the relation between deep learning (DL), machine learning (ML) and artificial intelligence (AI) referring to [92]. DL includes all DNN and convolutional neural network (CNN) architectures. ML also includes simple linear algebra models like PCA and support vector machnine (SVM) or k-means clustering whilst AI includes all kinds of different algorithms to create learning machines. Diagram adapted from [12].

areas of expertise where they excel. This observation has been known since the 1980s and is known as the Morvac paradox [95].

> It is comparatively easy to make computers exhibit adult-level performance on intelligence tests or playing checkers, and difficult or impossible to give them the skills of a one-year-old regarding perception and mobility.

The tasks computers outperform most humans nowadays are tasks that can be described by formal rules like chess or finding the shortest route on a map. This type of AI can already be achieved by programming decision trees with knowledge-based approaches. The way the computer outperforms humans, in that case, lies in the computational power, e.g. testing thousands of chess moves and their probability to improve the game in seconds, and not in the logical reasoning since the decisions are set by a programmer [1]. It is debatable to call such systems artificial intelligence. Nevertheless, these kinds of algorithms can make decisions based on some inputs. Therefore I will follow the definitions given by [92] where these types of algorithms are considered to be knowledge-based AI.

The restricted capabilities of knowledge-based, hard-coded intelligence suggest that another idea to create artificial intelligence is required. Machine learning (ML) describes the capability of machines to acquire their knowledge by using a fixed dataset from an experiment. One of the most straightforward and first applicable ML algorithms is logistic regression or a naive Bayes algorithm, e.g. in an E-mail spam filter. Both algorithms however suffer from the

[1] It should be mentioned that knowledge-based algorithms are outperformed by self-trained reinforcement algorithms [4] in many cases.

representation of the data in the dataset. To give useful information the algorithms need to be fed with several information, so-called **features** of the data. In the case of an E-Mail spam filter, it requires information about certain words or metadata to succeed. However, this representation of the data is implemented by a programmer and is hence vulnerable to biases. Furthermore sometimes it is not obvious which features are useful to identify a spam E-Mail or to identify a cat or dog from an image because the underlying correlations are unknown. The power of **representation learning**, consequently a subset of ML, tries to solve that by giving the algorithm the capabilities to find the underlying correlations (features) to make meaningful decisions. In the case of the E-Mail filter this might be the complete content of the E-Mail and the metadata. In the following, I will refer to both kinds of ML algorithms, those with given representations and those considered as representation learning, as machine learning (ML). Successful representation learning algorithms are PCA [96, 97], SVM [98, 99], and k-means clustering.

A subset of ML, or to be precise representation learning, is deep learning (DL). DL solves the problem of representing the features by combining features to new simpler features and recycles them as input for deeper models. This type of ML is accessible due to the advent of capable computational resources to process large datasets like image collections. The basis model for DL is the feedforward multilayer perceptron (MLP). A simple model, modeled by a function, that maps an input to a new set of features or a new representation of the data. The function itself consists of many smaller functions, all chained together. This deep chain of functions is the namesake of the field of DL. Today many different DL algorithms exist like DNNs or CNNs as a special version to work with images.

An overview of how AI, ML, and DL relate to each other is given in figure 2.1. In a nutshell AI can be seen as a collective term of different algorithms. It becomes evident that the only viable way to create an AI that can compete in real-world scenarios are ML methods. Most promising are DL algorithms allowing an abstraction of features from real-world data on several levels.

2.2 | Tasks

ML excels in different learning tasks and they can loosely be sorted into different categories. Here two categories are introduced, though, they are not complete and new categories emerge with the advances in the field.

Regression

Regression tasks are defined by their immediate relation between the input and output. However, the relation might be highly non-linear, and machine learning is required to model the

dependence. Regression problems try to map inputs $\boldsymbol{x}$ (features) to continuous outputs $\boldsymbol{y}$ (targets). The objective is to find the ML model that expresses the function $\boldsymbol{y} = f(\boldsymbol{x})$. Notice that the function itself is unknown. A recent useful example outside physics is predicting temperature or rainfall based on atmospheric observations [100]. Regression problems could also be beneficial in unsupervised learning schemes like anomaly detection described in chapter 3.3.2 which is applied to topological data from cold quantum gases in chapter 4.7.4. A subset of regression tasks is **dimensionality reduction**. It requires the dimension of the output to be smaller than the input $\dim(\boldsymbol{x}) > \dim(\boldsymbol{y})$. This becomes useful to reduce the size of data, like image compression, or get some insights into the data using clustering methods or just inspecting the lower dimensional representation of the data which is done for time of flight (ToF) images in chapter 4.7.1. A typical ML algorithm to solve this type of task is k-mean clustering. Based on the data each data is assigned to a certain class which leads to the second big group of tasks.

Classification

One of the most prototypical tasks is classification. In this task, an ML algorithm is trained to assign discrete labels to input data. In contrast to regression tasks, the desired output of such an algorithm is a vector $\boldsymbol{y}$ where each element is binary $\boldsymbol{y}_i \in \{0, 1\}$. The task is to find again a model for the unknown function $\boldsymbol{y} = f(\boldsymbol{x})$ where $\boldsymbol{x}$ is some type of input, e.g. images or time sequences. Each element in the vector $\boldsymbol{y}$ corresponds to a single pre-attached class and commonly only one class is allowed as an output. This encoding is called one hot encoding. Prominent examples of multi-class tasks are the Iris data set [101] from 1936 which is until today the textbook example for classification tasks. Classification can also be used for the detection of phase transitions in physics as it will be introduced in chapter 3.2 and later applied to data from cold quantum gases 4.6.

2.3 | Types of Learning

ML can be sorted into three categories regarding their learning procedure. Supervised, unsupervised, and reinforcement learning. The distinction lies in the way the training data $\boldsymbol{z} \in \mathcal{D}$ is presented to the ML algorithm. Based on the type of learning and the task to solve and the underlying data we can choose a suitable ML algorithm.

2.3.1 Supervised Learning

In supervised learning schemes the training data $\boldsymbol{z} \in \mathcal{D}$ are tuples $\boldsymbol{z}_i = (\boldsymbol{x}_i, \boldsymbol{y}_i)$ where $\boldsymbol{x}_i$ are the input values from an input space. Each element of $\boldsymbol{x}_i$ is referred to as a feature. This can be a pixel from an image or the number of atoms on a site of a lattice. $\boldsymbol{y}_i$ is the expected

output for the corresponding input $\boldsymbol{x}_i$. With this data, the ML algorithm is trained in a way that it learns to generalize. After training it can be applied to new unseen data. This type of learning is called supervised learning because a supervisor or teacher, i.e. the connection between input and output, is used as to teach the algorithm to reproduce a certain behavior. Notice that the training data samples $\boldsymbol{z}_i$ have to be created by hand like it is done for the MNIST dataset [102], or given by some prior algorithm. There exist different approaches to supervised learning, ranging from classic ML algorithms like SVMs to DL approaches like DNNs as described in chapter 2.6. In physics, supervised learning can help to pinpoint phase transition as described in chapter 3.2 and to map out parameter regimes that are unfeasible with classical methods as I will show in chapter 4.6.

2.3.2 Unsupervised Learning

Supervised learning has its confinements as it is restricted to the use of labeled data which is not always available for the desired task. Another approach is unsupervised learning where the data $\boldsymbol{z}_i$ exclusively consist of the features $\boldsymbol{x}_i$. The goal of unsupervised learning is to uncover hidden patterns, relationships, or structures within the data without explicit guidance or predefined categories. In unsupervised learning, the algorithm explores the data and tries to find meaningful patterns or groupings on its own. There are two main types of unsupervised learning. The first type is dimensionality reduction which involves reducing the number of features or variables in the data while preserving its important characteristics. Dimensionality reduction techniques help simplify the data and may reveal underlying patterns as shown in chapter 4.7.1 to identify different topological phases of matter. Methods range from PCA as explained in section 3.1.1 to autoencoders which are explained in chapter 2.6.2. The second one is clustering which is often used after the approach of dimensionality reduction. It involves grouping similar data points based on certain features or characteristics. The algorithm tries to identify inherent structures or clusters within the data. This technique can be used to reconstruct a phase diagram without knowledge of the underlying phases as shown in section 4.7.

2.3.3 Reinforcement Learning

Reinforcement learning is the third type of ML approach. In contrast to supervised and unsupervised learning, it requires no dataset at all. A ML agent can instead interact with an environment that is augmented with rewards if the last interaction was helpful or harmful in achieving a certain task. The history of interactions and rewards or penalties takes the place of the dataset. The ML agent has to learn a strategy to solve a certain task without prior knowledge of a strategy whatsoever. The scheme of reinforcement learning is not used in this thesis but has several applications. However, it can be acknowledged that it also has some

applications in science, for example in quantum error correction [103], quantum control [104, 105], and quantum circuit optimization [106].

2.4 | Training

At this point, the possible tasks and learning types have been introduced. This chapter will now describe the actual learning procedure. As mentioned earlier, ML models can be described as a function $\boldsymbol{y} = f(\boldsymbol{y})$ that is unknown. To maintain a better overview, only models that are parameterized by a parameter set $\boldsymbol{\theta}$ are examined here. Nevertheless, it should be noted that there are also parameter-free approaches, such as PCA which is described in detail in chapter 3.1.1. The general goal is to find the optimal parameters $\tilde{\boldsymbol{\theta}}$ to model the function $f(\boldsymbol{x})$. Notice that the model could be parameterized in many different ways. In Chapter 2.6, DNNs are introduced, which provide an excellent parametrizable approach to solving many different tasks.

The training itself starts with a general random initialization of the parameters, with some constraints regarding the probability distribution for the initialization which are beyond the scope of this thesis. For the interested reader, I refer to [92] for a detailed explanation. The actual learning progress is now to vary the parameters $\boldsymbol{\theta}$ until they converge to the optimal parameters $\tilde{\boldsymbol{\theta}}$ by minimizing a so-called **loss function** $\mathcal{L}(\mathcal{D}, \boldsymbol{\theta})$ which measures the success of the learning and can be calculated for the complete or subset of the dataset $\mathcal{D} = \{\boldsymbol{z}_i = (\boldsymbol{x}_i, \boldsymbol{y}_i)\}$ and the parameters $\boldsymbol{\theta}$ itself. Optimal in this context is not a definite choice. More about that in chapter 2.7 dedicated to the loss landscape. Several loss functions have been proven successful in ML applications. The most prominent ones are the mean square error (MSE) and the categorical cross entropy (CCE) which are used in chapter 4 to identify different phases of matter. The MSE loss function, which is commonly used for regression problems, is defined by

$$\mathcal{L}_{\text{MSE}}(\mathcal{D}, \boldsymbol{\theta}) = \frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} (y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i))^2 \tag{2.1}$$

where N_{data} is the number of samples inside the dataset. The CCE loss function can be calculated using one-hot encoding for N_{cat} categories using

$$\mathcal{L}_{\text{CCE}}(\mathcal{D}, \boldsymbol{\theta}) = -\frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} \sum_{c=1}^{N_{\text{cat}}} y_{i,c} \log\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\right) \tag{2.2}$$

where $y_{i,c}$ is the corresponding target value for the category c. The CCE can be derived from the Kullback-Leibler divergence, which will be discussed in the context of state tomography in chapter 5.

The optimization of the parameters $\boldsymbol{\theta}$ can be divided into gradient-based, gradient-free, and analytic approaches. Here only gradient-based methods will be discussed in further detail. A

Algorithm 1 Mini-Batch Stochastic Gradient Descent

Input: Training data: $\mathcal{D} = \{\boldsymbol{z}_i = (\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N_{\text{data}}}$
Input: Learning rate: λ, Batch size: N_{batch}, Number of epochs: N_{epochs}
Output: Optimized model parameters: $\tilde{\boldsymbol{\theta}}$

```
Initialize model parameters θ
for n_epoch = 1 to N_epoch do
    Randomly shuffle the training data
    for m = 1 to N_data by N_batch do
        Select mini-batch: B = {(x_i, y_i)}_m^{m+N_batch-1}
        Compute mini-batch gradient: ∇_θ L(B, θ) = 1/N_batch Σ_{z_i ∈ B} ∇_θ L(z_i, θ)
        Update parameters: θ ← θ − λ ∇_θ L(B, θ)
    end for
end for
```

gradient-based algorithm that is capable of training especially DNNs is the stochastic gradient descent (SGD) [107, 108] algorithm or more precisely the mini-batch SGD which is described formally in algorithm 1. The principles of the algorithm can be explained by discussing each part of the name. Gradient descent means, that the gradient of the loss function $\nabla_{\boldsymbol{\theta}}\mathcal{L}$ is calculated concerning the model parameters, and a path along the steepest descent of the loss landscape is selected. This is numerically achieved by using computational graphs and automatic differentiation [109] and is beyond the scope of this thesis and already solved in most libraries as discussed in section 2.8. The interested reader can learn more about the details in chapter 6.5 of [92]. The step size along the steepest descent is set by the **learning rate** λ. Another ingredient of the name is the mini-batch. In principle, it is possible to calculate the gradient for all samples of the dataset $\mathcal{D}$ at the same time. However, this is numerically inefficient and leads to bad generalization, as it is discussed in the next chapter 2.5. It is better to limit the updates of the parameters $\boldsymbol{\theta}$ to a subset of the original dataset, called **mini-batch** $\mathcal{B} \subset \mathcal{D}$ and update the parameters several times. Stochastic starts to play a role by shuffling the data after each **epoch**, i.e. after all mini-batches have been used to update the parameters. Sometimes SGD is expanded by the concept of momentum, which is similar to momentum in physics. The "movement" of the parameters, i.e. the change after calculating the gradient for the mini-batch, is influenced by the previous move, normally by just adding it to the current update, i.e. using a linear combination of the current and last update [110]. Next to SGD there exist other learning algorithms that excel in the context of DNNs like Adam [111]. There are also some non-gradient-based methods like simulated annealing and genetic algorithms [112] but are more designed to excel in specific regimes.

Almost all ML models and training algorithms are not only defined by the intrinsic parameters of the model $\boldsymbol{\theta}$. They always are highly influenced by numerous other factors, like the learning

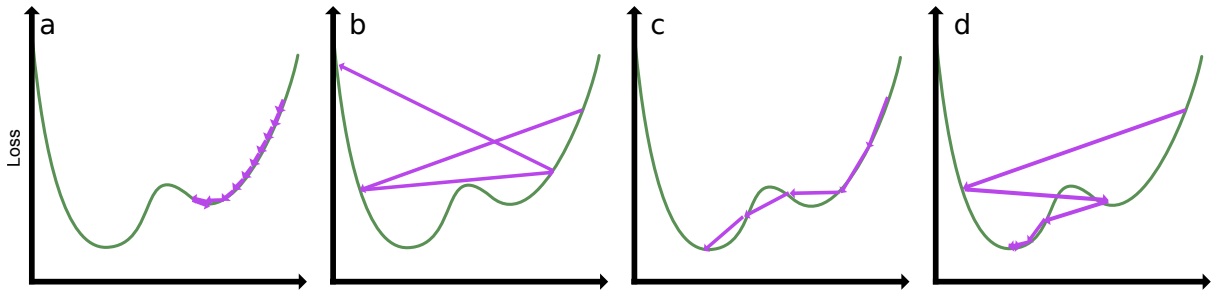

Figure 2.2: Different learning rates. (a) The learning rate is too small and convergence is very slow. Potential to get stuck in a local minimum. (b) The learning rate is too big and the algorithm jumps around in the loss landscape. Convergence is very unlikely. (c) The learning rate is set to a medium value and the training algorithm convergence in a reasonable amount of time. (d) Learning rate schedule. The learning rate is scheduled over the training iterations. This allows fast convergence and efficient fine-tuning of the parameters. All illustrations may lead to the assumption that the loss landscape is convex, which is not true in real-world examples.

rate λ, the mini-batch size N_{batch}, and how many deep layers or neurons to use in the context of DNNs. These parameters are so-called **hyper parameters**. Their choice is very delicate and there is no definite guide on how they can be determined. However, there are helpful libraries to solve such problems like the Optuna [113] library. In the end, the selection of such parameters relies on the experience of the programmer.

At least some intuition is possible for some hyperparameters, such as the learning rate λ whose choice is very delicate. It influences multiple aspects of the learning progress as well as the outcome. A small learning rate ensures stable training but slows down the convergence and thus is not very optimal. Furthermore, there exists a risk of getting stuck in a local minimum as visualized in figure 2.2a. This might not always be a disadvantage as it is discussed in chapter 2.7 but for now, it is considered not beneficial. In contrast, a high learning rate accelerates the convergence as the updates of the parameters are larger. However, if the learning rate is too high, it can lead to overshooting and cause divergence as illustrated in figure 2.2b. An optimal learning rate (figure 2.2c) lies somewhere in between. The most advanced technique is a learning rate schedule combining in general larger learning rates at the beginning and smaller ones at the end. The schedule is not required to be smooth. The learning rate schedule could also be a feedback parameter updated after each training epoch. Choosing a learning rate depends on the ML model and data and is in most cases a matter of experience. A good guidance can be found in this article [114].

2.5 | Generalization

So far it looks like ML is just a fancy word for function fitting. However, that is not true. The groundbreaking difference is the power of generalization, i.e. applying the learned algorithmic

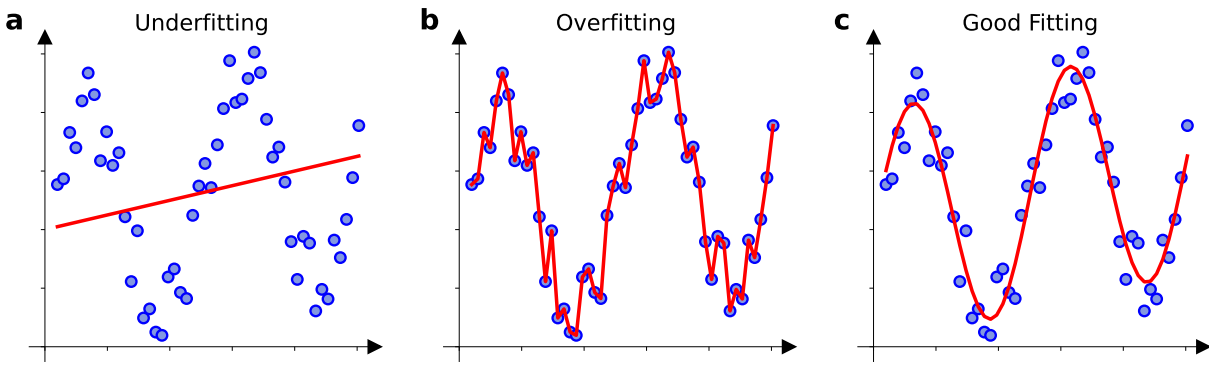

Figure 2.3: Visualization of underfitting just using a linear approximation (a), overfitting by memorizing each data point (b) and in comparison appropriate fitting using a combination of a cosine function with a linear function (c).

structures to data not used during the training progress successfully. To check the generalization of the ML model it is required to separate the original dataset $\mathcal{D}$ into two different subsets and another dataset from another source. It is noted here that this is not always possible, but the best-case scenario should be discussed here. The training set $\mathcal{D}_{\text{train}} \subset \mathcal{D}$ and the validation set $\mathcal{D}_{\text{val}} \subset \mathcal{D}$ are disjoint subsets of the original dataset $\mathcal{D}_{\text{train}} \cap \mathcal{D}_{\text{val}} = \varnothing$ and the test set $\mathcal{D}_{\text{test}}$ which in the best case stems from another source.

The difference between the loss function value of the training and the test set $g = \mathcal{L}(\mathcal{D}_{\text{train}}, \boldsymbol{\theta}) - \mathcal{L}(\mathcal{D}_{\text{test}}, \boldsymbol{\theta})$ is the so-called generalization error. Aside from obvious reasons for bad generalization, e.g. bad data sources, the large capacity of most DL models can lead to so-called overfitting. In such scenarios, the model just memorizes the complete training data, and as a result, badly generalizes. A visual idea of overfitting is given in figure 2.3b. To avoid overfitting the learning curve can be monitored. After each epoch the loss function value is calculated for the complete training and validation set. If the loss function of the training set is much smaller than the one of the validation set, the model is overfitting and training can be aborted. Notice that the validation set is used as a reference here. To decide if the model is over or underfitting some assumptions like independent and identically distributed data are required, i.e. that the samples have been drawn from the same probability distribution. The other edge case is underfitting, normally easily discovered if the loss function value is not decreased significantly. A visual concept of over- and underfitting is given in figure 2.3.

Over- or underfitting can be solved by employing regularization techniques. The most straightforward regularization is to vary the capacity of the model, i.e. adding or removing neurons in case of DNNs. However, reducing the capacity too much reduces the generalization capabilities of the model. It turned out to be more beneficial to use alternative techniques like L1 and L2 regularization which adds penalty terms to the loss functions that scale with their absolute value or their square value. Another option for DNNs are so-called dropout layers

which randomly deactivate neurons during training [115]. In this thesis, both techniques are employed.

2.6 | Neural Networks

At the forefront of achievements in AI, neural networks (NNs) continue to play a pivotal role in pushing the boundaries of what AI can achieve. They represent a fundamental paradigm in the field of artificial intelligence and machine learning, drawing inspiration from the biological neural networks found in the human brain. These computational models have gained immense popularity and significance in recent years due to their remarkable ability to solve complex tasks, ranging from image and speech recognition to natural language processing and autonomous decision-making.

At the heart of neural networks are so-called artificial neurons, which gather several inputs and weigh them to derive an output. Their concept dates back to the idea of perceptrons [116] and was implemented in the Mark I perceptron machine in 1957 [117]. Even quantum perceptrons are proposed in the early 90s [118]. Hence the concept itself is quite old. The original perceptron has binary outputs

$$y = \begin{cases} 1 & \sum_i w_i x_i + b_i \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

where w_i are the weights associated with each input x_i and b_i is a bias term. The neurons have been connected to bigger networks (inside analog computers) to mimic the decision processes in the human brain. The concept itself was forgotten for a long time until new advances in computational hardware. In modern NNs the neurons are combined in layers and their output is no longer binary and governed by activation functions shaping the nonlinear characteristics of NNs. The so-called fully connected layer function is

$$\boldsymbol{y} = g\left(\sum_i W_{i,\cdot} \cdot \boldsymbol{x} + \boldsymbol{b}\right) \tag{2.4}$$

where $\boldsymbol{y}$ is a n-dimensional vector of outputs, W is the $n \times m$ weight matrix and $\boldsymbol{x}$ is a m-dimensional vector of inputs. g is the so-called activation function. The most famous activation function is the sigmoid function

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.5}$$

which is similar to the hyperbolic tangent. Inspired by advances in neuroscience another activation function creating rectified linear units has been introduced [119]

$$\mathrm{ReLU}(x) = \max(0, x) \tag{2.6}$$

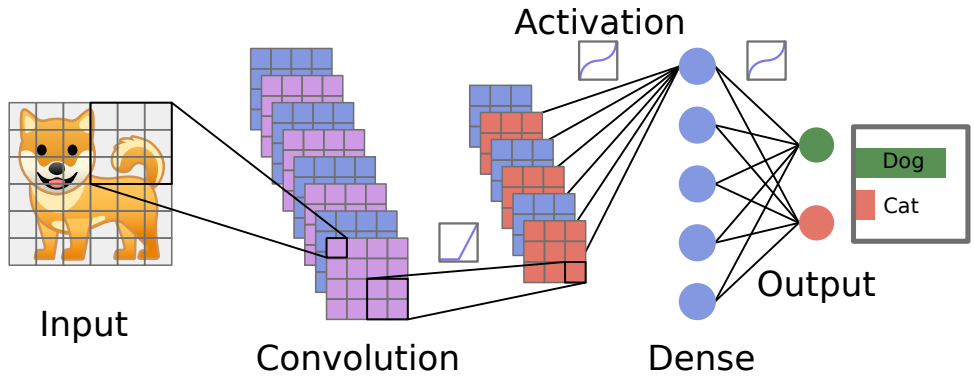

Figure 2.4: Structure of a feed-forward convolutional neural network (CNN). The input, here an image, is fed through several deep layers. Convolutions detect edges and structures and dense layers correlations of the features. All layers are connected layer by layer (feed-forward). Before the data is fed into the next layer a nonlinear activation function like ReLU or hyperbolic tangent is applied. The last layer returns a probability of the different classes or another value of interest.

and shortly after outperforms the sigmoid activation function [120]. Several other activation functions are available. However, the advantages are marginal compared to rectified linear unit (ReLU) and limited to special applications.

Employing several of such fully connected layers it is possible to build deep neural networks (DNNs). Such networks consist of an input layer and an output layer handling the inputs and outputting results and multiple so-called hidden layers in between. The input is forwarded to the first hidden layer and its outputs are forwarded to the next hidden layer until the output layer is reached. This forward passing of information raises the name of feed-forward NNs. In contrast recurrent neural networks (RNNs) [109] trained with long short-term memory (LSTM) [121] allows the output of neurons to be the input of the same neurons again. RNNs are not part of this thesis but they excel in many interesting applications like grammar learning [122], speech recognition [123], music composition [124] and are proven candidates to outperform prior ideas for NQSs [125, 126].

2.6.1 Convolutional Neural Networks

Working with image data presents special challenges for NNs. Although it is possible to process image data through fully connected layers, it is computationally expensive. An image of size 16×16 already requires 256 input neurons and an image of 265×265 with three color channels 196.608 neurons. A much better approach are convolutional neural networks (CNNs) [127] as visualized in figure 2.4. The trick of CNNs is to first detect features of the input I by multiple kernels K of the same dimensionality but usually smaller size. This can be done by applying a discrete convolution operation

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n) K(m,n) \tag{2.7}$$

where $*$ is the convolution operator and S is the output of the layer. Adding several convolutional stages allows the detection of edges and relations inside the image. Normally the found features are then correlated with multiple fully connected layers to form a CNN. The kernels K are part of the optimization parameters $\boldsymbol{\theta}$ of the NN and are optimized during the training. So far the input for the fully connected layers seems not to be reduced. This can be achieved by pooling operations like max-pooling [128] where a rectangular neighborhood of the input from the previous layer is combined to one new value with just the maximum value of the block. All ingredients together allow the creation of extremely powerful and efficient image recognition models as it is proven later for phase recognition in chapter 4.6.

2.6.2 Autoencoders

A special kind of DNNs are autoencoders (AEs) [129, 130] which can represent data in some kind of encoding with an informational bottleneck. Depending on the size of the bottleneck, autoencoders (AEs) can be sorted into three categories. Overcomplete AEs, i.e. the bottleneck is larger than the data, complete AEs, i.e. the bottleneck has the same size as the data, and undercomplete AEs, i.e. the bottleneck is smaller than the original data. Here we will focus on the latter. An AE consists of two NNs. An encoder is responsible for encoding the data into some kind of lower-dimensional representation and a decoder network is designed to reconstruct the original data from the lower-dimensional representation. A visualization of an AE is given in figure 2.5. AEs are trained by training the encoder and decoder simultaneously in an unsupervised fashion. The dataset $\mathcal{D} = \boldsymbol{x}_i$ contains just inputs that act as desired output at the same time, hence the learning scheme is unsupervised. The encoder $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ encodes the data into the lower dimensional representation $\tilde{\boldsymbol{x}}$ which afterward is decoded by the decoder $g_{\boldsymbol{\phi}}(\tilde{\boldsymbol{x}})$. Hence the complete AE is described by

$$\boldsymbol{x}_i' = g_{\boldsymbol{\phi}}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\right) \tag{2.8}$$

where $\boldsymbol{\phi}$ are the parameters of the decoder and $\boldsymbol{\theta}$ are the parameters of the encoder. After successful training of the AE, the latent space representation or bottleneck representation can be calculated by applying the encoder to the data. AEs can be employed in several ways like anomaly detection schemes [23] and dimensionality reduction schemes as introduced in section 3.3, where further unsupervised learning methods are used to identify different clusters, and for data augmentation in section 4.8. Furthermore, AEs can be useful to reduce noise [131].

Variational Autoencoders

Variational autoencoder (VAE) are a class of AE that are very different in their mathematical formulation [132–134]. Their original application is unsupervised learning approaches [135, 136]. However, they have been proven successful in semi-supervised [137, 138] and supervised

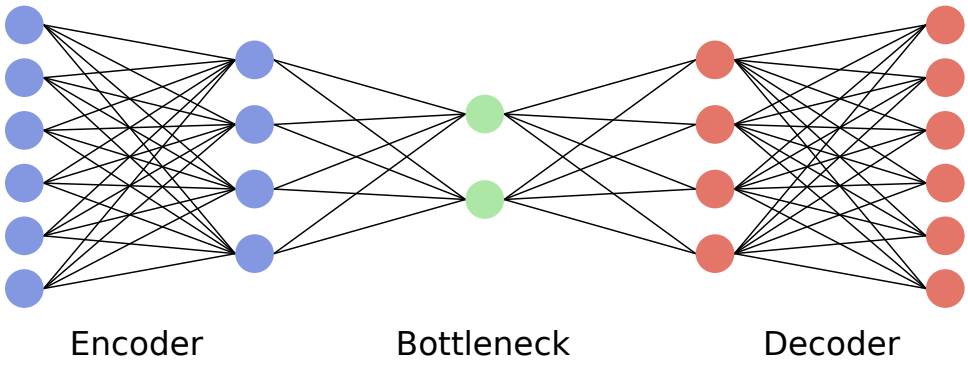

Figure 2.5: Structure of an undercomplete autoencoder (AE). An AE consists of an encoder (blue) and a decoder (red) connected with a bottleneck (green) forming a latent space.

[139] learning schemes. Variational autoencoders (VAEs) can be interpreted as normal AEs with a special regularization term to avoid overfitting.

The encoder network defines an approximate posterior probability distribution $q_\phi(\tilde{x}|x)$ which takes the inputs and outputs the parameters to define a conditional probability distribution for the latent representation $\tilde{x}$. For better clarity, the distribution here is Gaussian and represented by the mean and the log-variance for numerical stability. The job of the decoder network is to represent the outputs by defining another conditional probability distribution $p_\theta(x|\tilde{x})$ which models how likely it is to get x given $\tilde{x}$. Notice that here the input and output of the VAE are the same. Later we will see that this is not necessary in all cases and it might be helpful to train with different inputs and outputs. To actually train such models, the so-called reparametrization trick [132, 140] has to be employed. To sample from the distribution $q_\phi(\tilde{x}|x)$ random numbers ϵ_i are injected into the bottleneck of the AE. The structure is visualized in figure 2.6. Due to the specialties of VAEs the loss function needs to respect the character of the probability distribution. The trick is maximizing the evidence lower bound (ELBO). The overall idea is to optimize the decoder parameters θ to get low reconstruction error between the input and output, i.e. measured by MSE, and at the same time optimize the encoder parameters ϕ to make $q_\phi(\tilde{x}|x)$ as close as possible to $p_\theta(\tilde{x}|x)$. To model this properly without unnecessary assumptions expectation $\mathbb{E}$ needs to be introduced. The expectation of a function $f(x)$ concerning a probability distribution $p(x)$ is the average value that $f(x)$ takes when x is drawn from the probability distribution $p(x)$ which is calculated for continuous variables using the integral

$$\mathbb{E}_{x\sim p(x)}\left[f(x)\right] = \int dx p(x) f(x) \tag{2.9}$$

and a sum for discrete values

$$\mathbb{E}_{x\sim p(x)}\left[f(x)\right] = \sum_x p(x) f(x). \tag{2.10}$$

Notice that expectations are linear as the integral and sum are linear. With expectation at hand, the distance between two probability distributions, e.g. the two distributions used for

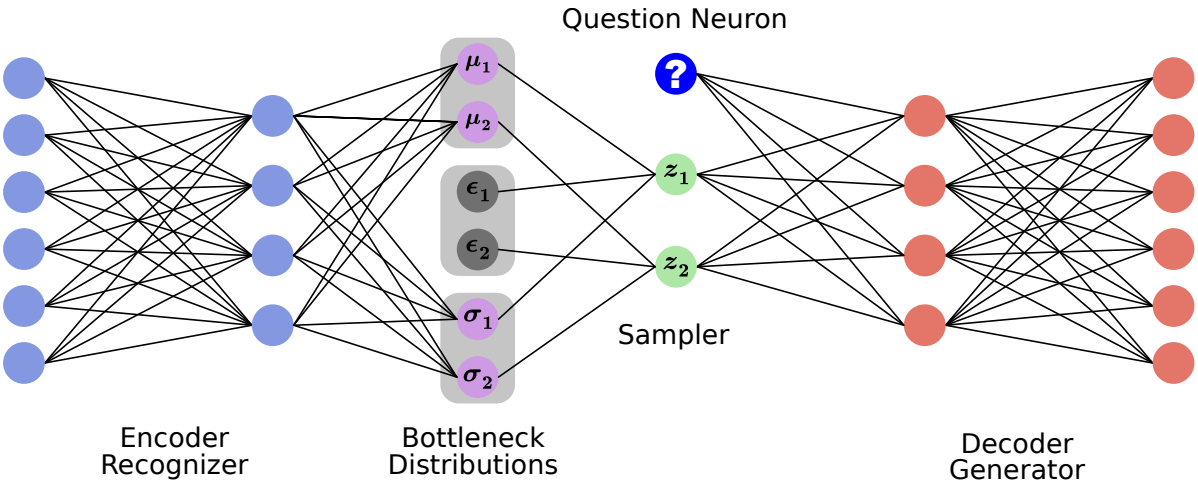

Figure 2.6: Structure of a variational autoencoder (VAE). The input is encoded by a recognizer model (light blue), deriving probability distributions with a mean value μ_i and variance σ_i (violet). The sampling of the distribution is done by sampling neurons z_i (green). Sampling is only possible by the reparameterization trick injecting the random number ϵ_i. An additional optional question neuron (blue) can be used to morph the data by a desired parameter. As a last step, a generator model decodes the data (red).

the VAE, is given by the Kullback-Leibler divergence

$$D_{KL}(q_\phi(\cdot|x) \parallel p_\theta(\cdot|x)) = \mathbb{E}_{\tilde{x}\sim q_\phi(\cdot|x)}\left[\log \frac{q_\phi(\tilde{x}|x)}{p_\theta(\tilde{x}|x)}\right] \tag{2.11}$$

$$= \mathbb{E}_{\tilde{x}\sim q_\phi(\cdot|x)}\left[\log \frac{q_\phi(\tilde{x}|x)p_\theta(x)}{p_\theta(x,\tilde{x})}\right] \tag{2.12}$$

$$= \log p_\theta(x) + \mathbb{E}_{\tilde{x}\sim q_\phi(\cdot|x)}\left[\log \frac{q_\phi(\tilde{x}|x)}{p_\theta(x,\tilde{x})}\right] \tag{2.13}$$

where from the first to the second line the definition of conditional probability is used and from the second to third line logarithmic laws. Using this definition the ELBO loss can be defined

$$\mathcal{L}_{\text{ELBO}}(x;\theta,\phi) = \mathbb{E}_{\tilde{x}\sim q_\phi(\cdot|x)}\left[\log \frac{q_\phi(\tilde{x}|x)}{p_\theta(x,\tilde{x})}\right] = D_{KL}(q_\phi(\cdot|x) \parallel p_\theta(\cdot|x)) - \log p_\theta(x). \tag{2.14}$$

Notice that the ELBO is the negative ELBO loss. However, we stick to the narrative of minimizing the loss function and define it correspondingly here. In practice, the probabilities are estimated with a single-sample Monte Carlo approach, which is proven sufficient. In figure 2.6 an additional question neuron is added to the VAE. So far only VAEs with the same input and output are considered. However VAEs are very powerful morphing properties of the data because their latent representation is smooth. Adding additional question neurons allows for morphing data properties. In this case, the VAE is trained with a dataset of inputs, desired outputs, and their morph value, i.e. the color, or noise filter strength. It is also helpful to morph dominant features in experimental data as shown in chapter 4.8.

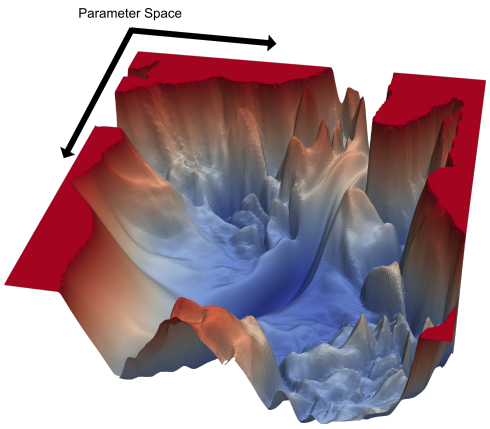

Figure 2.7: Low dimensional visualization of the loss landscape of ResNet-56. The loss landscape is non-convex and many local minima exist. Adapted from [145]

2.7 | Loss Landscape

To learn more about the black box of NN described so far it is helpful to investigate the highly non convec loss landscape further. The optimization of a NN challenges several difficulties. As shown in figure 2.7 a variety of different minima exist. Of course, the primary goal of the optimization is to find the minimum with the best generalization, however not all minima generalize at the same level. Generally, it is understood that flat and wide minima generalize better than sharp minima [141–143]. However, a developed concept of flatness in such highly dimensional landscapes does not exist [144]. Nevertheless, it makes sense to explore the surroundings of the local minimum and derive insights regarding the generalization capabilities of the trained NN. A helpful measure is the curvature of the loss landscape as introduced in the following section.

2.7.1 Curvature of the Loss

To investigate the environment around a found minimum of the loss function after training a NN, the Hessian of the loss can be calculated. It allows an analysis of the curvature of the loss landscape. The Hessian $H_{\tilde{\boldsymbol{\theta}}}$ of the loss function $\mathcal{L}$ after training is given by

$$H_{\tilde{\boldsymbol{\theta}},ij} = \frac{\partial^2}{\partial_{\theta_i}\partial_{\theta_j}}\mathcal{L}\mid_{\boldsymbol{\theta}=\tilde{\boldsymbol{\theta}}} \tag{2.15}$$

where $\tilde{\boldsymbol{\theta}}$ are the model parameter at the end of training. The largest positive eigenvalues and their corresponding eigenvectors of $H_{\tilde{\boldsymbol{\theta}}}$ correspond to the direction of the steepest ascent around the found minimum and vice versa.

As mentioned earlier, the parameters found in DL by training with SGD or other optimization algorithms do not necessarily correspond to the global minimum. Interestingly in most cases,

the training converges at local minima or saddle points [146–148]. Studying the loss landscape with the Hessian unveils more properties by looking at the spectrum. For large deep networks, the majority of eigenvalues is almost zero, indicating flat directions in the loss landscape. Even worse, some eigenvalues are negative, indicating negative curvature. This could be worrying since no global minimum nor a very steep minimum is found. However, studies involving spin-glass theory [149] show, that for larger networks most minima are equivalent and show similar performance regarding generalization, i.e. a low loss value for the test set. In practice, it is not beneficial to find the global minimum as it may lead to overfitting.

The investigation of curvature also allows for further insights into the influence of training data on the outcome. In the direction of high curvature, which corresponds to the eigenvectors associated with the largest eigenvalues, the training data has the greatest impact on the model parameters. This is immediately evident since a significant change in the loss function occurs along these directions. Furthermore, there are empirical studies that support this hypothesis. An analysis of the spectrum of the Hessian reveals that in classification problems, the number of directions with an increase equals the minimum of the number of classes reduced by one [147, 150, 151]. Additionally, gradients of the single sample loss function of training examples correspond to the direction of the highest curvature in nonlinear regression problems [152].

2.7.2 Similarity Measure and Influence Function

As discussed there is a strong correlation between the curvature of the loss landscape, the ability to generalize, and the training data. To understand the interplay better, the influence of the training data itself on the loss landscape can be studied. To do so leave one out (LOO) training is employed. One training point $\boldsymbol{x}_r$ of the dataset $\mathcal{D}$ is removed and the neural network is trained again. If the loss of the test set after training stays the same the removed point $\boldsymbol{z}_r$ was not helpful, i.e. not influential. If the training loss is higher, it was influential and helpful for the generalization of the model.

Unfortunately, retraining a large DNN model is very costly. Hence an approximation of the change of the loss landscape is helpful. This is achieved by employing the influence function $\mathcal{I}$ [153, 154]. The influence function measures the loss change for a given test point $\boldsymbol{x}_t$ if the point $\boldsymbol{x}_r$ is removed from the training set.

$$\mathcal{I}(\boldsymbol{x}_r, \boldsymbol{x}_t) = \frac{1}{N_{\text{data}}} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}_t, \tilde{\boldsymbol{\theta}})^T H_{\tilde{\boldsymbol{\theta}}}^{-1} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}_r, \tilde{\boldsymbol{\theta}}) \equiv \frac{1}{N_{\text{data}}} \nabla \mathcal{L}_t^T H_{\tilde{\boldsymbol{\theta}}}^{-1} \nabla \mathcal{L}_r \tag{2.16}$$

Notice that here N_{data} is the number of points in the original dataset $\mathcal{D}$. The loss function is now the loss for a single example from the dataset $\mathcal{L}_t = \mathcal{L}(\boldsymbol{x}_t, \tilde{\boldsymbol{\theta}})$. Notice that the influence function is an inner product between $\nabla \mathcal{L}_t$ and $H_\theta^{-1} \nabla \mathcal{L}_r$ [155] where $H_\theta^{-1} \nabla \mathcal{L}_r$ is the approximation of the parameter change from the removed data sample taking into account

the curvature of the loss landscape. This can be understood as a single Newton step towards the new minimum of the loss by training without $\boldsymbol{x}_r$.

Consequently, the influence function can be used to calculate the similarity between two samples from the dataset. If the removed sample from the training data and the test sample are very similar, i.e., similar to the underlying model, the influence function is high. This is caused by the alignment of $\nabla\mathcal{L}_t$ and $\nabla\mathcal{L}_r$ with a correction by the local curvature in the loss landscape. With these considerations, the similarity between two points from the dataset can be defined [152]

$$S(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\nabla\mathcal{L}_{\mathrm{t}}^T H_{\tilde{\boldsymbol{\theta}}}^{-1} \nabla\mathcal{L}_{\mathrm{r}}\right)^2 \propto \mathcal{I}(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 \tag{2.17}$$

which is advantageous for unsupervised ML approaches as described in section 3.3.

In a nutshell, the influence function is a tool to avoid retraining large DNNs by approximating the influence on the loss landscape using the local curvature. Furthermore, it helps to understand the relations between different data points from the perspective of the ML model, which can be used to identify faulty data or the right choice of training data.

2.8 | Practical Considerations

ML techniques and especially DNNs can be very large consisting of millions of parameters and thus are computationally costly. It is necessary to employ highly optimized algorithms to efficiently evaluate the value of the loss function and the gradient. Luckily there exist many libraries that have already implemented all the algorithms needed. In this thesis, Tensorflow [156], PyTorch [157] and Jax [158] are the main drivers behind the ML models. Depending on the size of the models it might be beneficial to use specialized processing units like GPUs to speed up linear algebra operations. However, they have to be used wisely since their natural main memory access bottleneck may lead to lower performance compared to CPUs. It is recommended to use performance metrics to analyze the performance to avoid costly operations. With the frameworks at hand, the implementation of neural networks is relatively easy, the design and tuning on the other hand needs a lot of experience and patience.

As ML and especially unsupervised ML with DNNs are data-driven approaches to problems, they highly relate to the data quality and quantity, i.e. if the data is bad it is not expected that ML will learn how to identify or separate it. Having that in mind it is good advice to always consider improving the data quality before trying to fix problems by applying ML algorithms. Sometimes this is not possible, as posed in section 4.7.2 but if it is, it should be considered first.

CHAPTER 3

Detection of Phase Transitions

The exploration of phase transitions has been a cornerstone in the understanding of the physical properties of matter, dating back to the early days of classical physics. As scientific inquiry progressed, the advent of quantum mechanics revolutionized our understanding of the microscopic world, leading to the emergence of quantum phase transitions (QPTs) as a distinct and captivating field of study. Detecting quantum phase transitions poses unique challenges compared to their classical counterparts due to the quantum nature of the systems involved. Normally modern QPT detection schemes employ so-called order parameters which are zero for one phase and finite for other phases [159]. Additionally, several experimental techniques have been developed to detect and characterize quantum phase transitions i.e. for the Haldane phase diagram realized with cold atoms [72, 87, 89, 90].

Both supervised and unsupervised machine learning techniques as introduced in 2.3 are instrumental to identify or pinpoint QPTs. Both have been applied to several different quantum systems and have been proven successful in fulfilling that task for numerical as well as experimental data [29, 31, 32, 34, 35, 40, 43, 160–171]. Notice that for most of the methods described here, there is no evidence that the return always corresponds to a QPTs. Nevertheless, a noticeable change in the underlying quantum mechanical state can be expected.

in this chapter, I will introduce how to reduce the dimensionality of data using ML techniques 3.1 to prepare it for clustering methods introduced in chapter 3.3.1. I will furthermore introduce the concepts of anomaly detection to differentiate different phases of matter in section 3.3.2.

3.1 | Dimensionality Reduction Methods

To detect different phases of matter from high dimensional data like ToF images from a cold quantum gas experiment, most methods require a dimensionality reduction applied to the data before analysis because the image data itself is too complex to identify. Dimensionality reduction is a lively field in ML and mathematics that tries to reduce highly dimensional data

like images, time series, and encoded text into a lower dimensional space ideally reducing the dimensions to represent the data to its intrinsic dimension I_d. The concept of the intrinsic dimensions arises from the correlations in natural data. Assuming a data set of pictures of circles with different radii. The data itself is highly dimensional, given the number of image pixels. Nevertheless, the underlying data itself lies in a one-dimensional manifold, the radius. The intrinsic dimension of a dataset can be estimated or in some cases analytically calculated and roughly corresponds to the minimum number of variables needed to describe a data set [172, 173]. The applications of finding intrinsic dimensions and using the extracted datasets range from protein sequence evolution [174], analysis of astrological spectrometric data [175], to analysis the functional variation from different plants [176] and the critical behavior in physical systems [177].

This thesis focuses on PCA as a linear dimensionality reduction method and the bottleneck analysis using AEs as a non-linear dimensionality reduction method. The techniques used here do not rely on finding the number of intrinsic dimensions but rather use the dimensionality-reduced data to find clusters or patterns that distinguish the different phases of matter. The lower dimensional representation of the data can be directly separated by direct detection or by the automated k-means clustering algorithm where different clusters can be associated with different phases of matter.

3.1.1 Principle Component Analysis

The most straightforward dimensionality reduction technique is the well-known principle component analysis (PCA) [96, 97]. This technique has been proven successful in one of the first publications to identify different phases of matter in an unsupervised manner [22]. More generally speaking PCA is in its bare form a linear dimensionality reduction technique commonly used in machine learning and data analysis. Its goal is to transform the original correlated features of a dataset into a new set of uncorrelated features called principal components, ordered by the amount of variance they capture. This reduces the dimensionality of the data while retaining as much of the original information as possible. Since PCA does not involve NNs it can be considered to be one of the most interpretable machine learning techniques. It is worth mentioning that even if the origins of PCA date back more than one hundred years it is an ongoing research topic and some of the ideas have influenced the choice of the PCA algorithms used in this thesis. Ongoing research includes non-linear PCAs as well as handling large data sets and detecting the number of relevant features [178–181].

PCA is sensitive to the scale of the features, hence it is common to standardize the data by subtracting the mean and dividing by the standard deviation for each feature. A feature could be a single correlation measurement or a site occupation in the context of physics. Assuming that the features of each sample are reshaped into a X_{ij} matrix which we will call a feature

matrix with i features and j samples. The standardization of data can be written as

$$Z_{ij} = \frac{X_{ij} - \mu_i}{\sigma_i} \tag{3.1}$$

where Z_{ij} are the standardized features, μ_i is the mean value and σ_i is the variance of the ith feature. It should be noted that depending on the underlying data this step might not be necessary. However, for image data, it is highly recommended. To find features that correlate most the covariance matrix C is calculated

$$C_{ik} = \mathrm{cov}(Z_i, Z_k) = \frac{1}{n-1} \sum_{j=1}^{n} (Z_{ij} - \bar{Z}_i)(Z_{kj} - \bar{Z}_k) \tag{3.2}$$

where $\bar{Z}_i$ is the mean of the standardized values for feature i and n is the number of samples. To find the shared directions of maximum variance the covariance matrix C is decomposed into its diagonal eigenvalue matrix Λ and its matrix of eigenvectors V

$$C = V \Lambda V^T \tag{3.3}$$

using a suitable singular value decomposition (SVD) algorithm. Notice that C is not required to be a square matrix. The most straightforward way to select the most relevant eigenvectors is by sorting them by the magnitude of the eigenvalues. If a dimensionality reduction to k features is desired the largest k eigenvalues and their corresponding eigenvectors form the direction matrix W. As a final step, each input data can be transformed into a lower dimensional representation

$$X_{\mathrm{new}} = Z \cdot W \tag{3.4}$$

where X_{new} holds the representations of each sample in terms of the selected k principal components. The so-called principal components are the directions in the original feature space along which the data varies the most.

3.1.2 Bottleneck Analysis

Another common method to reduce the dimensionality of the data is the bottleneck analysis. An undercomplete AE, i.e. the bottleneck has fewer dimensions than the input, is trained on the complete dataset as introduced in section 2.6.2. The choice of the AE is not important however AEs with shortcut connections [182] have been proven successful for small datasets. After training the complete AE the encoder part is used to calculate the lower dimensional representation of the data inside the bottleneck. Recall that an AE can be modeled by two NNs represented by highly non-linear functions. The encoder function $f_\theta(x)$ and the decoder function $g_\phi(y)$.

Bottleneck analysis assumes that the most dominant features of the data are central to the phase information. Since AEs compress the data and filter the most important information. Hence the information about the underlying phase is represented in the bottleneck of the AE.

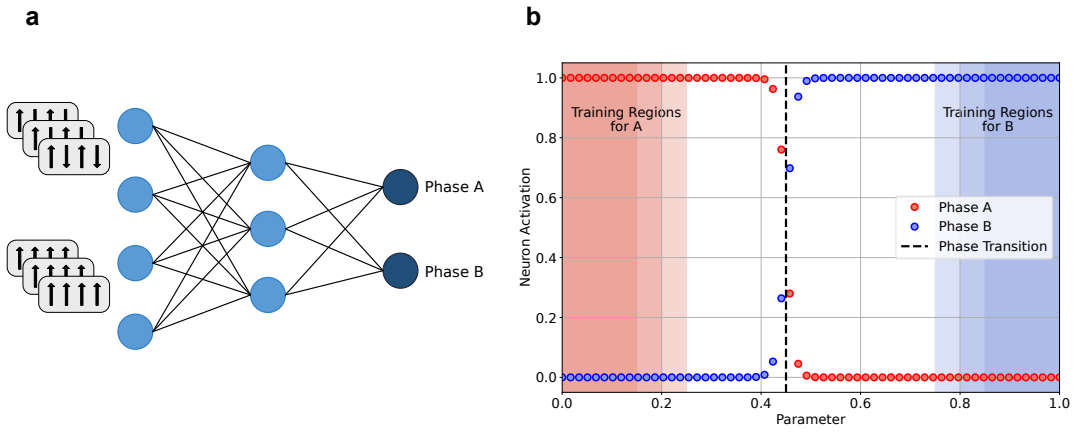

Figure 3.1: Supervised phase detection scheme. (a) Sampled data from the experiment or numerical calculations to train the NN to distinguish the two phases A and B i.e. spin configurations. The NN consists of multiple layers of neurons and or other building blocks like convolutions. (b) The two neurons can be mapped to a probability of being in phase A or phase B. At the point where the probabilities change from 1 to 0 or vice versa it is likely to have a phase transition. The shaded areas are the different training regions.

3.2 | Supervised Phase Boundary Detection

Supervised learning, 2.3.1, can be used to pinpoint phase transitions if the study of order parameters near the phase transition is not available. It already has proven successful for different kinds of physical systems [28–30, 34] and it is proven successful in section 4.6. We should recall from 2.1 that when using representation learning techniques we generally do not need to select helpful features from the dataset. Two datasets are required to run the phase classification. The training dataset $\mathcal{D}_{\text{train}}$, which can also be used as a source for the validation data, consists of measurements within regions we are confidently able to assign to the different phases. We label them according to different classes, i.e. that $\mathcal{D}_{\text{train}}$ includes samples containing the measured observable x and the information about the corresponding class c. These tuples $(x, c) \in \mathcal{D}_{\text{train}}$ are now used to train a classifier. Depending on the type of the data we can choose a suitable machine learning model. This could be support vector machines, random forests, deep neural networks, or other specialized models. The model selection itself is most of the time only driven by the type of data and not by the phase transition itself. Best performance is achieved by choosing DNNs because they generalize well. The second dataset includes measurements from the region we want to study. Normally that is the region around the expected phase transition. To check the performance and generalization of the model it is helpful to separate some of the training data as a validation set to crosscheck that the model is not overfitting. The analysis dataset $\mathcal{D}_{\text{analysis}}$ is afterward fed into the classifier which returns associated labels to the given observations.

In figure 3.1 the process of supervised learning is sketched. The classifier, here a DNN is trained with spin samples of two different phases of matter. Afterward, it is applied to the data between the two training regions and reveals a phase transition. Notice that the output of the two neurons for phase A and phase B are associated with probabilities in 3.1b because a softmax function is applied at the output layer. This allows us to estimate the width of the phase transition.

It is crucial to check if the choice of the training regions influences the position of the phase transition. Therefore it is helpful to choose different training regions symmetrically and asymmetric around the expected phase transition and validate the stability of the predicted transition. To extract the position of the phase transition several methods can be applied. Either a threshold value is set for the activation of the output neurons to decide if the system is in one of the phases or two hyperbolic tangents can be fit and their crossing point is the expected transition point.

3.3 | Unsupervised Phase Boundary Detection

Very often the underlying phases of matter are unknown so supervised learning is not feasible. In such cases, unsupervised ML can help to identify different phases of matter or at least guide to the interesting regions of the phase diagram. Unsupervised phase diagram discovery is a key ingredient for autonomous scientific discovery where AI augments, guides, and controls the experimental platform and autonomously discovers new phases of matter [183] in an efficient fashion. In that spirit, efficient means with as little data as needed.

Classically the study of phase transition is performed by studying a given model extensively and deriving observables that can be orchestrated to order parameters and are feasible for experiments. The idea of unsupervised ML approaches is to use accessible data and use general measuring protocols to avoid complex measurement protocols, that might not be feasible for all required regions of the phase diagram. As already discussed in section 2.3.2, the idea is to find patterns and representations of data without engineering special features.

The unsupervised ML approaches can be loosely sorted into two categories. Cluster-based ML approaches [22, 33, 169, 170, 184–188] and learning success-based methods [23, 24, 32, 34, 189, 190].

3.3.1 Cluster Based Approaches

At the heart of cluster-based approaches is the idea of separating the data into different clusters where each cluster represents one phase of the physical system. As cluster algorithms like k-means are not capable of successfully finding clusters from noisy experimental data or

very high dimensional numerical data it is required to use dimensionality reduction techniques introduced in 3.1 such as AEs, diffusion maps, t-distributed stochastic neighbour embedding (t-SNE) or PCA as a first step. The lower dimensional data can afterward be fed into a clustering algorithm.

K-means Clustering

K-means clustering [191] is a widely-used method in data analysis for partitioning a dataset into a set of k groups, or clusters, based on the features of the data. The primary goal is to divide the data points into clusters in which each point belongs to the cluster with the nearest mean value, thereby minimizing the intra-cluster variance.

The algorithm first randomly initializes k centroids, i.e. centers of the clusters, somewhere in the feature space of the data with the coordinates $\boldsymbol{\mu}_i$ for the i-th centroid. Afterward, each sample from the data represented by the coordinates $\boldsymbol{x}_i$ for the i-th sample is assigned to the nearest centroids. This is normally done by using the Euclidean metric, however, multiple other metrics have been proven successful. Afterward, each centroid is updated in each dimension with the mean value of all points within the cluster and the algorithm assigns all samples to the new centroids. A schematic overview can be found in algorithm 2. The objective of K-means is to minimize the sum of squared distances between each data point and its corresponding centroid. This is known as the inertia or within-cluster sum of squares (WCSS) and is given by

$$\mathcal{L}_{\text{WCSS}} = \sum_{j=1}^{k} \sum_{x_i \in C_j} ||x_i - \mu_j||^2. \tag{3.5}$$

Algorithm 2 K-Means Clustering

Input: Datapoints: $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^{N_{\text{data}}}$
Input: Number of clusters: k, Stop criterion: ϵ
Output: Assigned Clusters: $\mathcal{C} = C_{i\,i=1}^{k}$
  Randomly initialize centroids $\boldsymbol{\mu}_i$
  while $\mathcal{L}_{\text{WCSS}} > \epsilon$ **do**
    Assign each $\boldsymbol{x}_i$ to the nearest centroid C_i
    $\mathcal{L}_{\text{WCSS}} \leftarrow \sum_{j=1}^{k} \sum_{x_i \in C_j} ||x_i - \mu_j||^2$
    for each centroid C_i in $\mathcal{C}$ **do**
      $\mu_i \leftarrow \frac{1}{||C_i||} \sum_{x_j \in C_i} x_j$
    end for
  end while

Determining the right number of clusters k is a crucial part of k-means clustering. Several methods, like the Elbow Method [192] are used for this. Notice that the method of choice

is currently questioned [193] and more advanced methods for determining the number of clusters are proposed. Nevertheless, the elbow method is used here due to the actual parallel development and discussion of the new methods, which is beyond the scope of this thesis. The elbow method involves plotting the $\mathcal{L}_{\mathrm{WCSS}}$ against different values of k and looking for an elbow point where the rate of decrease sharply changes. This method is applied in section 4.10 to determine the correct number of clusters. It should be mentioned that k-means clustering does not work well for non-convex clusters or data with varying densities and it is sensitive to the initial choice of the centroids.

3.3.2 Learning Success Based Approaches

The other class of unsupervised ML approaches is success-based, i.e. how well or bad the model can learn given features or in general minimize a certain loss function. One of the first is the learning by confusion scheme [32] where a binary classifier, i.e. a DNN with two outputs is trained with data from different imposed phase transition points, trying to discriminate the two phases. When the learning is successful, i.e. a convergence of the loss can be observed, one can consider a phase transition. This method is appealing due to its straightforward idea, however, could not be proven successful for experimental data, and further methods have to be employed.

Anomaly Detection

Using machine learning for anomaly detection is very common in many industry areas ranging from credit card fraud detection to cybersecurity and machine failure prediction and many more [194–196]. The common goal is to identify anomalous behavior in unstructured data which does not reveal its nature by straight forward rule-based programming. The progress is similar in all application cases. Non-anomalous data is used to train a machine learning algorithm. In the course of this thesis, this will be an AE reproducing the same output as it gets as an input. Applying the AE after training to new unseen data during the training process the loss function, i.e. the ability of the AE to encode and decode the given unseen sample, holds information about the similarity of the new data compared to the data in the training set. Whenever the loss function is much larger than the average loss function of the validation set the data is anomalous as the data does not belong to the same subset of data the AE has seen during training. This method can directly be transferred to the idea of phase detection. Training such AE to auto-encode samples from an experiment or a numerical calculation within a limited parameter range of one phase, it is possible to detect different phases of matter. This has been proven successfully [197–199] and also leads to the detection of completely unknown phases before this kind of technology [23] existed. A pictorial scheme is shown in figure 3.2. The AE is trained in the region of phase A. Notice that it is not

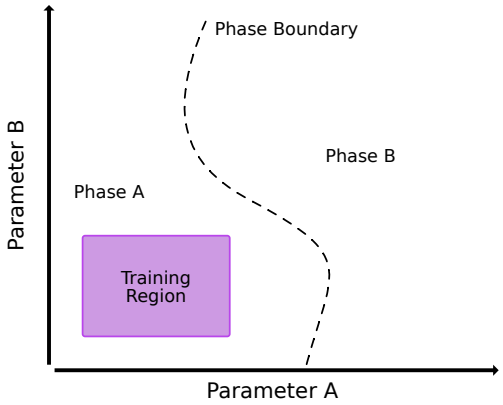

Figure 3.2: Anomaly detection scheme. A two-dimensional parameter space of a physical system is divided into two different phases. The AE is trained inside the purple region which is a subset of phase A. To detect the phase boundary the loss is calculated for all samples within parameter space. At the phase boundary, an anomaly is detected by an increase in the loss.

required that the training region covers the complete region. Afterward, reconstruction loss, i.e. the difference between the input and output data of the AE, is calculated for the data of the complete phase diagram. If the AE can reconstruct the data, i.e. the loss function for the single sample is at the same level as in the training region, the sample corresponds to the same class. For phase B the loss increases as the AE is not able to reconstruct samples from this area. The phase boundary can be then extracted by either setting a threshold value for the loss function or fitting a hyperbolic tangent as we will see in chapter 4.7.4. Similar to the training region variations for supervised phase detection in section 3.2 it is required to vary the training regions of the AE and crosscheck the stability of the extracted phase boundary.

Similarity Analysis

An alternative method of choice is the similarity analysis guided by the idea of influence function introduced in chapter 2.7.2. It has been proven successful in identifying different phases of matter and will be used to separate topological nontrivial classes in chapter 4.7.5. Recall that influence functions allow us to define a similarity measure between samples from a dataset in terms of a trained NN. Hence we use the feature space of the neural network to measure the distance between samples from the data. Notice that the similarity is not a metric in a mathematical sense. As the calculation of the influence function requires the supervised training of a classifier NN it is not completely unsupervised and can be considered a semi-supervised technique. Nevertheless, it can be used to verify classifications by cluster-based approaches from chapter 3.3.1 and anomaly detection.

First, a supervised classifier is trained similarly to the approach in 3.2 using the phase boundaries extracted by the previous methods. To now calculate the similarity of the underlying dataset a random test point is selected from one of the phases, e.g. detected by anomaly detection, and the similarity is calculated. It is expected that samples within the same class are similar to each other, i.e. their similarity value is approximately equal. If the phases discovered by anomaly detection or clustering methods are incomplete a change in similarity is expected.

CHAPTER 4

Haldane model

In this section of the thesis I will present the results of a collaborative project that has been published [43] together with Anna Dawid, Korbinian Kottmann, Maceij Lewenstein, Klaus Sengstock, Alexandre Dauphin and Christof Weitenberg. The experimental data used to produce the presented results were taken by Benno Rem, Luca Asteria and Matthias Tarnowski under the supervision of Christof Weitenberg and Klaus Sengstock at the Bose-fermi-mixture (BFM) experiment. The full implementation details are available in the publication notebooks [200].

Topology and quantum physics have been irretrievably linked in the early 1980s when Thouless et al. explained [84] in 1982 the integer quantum Hall effect [83]. The quantum Hall effect was measured in 1980 and allows high precision measurement of the fine structure constant. A short time later Michael Berry published his idea of a gauge-independent phase - which today is called Berry phase - in addition to the dynamical phase factor acquired by adiabatically transporting a quantum system on a closed loop in parameter space [201]. The link between topology and quantum systems was awarded the Nobel prize in 2016. Topology then gained an important role in many quantum physics experiments with cold atoms and has been extensively studied by many groups around the world [202].

In chapter 4.1 I will introduce the basic concepts of topology and its application to quantum physics. The concept is then applied to Bloch bands in chapter 4.2 and further extended to the actual physical implementation in Boron-Nitride lattices in chapter 4.3 before finally introducing the experimental protocol in chapter 4.4.

Topological properties like the Chern number have also been studied with other measurement protocols and I will mention a few in chapter 4.5. Afterward, I will demonstrate that supervised learning in chapter 4.6 and unsupervised learning in chapter 4.7 is capable of reconstructing the complete Haldane phase diagram from experimental data employing ML techniques like AEs for bottleneck analysis, anomaly detection, and influence functions.

4.1 | Topological Phases

One of the most famous examples to describe topology is the comparison between a doughnut and a coffee mug which belong to the same topological class and a sphere which does belong to another class. This seems non-intuitive at first glance but can be directly linked to properties of quantum physics. To understand the mathematical categorization of these objects one has to define the genus, sorting objects into different topological classes based on their topological properties. A straightforward concept of a topological property is the Euler-Poincaré characteristic which can be used to identify any geometrical object with a closed surface. The Euler-Poincaré characteristic is defined as

$$\chi(A) = \int_A G \mathrm{d}A \tag{4.1}$$

where A is a closed surface of a geometrical object and G is the Gaussian curvature of the surface. As long as the surface A is closed and smooth the value of $\chi \in \mathbb{N}$ is strictly integer-valued. The value is straightforward, however, it is much more convenient to define the genus of an object

$$g = \frac{1}{2}(\chi - 2) \tag{4.2}$$

which integer value is the number of holes of three-dimensional objects in the Euclidean world. This can be illustrated by the idea that forming holes enforces positive and negative values of the Gaussian curvature canceling each other, while a sphere, i.e. the surface of an object without any holes, adds up to a positive value of the Euler-Poincaré characteristic of 2. Notice that two objects are considered to be topological equivalent if the Euler-Poincaré characteristic is the same. Due to the properties of the integral and the Gaussian curvature smooth transformation of the object surface A does not change the value of the integral in equation 4.1. Hence the reason why a doughnut and a coffee mug are considered topologically equivalent results from the fact that both surfaces have the same number of holes and can be transformed smoothly into each other.

As we have seen curvature plays an important role to the topological identification of different geometrical objects. Another way to interpret the importance of curvature is the so-called Levi-Civita transport. First a vector $\boldsymbol{R}$ can be defined which is at each point orthogonal to the normal vector of the surface $\boldsymbol{R} \cdot \hat{\boldsymbol{n}} = 0$ and does not rotate around the normal vector $\mathrm{d}\boldsymbol{R} \cdot \hat{\boldsymbol{n}} = 0$. Notice that the normal vector itself is directly linked to the gradient $\hat{\boldsymbol{n}} = \nabla F(x, y, z)$ if the surface is given implicitly by points (x, y, z) with F satisfying $F(x, y, z) = 0$. Moving the vector $\boldsymbol{R}$ around on a closed loop on the surface of a sphere results in a phase, i.e. an angle α between the original orientation of $\boldsymbol{R}$ and the final orientation of $\boldsymbol{R}$, whenever curvature is present (see figure 4.1) in contrast to no surface curvature. Notice that when moving

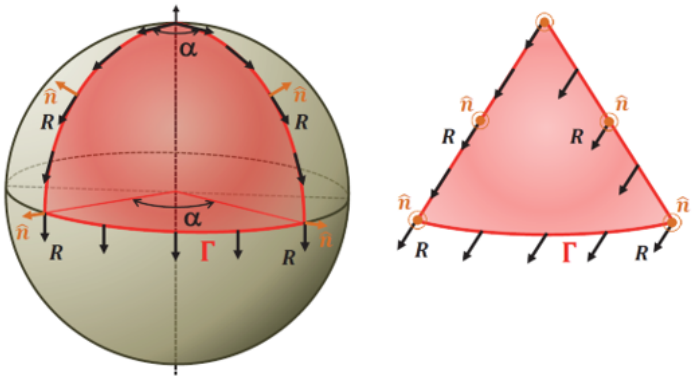

Figure 4.1: Visualization of the Levi-Civita transport on a surface (left) and on a flat cut without curvature (right). In the case of the sphere surface, a phase α is acquired moving $\boldsymbol{R}$ along the path. This graphic originates from [203].

the same vector on the unfolded flattened triangular-shaped part of the sphere, no phase is present, i.e. the vector points in the same direction.

The concept of the acquired phase can be projected onto the idea of quantum states. The existence of such phases was formalized by Michael Berry [201] and the existence of the Berry phase can be derived directly from the Schrödinger equation and the adiabatic theorem. Considering the time-independent Schrödinger equation

$$\hat{H}(\lambda(t))\,|n(\lambda(t))\rangle = E_n(\lambda(t))\,|n(\lambda(t))\rangle \tag{4.3}$$

where $\hat{H}$ is the Hamiltonian, governing the quantum system by the time-dependent parameter $\lambda(t)$ and $|n(\lambda)\rangle$ is the corresponding state associated with the energy spectrum $E_n(\lambda)$. The adiabatic theorem assumes that the quantum system state starts in an eigenstate of the Hamiltonian $|\psi(0)\rangle = |n(\lambda(0))\rangle$ and stays in an eigenstate if the parameters change slow enough in time. Nonetheless, the system can acquire a phase ϕ_n. The phase can be split into two parts $\phi_n = \delta_n + \beta_n$ where δ_n is the dynamic phase and β_n is the so-called geometrical phase or Berry phase. The Berry phase is the quantum physics pendant to the phase acquired during the Levi-Civita transport. Using this analogy one can derive the berry phase

$$\beta_n = i \oint_C \boldsymbol{A}(\lambda)\mathrm{d}\lambda \tag{4.4}$$

which is real-valued and gauge invariant and thus experimentally observable. Its definition is based on the so-called Berry connection

$$\boldsymbol{A}(\lambda) = \langle n(\lambda)|\nabla n(\lambda)\rangle \tag{4.5}$$

which is loosely analog to the vector $\boldsymbol{R}$ in the picture of Levi-Civita transport. Notice that the Hilbert space is complex-valued, hence a direct analog Euclidean interpretation is not possible.

Using Stokes theorem another quantity of interest can be derived

$$\beta_n = i \oint_C \boldsymbol{A} \mathrm{d}\lambda = \mathrm{Im}\left(\iint_{S(C)} (\nabla_\lambda \times \boldsymbol{A}(\lambda)) \cdot \mathrm{d}\boldsymbol{S} \right) = - \iint_{S(C)} \boldsymbol{\Omega}(\lambda) \cdot \mathrm{d}\boldsymbol{S} \tag{4.6}$$

the Berry curvature

$$\boldsymbol{\Omega}(\lambda) = \mathrm{Im}(\nabla_\lambda \times \boldsymbol{A}(\lambda)). \tag{4.7}$$

Stokes theorem also shows that the Berry phase and curvature are gauge-independent while the Berry connection is not. The proof can be done using a unitary transformation like $|n\rangle \rightarrow e^{i\theta} |n\rangle$. Since the structure of the formulas is very similar to the Vector field in Maxwell calculus the Berry curvature can be interpreted as a magnetic field in parameter space.

4.2 | Topology and Bloch Bands

As the system studied here is a lattice system, it is required to port the concept of topology to Bloch bands. Here I will follow the derivations in [201] and [204]. The Berry curvature introduced in equation 4.7 for Bloch bands is given by

$$\boldsymbol{\Omega}(\boldsymbol{q}) = \mathrm{Im}\left(\nabla \times \langle \boldsymbol{q}^n | \nabla \boldsymbol{q}^n \rangle\right) \tag{4.8}$$

where equation 4.5 is plugged into equation 4.7 and $|\boldsymbol{q}^n\rangle$ denotes the eigenstate of the band index n with quasi momentum $\boldsymbol{q}$. Notice that the original parameter defined in equation 4.3 now is the quasimomentum. Therefore the derivatives are also taken in quasimomentum space which may lead to problems in practice. Using the Schrödinger equation it is possible to derive a much more useful representation of $\langle \boldsymbol{q}^n | \nabla \boldsymbol{q}^n \rangle$ with the dual vector $\left\langle \boldsymbol{q}^{n'} \right|$

$$\left\langle \boldsymbol{q}^{n'} \middle| \nabla \boldsymbol{q}^n \right\rangle = \frac{\left\langle \boldsymbol{q}^{n'} \middle| \nabla \hat{H} \middle| \boldsymbol{q}^n \right\rangle}{E_q^n - E_q^{n'}} \tag{4.9}$$

which helps to interpret the Berry curvature. Using this relation the Berry curvature can be expressed differently

$$\begin{aligned}
\boldsymbol{\Omega}(\boldsymbol{q}) &= \mathrm{Im}\left(\nabla \times \langle \boldsymbol{q}^n | \nabla \boldsymbol{q}^n \rangle\right) && (4.10)\\
&= \mathrm{Im}\left(\langle \nabla \boldsymbol{q}^n | \times | \nabla \boldsymbol{q}^n \rangle\right) && (4.11)\\
&= \mathrm{Im}\left(\sum_{n \neq n'} \left\langle \nabla \boldsymbol{q}^n \middle| \boldsymbol{q}^{n'} \right\rangle \times \left\langle \boldsymbol{q}^{n'} \middle| \nabla \boldsymbol{q}^n \right\rangle \right) && (4.12)\\
&= \mathrm{Im}\left(\sum_{n \neq n'} \frac{\left\langle \boldsymbol{q}^n \middle| \nabla \hat{H} \middle| \boldsymbol{q}^{n'} \right\rangle \times \left\langle \boldsymbol{q}^{n'} \middle| \nabla \hat{H} \middle| \boldsymbol{q}^n \right\rangle}{(E_q^n - E_q^{n'})^2} \right) && (4.13)\\
& && (4.14)
\end{aligned}$$

where we used the orthogonality of the eigenbasis. The first observation that pops out of the equation is, that a least two bands are required for a minimal model to employ the concept of Berry curvature for Bloch bands. Furthermore, it allows the interpretation that Berry curvature exists due to virtual transitions into higher bands due to the varying Hamiltonian a hypothetical particle experiences [205]. Additionally, the inverse squared scaling of the Berry curvature by the energy gap shows that it converges for touching bands.

With the definition of Berry curvature for Bloch bands, it is now possible to define the topological property similar to the genus in the Euclidean world for a single Bloch band, the Chern number

$$C = \frac{1}{2\pi} \iint_{1.\mathrm{BZ}} \mathrm{d}\boldsymbol{S} \cdot \boldsymbol{\Omega}(\boldsymbol{q}) \tag{4.15}$$

which is the acquired Berry phase along a path enclosing the first Brillouin zone (BZ). The Chern number is topological invariant, i.e. smooth variations of the lattice Hamiltonian cannot change the Chern number of the band. However, when bands touch, i.e. the energy gap between the bands is zero, the Chern number of the bands can change. Nevertheless, the sum of the Chern number of the bands is still conserved [206].

Berry curvature and the Chern number can describe several macroscopic effects. According to the bulk-boundary correspondence, the Chern number counts the number of edge states [85]. Interpreting the Berry curvature in Bloch bands as a magnetic field, similar to the case of the general parameter interpretation given in chapter 4.1 interesting effects can be observed. In the presence of Berry curvature deflection can be measured for an accelerated wave packet in momentum space [87, 88, 205, 207] similar to the Hall drift.

Since the discussed effects arise from topological non-trivial regimes, it is important to understand circumstances that lead to a finite Berry curvature and nonzero Chern numbers. The time reversal operator $\mathcal{T}$ transforms Bloch states according to

$$\mathcal{T} \left| \boldsymbol{q}^n \right\rangle = \left| -\boldsymbol{q}^n \right\rangle^* \tag{4.16}$$

since $\boldsymbol{q} \to -\boldsymbol{q}$ and $t \to -t$. Consequently, the Berry curvature defined in equation 4.8 transforms like

$$\mathcal{T}\boldsymbol{\Omega}(\boldsymbol{q}) = -\boldsymbol{\Omega}(-\boldsymbol{q}). \tag{4.17}$$

Hence a system featuring time reversal symmetry features an odd Berry curvature. As the integral over the first BZ vanishes for odd functions, the Chern number of such systems is zero.

In contrast, the inversion symmetry operator transforms Bloch states according to

$$\mathcal{I} \left| \boldsymbol{q}^n \right\rangle = \left| -\boldsymbol{q}^n \right\rangle \tag{4.18}$$

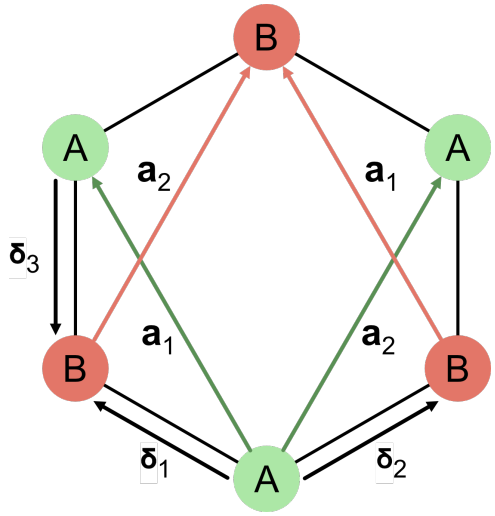

Figure 4.2: Sketech of the Boron Nitride lattice in real space. The two different sites A and B are connected by the basis vectors $\boldsymbol{\delta}_i$ and the corresponding tunnel elements $t_{AB,i}$. The lattice vectors $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ connect the A sites with the corresponding next neighbor tunneling elements $t_{AA,i}$ and the B sites with the tunneling amplitudes $t_{BB,i}$.

and consequently the Berry curvature like

$$\mathcal{I}\boldsymbol{\Omega}(\boldsymbol{q}) = \boldsymbol{\Omega}(-\boldsymbol{q}) \tag{4.19}$$

which leads to the observation that systems featuring time and inversion symmetry impose

$$\boldsymbol{\Omega}(\boldsymbol{q}) = \boldsymbol{\Omega}(-\boldsymbol{q}) = -\boldsymbol{\Omega}(-\boldsymbol{q}) \tag{4.20}$$

hence the Berry curvature has to be zero or singular. In conclusion, either time or inversion symmetry has to be broken to feature finite Berry curvature which is a key ingredient to get non-trivial Chern numbers.

4.3 | Topology in Boron-Nitride lattices

For now, all concepts have been introduced to work with the Berry curvature in Bloch bands but no specific Hamiltonian is considered. In this section, I will introduce the two-band tight-binding model for hexagonal lattices following the PhD thesis of Nick Fläschner [204] and [208, 209]. This model can feature topological trivial and non-trivial phases by applying Floquet engineering, i.e. breaking the time-reversal symmetry by accelerating the lattice. The real space lattice is a triangular lattice with a two-atomic basis also known as a Boron Nitride lattice. A sketch of the lattice can be found in figure 4.2. The corresponding tight-binding Hamiltonian can be written as

$$H(\boldsymbol{k}) = \begin{pmatrix} \frac{\Delta_{AB}}{2} + \sum_j 2t_{AA,j}\cos(\boldsymbol{k}\cdot\boldsymbol{a}_j) & \sum_j 2t_{AB,j}e^{-i\boldsymbol{k}\cdot\boldsymbol{\delta}_j} \\ \sum_j 2t_{AB,j}e^{-i\boldsymbol{k}\cdot\boldsymbol{\delta}_j} & -\frac{\Delta_{AB}}{2} + \sum_j 2t_{BB,j}\cos(\boldsymbol{k}\cdot\boldsymbol{a}_j) \end{pmatrix} \tag{4.21}$$

where Δ_{AB} is the energy offset between A and B sites, t_{AA} are the next neighbouring tunneling amplitudes between A sites and t_{BB} are the tunneling amplitudes for B sites and t_{AB} the neighbouring tunneling amplitudes between A and B sites. The lattice is spanned by the lattice vectors $\boldsymbol{a}_{1,2,3}$ and the basis vectors connecting the two sites of the two atomic basis $\boldsymbol{\delta}_{1,2,3}$. Notice that here the Hamiltonian is written in the basis of the two Bloch states $\hat{a}^\dagger_{\boldsymbol{k}} |0\rangle = |\boldsymbol{k}, A\rangle$ and $\hat{b}^\dagger_{\boldsymbol{k}} |0\rangle = |\boldsymbol{k}, B\rangle$ where $\hat{a}^\dagger_{\boldsymbol{k}}$ and $\hat{b}^\dagger_{\boldsymbol{k}}$ are creation operators of the Bloch state which are localized in momentum space and restricted to the sub-lattice spanned by the A or B sites. To illustrate interesting properties of the model, it is convenient to reformulate it with the pseudo-spin basis

$$H(\boldsymbol{k}) = \sum_{j=1}^{3} h_j(\boldsymbol{k})\sigma_j + h_0(\boldsymbol{k})\mathrm{Id}_{2\times 2} \tag{4.22}$$

where σ_j are the corresponding Pauli matrices from the Pauli vector $\boldsymbol{\sigma}$ and the 2-dimensional identity matrix $\mathrm{Id}_{2\times 2}$. Comparing the coefficients with the coefficients in the Bloch state basis in equation 4.21 allows us to derive the four coefficients

$$h_0 = h_0(\boldsymbol{k}) = \sum_{j=1}^{3} (t_{AA} + t_{BB}) \cos(\boldsymbol{k} \cdot \boldsymbol{a}_j) \tag{4.23}$$

$$h_1 = h_1(\boldsymbol{k}) = \sum_{j=1}^{3} t_{AB} \cos(\boldsymbol{k} \cdot \boldsymbol{\delta}_j) \tag{4.24}$$

$$h_2 = h_2(\boldsymbol{k}) = \sum_{j=1}^{3} t_{AB} \sin(\boldsymbol{k} \cdot \boldsymbol{\delta}_j) \tag{4.25}$$

$$h_3 = h_3(\boldsymbol{k}) = \frac{\Delta}{2} + \sum_{j=1}^{3} (t_{AA} - t_{BB}) \cos(\boldsymbol{k} \cdot \boldsymbol{a}_j). \tag{4.26}$$

These coefficients are handy as they allow us to define the mixing angle $\theta = \theta(\boldsymbol{k})$ and phase $\phi = \phi(\boldsymbol{k})$ belonging to the state vector inside the Bloch sphere. They can be derived once again by rewriting the Hamiltonian as

$$H(\boldsymbol{k}) = h_0 \mathrm{Id}_{2\times 2} + R \begin{pmatrix} \cos(\theta) & \sin(\theta) e^{-i\phi} \\ \sin(\theta) e^{i\phi} & -\cos(\theta) \end{pmatrix} \tag{4.27}$$

with $R = R(\boldsymbol{k}) = \sqrt{h_1^2 + h_2^2 + h_3^2}$. By comparing the entries in the Hamiltonian with the entries in equation 4.22 the angles θ and ϕ are defined by the three equations

$$\sin(\theta) = \frac{|h_1 + i h_2|}{R} \tag{4.28}$$

$$\cos(\theta) = \frac{h_3}{R} \tag{4.29}$$

$$\phi = \arg(h_1 + i h_2) \tag{4.30}$$

where $\arg$ denotes the argument of a complex number. The angle describes the eigenvectors of the lower (-) and upper (+) band as linear combinations of the Bloch states $|\boldsymbol{k}, A\rangle$ and $|\boldsymbol{k}, B\rangle$ which are at the north and south pole of the Bloch sphere

$$|\boldsymbol{k}, -\rangle = \sin\left(\frac{\theta}{2}\right) e^{-i\phi} |\boldsymbol{k}, A\rangle + \cos\left(\frac{\theta}{2}\right) |\boldsymbol{k}, B\rangle \tag{4.31}$$

$$|\boldsymbol{k}, +\rangle = \sin\left(\frac{\theta}{2}\right) e^{i\phi} |\boldsymbol{k}, B\rangle + \cos\left(\frac{\theta}{2}\right) |\boldsymbol{k}, A\rangle \,. \tag{4.32}$$

Using the definitions for the Berry connection the Berry curvature for this particular representation is

$$\boldsymbol{\Omega}(\boldsymbol{k}) = -\frac{1}{2}\sin(\theta)\left(\partial_x\theta\partial_y\phi - \partial_y\theta\partial_x\phi\right) \tag{4.33}$$

where ∂_i is the derivative in the corresponding direction in momentum space. A key insight is, that the Berry curvature for this particular model is given by just the two angles. This is important since this directly allows us to link the value of the Chern number to the properties of the pseudo-spin-operator $S = \hat{\boldsymbol{h}} \cdot \boldsymbol{\sigma}$ with

$$\hat{\boldsymbol{h}} = \begin{pmatrix} \sin(\theta)\cos(\phi) \\ \sin(\theta)\sin(\phi) \\ \cos(\theta) \end{pmatrix} . \tag{4.34}$$

The Chern number now counts how often the eigenvector wraps around the Bloch sphere [210].

Using these findings it is clear that it is possible to measure the Berry curvature and also the Chern number by using quench dynamics as introduced by [210]. As we describe in section 4.4 we use ToF images after different t_{hold} times in the shaken lattice. The resulting momentum space density following [210] at time t after the quench can be written as

$$n(\boldsymbol{k}, t) \propto 1 - \sin(\theta)\cos\left(\frac{t\Delta_{AB}}{\hbar} + \phi\right) \tag{4.35}$$

where we can extract the mixing angle and the phase to extract the Berry curvature and thus the Chern number as well. This is crucial since we can not expect ML to extract information from data where the required information is not included.

4.4 | Experimental Protocol

The experimental data used for the supervised and unsupervised machine-learning analysis of the topological phases stems from experiments with ultracold atoms in optical lattices [55, 62] realized in the BFM experiment [72]. Ultracold atoms trapped in optical potentials are a versatile and proven tool to study condensed matter phenomena and in particular topological

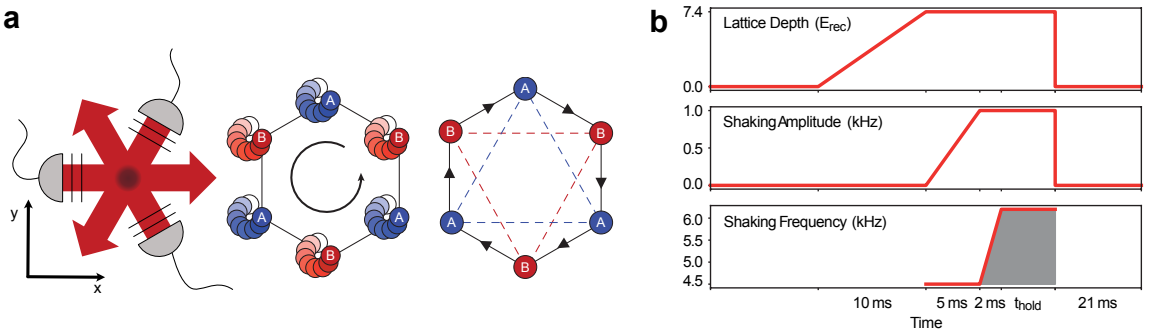

Figure 4.3: Overview of the experimental protocol. (a) Atoms are trapped and cooled with three interfering laser-beams. The angle between the laser beams is $120°$. Lattice shaking is used to create an effective model. The engineered model consists of a hexagonal lattice with A and B sites distinguished by their energy offset Δ_{AB} and the hopping amplitude between neighboring sites. (b) Schematic timeline of the experimental protocol and the relevant parameters. The grey area under the curve is the micromotion phase that can be varied by the hold time t_{hold}.

phases of matter [202, 211]. In the BFM experiment the topological Haldane model [91] is realized by trapping $^{40}\mathrm{K}$ with a mass of $m = 40\mathrm{u}$ atoms in an optical honeycomb lattice [72, 87] which is theoretically motivated in section 4.1 and an extension to the triangular lattices previously developed for quantum gas experiments [67, 71]. To realize the potential three interfering laser beams with a respective angle of $120°$ form an optical potential which is tuned by the polarization of the beams. The experiments start at a large offset of the two sublattices of $\Delta_{\mathrm{AB}} = 2\pi \cdot 6.1\mathrm{kHz}$. See figure 4.3a. The relative phases of the beams can be dynamically modified by acoustooptical elements which enables the possibility of lattice shaking. The lattice is accelerated on elliptical trajectories to break time-reversal symmetry, characterized by the shaking phase ϕ between the modulation in the x and y direction and the shaking frequency f_{sh}. The resulting Floquet engineered Haldane model for near-resonant shaking with a sublattice offset of $f_{\mathrm{sh}} \approx \frac{\Delta_{\mathrm{AB}}}{2\pi}$ features non-trivial topological properties that are identified by Chen numbers $C = \pm 1$. The resulting phase diagram 4.5b is closely related to the original phase diagram of the Haldane model [40, 89, 90].

In further detail, the potassium atoms $^{40}\mathrm{K}$ are prepared in the lowest band of the optical lattice formed by three laser beams with a wavelength of $\lambda = 1064\,\mathrm{nm}$ like in previous work on the BFM experiment [40, 72]. The defining energy scale of the system is given by the recoil energy

$$E_{\mathrm{rec}} = \frac{h^2}{2m\lambda^2} \tag{4.36}$$

where h is the Planck constant. The atom cloud is weakly confined in the transverse direction. After trapping and cooling the atoms the lattice depth is ramped up to $7.4E_{\mathrm{rec}}$. Afterwards, we start the Floquet driving. Figure 4.3b shows the timing of the different parameters to adiabatically prepare the lowest Floquet band. The Floquet drive ramps up in two different

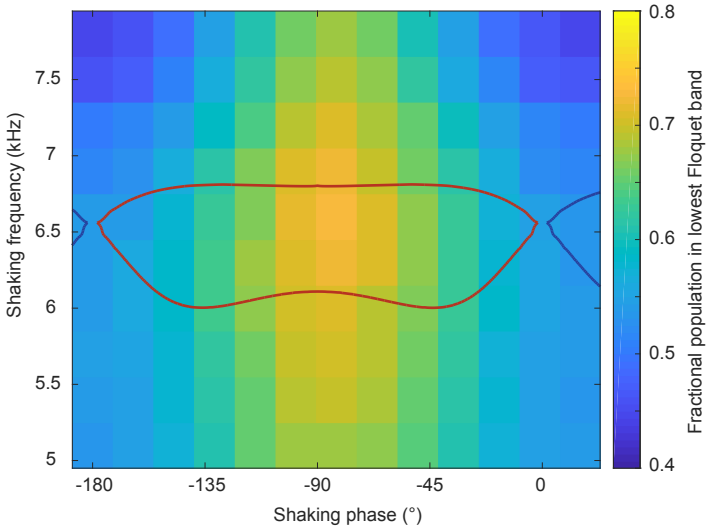

Figure 4.4: Population of the lowest Floquet band for different shaking frequencies f_{sh} and different shaking phases φ. As an orientation, the theoretical prediction is plotted by the solid line. Notice that this is only one half of the complete phase diagram. Due to the symmetry of the system, the band population will be similar on the other half of the complete Haldane phase diagram.

steps. First the shaking amplitude is ramped up to $1\,\mathrm{kHz}$ within $5\,\mathrm{ms}$ at the far off-resonant shaking frequency of $f_{\mathrm{sh}}^{\mathrm{ini}} = 4.5\,\mathrm{kHz}$. Afterwards, the shaking frequency is ramped up to its final value $f_{\mathrm{sh}}^{\mathrm{fin}}$ within $t_{\mathrm{ramp}} = 2\,\mathrm{ms}$ while the shaking amplitude is fixed. The protocol is well established via previous experimental and theoretical works [72, 90, 212] and keeps the band gaps as large as possible to avoid a population of higher bands. Nevertheless, Floquet heating leads to lower band populations of the lowest bands as shown in figure 4.4. Here we ramp the system back to the bare bands and perform band mapping of the two lowest bands to measure the relative population of the two bands. For shaking phases of around $\varphi = \pm 90°$ the band population of the lowest band is maximized around 75% while it drops to around 50% for shaking phases near $\varphi = 0°, \pm 180°$.

For the detection of the momentum space density distribution of the system, all potentials are switched off, leading to a free expansion of the system also known as time of flight (ToF). Due to the momentum of the system, the original momentum space distribution is mapped to a real space density that can be captured by absorption imaging. We expect topological information in the density distribution as the procedure can be related to Bloch state tomography [72, 90, 210] based on quench dynamics after projection onto the static lattice with large offsets Δ_{AB}. Although tomography requires the full quench dynamics to disentangle the parameters [210], the procedure is promising to contain information on the topological phases [40].

In the experiments, the hold time t_{hold} is varied and the atoms are held in the Floquet system for different times at the final shaking frequency $f_{\mathrm{sh}}^{\mathrm{fin}}$ in steps smaller than the Floquet period. This allows the analysis of samples at different instances of Floquet micromotion phases ϕ.

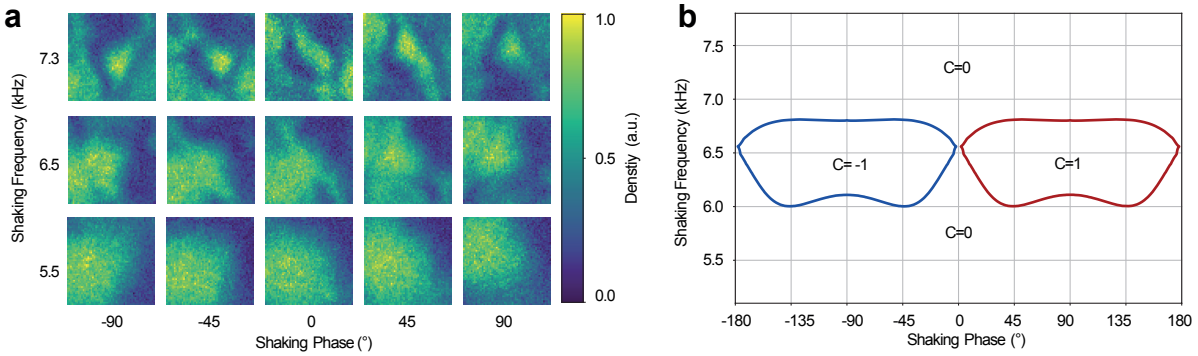

Figure 4.5: Measurement examples. (a) optical densities obtained by absorption imaging. The images are centered around zero momentum and their size corresponds to one reciprocal lattice vector length. The images are ordered to the corresponding parameters according to the phase diagram in (b). The expected topological phase diagram for the lowest band features two lobes of non-trivial topological phases associated with Chern number $C = -1$ for negative shaking phases and $C = 1$ for positive shaking phases. The theoretical predictions stem from a numerical Floquet calculation of a tight binding model.

Originally motivated by the Bloch state tomography as mentioned above, this variance in the micromotion phase is not beneficial for unsupervised learning procedures as we will see in section 4.7.2. The micromotion phase is given by the area under the shaking frequency curve, i.e. tracing the micromotion back to the start of the Floquet driving with a kick in a fixed direction, as given in figure 4.3b or by the formula

$$\phi = \left(\frac{t_{\mathrm{ramp}}}{2} + t_{\mathrm{hold}} \right) f_{\mathrm{sh}}^{\mathrm{fin}} + t_{\mathrm{ramp}} f_{\mathrm{sh}}^{\mathrm{ini}} \tag{4.37}$$

which allows to link the micromotion phase ϕ to the shaking frequency f_{sh}. While the micromotion, which is an intrinsic attribute of Floquet systems, often interferes with the studies of effective Floquet Hamiltonians [213, 214] it can also reveal new physics [215, 216]. As the first BZ theoretically contains all the information about the topological phase we restrict the images to a square region of 56×56 pixels centered around zero momentum of $k = 0$. The reciprocal lattice vector length is also 56 pixels. Sample ToF images are plotted in figure 4.5a. Furthermore, each image is scaled to values between 0 and 1 for the pixel values. In total $10,436$ images are part of the dataset evenly distributed in the parameter region of the phase diagram. Each parameter pair of shaking frequency and phase in the phase diagram contains between 3 and 7 images.

4.4.1 Numerical Predictions

We employ a Floquet calculation for a two-band tight-binding model of the hexagonal lattice based on the given shaking parameters and the calibrated parameters of the static lattice with a numerical framework introduced in the PhD thesis of Nick Fläschner [204]. The numerical phase diagram is plotted in figure 4.5b. Except for a slight shift of the topological phase boundary for higher frequencies towards higher frequencies, the predictions agree well with

previous works [40, 72, 89, 90]. The shift may be governed by uncertainties in the calibration of the static lattice or by influences of higher bands which are neglected in the tight-binding approximation. An uncertainty of $\pm 200\,\mathrm{Hz}$ of the numerically predicted phase transitions is expected due to the uncertainty of the polarisation of the lattice beams by 0.2° [40].

We suspect that the approximations of our numerical approach, in particular the neglect of higher bands, might lead to a systematic shift of the topological phase transitions. Such an effect was observed in [217]. We have therefore started to develop numerical calculation that takes higher bands into account. The master project of Corinna Menz [218] implemented a multi-band tight-binding model taken from [217] that can be fitted to the exact band structure and subsequently be used for the Floquet calculations. It turned out that fitting such multi-band tight-binding models is not very stable and the project did not reach a clear conclusion. The Bachelor project of Anna Lena Hauschild [219] implemented the complementary approach of avoiding tight-binding models and directly solving the Floquet matrix [220]. So far this method could not be tested in the parameter regime presented here.

4.5 | Measuring topological properties

To measure the Chern number from an experimental cold quantum gas system it is possible to employ measurement schemes that do not leverage machine learning methods [72, 87–90]. Each machine learning method has to compete with these methods in terms of data intensity, i.e. how much data is needed to get reliable results, and data accessibility. The method most comparable to the actual system that is analyzed in this thesis is the circular dichroism method by Luca Asteria et al. [89]. This method is based on spectroscopic signals of the Floquet bands revealing its underlying topological nature. The experimental protocol is fundamentally different from the measurement protocol used for the machine learning phase classification. The protocol here is similar to the measurement protocol used to measure the Berry curvature [72]. The reconstruction of the Berry curvature is however not accessible for regions of non-zero Chern numbers due to the high sensitivity to a residual population of the upper Floquet bands.

Figure 4.4 shows the population of the lowest Floquet band in the system. Aside from circular shaking, i.e. a shaking phase of $\varphi = \pm 90^\circ$ the band population is very low, i.e. below 70%. The population is measured by ramping the system back to bare bands after the ramp into the Floquet regime and performing adiabatic band mapping. Here only the first two BZs are counted. For linear shaking at $\varphi = 0^\circ, \pm 180^\circ$ the band population drastically drops which leads to poor performance in the measurement protocols for the Chern number. This problem can be resolved with ML methods as studied in the upcoming sections 4.6 and 4.7.

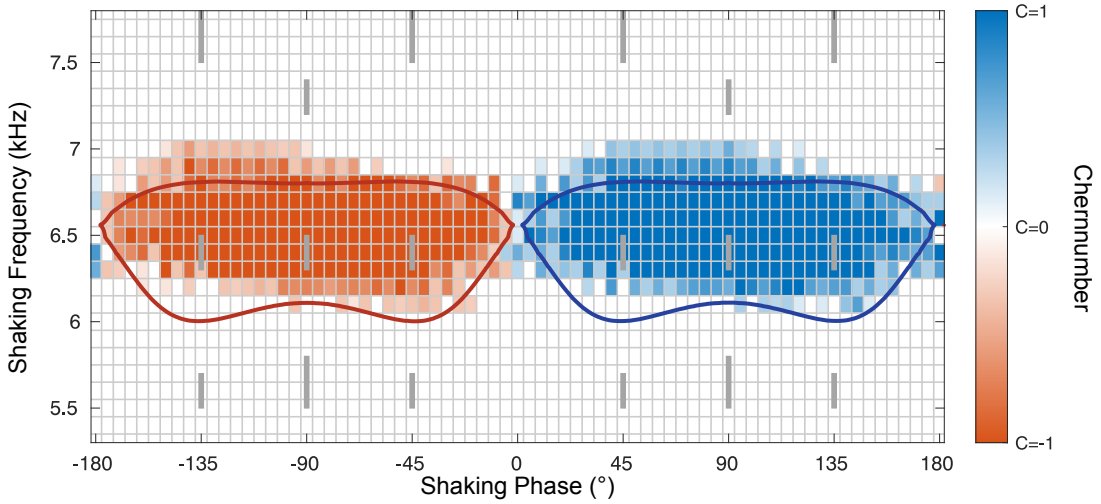

Figure 4.6: Supervised reconstruction of the Haldane phase diagram with a CNN. Each square in the grid corresponds to the average prediction of 3-7 images. The gray bars correspond to the training regions and are chosen far away from the theoretically predicted phase transitions (solid lines). The color bar is the Chern number average over predictions of the CNN. This figure is adapted from [40].

4.6 | Supervised Phase Detection

Prior to the unsupervised approaches employed in this thesis, we mapped out the complete Haldane phase diagram in a supervised manner [40]. As described in section 3.2 a CNN is employed and trained in a supervised scheme to detect the three phases with the different Chern numbers, i.e. $C = 0$ and $C = \pm 1$, of the phase diagram. In figure 4.6 the mapped-out phase diagram is shown. As for each point in the Haldane phase diagram, 3-7 images are available the plot illustrates the average predicted Chern number. The labeling of the data was conducted by the theoretical predictions and a separate data set was taken at the gray training regions far away from the expected transition regions. The additional training data consists of $15,963$ images covering only 3% of the complete area of the phase diagram. For this analysis, the images are not cropped to first BZ and have a size of 151×151 pixels. However, crosschecks with the smaller images show that only data from the first BZ is required to identify the different phases of matter as assumed in section 4.4.

The reconstructed phase diagram is in good agreement with the numerical predictions, even though the lower band population drops to about 50%. However, a small shift to higher frequencies in the upper frequency bound can be observed. Furthermore, some features like the apple shape structure of the two topological non-trivial regions are lacking. The reconstructed phase diagram proves that sufficient information about the topological phase exists in the underlying data and motivates the investigation of the dataset further with more advanced methods.

4.7 | Unsupervised Phase Detection

The supervised reconstruction of the Haldane phase diagram from experimental data is impressive considering the low population of the lowest band. However, it has one disadvantage. To train the CNN to classify the different phases of matter the approximate phase transitions have to be known from theory. In this section, we show that it is possible to reconstruct the phase diagram without prior knowledge of the phases.

4.7.1 Bottleneck Analysis

One of the most straightforward methods to analyze unknown data with unsupervised machine learning methods is the analysis of the neuron activations of an AE as introduced in section 2.6.2 at the bottleneck which is an effective tool for data dimension reduction [221]. In this case, the encoder and decoder consist of multiple convolutional layers and a bottleneck of two neurons as illustrated in figure 4.7a. Here we choose two bottleneck neurons because they are easily interpretable and it is the lower bound of bottleneck neurons. Choosing less bottleneck neurons leads to divergence of the training loss of the AE. A side effect is that two-dimensional data can be interpreted much more easily than more-dimensional data. The detailed structure of the AE can be found in appendix table C.1.

Extracting the activations of the two bottleneck neurons leads to a dense cloud of data points in two-dimensional space as shown in 4.7b where no clear signal can be extracted to identify the different Chern numbers nor any other clear clustering. Restricting the data to fixed shaking phases φ, i.e. vertical cuts through the Haldane phase diagram, guides to a much clearer result. The points cluster on elliptical-shaped patterns around the origin of the latent space as shown in figure 4.7c and d. To compare these structures each sub-dataset from a vertical cut is fitted with an ellipse using direct least-squares fitting [222]. Using a coordinate transformation into the associated elliptical coordinates, each data point can be associated with a radius r and an azimuthal angle θ measured from the major axis of the fitted ellipse. Plotting the different dependencies it becomes clear that the main signal originates from the micromotion phase ϕ as introduced in formula 4.37. It shows a clear linear dependence for a shaking phase of $\varphi = 90°$ as plotted in figure 4.7e. This correspondence can also be seen in the center of mass movement of the images. For details see appendix A. Analyzing the radial component of the data as plotted in figure 4.7f reveals some dependency on the topological phase indicated by plateaus along the shaking frequency axis. These plateaus are unfortunately not well separable, especially for other shaking phases aside from $\varphi = \pm 90°$. Hence it is likely that the AE identifies the center of mass as the main characteristic of the images and concentrates on the identification of the micromotion phase. The removal of the micromotion phase will be part of the next chapter. Nevertheless, it should be noted that this

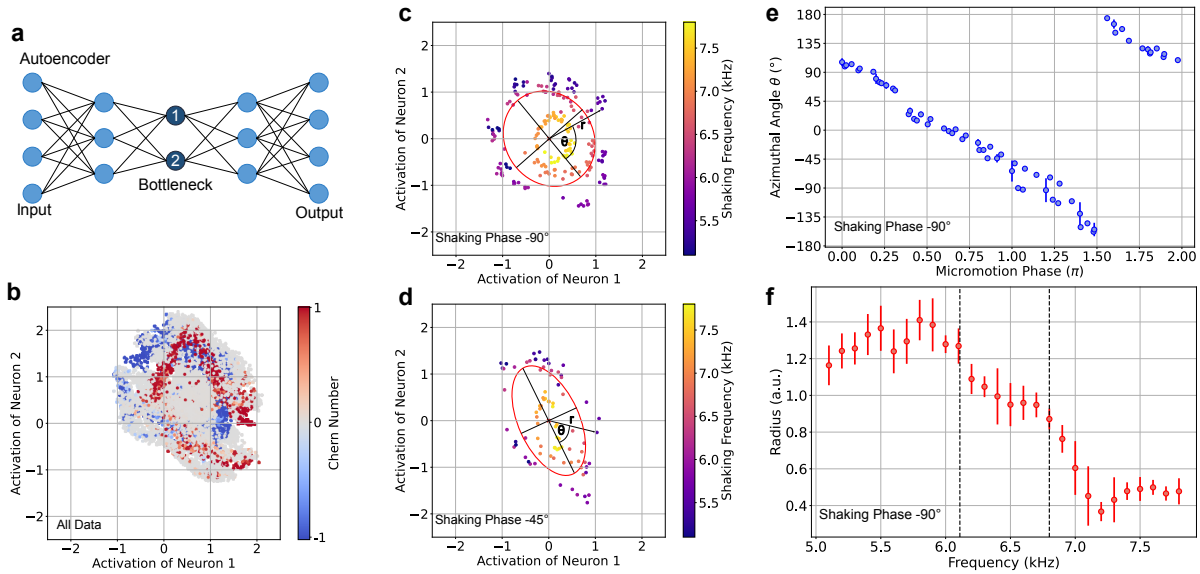

Figure 4.7: Bottleneck analysis. (a) Sketch of the autoencoder used to extract the two-dimensional latent space representation of the ToF images. (b) All two-dimensional representations of the experimental ToF images where each point corresponds to one image. The color coding stems from the numerical calculations to illustrate no clear clustering according to the Chern number. (c) and (d) the latent space representation and the corresponding fitted ellipses of two vertical cuts through the Haldane phase diagram for fixed shaking phases (c) $\varphi = -90°$ and (d) $\varphi = -45°$. (e) and (f) the resulting elliptical coordinates (e) θ and (f) r for a fixed shaking phase $\varphi = -90°$. (e) The azimuthal angle of the data has a clear linear dependence on the micromotion phase ϕ. (f) The radius includes information about the different topological phases by showing three plateaus. The dashed vertical lines indicate the numerical predictions for phase transitions.

method could work if the measurements were repeated with a protocol taking into account the micromotion phase. This could be done by modulating the hold time t_{hold} in the experimental procedure accordingly.

4.7.2 Micromotion Removal

The experimental realization of the Haldane model using Floquet engineering gives rise to a property that does not influence the underlying topological phases. Micromotion poses a hard challenge to employ ML algorithms originally designed to identify changes in the system. As shown in section 4.7.1 the machine learning algorithm focus is the micromotion phase however the information about the topological phases is highly suppressed. This can be changed drastically by using data with a fixed micromotion phase, i.e. the displacement of the center-of-mass (CoM) is in the same direction, as I will show in section 4.7.3. The most straightforward way to get rid of the micromotion phase is by employing a measurement protocol taking into account the micromotion phase and changing the hold time t_{hold} to result in a fixed value of the micromotion phase ϕ. Unfortunately, the BFM experiment undergoes some major redesign and the data could not be taken with this aspect in mind. As no analytic transformation of the ToF images between different micromotion phases exists, a machine learning approach is employed to post-process the data. The challenge of post-processing the data is similar to tasks like fringe removal in absorption imaging [223] in ultracold atoms experiments or the removal of jitter in pump-probe experiments [224]. VAEs are a versatile tool for data transformation and generation [132, 134]. As introduced in section 2.6.2 VAEs are a special kind of AEs that blend the concepts of AEs and probabilistic modeling and are very well suited to accomplish generative tasks. Here a VAE with an additional question neuron is used, i.e. an additional neuron is added to the bottleneck that can be set to the desired parameter. See inlet in figure 4.8d. This type of AE has already been proven successful in extracting physically relevant parameters [225].

The goal of the transforming ML ansatz is to transform any ToF image with a micromotion phase ϕ_{input} to a new, artificial ToF image with a new micromotion phase ϕ_{output}. The difference $\Delta\phi = \phi_{\mathrm{output}} - \phi_{\mathrm{input}}$ is the transformation parameter that is used as input for the question neuron of the VAE. The employed VAE is by far the largest machine learning model employed in the course of this thesis. The encoder consists of several convolutional stages and a staggered dense layer setup. The bottleneck is spanned by 26 fully connected neurons mimicking 13 uncorrelated Gaussian distributions, each defined by its mean and its log variance to enforce numerical stability. The decoder consists of several dense layers followed by several transposed convolutional stages. The details can be found in the appendix in table C.2. In total, the VAE has over three million trainable parameters. The structure and hyperparameters of the model are optimized and fine-tuned by the optimization library Optuna

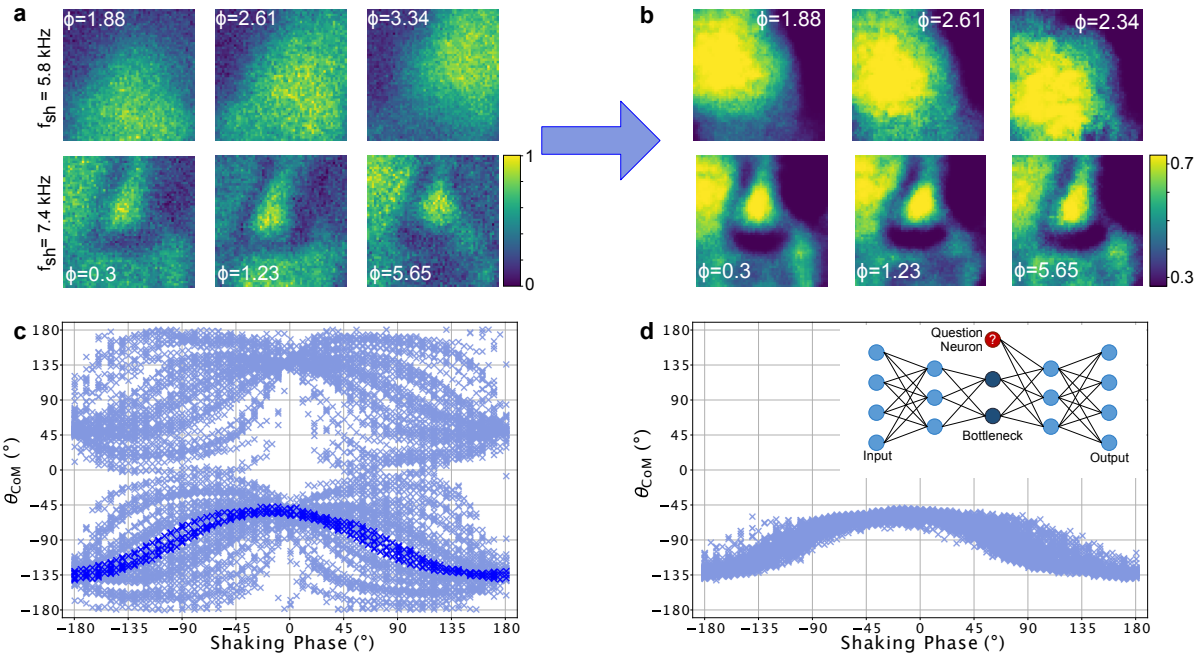

Figure 4.8: Using an additional question neuron to post-process the experimental ToF images to a fixed micromotion phase. (a) randomly selected sample images of $5.8\mathrm{kHz}$ and $7.4\mathrm{kHz}$ and a shaking frequency of $\varphi = 90°$ for different micromotion phases ϕ. Here the different CoM positions of the density distribution can be extracted by the eye. (b) The corresponding rephased images from sub-figure (a) after the micromotion transformation. Here a micromotion phase of $\phi_0 = 0.0$ is set with the VAE. The CoM is only weakly dependent on the original micromotion phase ϕ. (c) the original azimuthal angle of the CoM θ_{CoM} of all ToF images before transformation with the VAE are widely distributed for all shaking frequencies f_{sh}. (d) the distribution after transforming the data to a fixed micromotion phase is distributed along one path of θ_{CoM}. The inlet sketches the structure of the VAE with the additional question neuron.

[226]. Over 60.000 different network architectures have been trained. To identify the optimal network architecture and best choice of hyperparameters a good measure of performance is the structural similarity index [227]. The training data set is generated by the ToF images. For each point in the phase diagram, several measurements with varying hold times and fixed shaking phase and frequency are available. Examples are shown in figure 4.8a. They can be paired and complemented with their micromotion phase difference $\Delta\phi = \phi_{\text{output}} - \phi_{\text{input}}$ resulting in a dataset of 63.050 examples. Notice that in this case the input and output of the VAE is not always the same ToF image. However, both images contain the same topological information. For validation purposes, 10% of the training set is hidden during training.

After training the trained VAE can be used to set all ToF images to a desired micromotion phase. Without any preference $\phi = 0.0$ is chosen. The original image serves as input for the encoder and its negative micromotion phase $\Delta\phi = -\phi_{\text{input}}$ as input for the question neuron. The output of the decoder is a ToF image with the desired micromotion phase of $\phi_{\text{output}} = 0$. The transformed images are shown in figure 4.8b. Besides the transformation of the micromotion, the images lose some noise which is expected since it does not contribute to any physical properties of the underlying system, and undergo a squeezing of the pixel values for the images to a range of $0.3 - 0.7$ which can be attributed to the non-linear activation functions of the decoder.

The micromotion can be directly linked to the CoM of the images. Thus the success of micromotion removal can be seen by calculating the CoM of the images and comparing the two distributions before and after micromotion removal. In figure 4.8c the distribution before the removal is plotted. Highlighting only one micromotion phase reveals the underlying distribution. The azimuthal coordinate of the CoM θ_{CoM} is sinusoidal concerning the shaking phase φ and a fixed micromotion phase. Removing the micromotion phase reveals this structure but now for all ToF images from the dataset. To identify the phase transitions the data can now be set to a fixed micromotion phase focusing the attention of the machine learning algorithms to the phase information and reducing attention to the micromotion phase. To provide evidence that the micromotion removal is successful and no topological properties are lost in the process, we influence functions as introduced in 2.7.2. We expect that before the micromotion removal a supervised classification NN is mostly influenced by samples from the training data with the same micromotion phase. Consequently removing the micromotion phase leads to a distributed influence to identify the different samples correctly. First, a CNN is trained in a supervised manner to identify the different phases similar to section 4.6. However here we concentrate on a single vertical line of the Haldane phase diagram for a shaking phase of $\phi = 90°$. Here the Chern number changes from $C = 0$ to $C = -1$ to $C = 0$ with increasing shaking frequency f_{sh}. Therefore the labeled training set only contains samples from $C = 0$ and $C = -1$. The trained classifier can be used to calculate the influence function of the complete dataset for a single test point as shown in figure 4.9 (blue data). The test point is

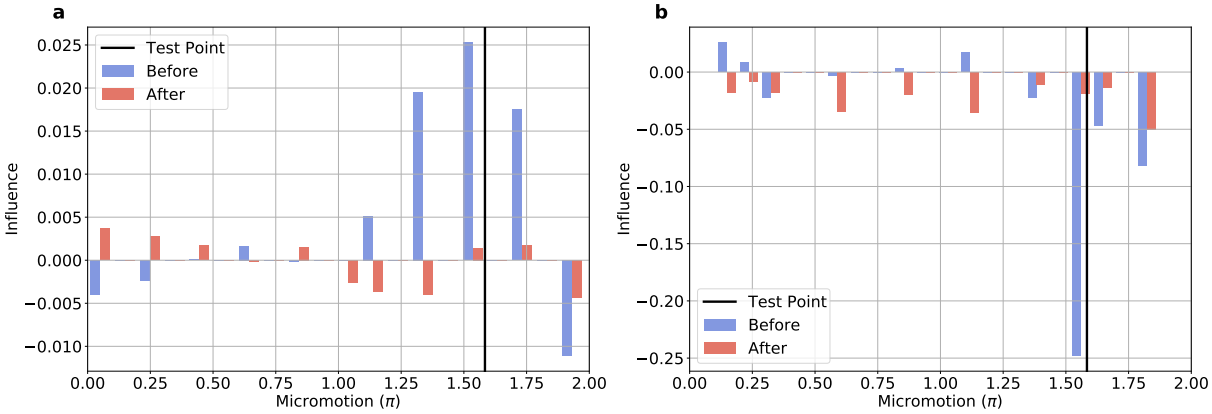

Figure 4.9: Distribution of influential points before and after the micromotion removal with the VAE for a shaking frequency of $f_{\mathrm{sh}} = 5.5$ (a) and $f_{\mathrm{sh}} = 6.3$ (b) and a fixed shaking phase of $\varphi = 90°$. The solid line is the corresponding test point $\boldsymbol{x}_{\mathrm{t}}$ to calculate the influence function $\mathcal{I}(\boldsymbol{x}_{\mathrm{r}}, \boldsymbol{x}_{\mathrm{t}})$.

marked with a black vertical line. The bars indicate the influence of the data from different micromotion phases and the same shaking frequency. The data shows that before the micromotion removal (blue), the influence distribution is maximized around the test point, hence the identification of the test point is highly influenced by data with the same micromotion phase. As we know micromotion does not pose any information of the underlying topological phase, the influence shows that even the supervised network is highly distracted by it, even though it can distinguish the different phases successfully.

The same analysis is employed for the data with a fixed micromotion phase. The data is plotted with the red bars in figure 4.9. In contrast to the influence before the micromotion fix, the influence to identify the test point given by the vertical line correctly is evenly distributed through all original micromotion phases. Note that the pre-processed images have only one micromotion phase, however, their original micromotion phase is plotted as a reference here.

Furthermore, it should be noted that the accuracy of the trained CNNs with the original and the fixed micromotion phase showed similar accuracy. As we see the supervised methods are also highly influenced by the different micromotion phases. However, due to the labeled training data and evenly distributed occurrence of different micromotion phases through the complete training set, the ML algorithm learns to ignore the micromotion phase as a relevant property for the different phases, which matches our expectations.

4.7.3 Bottleneck Analysis Revised

In this section, the bottleneck analysis of 4.7.1 is repeated with the ToF images with a fixed micromotion phase, i.e. the post-processed data from the VAE. The same AE is employed as in section 4.7.1. The dimensionality of the data is reduced by the trained auto-encoder to apply clustering. 10% of the complete post-processed data set is hidden during training of the AE as

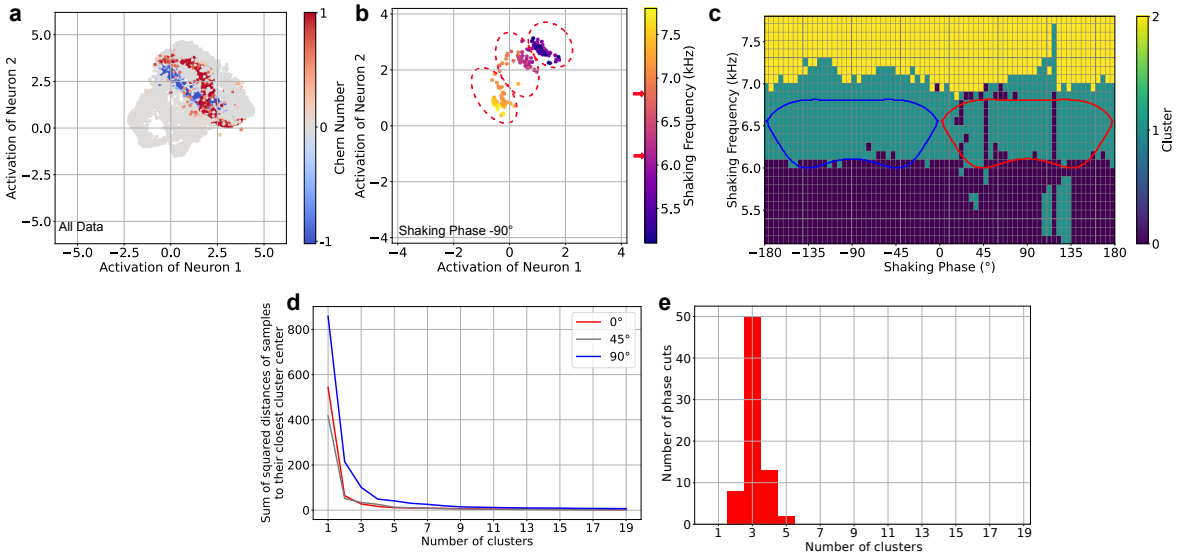

Figure 4.10: Bottleneck analysis of the rephased data from the Haldane model. (a) the new two-dimensional representation of all ToF images after rephasing to a fixed micromotion phase. The Chern number is assigned according to the theoretical predictions. (b) The bottleneck representation of data along a single shaking phase $\varphi = 90°$ cut. The clusters found by k-means clustering are noted by the red ellipses. The arrows on the color bar indicate the theoretical predictions. (c) The Haldane phase diagram after separating the bottleneck representation into three clusters by Kmeans clustering. (d) The inertia of the k-means clustering analysis as a function of the selected number of clusters for different shaking phases φ. (e) Histogram of how many phase cuts best fitted with the number of clusters according to the kneeling analysis.

a validation measure. The activation of the two bottleneck neurons is plotted in figure 4.10a for all data. The color encodes the Chern number according to the theoretical predictions. The complete dataset does not show any clusters, however, the data with topological non-trivial Chern numbers seem to cluster. As in section 4.7.1 the data is now restricted to single cuts for a fixed shaking phase. Figure 4.10b shows data belonging to a shaking phase of $\varphi = 90°$. Notice that this restriction does not allow us to extract clustering along the shaking phase axis of the phase diagram. We see three clusters that we can associate with the topological trivial and non-trivial regime of the phase diagram. We employ k-means clustering defined in algorithm 2 to associate each data point with a cluster. We use the implementation of k-means in the Scikit learn library [228] and set the number of clusters to 3 and the maximum number of iterations to 500. All other settings are left to the default values. Plotting each vertical line, i.e. for fixed shaking phases, in the Haldane phase diagram fashion, we can reconstruct the three different regions of the phase diagram as shown in figure 4.10c within good alignment of the theoretical predictions. For shaking phases around $120°$ and low shaking frequencies, there are some mismatched cluster assignments. We relate these to experimental noise as it is consistent with the analysis of anomaly detection in chapter 4.7.4 and does not appear around $-120°$ which would be expected due to the symmetry of the model.

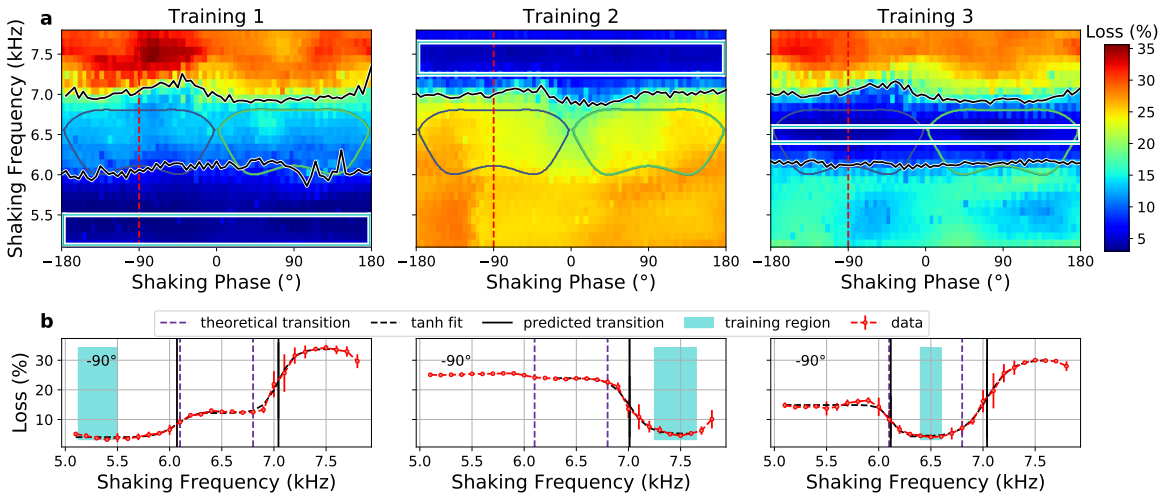

Figure 4.11: Anomaly detection scheme. The three different trainings (columns) correspond to different training regions, i.e. the AE is trained with data inside the blue box. In panel (a) the loss function of the complete Haldane phase diagram is color encoded. The solid lobes in the middle correspond to the theoretical predictions. The other lines are the extracted transitions. Panel (b) shows the cuts along a shaking phase of $\phi = -90°$ with three plateaus corresponding to the topological trivial and non-trivial regions.

The selection of clusters is crucial to k-means analysis and can be guided by the kneeling analysis that is based on finding the knee of the k-means inertia for different numbers of clusters. In figure 4.10d we show different inertia curves for three different shaking phase cuts along the phase diagram. They all have a clear knee at around 3 clusters which validates our first assumption of choosing three different clusters. Plotting a histogram of all shaking phase cuts even strengthens this assumption (see figure 4.10e).

In a nutshell, we can differentiate the topological trivial from the non-trivial regimes with the bottleneck analysis. We note that extraction of the different non-trivial Chern numbers $C = \pm 1$ is not possible as no clusters appear analyzing cuts for fixed shaking frequencies.

4.7.4 Anomaly Detection

As introduced in section 3.3.2 the learning success-based method of anomaly detection allows the detection of phase boundaries by training an AE in different regions of the phase diagram and comparing the reconstruction loss, i.e. in most cases the MSE loss $\mathcal{L}_{\mathrm{MSE}}$, inside the training region with the loss from other regions of the parameter space. If the loss value is significantly higher the data belongs to another phase. Here we use a AE with 50 bottleneck neurons. The full details of the AE structure can be found in the appendix C.4.

Here we start by training the AE for shaking frequencies f_{sh} from 5 kHz to 5.5 kHz and a shaking phase $\varphi \in [-180°, 180°]$ as indicated with the cyan boy in figure 4.11a in the column of training 1. The loss outside the regions reveals two plateaus between which we obtain two

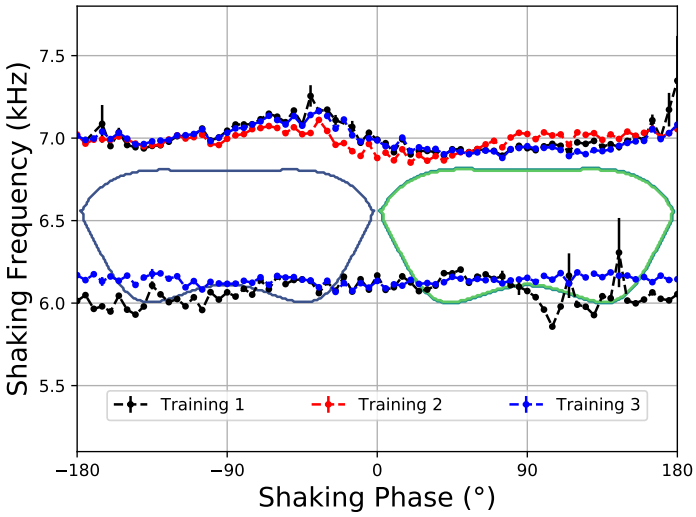

Figure 4.12: Reconstructed phase boundaries of the Haldane model with the anomaly detection scheme as in figure 4.11. All three results align with each other and lead to the assumption of three different phases. The phase boundaries are aligned with the theoretical predictions (solid lines), however, the higher frequency boundary is shifted to higher shaking frequencies f_{sh}.

boundaries by fitting hyperbolic tangent to the loss for each vertical cut through the phase diagram, i.e. for fixed shaking phases φ. The extracted phase boundaries are indicated by the black lines inside the two-dimensional phase diagram of the upper row for training 1 in figure 4.11. The single cut for a shaking phase of $\varphi = 90°$ in the lower row displays the details of the loss landscape and confirms the shift to higher shaking frequencies of the upper phase boundary.

The process is now guided unsupervised by selecting one of the plateaus as the new root to train to confirm the found phase transitions. The second training (center column of figure 4.11) is performed with data for shaking frequencies from 7.25 kHz to 7.65 kHz and again covering the complete shaking phase space from $\varphi = -180°$ for $\varphi = 180°$. The reverse picture of training 1 is revealed. Now plateaus arrive at similar regions as in training 1 with decreasing shaking frequency. The second plateau for frequencies below 6.1 kHz is not very present, and thus can not be extracted by the hyperbolic tangent model. However, the first training is confirmed.

Last but not least we perform training inside the intermediate region for shaking frequencies between 6.4 kHz and 6.6 kHz covering the complete shaking phase again. Two plateaus are revealed in the lower and upper-frequency region and two phase transitions can be extracted with the hyperbolic tangent method.

Notice that all three trainings are independent of each other, i.e. all parameters of the AE are reset after each training. All extracted phase boundaries align well with each other as shown in figure 4.12. Aside from some noise at shaking phases around $100°$ which aligns

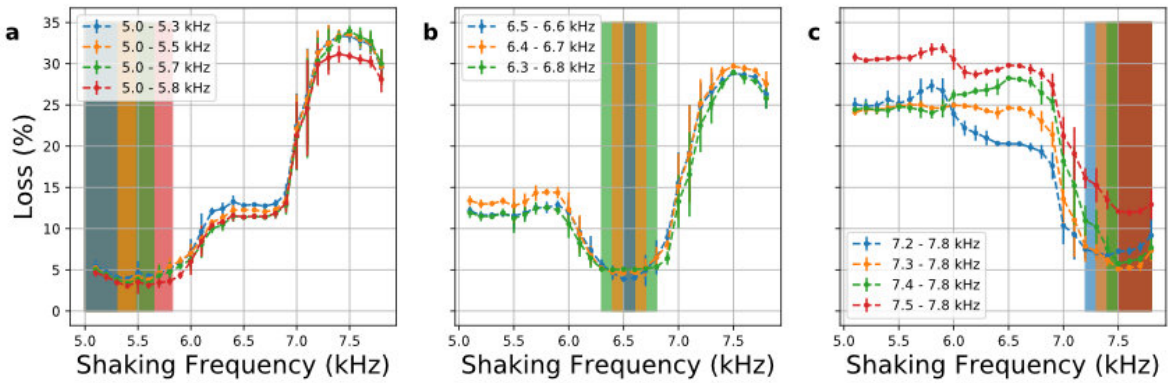

Figure 4.13: Influence of the training regions to the boundaries.

with the irregular clustering in chapter 4.7.3 the errors of the extracted boundaries are also very small. The good alignment also shows the validity of the method. The robustness of the phase boundaries can be improved by performing tests with different training regions varying the shaking frequency intervals. Figure 4.13 shows the robustness of the detected phase boundaries. The boxes show the training regions varied by independently training the AE. Although the training regions getting closer to the phase expected phase transition the plateaus do not change their position. The training in higher frequency in figure 4.12c shows some smaller instabilities concerning the overall loss, however, the expected phase transitions do not shift along the shaking frequency axis.

In a nutshell, anomaly detection can distinguish the topological trivial, i.e. $C = 0$, from topological non-trivial, i.e. $C = \pm 1$, regions. Unfortunately, it is not possible to distinguish between $C = 1$ and $C = -1$ with anomaly detection as the method does not produce consistent results. Details can be found in appendix B.2.

4.7.5 Data Similarity with Influence Functions

As the clustering approach in chapter 4.7.3 and the anomaly detection scheme described in chapter 4.7.4 are not able to distinguish the two different nontrivial regimes with the Chern number $C = \pm 1$ we employ the semi-supervised approach of influence functions as introduced in chapter 3.3.2. We employ a CNN (see table C.5) to classify the three different phases that the anomaly detection extracted in chapter 4.7.4, i.e. three different labels can be assigned to the data by the classifier. We then calculate the influence function $\mathcal{I}$ for a test point in each of the three regions as introduced in chapter 2.7.2. Notice that we choose the influence function and not the similarity here. As they are directly linked to each other, there is no disadvantage in using the influence function over the similarity measure.

In figure 4.14 the three different regions are analyzed. First, we choose a test point in the low-frequency regime (figure 4.14a). We see that inside the low and high-frequency regime

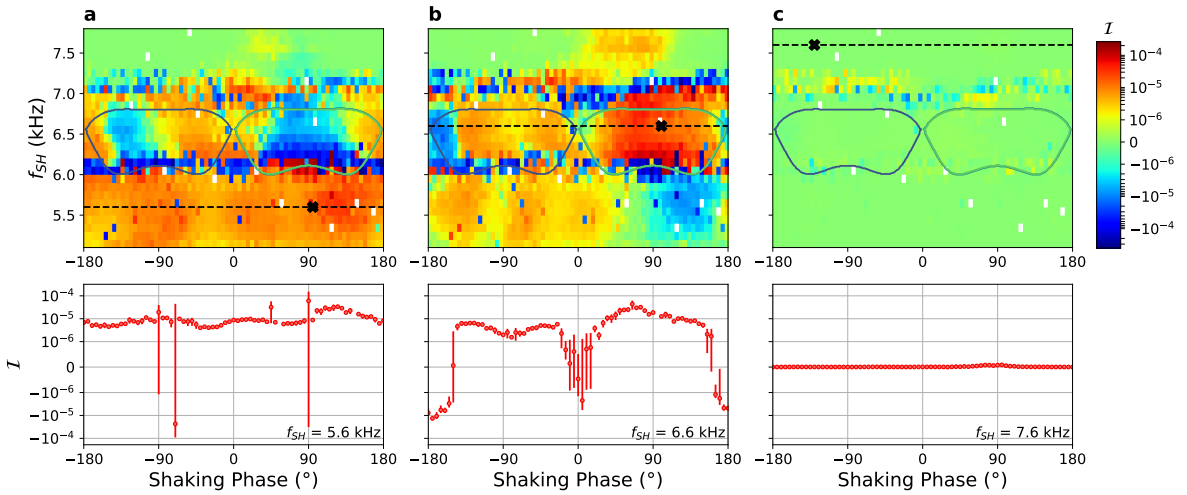

Figure 4.14: Similarity analysis of the data within the three phases detected by anomaly detection in chapter 4.7.4. In the upper row, the influence function is calculated for the complete Haldane phase diagram and color-coded. The test point is marked by a black cross. The test points belong to the low-frequency (a), intermediate frequency (b), and high-frequency regime (c) derived by the anomaly detection scheme. The lower row corresponds to horizontal lines through the phase diagram, i.e. for a fixed shaking phase f_{sh}, marked by the dashed line in the upper row. While (a) and (c) do not show any artifacts in the influence functions value, (b) has a dip at a shaking phase of $\phi = 0°$ and at the border of the phase diagram at $\phi = \pm 180°$ indicating a phase boundary.

plateaus of similar influence exist, whereas in the intermediate frequency regime between 6.0 to 7.0 kHz a distinction between positive and negative shaking phases appears. Looking at a single cut for a fixed shaking frequency of 5.5 kHz the influence also does not vary. Hence we conclude that only one phase is present in the lower frequency regime as the data is similar according to the CNN, i.e. the influence function is constant in the region.

The training in the intermediate shaking frequency region is plotted in figure 4.14b. Here we observe a clear distinction between the two regions that we expect for $C \pm 1$. Notice that the two regions are similar, however, they are separated by a dip with low similarity as posed in the single cut analysis for $f_{\mathrm{sh}} = 6.6$ kHz. Due to the periodicity of the model, the dips also occur at around $\varphi = \pm 180$ which strengthens the observation. The actual similarity values are higher for the region around the test point, i.e. the region for $C = 1$, in comparison to the $C = -1$ region. The average values of the plateaus differ by one order of magnitude which allows the conclusion that two different phases are present. Notice that this observation was not feasible with the other methods.

In panel c of figure 4.14 a test point in the high-frequency regime above 7 kHz is chosen. We observe a nearly uniform distribution of the influence function and hence similarity which is counterintuitive. Notice that the influence function in that case is almost zero everywhere. We can address this behavior to the fact that the test point is predicted with extremely high

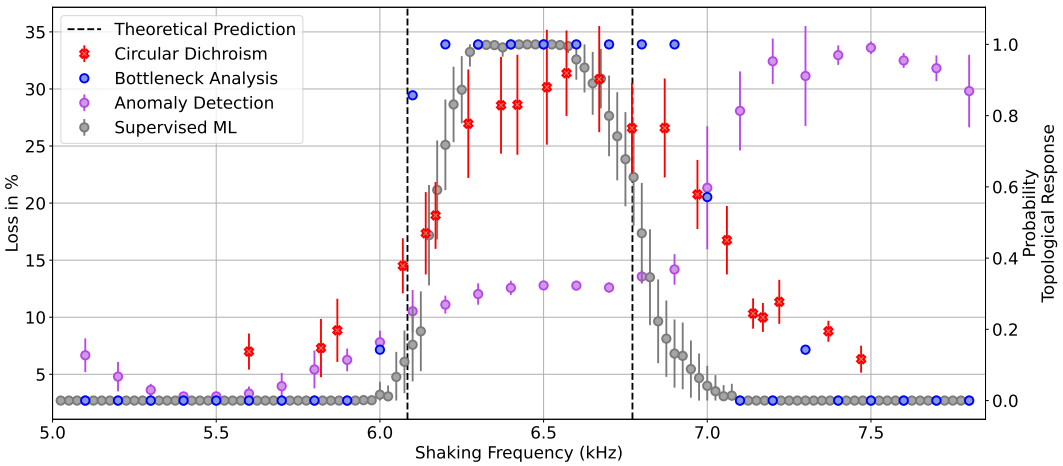

Figure 4.15: Comparison of the different methods to detect the topological phases of the Haldane model for a fixed shaking phase $\varphi = -90°$. The circular dichroism data is plotted as the topological response [89], the supervised learning [40] and bottleneck analysis with KMeans clustering [43] are given as probabilities and the anomaly detection data is given as loss percentage [43].

certainty of the CNN. This leads to low values of the influence function as it is proportional to the uncertainty of the prediction.

4.8 | Method Comparison

The previous sections presented a colorful mix of different ML methods that successfully reconstruct the Haldane phase diagram which is unfeasible with traditional methods. Figure 4.15 compares most methods along a single cut at a shaking phase of $\varphi = -90°$. A comparison of the complete Haldane diagram is still unfeasible because traditional methods do not provide the data for the parameter set studied here since they normally rely on the high population of the lowest Floquet band. Almost all methods tend to predict a slightly higher shaking frequency for the phase transitions at around 6.8 kHz. However, the machine learning predictions are in good agreement with the measurement of the circular dichroism. Therefore we think that there is no systematic shift in the phase ML models but instead a shift between the numerical predictions and the experimental situation. The shift to higher shaking frequencies may be a result of experiment calibration or numerical assumptions in the theory. Both explanations are under investigation but beyond the scope of this thesis.

The bottleneck analysis method lacks information about the error bars and is therefore not very well comparable. It also suffers from different steps that might fail if the algorithm is generalized. There is no guarantee that the data clusters at the bottleneck of the AE. The most successful unsupervised method is the anomaly detection scheme. It is rather general and does

not rely on clustering approaches like the bottleneck analysis approach. In comparison to the circular dichroism measurements, the anomaly detection scheme can find the two anomalies in the data differentiating the topological trivial and nontrivial regions with relatively small errors. Since the similarity comparison with influence functions relies on other unsupervised methods in the case of the experimentally realized Haldane model, it is not comparable in this particular context.

4.9 | Summary

In this section, we showed that the Haldane phase diagram can be reconstructed in an unsupervised fashion from experimental cold quantum gas data for the first time. We applied different ML methods to achieve the separation of topological phases by phase boundaries. To analyze the data with unsupervised machine learning techniques it has to be distraction-free. In our case, the micromotion was dominant in the data and it had to be removed by a VAE with an additional question neuron. We confirmed that the micromotion is successfully removed by using influence functions.

Assuming that there is no knowledge of the underlying phase diagram, the boundaries extracted by unsupervised machine learning methods separate four different regions. A low shaking frequency regime below roughly 5 kHz and a high shaking frequency regime above roughly 7 kHz which we successfully detected with clustering in the dimensionality reduced bottleneck data of an AE and with anomaly detection. The intermediate frequency regime between 5 and 7 kHz is separated into two distinct regions which we were able to distinguish by similarity analysis using an influence function approach. Notice that this extraction could have been possible without any prior knowledge of the underlying physics. Since the associated phases of the different regions are known, the regions can be associated with their topological indices. For the low and high-frequency regime, the Chern number is trivial $C = 0$ and for the distinct two regions in between the non-trivial Chern numbers $C = \pm 1$ can be assigned. Hence the complete phase diagram can be extracted in an unsupervised manner by combining the different methods.

We showed that unsupervised ML can identify phase boundaries even from noisy experimental data despite the low band population of the lowest band which is unfeasible with traditional methods. In the future, strongly correlated systems can be analyzed driven by experiments to identify new exotic phases of matter with hidden order [42] or explore numerical unfeasible regions in the parameter space.

CHAPTER 5

State Tomography

In this chapter, we will propose a new algorithm to perform quantum state tomography from random quench dynamics. The work in this chapter has not been published yet. The idea stems from a collaborative team including Alexandre Dauphin, Paolo Stornati under the supervision of Christof Weitenberg, Klaus Sengstock and Maceij Lewenstein. Due to the lack of time, it was not possible to prove the algorithm in its general form. Nevertheless, I will discuss the idea and the possible advantages of the new proposed method and the obstacles we encountered during the implementation.

In the first section 5.1 of this chapter I will introduce the concept of neural quantum state (NQS) that evolved as a subset of variational methods [14]. I will describe the needed mathematical framework to work with such states including sampling and time evolution.

In section 5.2 I will describe how phase tomography is done with NQS [15]. I will expand this method further in section 5.3 where I introduce a new ansatz for state tomography and discuss the challenges that appear when employing it.

5.1 | Neural Network Quantum States

At the heart of quantum mechanics is the Schrödinger equation exposing the wave function living in a Hilbert space governed by a Hamiltonian as a description of quantum mechanical states. The elegant description with just one equation however encounters some problems when it has to be solved for many interacting particles and it quickly became clear that approximation methods are needed to tackle these problems [229]. The problem of exploding Hilbert space can be referred to as the quantum many-body problem. NQS as a subset of variational states can solve this issue [14] and are, together with their mathematical framework, introduced in this chapter. As the quantum many-body problem poses exponential growth of the Hilbert space, it is noted that physically relevant quantum states only live in small corners of the Hilbert space.

5.1.1 Variational States

Variational states utilize the fact that only small parts of the Hilbert space are of physical interest. They represent the wave function by a parameterized function with a polynomial number of parameters $\boldsymbol{\theta}$ in system size. Hence the state can be stored on classical computers with limited memory. Variational states in general can be expanded to the computational basis by

$$|\psi_{\boldsymbol{\theta}}\rangle = \sum_{s} \psi_{\boldsymbol{\theta}}(s) \, |s\rangle \tag{5.1}$$

where $|\psi_{\boldsymbol{\theta}}\rangle$ is the variational state and $\psi_{\boldsymbol{\theta}}(s) = \langle s|\psi_{\boldsymbol{\theta}}\rangle$ is the amplitude of the corresponding basis state $|s\rangle$. Similar to the training of ML models as introduced in section 2.4 the task is to find the optimal parameters $\boldsymbol{\theta}$ that describe the quantum state best.

Variational states that are practicable, i.e. polynomial in time to calculate expectation values, can be sorted into two families. Those with the ability to calculate expectation values exactly, e.g. the mean field ansatz or matrix product states [230–232], and those where the expectation values are only calculated approximately, e.g. the Jastrow ansatz [233] based on the assumption that two-body interactions are the most physically relevant ones and NQS [14] based on the observation that NNs can represent entanglement and higher-order correlations and autonomously finding the corresponding features. The latter are the ones of interest in this thesis. Sampling however poses a source of error which can be reduced to additional computational cost, i.e. sampling more.

Here only computationally tractable states are discussed [234]. According to the definitions in the article, two conditions are met for computationally tractable states. Firstly it has to be possible to generate samples from the Born distribution

$$p(s) = \frac{|\langle s|\psi_{\boldsymbol{\theta}}\rangle|^2}{\langle \psi_{\boldsymbol{\theta}}|\psi_{\boldsymbol{\theta}}\rangle} = \frac{|\langle s|\psi_{\boldsymbol{\theta}}\rangle|^2}{\sum_s |\langle s|\psi_{\boldsymbol{\theta}}\rangle|^2} \tag{5.2}$$

where we inserted $\sum_s |s\rangle \langle s| = 1$ in the denominator and secondly the complex amplitudes for single basis elements $\psi_{\boldsymbol{\theta}}(s) = \langle s|\psi_{\boldsymbol{\theta}}\rangle$ can be computed efficiently, i.e. in polynomial time. Fulfilling both conditions it is possible to estimate expectation values of k-local operators, i.e. an operator acting on at least k local quantum numbers. The statistical error can be efficiently controlled by increasing the number of samples.

Defining the local estimator

$$O_{\mathrm{loc}}(s) = \sum_{s'} \left\langle s \middle| \hat{O} \middle| s' \right\rangle \frac{\langle s'|\psi_{\boldsymbol{\theta}}\rangle}{\langle s|\psi_{\boldsymbol{\theta}}\rangle} \tag{5.3}$$

of the k local operator $\hat{O}$ it can be seen that the estimation of the expectation value is given by

$$\langle \hat{O} \rangle = \frac{\left\langle \psi_{\boldsymbol{\theta}} \middle| \hat{O} \middle| \psi_{\boldsymbol{\theta}} \right\rangle}{\langle \psi_{\boldsymbol{\theta}} | \psi_{\boldsymbol{\theta}} \rangle} = \sum_s p(s) O_{\mathrm{loc}}(s) \equiv \langle O_{\mathrm{loc}} \rangle_p. \tag{5.4}$$

Notice that this is only tractable if the number of non-zero matrix elements of the operator $\hat{O}$, i.e. $\left\langle s' \middle| \hat{O} \middle| s \right\rangle \neq 0$, is at most polynomial. For k-local operators, this is fulfilled. An efficient and controllable way to calculate the expectation value is the estimation

$$\langle \hat{O} \rangle \approx \frac{1}{n} \sum_{i=1}^{n} O_{\mathrm{loc}}(s^i) \tag{5.5}$$

where s^i is a series of n sampled states from the born distribution $p(s)$. Notice that the statistical error of this estimate is $\epsilon = \sqrt{\frac{\sigma^2}{n}}$ where σ^2 is the variance of O_{loc}. It is possible to show that σ^2 is strictly finite for k local operators, hence the error scales with $\epsilon \propto \frac{1}{\sqrt{n}}$ and the error can be diminished with a larger sample set.

It should be noted here that drawing samples from the Born distribution is not in all cases a computational cheap operation. It is useful to employ Markov chain Monte Carlo (MCMC) methods, e.g. the Metropolis-Hasting algorithm [235], to draw a sequence of samples from $p(s)$. The details of this algorithm are beyond the scope of this thesis, however well explained in [12].

5.1.2 Neural Quantum States

A special class of variational ansätze are the neural quantum states (NQSs) which have been introduced with restricted boltzmann machines (RBMs) by Carleo and Troyer in 2017 [14]. Instead of using assumptions about the inner structure of the state, the inner structure is governed by NNs as introduced in chapter 2.6. Different NN architectures including RBMs [14] and autoregressive neural networks (ARNNs) [125, 236, 237], have been proven successful in representing a plethora of different quantum systems. The idea is to represent the wave function by a NN, e.g. with a feed-forward NN where $\psi_{\boldsymbol{\theta}}(s)$ corresponds to the output and s is the input to the network. Notice that NN can by design not preserve the structure of the Hilbert space as they are not linear. Hence it is not possible to add two quantum states and receive a valid superposition of both.

As discussed NQS employ a versatile ansatz to represent many different quantum systems. This raises the question of how they compete with other methods of choice like tensor network states (TNSs). As they relate very closely and solve similar problems, the relation between TNSs and NQSs are extensively studied [238–240]. It is possible to prove that for each TNS there exists a NQS of polynomial size approximating it with arbitrarily high precision [240]. Therefore TNSs are a subset of NQSs.

Restricted Boltzmann Machines

As an introduction to all possible NQS ansätze is beyond the scope of this thesis, hence I will only introduce RBMs. A broader overview of the different methods can be found here [12]. RBMs are a special kind of NN introduced as harmonium [241] and later successfully trained by contrastive divergence [242] before the advent of DNNs started. In fact RBMs are so-called energy-driven models because their structure allows one to assign a model energy which is beneficial interpretability in contrast to modern DNNs. They can be considered as a special class of Hopfield network - a type of spin glass - which is discussed in detail here [243].

RBMs featuring two fully connected layers. A visible layer and a hidden layer modeled by the in general complex-valued bias terms $\boldsymbol{b}_v$ for the visible layer and $\boldsymbol{b}_h$ for the hidden layer and their weight matrix $\boldsymbol{W}$ connecting the n_v neurons of the visible and n_h neurons of the hidden layer. The wave function of the variational ansatz for RBMs is given by

$$\psi_{\boldsymbol{\theta}}(\boldsymbol{s}) = \sum_{\boldsymbol{h}\in\mathcal{H}} e^{\boldsymbol{b}_v^\dagger \boldsymbol{s} + \boldsymbol{b}_h \boldsymbol{h} + \boldsymbol{h}^\dagger \boldsymbol{W} \boldsymbol{v}} \tag{5.6}$$

where $\boldsymbol{s}$ is the sampled state s, now represented as a vector of quantum numbers, e.g. the spin on each site, and $\boldsymbol{h}$ is one hidden state of the RBM from the set of all possible hidden states $\mathcal{H}$. At first glance that might pose issues because the sum of all hidden states may be computationally costly. Fortunately RBMs are designed to tackle that problem efficiently allowing only binary hidden units, i.e. $h_i \in \{-1, 1\}$. It is straightforward to show that

$$\psi_{\boldsymbol{\theta}}(\boldsymbol{s}) = e^{\boldsymbol{b}_v^\dagger \boldsymbol{s}} \prod_{i=1}^{n_h} 2\cosh(b_{h,i} + \boldsymbol{W}_{i,\cdot}\boldsymbol{s}) \tag{5.7}$$

where $\boldsymbol{W}_{i,\cdot}$ denotes the i-th row in the weight matrix. This analytical form with complex weights and biases allows the full representation of a quantum state featuring interesting properties regarding the capacity and quantum entanglement [238, 244].

5.1.3 Ground State Search

It appears natural to use NQS to find a ground state, i.e. minimizing the energy, since minimizing a loss function is at the heart of most ML models. As already introduced in section 5.1.1 it is possible to calculate the expectation value of a k local operator and thus many Hamiltonians

$$\mathcal{L}(\boldsymbol{\theta}) = E(\boldsymbol{\theta}) = \left\langle \psi_{\boldsymbol{\theta}} \middle| \hat{H} \middle| \psi_{\boldsymbol{\theta}} \right\rangle \tag{5.8}$$

where $E(\boldsymbol{\theta})$ is the energy of the NQS for a given parameter set $\boldsymbol{\theta}$. This follows the variational principle of quantum mechanics which states that given a Hamiltonian $\hat{H}$, the energy of the

variational wave function $\psi_{\boldsymbol{\theta}}$ is greater or equal to the energy of the real ground state E_0

$$E(\boldsymbol{\theta}) = \frac{\left\langle \psi_{\boldsymbol{\theta}} \middle| \hat{H} \middle| \psi_{\boldsymbol{\theta}} \right\rangle}{\langle \psi_{\boldsymbol{\theta}} | \psi_{\boldsymbol{\theta}} \rangle} \geq E_0 \tag{5.9}$$

and minimizing the proposed loss function is consequently a good choice. Notice that calculating the energy of a variational ansatz is also helpful in comparing them with other variational methods. If the reached energy is lower than achieved with other employed methods, it can be considered to be better in representing the ground state.

The local estimator introduced in equation 5.3 of the Hamiltonian is

$$E_{\mathrm{loc}}(s) = \sum_{s'} \left\langle s \middle| \hat{H} \middle| s' \right\rangle \frac{\langle s' | \psi_{\boldsymbol{\theta}} \rangle}{\langle s | \psi_{\boldsymbol{\theta}} \rangle} \tag{5.10}$$

the variational energy can be approximated by using equation 5.5. Defining the diagonal operator

$$O_m(s) = \frac{\partial}{\partial \theta_m} \log(\langle s | \psi_{\boldsymbol{\theta}} \rangle) = \left\langle s \middle| \hat{O}_m \middle| s \right\rangle \tag{5.11}$$

for the m-th parameter of the ansatz θ_m it is possible to express the gradient of the approximated energy by

$$\frac{\partial E(\boldsymbol{\theta})}{\partial \theta_m} = 2\Re \left(\langle E_{\mathrm{loc}}(s) O_m^*(s) \rangle - \langle E_{\mathrm{loc}}(s) \rangle \langle O_m^*(s) \rangle \right) \tag{5.12}$$

where real-valued parameters, i.e. $\theta \in \mathbb{R}$ and $\Re$ denotes the real part. Notice that it is possible to also derive a similar expression for complex-valued parameters, i.e. $\theta \in \mathbb{C}$, however, one has to be careful with non-holomorphic ansätze like ARNNs.

The optimization and thus the search for the ground state follows the same structure as SGD introduced in algorithm 1 and is straight forward.

5.1.4 Real Time Evolution

As NQS are variational states they can perform variational real-time evolution using time-dependent variational Monte Carlo (t-VMC) [14, 245, 246], which is based on the Dirac-Frenkel variational principle. In particular, the time-dependent Schrödinger equation needs to be solved for a NQS $|\psi_{\boldsymbol{\theta}}(t)\rangle$

$$i \frac{d}{dt} |\psi_{\boldsymbol{\theta}}(t)\rangle = \hat{H} |\psi_{\boldsymbol{\theta}}(t)\rangle \tag{5.13}$$

governed by a Hamiltonian $\hat{H}$ and $\hbar = 1$. To achieve this the Schrödinger equation is expanded in the first order of the time step τ and multiplied by the basis state $\langle s|$ we receive

$$\psi_{\boldsymbol{\theta}}(t+\tau)(s) = 1 - \hbar\tau \left\langle s \middle| \hat{H} \middle| \psi_{\boldsymbol{\theta}}(t) \right\rangle + \mathcal{O}(\tau^2) \tag{5.14}$$

$$= 1 - i E_{\mathrm{loc}}(s) + \mathcal{O}(\tau^2) \tag{5.15}$$

where

$$E_{\mathrm{loc}} = \sum_{s'} \left\langle s \middle| \hat{H} \middle| s \right\rangle \frac{\langle s'|\psi\rangle}{\langle s|\psi\rangle} \tag{5.16}$$

is the local estimator of the energy. Notice that numerically that is appealing since only nonvanishing matrix elements of the Hamiltonian need to be calculated. As in most ML approaches and thus true for NQS it is helpful to define a loss function to minimize. Here the loss function of the new parameters after the times step τ are $\boldsymbol{\theta}'$, so the loss function reads

$$\mathcal{L}(\boldsymbol{\theta}') = \mathrm{dist}\left(\psi_{\boldsymbol{\theta}'}, \psi_{\boldsymbol{\theta}}(t+\tau)\right). \tag{5.17}$$

Notice that the loss function can be minimized analytically by selecting a timestep τ that is sufficiently small. In that case the new variational parameters $\boldsymbol{\theta}'$ can be written in first-order

$$\boldsymbol{\theta}' = \boldsymbol{\theta} + \tau\dot{\boldsymbol{\theta}} + \mathcal{O}(\tau^2). \tag{5.18}$$

With this observation the new state $\psi_{\boldsymbol{\theta}'}$ can be expanded at first order in τ

$$\psi_{\boldsymbol{\theta}+\tau\dot{\boldsymbol{\theta}}}(s) = \left(1 - \tau\dot{\boldsymbol{\theta}}\right)\psi_{\boldsymbol{\theta}}(s) + \mathcal{O}(\tau^2) \tag{5.19}$$

where we multiplied with $\langle s|$. The choice of the distance measure varies the underlying variational principle [247]. Taking into account that most NQS are not normalized the choice here is

$$\mathrm{dist}(\psi, \phi) = 1 - \frac{\langle\psi|\phi\rangle\,\langle\phi|\psi\rangle}{\langle\psi|\psi\rangle\,\langle\phi|\phi\rangle} \tag{5.20}$$

for two wavefunctions ψ and ϕ, which is similar to the Fubini-Study metric normally used for NQS. Combining equation 5.14 and 5.19 with the distance measure and only keeping the leading terms in τ we obtain an equation to calculate $\dot{\boldsymbol{\theta}}$

$$\boldsymbol{S}\dot{\boldsymbol{\theta}} = -i\boldsymbol{f} \tag{5.21}$$

with the quantum geometric tensor S and the gradient of the local energy with respect to the parameters $\boldsymbol{\theta}$. The elements of the quantum geometric tensor are given by

$$\boldsymbol{S}_{s,s'} = \langle O_s^* O_{s'}\rangle - \langle O_s^*\rangle\langle O_{s'}\rangle \tag{5.22}$$

and the gradient

$$\boldsymbol{f}_s = \langle E_{\mathrm{loc}} O_s^*\rangle - \langle E_{\mathrm{loc}}\rangle\langle O_s^*\rangle \tag{5.23}$$

which is also known as the vector of forces in this context. The system of linear equations raised by equation 5.21 can be solved by inverting $\boldsymbol{S}$ to get an explicit form of the parameter time derivative

$$\dot{\boldsymbol{\theta}} = -i\boldsymbol{S}^{-1}\boldsymbol{f} \tag{5.24}$$

which now can be used to formulate an iterative approach for the time evolution of the NQS parameters choosing a time step size of Δt

$$\boldsymbol{\theta}'(\boldsymbol{\theta}) = \boldsymbol{\theta} - i\Delta t \boldsymbol{S}^{-1}\boldsymbol{f} \tag{5.25}$$

where $\boldsymbol{\theta}'$ are the parameters after the time step. In practice inverting $\boldsymbol{S}$ can be achieved by using a direct solver like QR factorization. However, in some cases that poses problems as $\boldsymbol{S}$ is often singular. Multiple different techniques have been developed to overcome that problem, but hence $\boldsymbol{S}$ and $\boldsymbol{f}$ are sampled stochastically stable long-time evolutions are challenging for NQS [248].

Instead of the variational principle as introduced in the section 5.1.3 to find the ground state, it is clear that imaginary time evolution using Wicks theorem and replacing $\tau \to i\tau$ which leads to the evolution of theta defined by

$$\dot{\boldsymbol{\theta}} = -\boldsymbol{S}^{-1}\boldsymbol{f}. \tag{5.26}$$

Employing this method normally involves regularizing $\boldsymbol{S} \to \boldsymbol{S} + \lambda \mathrm{Id}$ with a constant term λ on the diagonal. This algorithm is also known as stochastic reconfiguration [249].

5.2 | Quantum State Tomography

NQS can be employed to perform quantum state tomography [15]. The formal task is to reconstruct a wave function ψ from a limited amount of projective measurements $|\psi(s)|^2$ in a basis spanned by $|s\rangle = |s_1, s_2, \cdots, s_N\rangle$ with s_i is some quantum number and N is the system size. In terms of ML as introduced in chapter 2.4 the optimization is guided by simply minimizing

$$\min_{\boldsymbol{\theta}} \mathrm{dist}(\psi, \psi_{\boldsymbol{\theta}}) \tag{5.27}$$

where $\psi_{\boldsymbol{\theta}}$ is a variational state, e.g. a NQS, with some parametrization $\boldsymbol{\theta}$. For now, the choice of the ansatz or NN representing the variational state is left open, however in the proposed tomography ansatz it has to be constrained (see section 5.3). The distance between the two states can be measured by different functions. The Kullback-Leiber divergence, i.e. a measure for the distance between two probability distributions, is employed here and is proven successful in the field of state tomography with NQSs [15]. Recall from equation 2.13 that the Kullback-Leibler divergence for two probability distributions p and q reads as

$$D_{KL}(p \parallel q) = \mathbb{E}_{x\sim p(x)}\left[\log \frac{p(x)}{q(x)}\right] \tag{5.28}$$

$$= \sum_{x\in X} p(x) \log \frac{p(x)}{q(x)} \tag{5.29}$$

as we work with discrete probability distributions here. The Born rule can be used to get probability distributions from the two states that should be optimized here, i.e. $p_\theta(s) = |\psi_\theta(s)|^2$ and $q(s) = |\psi(s)|^2$ where s is now a measurement, a try to define tomography could be

$$D_{KL}(\psi_{\boldsymbol{\theta}} \parallel \psi) = D_{KL}(\boldsymbol{\theta}) = \sum_{s \in S} |\psi_{\boldsymbol{\theta}}(s)|^2 \log \frac{|\psi_{\boldsymbol{\theta}}(s)|^2}{|\psi(s)|^2}. \tag{5.30}$$

However quantum states are in most cases not simple probability distributions due to their complex-valued nature. Whenever we use the Born rule to get the probability distribution from a quantum state the probability distribution can not be transformed back into the quantum state. This can be shown as representing a quantum state in its amplitude phase structure we encountered earlier $\psi(s) = \sqrt{p(s)}e^{i\phi(s)}$. By transforming the state into a probability distribution the information of the phase $\phi(s)$ is lost.

A solution to overcome the problem of lost phase information is to measure in different basis $B \in \mathcal{B}$ which is proven to be efficient [250]. So instead of minimizing the Kullback-Leibler divergence, it is generalized to different basis sets

$$\mathcal{D}_{KL}(\boldsymbol{\theta}) = \sum_{B \in \mathcal{B}} \sum_{s \in S_B} |\psi^B_{\boldsymbol{\theta}}(s)|^2 \log \frac{|\psi^B_{\boldsymbol{\theta}}(s)|^2}{|\psi^B(s)|^2} \tag{5.31}$$

where S_B is a set of measurements of the state in the basis B and $\psi^B(s) = \left\langle s \middle| \hat{U}_B \middle| \psi \right\rangle$ is the quantum state in the basis B. $\hat{U}_B$ is a unitary operator. The modified Kullback-Leibler divergence can now serve as a loss function and in terms of ML approaches it can be minimized with SGD and the gradients can be easily calculated using automatic differentiation in most cases.

5.2.1 Applications and Limitations

Efficient quantum state tomography is of high interest for experiments as it allows us to measure arbitrary observables from the reconstructed state. This was proven successful for an array of rubidium atoms [16] using an RBM ansatz for the NQS and recently on a 20 qubit trapped ion system [251] comparing different architectures. It should be mentioned here that noise poses a big problem for state tomography with NQS in its bare form. At the cost of additional parameters of the NN representing the NQS it is possible to add a noise layer to circumvent these issues [16].

In general state tomography from experimental data is challenging. Several proposals try to circumvent issues raised with experimental data [252–256]. In the end however it should be possible to perform state tomography with a polynomial number of different bases and snapshots for a special class. For now, this class is unknown and is subject to current research.

One large limitation of the state tomography protocol using different bases as the training set to learn the phase information of the wave function is the unitary basis transformation $\hat{U}_B$ introduced in equation 5.31. To calculate $\psi^B_{\boldsymbol{\theta}}(s)$ efficiently it is strictly required that $\hat{U}$ acts non-trivially on a limited number of quantum numbers, i.e. single qubits or spins. Otherwise, the computational costs scale exponentially with system size. See the appendix of [15] for details. This restriction limits the method to experimental systems where local unitary operations are feasible, e.g. Rydberg atom tweezer arrays like in [16].

5.3 | State Tomography from Random Quench Dynamics

As already seen in section 5.2 NQSs pose a versatile framework to perform quantum state tomography and have already been proven successful. However, they are often limited to experimental systems where only local unitary operations are feasible, due to the limited ability to transform the variational state $\psi_{\boldsymbol{\theta}}$ computationally. Here we propose a new ansatz to solve that problem.

We start with the ground state $|\psi_0\rangle$ of some quantum system defined by $\hat{H}$. Then we quench the system with a quench Hamiltonian $\hat{H}_q$ for a given time t_q. The evolution of the quantum state is governed by the time-dependent Schrödinger equation

$$i\hbar\frac{d}{dt}|\psi(t)\rangle = \hat{H}_q|\psi(t)\rangle \tag{5.32}$$

where $|\psi(t)\rangle$ is the time-dependent state. Afterwards, we measure the state by sampling $|\psi(s,t_q)|^2$. Instead of using different bases to avoid loss of phase information in the tomography scheme, we propose to use the quench Hamiltonian to sample sufficiently inside the Hilbert space. We do not specify the quench Hamiltonian $\hat{H}_q$ in further detail here, instead, we assume a random set of different quench Hamiltonians $\hat{H}_q \in \mathcal{H}_q$. The Kullback-Leibler divergence can now be adopted to that case and reads

$$\mathcal{D}_{KL}(\boldsymbol{\theta}(t_q)) = \sum_{H_q\in\mathcal{H}_q}\sum_{s\in S_{H_q}} |\psi_{\boldsymbol{\theta}(\boldsymbol{t_q})}(s)|^2 \log\frac{|\psi_{\boldsymbol{\theta}(\boldsymbol{t_q})}(s)|^2}{|\psi(s,t_q)|^2} \tag{5.33}$$

with $\boldsymbol{\theta}(t)$ now being the time-dependent parameters of the NQS and $\psi(s,t_q) = \langle s|\psi(t_q)\rangle$. Notice that the time dependence of the NQS is included in the parameters $\boldsymbol{\theta}(t)$ which can be derived from the time evolution introduced in 5.1.4. This poses a big problem in deriving the gradients from the loss function and use SGD to optimize the parameters of the variational NQS ansatz due to numerical limitations as discussed later in section 5.3.1. However, this ansatz allows to leverage of experimental platforms that are not able to apply local unitary operations to the system. We also hope that our approach allows state tomography of bosonic systems which are so far not feasible for state tomography with NQS.

Similar to the tomography approaches using time evolutions [210] we can extend our approach to different quench times to capture more parts of the Hilbert space. We extend our idea by adding several quench times $t_q \in \mathcal{T}_q$ and we propose the overall loss function for our tomography ansatz

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{t_q \in \mathcal{T}_q} \mathcal{D}_{KL}(\boldsymbol{\theta}(t_q)) = \sum_{t_q \in \mathcal{T}_q} \sum_{H_q \in \mathcal{H}_q} \sum_{s \in S_{H_q}} |\psi_{\boldsymbol{\theta}(\boldsymbol{t_q})}(s)|^2 \log \frac{|\psi_{\boldsymbol{\theta}(\boldsymbol{t_q})}(s)|^2}{|\psi(s, t_q)|^2} \tag{5.34}$$

to optimize the NQS. Notice that this approach in contrast to equation 5.31 does not require local $\hat{U}$ which is an advantage as we only rely on the quench Hamiltonian and the stable time evolution of the NQS parameters.

5.3.1 Results and Technical Difficulties

At the time of writing this thesis, it is not possible to perform quantum state tomography of non positive wave functions, i.e. wave functions containing phase information, with our ansatz proposed in equation 5.34. The root of the problems stems from the differentiation of the NQS parameters. As we are interested in the NQS parameters before the quench $\boldsymbol{\theta}$ each value and hence each gradient of the parameters has to be computed concerning the time evolution. To solve this we implemented two different methods which both could not be proven successful so far.

Our test setup uses RBMs to build the NQS as they are naturally good candidates for performing time evolution with t-VMC as they have complex valued parameters, i.e. $\theta \in \mathbb{C}$. This is needed since for real-time evolution the parameters are updated with complex values due to the imaginary unit in equation 5.21. As a test model, we use the one-dimensional XXZ model

$$\hat{H} = \sum_{<i,j>} J_x \sigma_i^x \sigma_j^x + J_y \sigma_i^y \sigma_j^y + J_z \sigma_i^z \sigma_j^z - \frac{h}{2} \sum_i \sigma_i^z \tag{5.35}$$

with three sites which we solve using exact diagonalization (ED). J_i are the coupling terms between neighboring sites and h is an external field. The Hilbert space has 8 basis states in that case. We start with a positive wave function, i.e. no phase information has to be obtained by the state tomography algorithm, by setting $J_x = 1.0$ and $h = 1.0$. The other coupling terms are set to 0. We checked that the RBM can learn the original state by performing state tomography using equation 5.30. Notice that for real-valued wave function, no local unitaries are required to perform a full tomography [15]. For the quenches, we vary $J_i \in [0, 1]$ and $h \in [0, 1]$ randomly.

The first approach includes going forward and backward in time. We start with randomly initialized parameters for the state at $t = 0$. Then we select a quench Hamiltonian $\hat{H}_q$ and

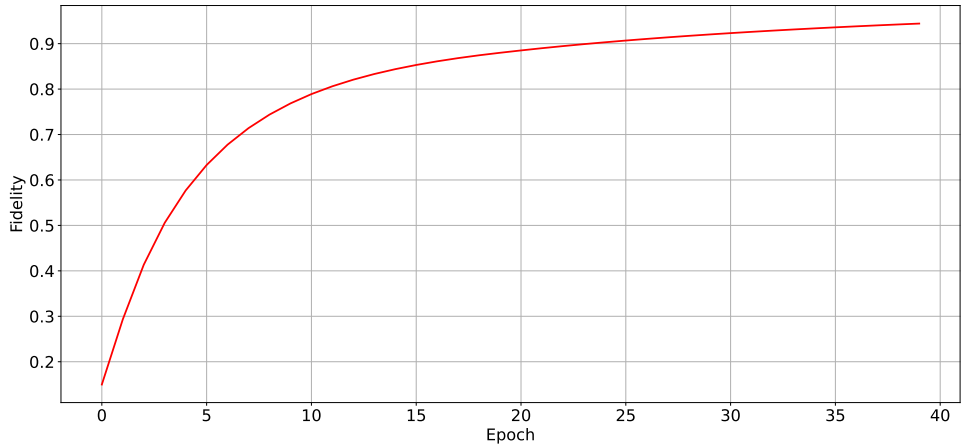

Figure 5.1: Fidelity of the ED wave function ψ_{ED} and the exact NQS wave function $\psi_{\boldsymbol{\theta}}$. Here two random quench Hamiltonians are used. The quench time is fixed and set to $t_q = 0.4$ reached with 30 time steps. A final fidelity of $\mathcal{F} = 0.945$ is achieved.

use real-time evolution to evolve the parameters to time t_q as introduced in chapter 5.1.4 with a sufficiently small time step τ. We now draw samples from the experiment at a t_q and minimize the loss function $\mathcal{D}_{KL}(\boldsymbol{\theta}(t_q))$ at the given time step of the quench. After minimizing the loss function we propagate the parameters back in time to $t = 0$ and repeat with another quench Hamiltonian and other quench time. This however poses multiple difficulties. First of all the time evolution is challenging and unstable for long-time evolutions [248]. In a pictorial picture, the information of the first optimization depletes from the NQS as more single times steps are made. Secondly, the SGD lives from the shuffling of data it gets presented in each mini-batch. Here we restrict each epoch of the gradient descent step to one time and quench Hamiltonian.

The second approach we tried so far includes differentiation through the complete time evolution of the parameters. Instead of training each time step individually we now calculate the backpropagation, i.e. building a complete computational graph, through all calculations involving the time evolution which allows us to still shuffle the data after each epoch for SGD. This method is more promising but poses computational difficulties as for each time step we need to calculate the quantum geometric tensor $\boldsymbol{S}$ and the gradient $\boldsymbol{f}$, i.e. the vector of forces, which involves sampling. The size of the computational graph explodes quickly reaching several gigabytes which makes this approach unfeasible. As a proof of concept, we also implemented an analytical approach. For small Hilbert spaces, i.e. small system sizes, it is possible to calculate all the Monte Carlo averages exactly without sampling.

Employing this ansatz we were able to perform state tomography for the amplitude of the wave function as shown in figure 5.1. Here the analytic RBM ansatz contained $n_v = 3$ visible neurons, i.e. one for each site of the chain, and $n_h = 9$ hidden units. The weights and biases are initialized by a normal distribution. Two random quench Hamiltonians are sampled. We

train at only one-time point $t_q = 0.4$ using 30 time steps. The learning rate is set to $\lambda = 0.01$. The fidelity curve increases for each epoch until we reach a final fidelity of $\mathcal{F} = |\langle\psi_{\boldsymbol{\theta}}|\psi_{\mathrm{ED}}\rangle|^2$. Unfortunately, it is so far not possible to learn the complex part of a wave function as we always observe divergence of the parameters in that case. We note that this ansatz is not scalable due to the analytical form. So other methods have to be developed to overcome that problem.

5.4 | Outlook

In this section, we introduced a new ansatz for quantum state tomography with NQS using random quench dynamics. We proved our method successful for wave functions without phase information and analytical RBMs. In the future, several properties have to be investigated further. As time evolution poses critical limitations to our method it is required to use different approaches to evolve in time which are tailored for NQSs [246]. Furthermore, we need to understand why wave functions containing phase information lead to a divergence of the parameters. This might be solved by using advanced regularization schemes for the quantum geometric tensor as introduced in [245, 248]. In addition different NQS architectures like ARNNs may overcome some of the encountered problems as they do not rely on Monte Carlo methods for sampling. In general, our ansatz poses new possibilities to perform quantum state tomography with NQS, including state reconstructions from bosonic systems which is unfeasible so far.

CHAPTER 6

Conclusion and Outlook

This thesis proves several ML methods successful in identifying different phases of matter from experimental data. The full unsupervised reconstruction of the Haldane phase diagram in chapter 4 proves, that it is possible to determine and distinguish different phases of matter without knowledge of the underlying phases from cold quantum gas experiments. The Haldane phase diagram poses three different topological regions. For high and low shaking frequencies topologic trivial phases, i.e. a zero-valued Chern number, and topological nontrivial phases in an intermediate shaking frequency regime, identified with nonzero Chern number. It was demonstrated that bottleneck analysis and anomaly detection can distinguish topological trivial from topological nontrivial regimes, hence the intermediate regime from the rest. The different signs of the Chern number correlated with the sign of the shaking phase, i.e. for positive shaking phases the Chern number is also positive and vice versa, could be extracted by employing influence functions.

To successfully apply unsupervised ML techniques it is required to reduce distraction inside the data set, e.g. the micromotion phase. The micromotion phase can be fixed to a single micromotion phase employing a generative model, namely VAE with an additional question neuron. It was possible to train a VAE which can transform each ToF image to a desired micromotion phase. This method of data post-processing of experimental data is on its own an exciting tool that might be applied to other fields of interest like fringe removal or noise reduction. Further applications also include the reconstruction of quantum gas microscope images [47].

The successful implementation of the unsupervised ML methods for noisy data poses new possibilities in analyzing experimental data even beyond cold quantum gases. In this context, the investigation of how noise influences phase classification is also of higher interest and should be researched in the future. In the future, the methods may be applied to complete unknown phase diagrams guiding theoretical predictions or to understand exotic quantum many-body systems or hidden orders [42, 46]. Unsupervised phase discovery is one crucial ingredient to self-driven experiments autonomously discovering new phases of matter and new functional materials [257].

NQS are a versatile tool and some works have proven their capability to represent volume-law entanglement, which means they will outperform TNS in two and three-dimensional systems [160, 240, 258]. Already today NQS outperform state-of-the-art techniques like projected entangled pair states (PEPS) [259]. This is a huge step forward since it allows quantum state tomography for such higher dimensional systems. In chapter 5 a novel ansatz for quantum state tomography for NQS is introduced and proven successful for wave functions without phase information. As of today, state tomography with NQS for experimental data [16, 251] already works and further advantages in the field will allow the proposed method to excel in different environments. The promising experimental idea of the quantum gas magnifier [81] may allow us to access the necessary information from quench dynamics to perform state reconstruction with our approach.

In conclusion ML will change the way scientific discovery is achieved. Even today scientists can get valuable new ideas and concepts from ML approaches which can lead to new scientific discoveries [260, 261]. Identifying trends in the scientific community helps to map out possible focus topics for policymakers [262]. At the heart of science is intuition and understanding of the observed effects which can be challenging and be assisted by AI [263, 264]. The ingredients posed by this thesis are desireable to design new AI driven laboratories [183]. With the introduction of large language models [1] a new era of AI assisted technology started, enabling the implementation of autonomous research assistants that assist the daily work of scientists in the lab [265, 266]. The progress of DL models propelled by large data sets and computational progress with all its benefits and possible risks will not only shape the scientific landscape. As concrete scientific applications of today DL models will arise, it will be interesting how for example large language models find their way into quantum physics.

APPENDIX A

Micromotion and Center of Mass

In figure A.1 we show the dependence of the azimuthal coordinate of the CoM on the micromotion phase. For circular shaking, i.e. a shaking phase of $\pm 90°$, the CoM moves in a circular fashion yielding a linear dependence between the azimuthal coordinate of the CoM and the micromotion phase. For linear shaking, i.e. a shaking phase of $0°$ and $180°$, the CoM moves along a diagonal line yielding a constant azimuthal coordinate of the CoM at $\pm 45°$, with a phase jump of $180°$ due to small disturbances in the experiment forcing an exchange of the long and short axis in the fitting scheme of the ellipse. Other shaking phases interpolate between these two behaviors. In conclusion, the movement of the CoM of the momentum distribution follows the shaking trajectories as expected.

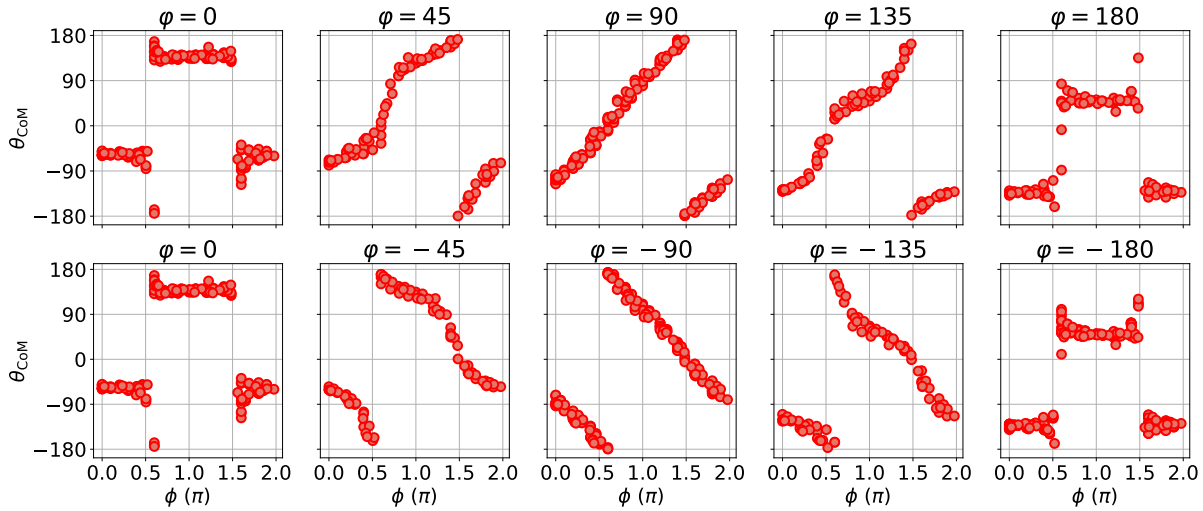

Figure A.1: Dependence of the azimuthal coordinate θ_{CoM} of the CoM and the micromotion phase ϕ. The dependence can be explained by the elliptical shaking. For the shaking phases $\varphi = 0, \pm 180$ the shaking is linear, thus the cloud can only be displaced in k-space along the direction of the shaking. For a shaking phase of $\varphi = \pm 90$ the shaking is circular and thus the dependence is linear. The sign of φ decides on the direction of shaking which is encoded in the phase jump and the direction of the slope.

APPENDIX B

Haldane Model

B.1 | PCA

The easiest approach to analyze experimental and theoretical data is the PCA which is described in detail in chapter 3.1.1. In contrast to the clear signals achieved by feeding the algorithm with theoretical data like spin configurations [22] it is hard to link the components to the different phase transitions for the experimental data of the Haldane model. In figure

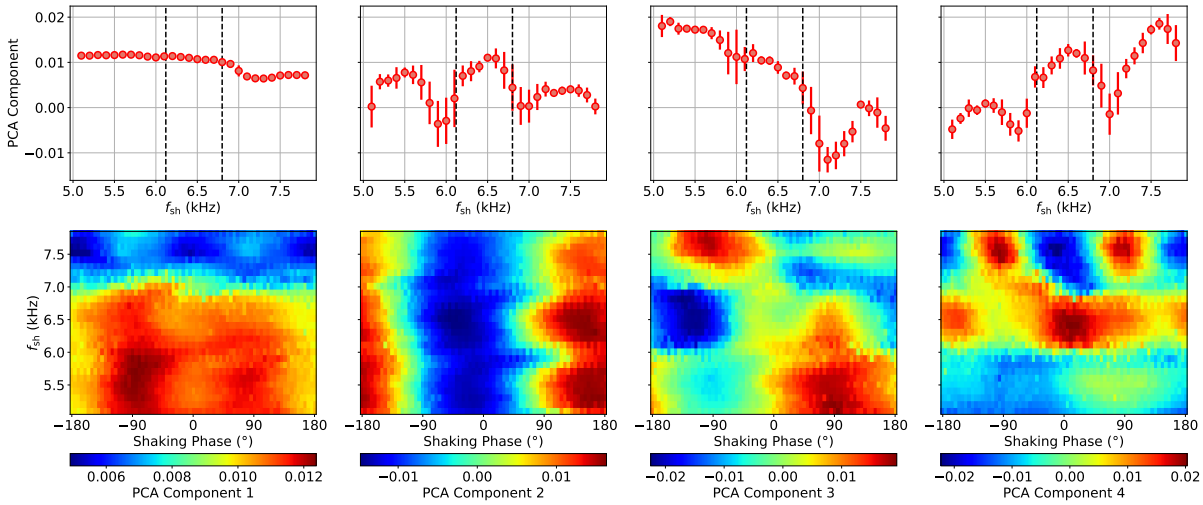

Figure B.1: PCA analysis of the time of flight images with a fixed micromotion phase. From left to right PCA components 1,2,6 and 8 which are selected arbitrarily. Upper row: The PCA components for a shaking phase of $\varphi = 90°$. The error bars are the standard deviation of the mean value. The dashed lines are the theoretically expected transitions. Lower row: Averaged PCA components plotted in the Haldane phase diagram fashion.

B.1 different PCA components of the micromotion fixed data are plotted. Looking at single shaking frequency cuts reveals some features related to the phase transitions. However, the data also provides features not related to phase transitions and do not provide a clear recipe for how to extract them.

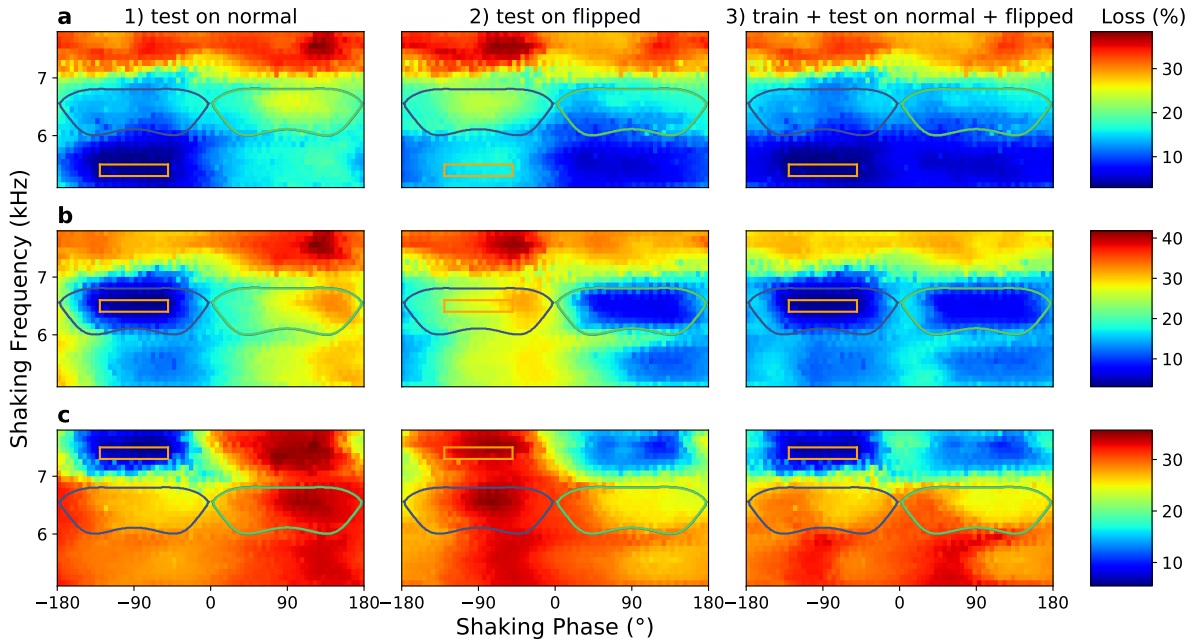

Figure B.2: (a)-(c)Training within small boxes of the phase diagram with only negative shaking phases. (1) we test on the normal images. (2) we flip the images performing a phase transformation $(k_x, k_y) \rightarrow (-k_x, k_y)$ which flips the sign of the shaking phase. (3) here we use a dataset consisting of both, flipped and not flipped ToF images.

B.2 | Anomaly Detection in Shaking Phase Direction

Figure B.2 shows the training with smaller boxes for anomaly detection to separate the two topological nontrivial phases from each other. For smaller boxes only trained on one side of the phase diagram, the anomaly detection scheme also distinguished in the topological trivial phases two different phases of matter which is not expected. The emerging patterns seem proportional to the band population and hence are sufficient to differentiate the two nontrivial phases. We also note that the box size in this case influences the found signal from the anomaly detection. we also use flipping the images to investigate this behavior further. Flipping the images corresponds to a transformation in the momentum space $(k_x, k_y) \rightarrow (-k_x, k_y)$ and flips the sign of the shaking phase, and hence the sign Chern number as well. We can see that the flipping also changes the expected reconstruction loss. In section 3 of figure B.2 we train with a mixed set of flipped and not flipped images. Notice that there is still a differentiation in the topological trivial regions.

APPENDIX C

Neural Network Architectures and Training

Table C.1: Architecture for the AE to perform the bottleneck cluster analysis of the Haldane model. Realized in Tensorflow [156]. Learning rate 0.001. Optimized with Adam for 100 epochs and a mini-batch size of 100.

Layer (type)	Output Shape	Number of Params
conv2d (Conv2D)	(None, 56, 56, 64)	320
maxpooling2d (MaxPooling2D)	(None, 28, 28, 64)	0
conv2d1 (Conv2D)	(None, 28, 28, 32)	8224
maxpooling2d1 (MaxPooling 2D)	(None, 14, 14, 32)	0
flatten (Flatten)	(None, 6272)	0
bottleneck dense (Dense)	(None, 2)	1254
dense1 (Dense)	(None, 196)	588
reshape (Reshape)	(None, 14, 14, 1)	0
conv2d2 (Conv2D)	(None, 14, 14, 32)	320
upsampling2d (UpSampling2D)	(None, 28, 28, 32)	0
conv2d3 (Conv2D)	(None, 28, 28, 32)	9248
upsampling2d1 (UpSampling2D)	(None, 56, 56, 32)	0
conv2d4 (Conv2D)	(None, 56, 56, 1)	129

Table C.2: Architecture of the VAE encoder for the micromotion removal. Realized in Tensorflow [156]. Hyperparameters obtained by [226]. Full details can be found in the notebooks [200]

Layer (type)	Output Shape	Number of Params
enconv0 (Conv2D)	(None, 56, 56, 90)	1530
batchnormalization (BatchNormalization)	(None, 56, 56, 90)	360
leakyrelu (LeakyReLU)	(None, 56, 56, 90) 0	
enpool0 (MaxPooling2D)	(None, 14, 14, 90)	0
enconv1 (Conv2D)	(None, 14, 14, 50)	162050
batchnormalization1 (BatchNormalization)	(None, 14, 14, 50)	200
leakyrelu1 (LeakyReLU)	(None, 14, 14, 50)	0
enpool1 (MaxPooling2D)	(None, 4, 4, 50)	0
enconv2 (Conv2D)	(None, 4, 4, 160)	288160
batchnormalization2 (BatchNormalization)	(None, 4, 4, 160)	640
leakyrelu2 (LeakyReLU)	(None, 4, 4, 160)	0
enpool2 (MaxPooling2D)	(None, 1, 1, 160)	0
flatten (Flatten)	(None, 160)	0
enfc0 (Dense)	(None, 260)	41860
batchnormalization3 (BatchNormalization)	(None, 260)	1040
leakyrelu3 (LeakyReLU)	(None, 260)	0
enfc1 (Dense)	(None, 180)	46980
batchnormalization4 (BatchNormalization)	(None, 180)	720
leakyrelu4 (LeakyReLU)	(None, 180)	0
enfc2 (Dense)	(None, 320)	57920
batchnormalization5 (BatchNormalization)	(None, 320)	1280
leakyrelu5 (LeakyReLU)	(None, 320)	0
enfc3 (Dense)	(None, 480)	154080
batchnormalization6 (BatchNormalization)	(None, 480)	1920
leakyrelu6 (LeakyReLU)	(None, 480)	0
fcenlatenetspace (Dense)	(None, 26)	12506

Table C.3: Architecture of the VAE decoder for the micromotion removal. Realized in Tensorflow [156]. Hyperparameters obtained by [226]. Full details can be found in the notebooks [200]

Layer (type)	Output Shape	Number of Params
defc0 (Dense)	(None, 250)	3750
batchnormalization7 (BatchNormalization)	(None, 250)	1000
dropout (Dropout)	(None, 250)	0
leakyrelu7 (LeakyReLU)	(None, 250)	0
defc1 (Dense)	(None, 250)	62750
batchnormalization8 (BatchNormalization)	(None, 250)	1000
dropout1 (Dropout)	(None, 250)	0
leakyrelu8 (LeakyReLU)	(None, 250)	0
defc2 (Dense)	(None, 230)	57730
batchnormalization9 (BatchNormalization)	(None, 230)	920
dropout2 (Dropout)	(None, 230)	0
leakyrelu9 (LeakyReLU)	(None, 230)	0
defcshaping (Dense)	(None, 40)	9240
batchnormalization10 (BatchNormalization)	(None, 40)	160
leakyrelu10 (LeakyReLU)	(None, 40)	0
reshape (Reshape)	(None, 2, 2, 10)	0
detconv0 (Conv2DTranspose)	(None, 4, 4, 140)	22540
batchnormalization11 (BatchNormalization)	(None, 4, 4, 140)	560
dropout3 (Dropout)	(None, 4, 4, 140)	0
leakyrelu11 (LeakyReLU)	(None, 4, 4, 140)	0
detconv1 (Conv2DTranspose)	(None, 8, 8, 180)	907380
batchnormalization12 (BatchNormalization)	(None, 8, 8, 180)	720
dropout4 (Dropout)	(None, 8, 8, 180)	0
leakyrelu12 (LeakyReLU)	(None, 8, 8, 180)	0
detconv2 (Conv2DTranspose)	(None, 56, 56, 200)	1296200
batchnormalization13 (Bat	(None, 56, 56, 200)	800
dropout5 (Dropout)	(None, 56, 56, 200)	0
leakyrelu13 (LeakyReLU)	(None, 56, 56, 200)	0
dropout (Conv2DTranspose)	(None, 56, 56, 1)	7201

Table C.4: Architecture of the AE for the anomaly detection scheme of the Haldane model. The horizontal line is the bottleneck. The model is implemented in Tensorflow [156]. The Adam optimizer trains the model in 10 epochs and a batch size of 128.

Layer (type)	Output Shape	Number of Params
input1 (InputLayer)	(None, 56, 56, 1)	0
conv2d (Conv2D)	(None, 56, 56, 16)	160
maxpooling2d (MaxPooling2D)	(None, 28, 28, 16)	0
conv2d1 (Conv2D)	(None, 28, 28, 16)	2320
maxpooling2d1 (MaxPooling2D)	(None, 14, 14, 16)	0
conv2d2 (Conv2D)	(None, 14, 14, 16)	2320
flatten (Flatten)	(None, 3136)	0
dense (Dense)	(None, 50)	156850
dropout (Dropout)	(None, 50)	0
dense1 (Dense)	(None, 3136)	159936
dropout1 (Dropout)	(None, 3136)	0
reshape (Reshape)	(None, 14, 14, 16)	0
upsampling2d (UpSampling2D)	(None, 28, 28, 16)	0
conv2d3 (Conv2D)	(None, 28, 28, 16)	2320
upsampling2d1 (UpSampling2D)	(None, 56, 56, 16)	0
conv2d4 (Conv2D)	(None, 56, 56, 1)	145

Table C.5: Architecture of the CNN to classify the different snapshots for the influence function approach. The network is implemented in PyTorch [157].

Name	Output Shape
input1 (InputLayer)	(None, 56, 56, 1)
conv1 (Conv2D)	(None, 52, 52, 1)
maxpool1 (MaxPooling2D)	(None, 25, 25, 1)
conv2 (Conv2D)	(None, 21, 21, 1)
avgpool2 (AveragePooling2D)	(None, 5, 5, 1)
dense (Dense)	(None, 200)
dense (Dense)	(None, 2)
output (Softmax)	(None, 2)

Bibliography

[1] T. Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901 (cit. on pp. 1, 5, 72).

[2] D. Silver et al. "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529.7587 (2016), pp. 484–489. ISSN: 0028-0836. DOI: 10.1038/nature16961 (cit. on p. 1).

[3] D. Silver et al. "Mastering the game of Go without human knowledge". In: *Nature* 550.7676 (2017), pp. 354–359. ISSN: 0028-0836. DOI: 10.1038/nature24270 (cit. on p. 1).

[4] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis. "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play". In: *Science* 362.6419 (2018), pp. 1140–1144. ISSN: 0036-8075. DOI: 10.1126/science.aar6404 (cit. on pp. 1, 6).

[5] A. Fawzi, M. Balog, A. Huang, T. Hubert, B. Romera-Paredes, M. Barekatain, A. Novikov, F. J. R. Ruiz, J. Schrittwieser, G. Swirszcz, D. Silver, D. Hassabis, and P. Kohli. "Discovering faster matrix multiplication algorithms with reinforcement learning". In: *Nature* 610.7930 (2022), pp. 47–53. ISSN: 0028-0836. DOI: 10.1038/s41586-022-05172-4 (cit. on p. 1).

[6] Y. Li et al. "Competition-level code generation with AlphaCode". In: *Science* 378.6624 (2022), pp. 1092–1097. ISSN: 0036-8075. DOI: 10.1126/science.abq1158 (cit. on p. 1).

[7] J. Jumper et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589. ISSN: 0028-0836. DOI: 10.1038/s41586-021-03819-2 (cit. on p. 1).

[8] G. Ahdritz et al. "OpenFold: Retraining AlphaFold2 yields new insights into its learning mechanisms and capacity for generalization". In: *bioRxiv* (2023), p. 2022.11.20.517210. DOI: 10.1101/2022.11.20.517210 (cit. on p. 1).

[9] M. Varadi and S. Velankar. "The impact of AlphaFold Protein Structure Database on the fields of life sciences". In: *PROTEOMICS* 23.17 (2023), e2200128. ISSN: 1615-9853. DOI: 10.1002/pmic.202200128 (cit. on p. 1).

[10] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová. "Machine learning and the physical sciences*". In: *Reviews of Modern Physics* 91.4 (2019), p. 045002. ISSN: 0034-6861. DOI: 10.1103/revmodphys.91.045002 (cit. on pp. 1, 5).

[11] J. Carrasquilla. "Machine learning for quantum matter". In: *Advances in Physics: X* 5.1 (2020), p. 1797528. DOI: 10.1080/23746149.2020.1797528 (cit. on p. 1).

[12] A. Dawid et al. "Modern applications of machine learning in quantum sciences". In: *arXiv* (2022). DOI: 10.48550/arxiv.2204.04198 (cit. on pp. 1, 5, 6, 61, 62).

[13] M. Krenn, J. Landgraf, T. Foesel, and F. Marquardt. "Artificial intelligence and machine learning for quantum technologies". In: *Physical Review A* 107.1 (2023), p. 010101. ISSN: 2469-9926. DOI: 10.1103/physreva.107.010101 (cit. on p. 1).

[14] G. Carleo and M. Troyer. "Solving the quantum many-body problem with artificial neural networks". In: *Science* 355.6325 (2017), pp. 602–606. ISSN: 0036-8075. DOI: 10.1126/science.aag2302 (cit. on pp. 1, 59–61, 63).

[15] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo. "Neural-network quantum state tomography". In: *Nature Physics* 14.5 (2018), pp. 447–450. ISSN: 1745-2473. DOI: 10.1038/s41567-018-0048-5 (cit. on pp. 1, 59, 65, 67, 68).

[16] G. Torlai, B. Timar, E. P. L. v. Nieuwenburg, H. Levine, A. Omran, A. Keesling, H. Bernien, M. Greiner, V. Vuletić, M. D. Lukin, R. G. Melko, and M. Endres. "Integrating Neural Networks with a Quantum Simulator for State Reconstruction". In: *Physical Review Letters* 123.23 (2019), p. 230504. ISSN: 0031-9007. DOI: 10.1103/physrevlett.123.230504 (cit. on pp. 1, 66, 67, 72).

[17] M. Neugebauer, L. Fischer, A. Jäger, S. Czischek, S. Jochim, M. Weidemüller, and M. Gärttner. "Neural-network quantum state tomography in a two-qubit experiment". In: *Physical Review A* 102.4 (2020), p. 042604. ISSN: 2469-9926. DOI: 10.1103/physreva.102.042604 (cit. on p. 1).

[18] P. B. Wigley, P. J. Everitt, A. v. d. Hengel, J. W. Bastian, M. A. Sooriyabandara, G. D. McDonald, K. S. Hardman, C. D. Quinlivan, P. Manju, C. C. N. Kuhn, I. R. Petersen, A. N. Luiten, J. J. Hope, N. P. Robins, and M. R. Hush. "Fast machine-learning online optimization of ultra-cold-atom experiments". In: *Scientific Reports* 6.1 (2016), p. 25890. DOI: 10.1038/srep25890 (cit. on p. 1).

[19] A. D. Tranter, H. J. Slatyer, M. R. Hush, A. C. Leung, J. L. Everett, K. V. Paul, P. Vernaz-Gris, P. K. Lam, B. C. Buchler, and G. T. Campbell. "Multiparameter optimisation of a magneto-optical trap using deep learning". In: *Nature Communications* 9.1 (2018), p. 4360. DOI: 10.1038/s41467-018-06847-1 (cit. on p. 1).

[20] E. T. Davletov, V. V. Tsyganok, V. A. Khlebnikov, D. A. Pershin, D. V. Shaykin, and A. V. Akimov. "Machine learning for achieving Bose-Einstein condensation of thulium atoms". In: *Physical Review A* 102.1 (2020), p. 011302. ISSN: 2469-9926. DOI: 10.1103/physreva.102.011302 (cit. on p. 1).

[21] N. Milson, A. Tashchilina, T. Ooi, A. Czarnecka, Z. F. Ahmad, and L. J. LeBlanc. "High-dimensional reinforcement learning for optimization and control of ultracold quantum gases". In: *Machine Learning: Science and Technology* 4.4 (2023), p. 045057. DOI: 10.1088/2632-2153/ad1437 (cit. on p. 1).

[22] L. Wang. "Discovering phase transitions with unsupervised learning". In: *Physical Review B* 94.19 (2016), p. 195105. ISSN: 2469-9950. DOI: 10.1103/physrevb.94.195105 (cit. on pp. 1, 24, 27, 74).

[23] K. Kottmann, P. Huembeli, M. Lewenstein, and A. Acín. "Unsupervised Phase Discovery with Deep Anomaly Detection". In: *Physical Review Letters* 125.17 (2020), p. 170603. ISSN: 0031-9007. DOI: 10.1103/physrevlett.125.170603 (cit. on pp. 1, 16, 27, 29).

[24] K. Kottmann, P. Corboz, M. Lewenstein, and A. Acín. "Unsupervised mapping of phase diagrams of 2D systems from infinite projected entangled-pair states via deep anomaly detection". In: *SciPost Physics* 11.2 (2021), p. 025. ISSN: 2542-4653. DOI: 10.21468/scipostphys.11.2.025 (cit. on pp. 1, 27).

[25] Z. Patel, E. Merali, and S. J. Wetzel. "Unsupervised learning of Rydberg atom array phase diagram with Siamese neural networks". In: *New Journal of Physics* 24.11 (2022), p. 113021. DOI: 10.1088/1367-2630/ac9c7a (cit. on p. 1).

[26] D. Contessi, E. Ricci, A. Recati, and M. Rizzi. "Detection of Berezinskii-Kosterlitz-Thouless transition via Generative Adversarial Networks". In: *SciPost Physics* 12.3 (2022), p. 107. ISSN: 2542-4653. DOI: 10.21468/scipostphys.12.3.107 (cit. on p. 1).

[27] M.-C. Chung, G.-Y. Huang, I. McCulloch, and Y.-H. Tsai. "Deep Learning of Phase Transitions for Quantum Spin Chains from Correlation Aspects". In: *arXiv* (2023). DOI: 10.48550/arxiv.2301.06669 (cit. on p. 1).

[28] T. Ohtsuki and T. Ohtsuki. "Deep Learning the Quantum Phase Transitions in Random Two-Dimensional Electron Systems". In: *Journal of the Physical Society of Japan* 85.12 (2016), p. 123706. ISSN: 0031-9015. DOI: 10.7566/jpsj.85.123706 (cit. on pp. 1, 26).

[29] J. Carrasquilla and R. G. Melko. "Machine learning phases of matter". In: *Nature Physics* 13.5 (2017), pp. 431–434. ISSN: 1745-2473. DOI: 10.1038/nphys4035 (cit. on pp. 1, 23, 26).

[30] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami. "Machine Learning Phases of Strongly Correlated Fermions". In: *Physical Review X* 7.3 (2017), p. 031038. ISSN: 1068-9265. DOI: 10.1103/physrevx.7.031038 (cit. on pp. 1, 26).

[31] P. Broecker, J. Carrasquilla, R. G. Melko, and S. Trebst. "Machine learning quantum phases of matter beyond the fermion sign problem". In: *Scientific Reports* 7.1 (2017), p. 8823. DOI: 10.1038/s41598-017-09098-0 (cit. on pp. 1, 23).

[32] E. P. L. v. Nieuwenburg, Y.-H. Liu, and S. D. Huber. "Learning phase transitions by confusion". In: *Nature Physics* 13.5 (2017), pp. 435–439. ISSN: 1745-2473. DOI: 10.1038/nphys4037 (cit. on pp. 1, 23, 27, 29).

[33] C. Wang and H. Zhai. "Machine learning of frustrated classical spin models. I. Principal component analysis". In: *Physical Review B* 96.14 (2017), p. 144432. ISSN: 2469-9950. DOI: 10.1103/physrevb.96.144432 (cit. on pp. 1, 27).

[34] P. Huembeli, A. Dauphin, and P. Wittek. "Identifying quantum phase transitions with adversarial neural networks". In: *Physical Review B* 97.13 (2018), p. 134109. ISSN: 2469-9950. DOI: 10.1103/physrevb.97.134109 (cit. on pp. 1, 23, 26, 27).

[35] P. Huembeli, A. Dauphin, P. Wittek, and C. Gogolin. "Automated discovery of characteristic features of phase transitions in many-body localization". In: *Physical Review B* 99.10 (2019), p. 104106. ISSN: 2469-9950. DOI: 10.1103/physrevb.99.104106 (cit. on pp. 1, 23).

[36] M. Ziatdinov, A. Maksov, L. Li, A. S. Sefat, P. Maksymovych, and S. V. Kalinin. "Deep data mining in a real space: separation of intertwined electronic responses in a lightly doped BaFe2As2". In: *Nanotechnology* 27.47 (2016), p. 475706. ISSN: 0957-4484. DOI: 10.1088/0957-4484/27/47/475706 (cit. on p. 1).

[37] Y. Zhang, A. Mesaros, K. Fujita, S. D. Edkins, M. H. Hamidian, K. Ch'ng, H. Eisaki, S. Uchida, J. C. S. Davis, E. Khatami, and E.-A. Kim. "Machine learning in electronic-quantum-matter imaging experiments". In: *Nature* 570.7762 (2019), pp. 484–490. ISSN: 0028-0836. DOI: 10.1038/s41586-019-1319-8 (cit. on p. 1).

[38] A. M. Samarakoon, K. Barros, Y. W. Li, M. Eisenbach, Q. Zhang, F. Ye, V. Sharma, Z. L. Dun, H. Zhou, S. A. Grigera, C. D. Batista, and D. A. Tennant. "Machine-learning-assisted insight into spin ice Dy2Ti2O7". In: *Nature Communications* 11.1 (2020), p. 892. ISSN: 20411723. DOI: 10.1038/s41467-020-14660-y (cit. on p. 1).

[39] M. Kim, J. Kwon, T. Rabga, and Y. Shin. "Vortex detection in atomic Bose–Einstein condensates using neural networks trained on synthetic images". In: *Machine Learning: Science and Technology* 4.4 (2023), p. 045017. DOI: 10.1088/2632-2153/ad03ad (cit. on p. 1).

[40] 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". In: *Nature Physics* 15.9 (2019), pp. 917–920. ISSN: 1745-2473. DOI: 10.1038/s41567-019-0554-0 (cit. on pp. 1, 3, 4, 23, 41, 42, 44, 45, 57).

[41] 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". In: *Nature Physics* 15.9 (2019), pp. 921–924. ISSN: 1745-2473. DOI: 10.1038/s41567-019-0565-x (cit. on p. 1).

[42] E. Khatami, E. Guardado-Sanchez, B. M. Spar, J. F. Carrasquilla, W. S. Bakr, and R. T. Scalettar. "Visualizing strange metallic correlations in the two-dimensional Fermi-Hubbard model with artificial intelligence". In: *Physical Review A* 102.3 (2020), p. 033326. ISSN: 2469-9926. DOI: 10.1103/physreva.102.033326 (cit. on pp. 1, 58, 71).

[43] N. Käming, A. Dawid, K. Kottmann, M. Lewenstein, K. Sengstock, A. Dauphin, and C. Weitenberg. "Unsupervised machine learning of topological phase transitions from experimental data". In: *Machine Learning: Science and Technology* 2.3 (2021), p. 035037. DOI: 10.1088/2632-2153/abffe7 (cit. on pp. 1, 3, 4, 23, 33, 57).

[44] E. Zhao, T. H. Mak, C. He, Z. Ren, K. K. Pak, Y.-J. Liu, and G.-B. Jo. "Observing a topological phase transition with deep neural networks from experimental images of ultracold atoms". In: *Optics Express* 30.21 (2022), p. 37786. DOI: 10.1364/oe.473770 (cit. on p. 1).

[45] D. Eberz, M. Link, A. Kell, M. Breyer, K. Gao, and M. Köhl. "Detecting the phase transition in a strongly interacting Fermi gas by unsupervised machine learning". In: *Physical Review A* 108.6 (2023), p. 063303. ISSN: 2469-9926. DOI: 10.1103/physreva.108.063303 (cit. on p. 1).

[46] C. Miles, A. Bohrdt, R. Wu, C. Chiu, M. Xu, G. Ji, M. Greiner, K. Q. Weinberger, E. Demler, and E.-A. Kim. "Correlator convolutional neural networks as an interpretable architecture for image-like quantum matter data". In: *Nature Communications* 12.1 (2021), p. 3905. DOI: 10.1038/s41467-021-23952-w (cit. on pp. 1, 71).

[47] A. Impertro, J. F. Wienand, S. Häfele, H. v. Raven, S. Hubele, T. Klostermann, C. R. Cabrera, I. Bloch, and M. Aidelsburger. "An unsupervised deep learning algorithm for single-site reconstruction in quantum gas microscopes". In: *Communications Physics* 6.1 (2023), p. 166. DOI: 10.1038/s42005-023-01287-w (cit. on pp. 2, 71).

[48] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik. "The theory of variational hybrid quantum-classical algorithms". In: *New Journal of Physics* 18.2 (2016), p. 023023. DOI: 10.1088/1367-2630/18/2/023023 (cit. on p. 2).

[49] J. Romero, J. P. Olson, and A. Aspuru-Guzik. "Quantum autoencoders for efficient compression of quantum data". In: *Quantum Science and Technology* 2.4 (2017), p. 045001. DOI: 10.1088/2058-9565/aa8072 (cit. on p. 2).

[50] J. Stokes, J. Izaac, N. Killoran, and G. Carleo. "Quantum Natural Gradient". In: *Quantum* 4 (2020), p. 269. DOI: 10.22331/q-2020-05-25-269 (cit. on p. 2).

[51] C. Bravo-Prieto. "Quantum autoencoders with enhanced data encoding". In: *Machine Learning: Science and Technology* 2.3 (2021), p. 035028. DOI: 10.1088/2632-2153/ac0616 (cit. on p. 2).

[52] K. Kottmann, F. Metz, J. Fraxanet, and N. Baldelli. "Variational quantum anomaly detection: Unsupervised mapping of phase diagrams on a physical quantum computer". In: *Physical Review Research* 3.4 (2021), p. 043184. DOI: 10.1103/physrevresearch.3.043184 (cit. on p. 2).

[53] R. P. Feynman. "Simulating physics with computers". In: *International Journal of Theoretical Physics* 21.6–7 (1982), pp. 467–488. ISSN: 0020-7748. DOI: 10.1007/bf02650179 (cit. on p. 2).

[54] J. I. Cirac and P. Zoller. "Goals and opportunities in quantum simulation". In: *Nature Physics* 8.4 (2012), pp. 264–266. ISSN: 1745-2473. DOI: 10.1038/nphys2275 (cit. on p. 2).

[55] M. Lewenstein, A. Sanpera, and V. Ahufinger. *Ultracold Atoms in Optical Lattices: Simulating quantum many-body systems*. Oxford University Press, 2012 (cit. on pp. 2, 40).

[56] I. Bloch, J. Dalibard, and S. Nascimbène. "Quantum simulations with ultracold quantum gases". In: *Nature Physics* 8.4 (2012), pp. 267–276. ISSN: 1745-2473. DOI: 10.1038/nphys2259 (cit. on p. 2).

[57] R. Blatt and C. F. Roos. "Quantum simulations with trapped ions". In: *Nature Physics* 8.4 (2012), pp. 277–284. ISSN: 1745-2473. DOI: 10.1038/nphys2252 (cit. on p. 2).

[58] A. Aspuru-Guzik and P. Walther. "Photonic quantum simulators". In: *Nature Physics* 8.4 (2012), pp. 285–291. ISSN: 1745-2473. DOI: 10.1038/nphys2253 (cit. on p. 2).

[59] Davis, Mewes, Andrews, v. D. NJ, Durfee, Kurn, and Ketterle. "Bose-Einstein condensation in a gas of sodium atoms." In: *Physical review letters* 75.22 (1995), pp. 3969–3973. ISSN: 0031-9007. DOI: 10.1103/physrevlett.75.3969 (cit. on p. 2).

[60] M. H. Anderson, J. R. Ensher, M. R. Matthews, C. E. Wieman, and E. A. Cornell. "Observation of bose-einstein condensation in a dilute atomic vapor." In: *Science (New York, N.Y.)* 269.5221 (1995), pp. 198–201. ISSN: 0036-8075. DOI: 10.1126/science.269.5221.198 (cit. on p. 2).

[61] DeMarco and Jin. "Onset of fermi degeneracy in a trapped atomic Gas". In: *Science (New York, N.Y.)* 285.5434 (1999), pp. 1703–6. ISSN: 0036-8075. DOI: 10.1126/science.285.5434.1703 (cit. on p. 2).

[62] I. Bloch, J. Dalibard, and W. Zwerger. "Many-body physics with ultracold gases". In: *Reviews of Modern Physics* 80.3 (2008), pp. 885–964. ISSN: 0034-6861. DOI: 10.1103/revmodphys.80.885 (cit. on pp. 2, 40).

[63] C. Chin, R. Grimm, P. Julienne, and E. Tiesinga. "Feshbach resonances in ultracold gases". In: *Reviews of Modern Physics* 82.2 (2010), pp. 1225–1286. ISSN: 0034-6861. DOI: 10.1103/revmodphys.82.1225 (cit. on p. 2).

[64] M. Greiner, O. Mandel, T. Esslinger, T. W. Hänsch, and I. Bloch. "Quantum phase transition from a superfluid to a Mott insulator in a gas of ultracold atoms". In: *Nature* 415.6867 (2002), pp. 39–44. ISSN: 0028-0836. DOI: 10.1038/415039a (cit. on p. 2).

[65] R. Jördens, N. Strohmaier, K. Günter, H. Moritz, and T. Esslinger. "A Mott insulator of fermionic atoms in an optical lattice". In: *Nature* 455.7210 (2008), pp. 204–207. ISSN: 0028-0836. DOI: 10.1038/nature07244 (cit. on p. 2).

[66] U. Schneider, L. Hackermüller, S. Will, T. Best, I. Bloch, T. A. Costi, R. W. Helmes, D. Rasch, and A. Rosch. "Metallic and insulating phases of repulsively interacting fermions in a 3D optical lattice." In: *Science (New York, N.Y.)* 322.5907 (2008), pp. 1520–5. ISSN: 0036-8075. DOI: 10.1126/science.1165449 (cit. on p. 2).

[67] J. Struck, C. Ölschläger, R. L. Targat, P. Soltan-Panahi, A. Eckardt, M. Lewenstein, P. Windpassinger, and K. Sengstock. "Quantum simulation of frustrated classical magnetism in triangular optical lattices." In: *Science (New York, N.Y.)* 333.6045 (2011), pp. 996–9. ISSN: 0036-8075. DOI: 10.1126/science.1207239 (cit. on pp. 2, 41).

[68] J. Struck, M. Weinberg, C. Ölschläger, P. Windpassinger, J. Simonet, K. Sengstock, R. Höppner, P. Hauke, A. Eckardt, M. Lewenstein, and L. Mathey. "Engineering Ising-XY spin-models in a triangular lattice using tunable artificial gauge fields". In: *Nature Physics* 9.11 (2013), pp. 738–743. ISSN: 1745-2473. DOI: 10.1038/nphys2750 (cit. on p. 2).

[69] L. Tarruell, D. Greif, T. Uehlinger, G. Jotzu, and T. Esslinger. "Creating, moving and merging Dirac points with a Fermi gas in a tunable honeycomb lattice". In: *Nature* 483.7389 (2012), pp. 302–305. ISSN: 0028-0836. DOI: 10.1038/nature10871 (cit. on p. 2).

[70] T. Uehlinger, G. Jotzu, M. Messer, D. Greif, W. Hofstetter, U. Bissbort, and T. Esslinger. "Artificial graphene with tunable interactions." In: *Physical review letters* 111.18 (2013), p. 185307. ISSN: 0031-9007. DOI: 10.1103/physrevlett.111.185307 (cit. on p. 2).

[71] C. Becker, P. Soltan-Panahi, J. Kronjäger, S. Dörscher, K. Bongs, and K. Sengstock. "Ultracold quantum gases in triangular optical lattices". In: *New Journal of Physics* 12.6 (2010), p. 065025. DOI: 10.1088/1367-2630/12/6/065025 (cit. on pp. 2, 41).

[72] N. Fläschner, B. S. Rem, M. Tarnowski, D. Vogel, D.-S. Lühmann, K. Sengstock, and C. Weitenberg. "Experimental reconstruction of the Berry curvature in a Floquet Bloch band". In: *Science* 352.6289 (2016), pp. 1091–1094. ISSN: 0036-8075. DOI: 10.1126/science.aad4568 (cit. on pp. 2, 3, 23, 40–42, 44).

[73] L. Duca, T. Li, M. Reitter, I. Bloch, M. Schleier-Smith, and U. Schneider. "An Aharonov-Bohm interferometer for determining Bloch band topology". In: *Science* 347.6219 (2015), pp. 288–292. ISSN: 0036-8075. DOI: 10.1126/science.1259052 (cit. on p. 2).

[74] G. Wirth, M. Ölschläger, and A. Hemmerich. "Evidence for orbital superfluidity in the P-band of a bipartite optical square lattice". In: *Nature Physics* 7.2 (2011), pp. 147–153. ISSN: 1745-2473. DOI: 10.1038/nphys1857 (cit. on p. 2).

[75] S. Taie, H. Ozawa, T. Ichinose, T. Nishio, S. Nakajima, and Y. Takahashi. "Coherent driving and freezing of bosonic matter wave in an optical Lieb lattice". In: *Science Advances* 1.10 (2015), e1500854. ISSN: 2375-2548. DOI: 10.1126/sciadv.1500854 (cit. on p. 2).

[76] G.-B. Jo, J. Guzman, C. K. Thomas, P. Hosur, A. Vishwanath, and D. M. Stamper-Kurn. "Ultracold Atoms in a Tunable Optical Kagome Lattice". In: *Physical Review Letters* 108.4 (2012), p. 045305. ISSN: 0031-9007. DOI: 10.1103/physrevlett.108.045305 (cit. on p. 2).

[77] K. Viebahn, M. Sbroscia, E. Carter, J.-C. Yu, and U. Schneider. "Matter-Wave Diffraction from a Quasicrystalline Optical Lattice". In: *Physical Review Letters* 122.11 (2019), p. 110404. ISSN: 0031-9007. DOI: 10.1103/physrevlett.122.110404 (cit. on p. 2).

[78] W. S. Bakr, A. Peng, M. E. Tai, R. Ma, J. Simon, J. I. Gillen, S. Fölling, L. Pollet, and M. Greiner. "Probing the Superfluid–to–Mott Insulator Transition at the Single-Atom Level". In: *Science* 329.5991 (2010), pp. 547–550. ISSN: 0036-8075. DOI: 10.1126/science.1192368 (cit. on p. 2).

[79] J. F. Sherson, C. Weitenberg, M. Endres, M. Cheneau, I. Bloch, and S. Kuhr. "Single-Atom Resolved Fluorescence Imaging of an Atomic Mott Insulator". In: *arXiv* 467.7311 (2010), pp. 68–72. ISSN: 0028-0836. DOI: 10.1038/nature09378 (cit. on p. 2).

[80] S. Kuhr. "Quantum-gas microscopes: a new tool for cold-atom quantum simulators". In: *National Science Review* 3.2 (2016), pp. 170–172. ISSN: 2095-5138. DOI: 10.1093/nsr/nww023 (cit. on p. 2).

[81] L. Asteria, H. P. Zahn, M. N. Kosch, K. Sengstock, and C. Weitenberg. "Quantum gas magnifier for sub-lattice-resolved imaging of 3D quantum systems". In: *Nature* 599.7886 (2021), pp. 571–575. ISSN: 0028-0836. DOI: 10.1038/s41586-021-04011-2 (cit. on pp. 3, 72).

[82] Q. L. He, T. L. Hughes, N. P. Armitage, Y. Tokura, and K. L. Wang. "Topological spintronics and magnetoelectronics". In: *Nature Materials* 21.1 (2022), pp. 15–23. ISSN: 1476-1122. DOI: 10.1038/s41563-021-01138-5 (cit. on p. 3).

[83] K. v. Klitzing, G. Dorda, and M. Pepper. "New Method for High-Accuracy Determination of the Fine-Structure Constant Based on Quantized Hall Resistance". In: *Physical Review Letters* 45.6 (1980), pp. 494–497. ISSN: 0031-9007. DOI: 10.1103/physrevlett.45.494 (cit. on pp. 3, 33).

[84] D. J. Thouless, M. Kohmoto, M. P. Nightingale, and M. d. Nijs. "Quantized Hall Conductance in a Two-Dimensional Periodic Potential". In: *Physical Review Letters* 49.6 (1982), pp. 405–408. ISSN: 0031-9007. DOI: 10.1103/physrevlett.49.405 (cit. on pp. 3, 33).

[85] M. Z. Hasan and C. L. Kane. "Colloquium: Topological insulators". In: *Reviews of Modern Physics* 82.4 (2010), pp. 3045–3067. ISSN: 0034-6861. DOI: 10.1103/revmodphys.82.3045 (cit. on pp. 3, 37).

[86] X.-L. Qi and S.-C. Zhang. "Topological insulators and superconductors". In: *Reviews of Modern Physics* 83.4 (2011), pp. 1057–1110. ISSN: 0034-6861. DOI: 10.1103/revmodphys.83.1057 (cit. on p. 3).

[87] G. Jotzu, M. Messer, R. Desbuquois, M. Lebrat, T. Uehlinger, D. Greif, and T. Esslinger. "Experimental realization of the topological Haldane model with ultracold fermions". In: *Nature* 515.7526 (2014), pp. 237–240. ISSN: 0028-0836. DOI: 10.1038/nature13915 (cit. on pp. 3, 23, 37, 41, 44).

[88] M. Aidelsburger, M. Lohse, C. Schweizer, M. Atala, J. T. Barreiro, S. Nascimbène, N. R. Cooper, I. Bloch, and N. Goldman. "Measuring the Chern number of Hofstadter bands with ultracold bosonic atoms". In: *Nature Physics* 11.2 (2014), pp. 162–166. ISSN: 1745-2473. DOI: 10.1038/nphys3171 (cit. on pp. 3, 37, 44).

[89] L. Asteria, D. T. Tran, T. Ozawa, M. Tarnowski, B. S. Rem, N. Fläschner, K. Sengstock, N. Goldman, and C. Weitenberg. "Measuring quantized circular dichroism in ultracold topological matter". In: *Nature Physics* 15.5 (2019), pp. 449–454. ISSN: 1745-2473. DOI: 10.1038/s41567-019-0417-8 (cit. on pp. 3, 23, 41, 44, 57).

[90] M. Tarnowski, F. N. Ünal, N. Fläschner, B. S. Rem, A. Eckardt, K. Sengstock, and C. Weitenberg. "Measuring topology from dynamics by obtaining the Chern number from a linking number". In: *Nature Communications* 10.1 (2019), p. 1728. DOI: 10.1038/s41467-019-09668-y (cit. on pp. 3, 23, 41, 42, 44).

[91] F. D. M. Haldane. "Model for a Quantum Hall Effect without Landau Levels: Condensed-Matter Realization of the "Parity Anomaly"". In: *Physical Review Letters* 61.18 (1988), pp. 2015–2018. ISSN: 0031-9007. DOI: 10.1103/physrevlett.61.2015 (cit. on pp. 3, 41).

[92] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016 (cit. on pp. 5, 6, 10, 11).

[93] P. Mehta, M. Bukov, C.-H. Wang, A. G. Day, C. Richardson, C. K. Fisher, and D. J. Schwab. "A high-bias, low-variance introduction to Machine Learning for physicists". In: *Physics Reports* 810 (2019), pp. 1–124. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2019.03.001 (cit. on p. 5).

[94] V. Dunjko and H. J. Briegel. "Machine learning and artificial intelligence in the quantum domain: a review of recent progress". In: *Reports on Progress in Physics* 81.7 (2018), p. 074001. ISSN: 0034-4885. DOI: 10.1088/1361-6633/aab406 (cit. on p. 5).

[95] F. Truck and H. Moravec. "Mind Children: The Future of Robot and Human Intelligence". In: *Leonardo* 24.2 (1991), p. 242. ISSN: 0024-094X. DOI: 10.2307/1575314 (cit. on p. 6).

[96] K. P. F.R.S. "LIII. On lines and planes of closest fit to systems of points in space". In: *Philosophical Magazine Series 6* 2.11 (1901), pp. 559–572. ISSN: 1941-5982. DOI: 10.1080/14786440109462720 (cit. on pp. 7, 24).

[97] I. T. Jolliffe and J. Cadima. "Principal component analysis: a review and recent developments". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374.2065 (2016), p. 20150202. ISSN: 1364-503X. DOI: 10.1098/rsta.2015.0202 (cit. on pp. 7, 24).

[98] D. Haussler, B. E. Boser, I. M. Guyon, and V. N. Vapnik. "A training algorithm for optimal margin classifiers". In: *Proceedings of the fifth annual workshop on Computational learning theory* (1992), pp. 144–152. DOI: 10.1145/130385.130401 (cit. on p. 7).

[99] C. Cortes and V. Vapnik. "Support-vector networks". In: *Machine Learning* 20.3 (1995), pp. 273–297. ISSN: 0885-6125. DOI: 10.1007/bf00994018 (cit. on p. 7).

[100] R. Lam et al. "Learning skillful medium-range global weather forecasting". In: *Science* 382.6677 (2023), pp. 1416–1421. ISSN: 0036-8075. DOI: 10.1126/science.adi2336 (cit. on p. 8).

[101] R. A. FISHER. "THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS". In: *Annals of Eugenics* 7.2 (1936), pp. 179–188. ISSN: 2050-1420. DOI: 10.1111/j.1469-1809.1936.tb02137.x (cit. on p. 8).

[102] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. ISSN: 0018-9219. DOI: 10.1109/5.726791 (cit. on p. 9).

[103] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt. "Reinforcement Learning with Neural Networks for Quantum Feedback". In: *Physical Review X* 8.3 (2018), p. 031084. DOI: 10.1103/physrevx.8.031084 (cit. on p. 10).

[104] S. Borah, B. Sarma, M. Kewming, G. J. Milburn, and J. Twamley. "Measurement-Based Feedback Quantum Control with Deep Reinforcement Learning for a Double-Well Nonlinear Potential". In: *Physical Review Letters* 127.19 (2021), p. 190403. ISSN: 0031-9007. DOI: 10.1103/physrevlett.127.190403 (cit. on p. 10).

[105] V. Nguyen, S. B. Orbell, D. T. Lennon, H. Moon, F. Vigneau, L. C. Camenzind, L. Yu, D. M. Zumbühl, G. A. D. Briggs, M. A. Osborne, D. Sejdinovic, and N. Ares. "Deep reinforcement learning for efficient measurement of quantum devices". In: *npj Quantum Information* 7.1 (2021), p. 100. DOI: 10.1038/s41534-021-00434-x (cit. on p. 10).

[106] T. Fösel, M. Y. Niu, F. Marquardt, and L. Li. "Quantum circuit optimization with deep reinforcement learning". In: *arXiv* (2021). DOI: 10.48550/arxiv.2103.07585 (cit. on p. 10).

[107] H. Robbins and S. Monro. "A Stochastic Approximation Method". In: *The Annals of Mathematical Statistics* 22.3 (1951), pp. 400–407. ISSN: 0003-4851. DOI: 10.1214/aoms/1177729586 (cit. on p. 11).

[108] L. Bottou and Y. Cun. "Large Scale Online Learning". In: *Advances in Neural Information Processing Systems*. Ed. by S. Thrun, L. Saul, and B. Schölkopf. Vol. 16. MIT Press, 2003 (cit. on p. 11).

[109] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning representations by back-propagating errors". In: *Nature* 323.6088 (1986), pp. 533–536. ISSN: 0028-0836. DOI: 10.1038/323533a0 (cit. on pp. 11, 15).

[110] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. "On the importance of initialization and momentum in deep learning". In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by S. Dasgupta and D. McAllester. Vol. 28. Proceedings of Machine Learning Research 3. Atlanta, Georgia, USA: PMLR, June 2013, pp. 1139–1147 (cit. on p. 11).

[111] D. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations (ICLR)*. San Diega, CA, USA, 2015 (cit. on p. 11).

[112] L. M. Rios and N. V. Sahinidis. "Derivative-free optimization: a review of algorithms and comparison of software implementations". In: *Journal of Global Optimization* 56.3 (2013), pp. 1247–1293. ISSN: 0925-5001. DOI: 10.1007/s10898-012-9951-y (cit. on p. 11).

[113] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019 (cit. on p. 12).

[114] Y. Bengio. "Neural Networks: Tricks of the Trade, Second Edition". In: *Lecture Notes in Computer Science* (2012), pp. 437–478. ISSN: 0302-9743. DOI: 10.1007/978-3-642-35289-8_26 (cit. on p. 12).

[115] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958 (cit. on p. 14).

[116] W. S. McCulloch and W. Pitts. "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133. ISSN: 0007-4985. DOI: 10.1007/bf02478259 (cit. on p. 14).

[117] F. Rosenblatt. "The perceptron: A probabilistic model for information storage and organization in the brain". In: *Phys. Rev.* 65 (1958), p. 386 (cit. on p. 14).

[118] M. Lewenstein. "Quantum Perceptrons". In: *Journal of Modern Optics* 41.12 (1994), pp. 2491–2501. ISSN: 0950-0340. DOI: 10.1080/09500349414552331 (cit. on p. 14).

[119] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung. "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit." In: *Nature* 405.6789 (2000), pp. 947–51. ISSN: 0028-0836. DOI: 10.1038/35016072 (cit. on p. 14).

[120] X. Glorot, A. Bordes, and Y. Bengio. "Deep Sparse Rectifier Neural Networks". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Ed. by G. Gordon, D. Dunson, and M. Dudík. Vol. 15. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, Apr. 2011, pp. 315–323 (cit. on p. 15).

[121] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735 (cit. on p. 15).

[122] F. A. Gers and J. Schmidhuber. "LSTM Recurrent Networks Learn Simple Context-Free and Context-Sensitive Languages". In: *IEEE Transactions on Neural Networks* 12.6 (2001), pp. 1333–1340. ISSN: 1045-9227. DOI: 10.1109/72.963769 (cit. on p. 15).

[123] A. Graves and J. Schmidhuber. "Framewise phoneme classification with bidirectional LSTM and other neural network architectures". In: *Neural Networks* 18.5–6 (2005), pp. 602–610. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2005.06.042 (cit. on p. 15).

[124] D. Eck and J. Schmidhuber. "Artificial Neural Networks — ICANN 2002, International Conference Madrid, Spain, August 28–30, 2002 Proceedings". In: *Lecture Notes in Computer Science* (2002), pp. 284–289. ISSN: 0302-9743. DOI: 10.1007/3-540-46084-5_47 (cit. on p. 15).

[125] O. Sharir, Y. Levine, N. Wies, G. Carleo, and A. Shashua. "Deep Autoregressive Models for the Efficient Variational Simulation of Many-Body Quantum Systems". In: *Physical Review Letters* 124.2 (2020), p. 020503. ISSN: 0031-9007. DOI: 10.1103/physrevlett.124.020503 (cit. on pp. 15, 61).

[126] D. Wu, R. Rossi, F. Vicentini, and G. Carleo. "From tensor-network quantum states to tensorial recurrent neural networks". In: *Physical Review Research* 5.3 (2023), p. L032001. DOI: 10.1103/physrevresearch.5.l032001 (cit. on p. 15).

[127] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 1.4 (1989), pp. 541–551. ISSN: 0899-7667. DOI: 10.1162/neco.1989.1.4.541 (cit. on p. 15).

[128] Zhou and Chellappa. "Computation of optical flow using a neural network". In: *IEEE International Conference on Neural Networks* (1988), 71–78 vol.2. DOI: 10.1109/icnn.1988.23914 (cit. on p. 16).

[129] H. Bourlard and Y. Kamp. "Auto-association by multilayer perceptrons and singular value decomposition". In: *Biological Cybernetics* 59.4–5 (1988), pp. 291–294. ISSN: 0340-1200. DOI: 10.1007/bf00332918 (cit. on p. 16).

[130] G. E. Hinton and R. Zemel. "Autoencoders, Minimum Description Length and Helmholtz Free Energy". In: *Advances in Neural Information Processing Systems*. Ed. by J. Cowan, G. Tesauro, and J. Alspector. Vol. 6. Morgan-Kaufmann, 1993 (cit. on p. 16).

[131] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion". In: *Journal of Machine Learning Research* 11.110 (2010), pp. 3371–3408 (cit. on p. 16).

[132] D. P. Kingma and M. Welling. "Auto-Encoding Variational Bayes". In: *arXiv* (2013). DOI: 10.48550/arxiv.1312.6114 (cit. on pp. 16, 17, 48).

[133] D. P. Kingma and M. Welling. "An Introduction to Variational Autoencoders". In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392. ISSN: 1935-8237. DOI: 10.1561/2200000056 (cit. on p. 16).

[134] D. J. Rezende, S. Mohamed, and D. Wierstra. "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". In: *arXiv* (2014). DOI: 10.48550/arxiv.1401.4082 (cit. on pp. 16, 48).

[135] N. Dilokthanakul, P. A. M. Mediano, M. Garnelo, M. C. H. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan. "Deep Unsupervised Clustering with Gaussian Mixture Variational Autoencoders". In: *arXiv* (2016). DOI: 10.48550/arXiv.1611.02648 (cit. on p. 16).

[136] W.-N. Hsu, Y. Zhang, and J. R. Glass. "Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation". In: *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* (2017), pp. 16–23 (cit. on p. 16).

[137] M. Ehsan Abbasnejad, A. Dick, and A. van den Hengel. "Infinite Variational Autoencoder for Semi-Supervised Learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017 (cit. on p. 16).

[138] W. Xu, H. Sun, C. Deng, and Y. Tan. "Variational Autoencoder for Semi-Supervised Text Classification". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 31.1 (2017). ISSN: 2159-5399. DOI: 10.1609/aaai.v31i1.10966 (cit. on p. 16).

[139] H. Kameoka, L. Li, S. Inoue, and S. Makino. "Supervised Determined Source Separation with Multichannel Variational Autoencoder". In: *Neural Computation* 31.9 (2019), pp. 1891–1914. ISSN: 0899-7667. DOI: 10.1162/neco_a_01217 (cit. on p. 17).

[140] Y. Bengio, A. Courville, and P. Vincent. "Representation Learning: A Review and New Perspectives". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1798–1828. ISSN: 0162-8828. DOI: 10.1109/tpami.2013.50 (cit. on p. 17).

[141] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima". In: *arXiv* (2016). DOI: 10.48550/arxiv.1609.04836 (cit. on p. 19).

[142] L. Wu, Z. Zhu, and W. E. "Towards Understanding Generalization of Deep Learning: Perspective of Loss Landscapes". In: *arXiv* (2017). DOI: 10.48550/arxiv.1706.10239 (cit. on p. 19).

[143] P. Izmailov, D. Podoprikhin, T. Garipov, D. P. Vetrov, and A. G. Wilson. "Averaging Weights Leads to Wider Optima and Better Generalization". In: *Conference on Uncertainty in Artificial Intelligence*. 2018 (cit. on p. 19).

[144] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio. "Sharp Minima Can Generalize For Deep Nets". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by D. Precup and Y. W. Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 1019–1028 (cit. on p. 19).

[145] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein. "Visualizing the Loss Landscape of Neural Nets". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., 2018 (cit. on p. 19).

[146] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio. "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger. Vol. 27. Curran Associates, Inc., 2014 (cit. on p. 20).

[147] L. Sagun, L. Bottou, and Y. LeCun. "Eigenvalues of the Hessian in Deep Learning: Singularity and Beyond". In: *arXiv* (2016). DOI: 10.48550/arxiv.1611.07476 (cit. on p. 20).

[148] G. Alain, N. L. Roux, and P.-A. Manzagol. "Negative eigenvalues of the Hessian in deep neural networks". In: *arXiv* (2019). DOI: 10.48550/arxiv.1902.02366 (cit. on p. 20).

[149] A. Choromanska, M. Henaff, M. Mathieu, G. Ben Arous, and Y. LeCun. "The Loss Surfaces of Multilayer Networks". In: *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*. Ed. by G. Lebanon and S. V. N. Vishwanathan. Vol. 38. Proceedings of Machine Learning Research. San Diego, California, USA: PMLR, May 2015, pp. 192–204 (cit. on p. 20).

[150] L. Sagun, U. Evci, V. U. Guney, Y. Dauphin, and L. Bottou. "Empirical Analysis of the Hessian of Over-Parametrized Neural Networks". In: *arXiv* (2017). DOI: 10.48550/arxiv.1706.04454 (cit. on p. 20).

[151] B. Ghorbani, S. Krishnan, and Y. Xiao. "An Investigation into Neural Net Optimization via Hessian Eigenvalue Density". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 2019, pp. 2232–2241 (cit. on p. 20).

[152] P. Schulam and S. Saria. "Can You Trust This Prediction? Auditing Pointwise Reliability After Learning". In: *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*. Ed. by K. Chaudhuri and M. Sugiyama. Vol. 89. Proceedings of Machine Learning Research. PMLR, Apr. 2019, pp. 1022–1031 (cit. on pp. 20, 21).

[153] P. W. Koh and P. Liang. "Understanding Black-box Predictions via Influence Functions". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by D. Precup and Y. W. Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, Aug. 2017, pp. 1885–1894 (cit. on p. 20).

[154] P. W. W. Koh, K.-S. Ang, H. Teo, and P. S. Liang. "On the Accuracy of Influence Functions for Measuring Group Effects". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019 (cit. on p. 20).

[155] E. Barshan, M.-E. Brunet, and G. K. Dziugaite. "RelatIF: Identifying Explanatory Training Samples via Relative Influence". In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Ed. by S. Chiappa and R. Calandra. Vol. 108. Proceedings of Machine Learning Research. PMLR, Aug. 2020, pp. 1899–1909 (cit. on p. 20).

[156] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015 (cit. on pp. 21, 76–79).

[157] A. Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035 (cit. on pp. 21, 79).

[158] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. *JAX: composable transformations of Python+NumPy programs*. Version 0.3.13. 2018 (cit. on p. 21).

[159] J. B. Clarke, J. W. Hastie, L. H. E. Kihlborg, R. Metselaar, and M. M. Thackeray. "Definitions of terms relating to phase transitions of the solid state (IUPAC Recommendations 1994)". In: *Pure and Applied Chemistry* 66.3 (1994), pp. 577–594. ISSN: 0033-4545. DOI: 10.1351/pac199466030577 (cit. on p. 23).

[160] D.-L. Deng, X. Li, and S. D. Sarma. "Machine learning topological states". In: *Physical Review B* 96.19 (2017), p. 195145. ISSN: 2469-9950. DOI: 10.1103/physrevb.96.195145 (cit. on pp. 23, 72).

[161] P. Zhang, H. Shen, and H. Zhai. "Machine Learning Topological Invariants with Neural Networks". In: *Physical Review Letters* 120.6 (2018), p. 066401. ISSN: 0031-9007. DOI: 10.1103/physrevlett.120.066401 (cit. on p. 23).

[162] Y.-H. Tsai, M.-Z. Yu, Y.-H. Hsu, and M.-C. Chung. "Deep learning of topological phase transitions from entanglement aspects". In: *Physical Review B* 102.5 (2020), p. 054512. ISSN: 2469-9950. DOI: 10.1103/physrevb.102.054512 (cit. on p. 23).

[163] K. Shinjo, K. Sasaki, S. Hase, S. Sota, S. Ejima, S. Yunoki, and T. Tohyama. "Machine Learning Phase Diagram in the Half-filled One-dimensional Extended Hubbard Model". In: *Journal of the Physical Society of Japan* 88.6 (2019), p. 065001. ISSN: 0031-9015. DOI: 10.7566/jpsj.88.065001 (cit. on p. 23).

[164] H. Théveniaut and F. Alet. "Neural network setups for a precise detection of the many-body localization transition: Finite-size scaling and limitations". In: *Physical Review B* 100.22 (2019), p. 224202. ISSN: 2469-9950. DOI: 10.1103/physrevb.100.224202 (cit. on p. 23).

[165] X.-Y. Dong, F. Pollmann, and X.-F. Zhang. "Machine learning of quantum phase transitions". In: *Physical Review B* 99.12 (2019), p. 121104. ISSN: 2469-9950. DOI: 10.1103/physrevb.99.121104 (cit. on p. 23).

[166] K. Kawaki, Y. Kuno, and I. Ichinose. "Phase diagrams of the extended Bose-Hubbard model in one dimension by Monte-Carlo simulation with the help of a stochastic-series expansion". In: *Physical Review B* 95.19 (2017), p. 195101. ISSN: 2469-9950. DOI: 10.1103/physrevb.95.195101 (cit. on p. 23).

[167] F. Schindler, N. Regnault, and T. Neupert. "Probing many-body localization with neural networks". In: *Physical Review B* 95.24 (2017), p. 245134. ISSN: 2469-9950. DOI: 10.1103/physrevb.95.245134 (cit. on p. 23).

[168] Y.-H. Liu and E. P. L. v. Nieuwenburg. "Discriminative Cooperative Networks for Detecting Phase Transitions". In: *Physical Review Letters* 120.17 (2018), p. 176401. ISSN: 0031-9007. DOI: 10.1103/physrevlett.120.176401 (cit. on p. 23).

[169] S. J. Wetzel. "Unsupervised learning of phase transitions: From principal component analysis to variational autoencoders". In: *Physical Review E* 96.2 (2017), p. 022140. ISSN: 2470-0045. DOI: 10.1103/physreve.96.022140 (cit. on pp. 23, 27).

[170] K. Ch'ng, N. Vazquez, and E. Khatami. "Unsupervised machine learning account of magnetic transitions in the Hubbard model". In: *Physical Review E* 97.1 (2018), p. 013306. ISSN: 2470-0045. DOI: 10.1103/physreve.97.013306 (cit. on pp. 23, 27).

[171] M. Koch-Janusz and Z. Ringel. "Mutual information, neural networks and the renormalization group". In: *Nature Physics* 14.6 (2018), pp. 578–582. ISSN: 1745-2473. DOI: 10.1038/s41567-018-0081-4 (cit. on p. 23).

[172] P. Campadelli, E. Casiraghi, C. Ceruti, and A. Rozza. "Intrinsic Dimension Estimation: Relevant Techniques and a Benchmark Framework". In: *Mathematical Problems in Engineering* 2015 (2015), pp. 1–21. ISSN: 1024-123X. DOI: 10.1155/2015/759567 (cit. on p. 24).

[173] E. Facco, M. d'Errico, A. Rodriguez, and A. Laio. "Estimating the intrinsic dimension of datasets by a minimal neighborhood information". In: *Scientific Reports* 7.1 (2017), p. 12140. DOI: 10.1038/s41598-017-11873-y (cit. on p. 24).

[174] E. Facco, A. Pagnani, E. T. Russo, and A. Laio. "The intrinsic dimension of protein sequence evolution". In: *PLoS Computational Biology* 15.4 (2019), e1006767. ISSN: 1553-734X. DOI: 10.1371/journal.pcbi.1006767 (cit. on p. 24).

[175] A. A. Ramos, H. Socas-Navarro, A. L. Ariste, and M. J. M. González. "The Intrinsic Dimensionality of Spectropolarimetric Data". In: *The Astrophysical Journal* 660.2 (2007), pp. 1690–1699. ISSN: 0004-637X. DOI: 10.1086/513069 (cit. on p. 24).

[176] D. C. Laughlin. "The intrinsic dimensionality of plant traits and its relevance to community assembly". In: *Journal of Ecology* 102.1 (2014), pp. 186–193. ISSN: 0022-0477. DOI: 10.1111/1365-2745.12187 (cit. on p. 24).

[177] T. Mendes-Santos, X. Turkeshi, M. Dalmonte, and A. Rodriguez. "Unsupervised Learning Universal Critical Behavior via the Intrinsic Dimension". In: *Physical Review X* 11.1 (2021), p. 011040. DOI: 10.1103/physrevx.11.011040 (cit. on p. 24).

[178] M. E. Tipping and C. M. Bishop. "Mixtures of Probabilistic Principal Component Analyzers". In: *Neural Computation* 11.2 (1999), pp. 443–482. ISSN: 0899-7667. DOI: 10.1162/089976699300016728 (cit. on p. 24).

[179] B. Schölkopf, A. Smola, and K.-R. Müller. "Artificial Neural Networks — ICANN'97". In: *Lecture Notes in Computer Science* (1997), pp. 583–588. ISSN: 0302-9743. DOI: 10.1007/bfb0020217 (cit. on p. 24).

[180] P.-G. Martinsson, V. Rokhlin, and M. Tygert. "A randomized algorithm for the decomposition of matrices". In: *Applied and Computational Harmonic Analysis* 30.1 (2011), pp. 47–68. ISSN: 1063-5203. DOI: 10.1016/j.acha.2010.02.003 (cit. on p. 24).

[181] N. Halko, P. G. Martinsson, and J. A. Tropp. "Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions". In: *SIAM Review* 53.2 (2011), pp. 217–288. ISSN: 0036-1445. DOI: 10.1137/090771806 (cit. on p. 24).

[182] J. Dong, X.-J. Mao, C. Shen, and Y.-B. Yang. "Learning Deep Representations Using Convolutional Auto-encoders with Symmetric Skip Connections". In: *arXiv* (2016). DOI: 10.48550/arxiv.1611.09119 (cit. on p. 25).

[183] F. Häse, L. M. Roch, and A. Aspuru-Guzik. "Next-Generation Experimentation with Self-Driving Laboratories". In: *Trends in Chemistry* 1.3 (2019), pp. 282–291. ISSN: 2589-5974. DOI: 10.1016/j.trechm.2019.02.007 (cit. on pp. 27, 72).

[184] A. Lidiak and Z. Gong. "Unsupervised Machine Learning of Quantum Phase Transitions Using Diffusion Maps". In: *Physical Review Letters* 125.22 (2020), p. 225701. ISSN: 0031-9007. DOI: 10.1103/physrevlett.125.225701 (cit. on p. 27).

[185] J. F. Rodriguez-Nieva and M. S. Scheurer. "Identifying topological order through unsupervised machine learning". In: *Nature Physics* 15.8 (2019), pp. 790–795. ISSN: 1745-2473. DOI: 10.1038/s41567-019-0512-x (cit. on p. 27).

[186] W. Hu, R. R. P. Singh, and R. T. Scalettar. "Discovering phases, phase transitions, and crossovers through unsupervised machine learning: A critical examination". In: *Physical Review E* 95.6 (2017), p. 062122. ISSN: 2470-0045. DOI: 10.1103/physreve.95.062122 (cit. on p. 27).

[187] Y. Ming, C.-T. Lin, S. D. Bartlett, and W.-W. Zhang. "Quantum topology identification with deep neural networks and quantum walks". In: *npj Computational Materials* 5.1 (2019), p. 88. ISSN: 20573960. DOI: 10.1038/s41524-019-0224-x (cit. on p. 27).

[188] P. Rosson, M. Kiffner, J. Mur-Petit, and D. Jaksch. "Characterizing the phase diagram of finite-size dipolar Bose-Hubbard systems". In: *Physical Review A* 101.1 (2020), p. 013616. ISSN: 2469-9926. DOI: 10.1103/physreva.101.013616 (cit. on p. 27).

[189] P. Broecker, F. F. Assaad, and S. Trebst. "Altered leg muscle activity in volleyball players with functional ankle instability during a sideward lateral cutting movement". In: *Physical Therapy in Sport* 12.4 (2017), pp. 164–170. ISSN: 1466-853X. DOI: 10.1016/j.ptsp.2011.01.003 (cit. on p. 27).

[190] E. Greplova, A. Valenti, G. Boschung, F. Schäfer, N. Lörch, and S. D. Huber. "Unsupervised identification of topological phase transitions using predictive models". In: *New Journal of Physics* 22.4 (2020), p. 045003. ISSN: 13672630. DOI: 10.1088/1367-2630/ab7771 (cit. on p. 27).

[191] J. MacQueen et al. "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297 (cit. on p. 28).

[192] R. L. Thorndike. "Who belongs in the family?" In: *Psychometrika* 18.4 (1953), pp. 267–276. ISSN: 0033-3123. DOI: 10.1007/bf02289263 (cit. on p. 28).

[193] E. Schubert. "Stop using the elbow criterion for k-means and how to choose the number of clusters instead". In: *ACM SIGKDD Explorations Newsletter* 25.1 (2023), pp. 36–42. ISSN: 1931-0145. DOI: 10.1145/3606274.3606278 (cit. on p. 29).

[194] R. Chalapathy and S. Chawla. "Deep Learning for Anomaly Detection: A Survey". In: *arXiv* (2019). DOI: 10.48550/arxiv.1901.03407 (cit. on p. 29).

[195] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim. "A survey of deep learning-based network anomaly detection". In: *Cluster Computing* 22.Suppl 1 (2019), pp. 949–961. ISSN: 1386-7857. DOI: 10.1007/s10586-017-1117-8 (cit. on p. 29).

[196] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano, and L. Benini. "Anomaly Detection Using Autoencoders in High Performance Computing Systems". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (2019), pp. 9428–9433. ISSN: 2159-5399. DOI: 10.1609/aaai.v33i01.33019428 (cit. on p. 29).

[197] S. Acevedo, M. Arlego, and C. A. Lamas. "Phase diagram study of a two-dimensional frustrated antiferromagnet via unsupervised machine learning". In: *Physical Review B* 103.13 (2021), p. 134422. ISSN: 2469-9950. DOI: 10.1103/physrevb.103.134422 (cit. on p. 29).

[198] G.-Q. Zhang, L.-Z. Tang, L.-F. Zhang, D.-W. Zhang, and S.-L. Zhu. "Connecting topological Anderson and Mott insulators in disordered interacting fermionic systems". In: *Physical Review B* 104.16 (2021), p. L161118. ISSN: 2469-9950. DOI: 10.1103/physrevb.104.l161118 (cit. on p. 29).

[199] G. Muñoz-Gil, G. G. i. Corominas, and M. Lewenstein. "Unsupervised learning of anomalous diffusion data: an anomaly detection approach". In: *Journal of Physics A: Mathematical and Theoretical* 54.50 (2021), p. 504001. ISSN: 1751-8113. DOI: 10.1088/1751-8121/ac3786 (cit. on p. 29).

[200] N. Käming, A. Dawid, K. Kottmann, M. Lewenstein, K. Sengstock, A. Dauphin, and C. Weitenberg. "Code and data for Unsupervised machine learning of topological phase transitions from experimental data". In: (2021). DOI: 10.5281/zenodo.4700379 (cit. on pp. 33, 77, 78).

[201] M. V. Berry. "Quantal phase factors accompanying adiabatic changes". In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 392.1802 (1984), pp. 45–57. ISSN: 0080-4630. DOI: 10.1098/rspa.1984.0023 (cit. on pp. 33, 35, 36).

[202] N. R. Cooper, J. Dalibard, and I. B. Spielman. "Topological bands for ultracold atoms". In: *Reviews of Modern Physics* 91.1 (2019), p. 015005. ISSN: 0034-6861. DOI: 10.1103/revmodphys.91.015005 (cit. on pp. 33, 41).

[203] R. Gross and A. Marx. *Festkörperphysik*. München: De Gruyter Oldenbourg, 2014. ISBN: 9783110358704. DOI: doi:10.1524/9783110358704 (cit. on p. 35).

[204] N. Fläschner. "Ultracold Fermions in Tunable Hexagonal Lattices: From High-Precision Spectroscopy to the Measurement of Berry Curvature". PhD Thesis. Universität Hamburg, 2016 (cit. on pp. 36, 38, 43).

[205] J. N. Fuchs, F. Piéchon, M. O. Goerbig, and G. Montambaux. "Topological Berry phase and semiclassical quantization of cyclotron orbits for two dimensional electrons in coupled band models". In: *The European Physical Journal B* 77.3 (2010), pp. 351–362. ISSN: 1434-6028. DOI: 10.1140/epjb/e2010-00259-2 (cit. on p. 37).

[206] J. E. Avron, R. Seiler, and B. Simon. "Homotopy and Quantization in Condensed Matter Physics". In: *Physical Review Letters* 51.1 (1983), pp. 51–53. ISSN: 0031-9007. DOI: 10.1103/physrevlett.51.51 (cit. on p. 37).

[207] H. M. Price and N. R. Cooper. "Mapping the Berry curvature from semiclassical dynamics in optical lattices". In: *Physical Review A* 85.3 (2012), p. 033620. ISSN: 1050-2947. DOI: 10.1103/physreva.85.033620 (cit. on p. 37).

[208] C. Bena and G. Montambaux. "Remarks on the tight-binding model of graphene". In: *New Journal of Physics* 11.9 (2009), p. 095003. DOI: 10.1088/1367-2630/11/9/095003 (cit. on p. 38).

[209] J. Ibañez-Azpiroz, A. Eiguren, A. Bergara, G. Pettini, and M. Modugno. "Tight-binding models for ultracold atoms in honeycomb optical lattices". In: *Physical Review A* 87.1 (2013), p. 011602. ISSN: 1050-2947. DOI: 10.1103/physreva.87.011602 (cit. on p. 38).

[210] P. Hauke, M. Lewenstein, and A. Eckardt. "Tomography of Band Insulators from Quench Dynamics". In: *Physical Review Letters* 113.4 (2014), p. 045303. ISSN: 0031-9007. DOI: 10.1103/physrevlett.113.045303 (cit. on pp. 40, 42, 68).

[211] J. Dalibard, F. Gerbier, G. Juzeliūnas, and P. Öhberg. "Colloquium: Artificial gauge potentials for neutral atoms". In: *Reviews of Modern Physics* 83.4 (2011), pp. 1523–1543. ISSN: 0034-6861. DOI: 10.1103/revmodphys.83.1523 (cit. on p. 41).

[212] N. Fläschner, D. Vogel, M. Tarnowski, B. S. Rem, D.-S. Lühmann, M. Heyl, J. C. Budich, L. Mathey, K. Sengstock, and C. Weitenberg. "Observation of dynamical vortices after quenches in a system with topology". In: *Nature Physics* 14.3 (2018), pp. 265–268. ISSN: 1745-2473. DOI: 10.1038/s41567-017-0013-8 (cit. on p. 42).

[213] A. Dauphin, D.-T. Tran, M. Lewenstein, and N. Goldman. "Loading ultracold gases in topological Floquet bands: the fate of current and center-of-mass responses". In: *2D Materials* 4.2 (2017), p. 024010. ISSN: 2053-1583. DOI: 10.1088/2053-1583/aa6a3b (cit. on p. 43).

[214] A. Kumar, M. Rodriguez-Vega, T. Pereg-Barnea, and B. Seradjeh. "Linear response theory and optical conductivity of Floquet topological insulators". In: *Physical Review B* 101.17 (2020), p. 174314. ISSN: 2469-9950. DOI: 10.1103/physrevb.101.174314 (cit. on p. 43).

[215] T. Kitagawa, E. Berg, M. Rudner, and E. Demler. "Topological characterization of periodically driven quantum systems". In: *Physical Review B* 82.23 (2010), p. 235114. ISSN: 1098-0121. DOI: 10.1103/physrevb.82.235114 (cit. on p. 43).

[216] M. S. Rudner, N. H. Lindner, E. Berg, and M. Levin. "Anomalous Edge States and the Bulk-Edge Correspondence for Periodically Driven Two-Dimensional Systems". In: *Physical Review X* 3.3 (2013), p. 031005. DOI: 10.1103/physrevx.3.031005 (cit. on p. 43).

[217] K. M. M. Wintersperger. "Realization of Floquet topological systems with ultracold atoms in optical honeycomb lattices". PhD Thesis. LMU München, 2020 (cit. on p. 44).

[218] C. B. Menz. "Analysis of Topological Phases in Optical Lattices Using Floquet Theory". Masters's Thesis. Universität Hamburg, 2023 (cit. on p. 44).

[219] A. L. Hauschild. "Implementierung des Floquet-Matrix-Formalismus für getriebene optische Gitter". Bachelor's Thesis. Universität Hamburg, 2023 (cit. on p. 44).

[220] C. Ölschläger. "Floquet Engineering of Bloch Bands in Shaken Optical Lattices". PhD Thesis. Universität Hamburg, 2016 (cit. on p. 44).

[221] G. E. Hinton and R. R. Salakhutdinov. "Reducing the Dimensionality of Data with Neural Networks". In: *Science* 313.5786 (2006), pp. 504–507. ISSN: 0036-8075. DOI: 10.1126/science.1127647 (cit. on p. 46).

[222] A. Fitzgibbon, M. Pilu, and R. Fisher. "Direct least squares fitting of ellipses". In: *Proceedings of 13th International Conference on Pattern Recognition* 1 (1996), 253–257 vol.1. ISSN: 1051-4651. DOI: 10.1109/icpr.1996.546029 (cit. on p. 46).

[223] C. F. Ockeloen, A. F. Tauschinsky, R. J. C. Spreeuw, and S. Whitlock. "Detection of small atom numbers through image processing". In: *Physical Review A* 82.6 (2010), p. 061606. ISSN: 1050-2947. DOI: 10.1103/physreva.82.061606 (cit. on p. 48).

[224] R. Fung, A. M. Hanna, O. Vendrell, S. Ramakrishna, T. Seideman, R. Santra, A. Ourmazd, and A. Ourmazd. "Dynamics from noisy data with extreme timing uncertainty". In: *Nature* 532.7600 (2016), pp. 471–475. ISSN: 0028-0836. DOI: 10.1038/nature17627 (cit. on p. 48).

[225] R. Iten, T. Metger, H. Wilming, L. d. Rio, and R. Renner. "Discovering Physical Concepts with Neural Networks". In: *Physical Review Letters* 124.1 (2020), p. 010508. ISSN: 0031-9007. DOI: 10.1103/physrevlett.124.010508 (cit. on p. 48).

[226] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019 (cit. on pp. 50, 77, 78).

[227] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. "Image Quality Assessment: From Error Visibility to Structural Similarity". In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612. ISSN: 1057-7149. DOI: 10.1109/tip.2003.819861 (cit. on p. 50).

[228] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (cit. on p. 52).

[229] P. A. M. Dirac. "Quantum mechanics of many-electron systems". In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 123.792 (1929), pp. 714–733. ISSN: 0950-1207. DOI: 10.1098/rspa.1929.0094 (cit. on p. 59).

[230] S. R. White. "Density matrix formulation for quantum renormalization groups". In: *Physical Review Letters* 69.19 (1992), pp. 2863–2866. ISSN: 0031-9007. DOI: 10.1103/physrevlett.69.2863 (cit. on p. 60).

[231] S. Rommer and S. Östlund. "Class of ansatz wave functions for one-dimensional spin systems and their relation to the density matrix renormalization group". In: *Physical Review B* 55.4 (1997), pp. 2164–2181. ISSN: 1098-0121. DOI: 10.1103/physrevb.55.2164 (cit. on p. 60).

[232] U. Schollwöck. "The density-matrix renormalization group". In: *Reviews of Modern Physics* 77.1 (2005), pp. 259–315. ISSN: 0034-6861. DOI: 10.1103/revmodphys.77.259 (cit. on p. 60).

[233] R. Jastrow. "Many-Body Problem with Strong Forces". In: *Physical Review* 98.5 (1955), pp. 1479–1484. ISSN: 0031-899X. DOI: 10.1103/physrev.98.1479 (cit. on p. 60).

[234] M. V. d. Nest. "Simulating quantum computers with probabilistic methods". In: *arXiv* (2009). DOI: 10.48550/arxiv.0911.1624 (cit. on p. 60).

[235] W. K. Hastings. "Monte Carlo sampling methods using Markov chains and their applications". In: *Biometrika* 57.1 (1970), pp. 97–109. ISSN: 0006-3444. DOI: 10.1093/biomet/57.1.97 (cit. on p. 61).

[236] D. Luo, G. Carleo, B. K. Clark, and J. Stokes. "Gauge Equivariant Neural Networks for Quantum Lattice Gauge Theories". In: *Physical Review Letters* 127.27 (2021), p. 276402. ISSN: 0031-9007. DOI: 10.1103/physrevlett.127.276402 (cit. on p. 61).

[237] D. Wu, L. Wang, and P. Zhang. "Solving Statistical Mechanics Using Variational Autoregressive Networks". In: *Physical Review Letters* 122.8 (2019), p. 080602. ISSN: 0031-9007. DOI: 10.1103/physrevlett.122.080602 (cit. on p. 61).

[238] J. Chen, S. Cheng, H. Xie, L. Wang, and T. Xiang. "Equivalence of restricted Boltzmann machines and tensor network states". In: *Physical Review B* 97.8 (2018), p. 085104. ISSN: 2469-9950. DOI: 10.1103/physrevb.97.085104 (cit. on pp. 61, 62).

[239] I. Glasser, N. Pancotti, M. August, I. D. Rodriguez, and J. I. Cirac. "Neural-Network Quantum States, String-Bond States, and Chiral Topological States". In: *Physical Review X* 8.1 (2018), p. 011006. DOI: 10.1103/physrevx.8.011006 (cit. on p. 61).

[240] O. Sharir, A. Shashua, and G. Carleo. "Neural tensor contractions and the expressive power of deep neural quantum states". In: *Physical Review B* 106.20 (2022), p. 205136. ISSN: 2469-9950. DOI: 10.1103/physrevb.106.205136 (cit. on pp. 61, 72).

[241] P. Smolensky. "Information processing in dynamical systems: foundations of Harmony Theory". In: *Parallel Distributed Processing: explorations in the microstructure of cognition*. Ed. by D. Rumelhart, J. L. McClelland, and the PDP Research Group. Vol. 1. MIT Press, 1986. Chap. 6, pp. 194–281 (cit. on p. 62).

[242] G. E. Hinton. "Training Products of Experts by Minimizing Contrastive Divergence". In: *Neural computation* 14.8 (2002), pp. 1771–1800. ISSN: 0899-7667. DOI: 10.1162/089976602760128018 (cit. on p. 62).

[243] A. Barra, A. Bernacchia, E. Santucci, and P. Contucci. "On the equivalence of Hopfield networks and Boltzmann Machines". In: *Neural Networks* 34 (2012), pp. 1–9. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2012.06.003 (cit. on p. 62).

[244] D.-L. Deng, X. Li, and S. D. Sarma. "Quantum Entanglement in Neural Network States". In: *Physical Review X* 7.2 (2017), p. 021021. DOI: 10.1103/physrevx.7.021021 (cit. on p. 62).

[245] M. Schmitt and M. Heyl. "Quantum Many-Body Dynamics in Two Dimensions with Artificial Neural Networks". In: *Physical Review Letters* 125.10 (2020), p. 100503. ISSN: 0031-9007. DOI: 10.1103/physrevlett.125.100503 (cit. on pp. 63, 70).

[246] I. L. Gutiérrez and C. B. Mendl. "Real time evolution with neural-network quantum states". In: *Quantum* 6 (2022), p. 627. DOI: 10.22331/q-2022-01-20-627 (cit. on pp. 63, 70).

[247] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. C. Benjamin. "Theory of variational quantum simulation". In: *Quantum* 3 (2019), p. 191. DOI: 10.22331/q-2019-10-07-191 (cit. on p. 64).

[248] 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". In: *SciPost Physics* 12.5 (2022), p. 165. ISSN: 2542-4653. DOI: 10.21468/scipostphys.12.5.165 (cit. on pp. 65, 69, 70).

[249] S. Sorella. "Green Function Monte Carlo with Stochastic Reconfiguration". In: *Physical Review Letters* 80.20 (1998), pp. 4558–4561. ISSN: 0031-9007. DOI: 10.1103/physrevlett.80.4558 (cit. on p. 65).

[250] M. Cramer, M. B. Plenio, S. T. Flammia, R. Somma, D. Gross, S. D. Bartlett, O. Landon-Cardinal, D. Poulin, and Y.-K. Liu. "Efficient quantum state tomography". In: *Nature Communications* 1.1 (2010), p. 149. DOI: 10.1038/ncomms1147 (cit. on p. 66).

[251] M. K. Kurmapu, V. Tiunova, E. Tiunov, M. Ringbauer, C. Maier, R. Blatt, T. Monz, A. K. Fedorov, and A. Lvovsky. "Reconstructing Complex States of a 20-Qubit Quantum Simulator". In: *PRX Quantum* 4.4 (2023), p. 040345. DOI: 10.1103/prxquantum.4.040345 (cit. on pp. 66, 72).

[252] S. Lohani, B. T. Kirby, M. Brodsky, O. Danaci, and R. T. Glasser. "Machine learning assisted quantum state estimation". In: *Machine Learning: Science and Technology* 1.3 (2020), p. 035007. DOI: 10.1088/2632-2153/ab9a21 (cit. on p. 66).

[253] S. Lohani, T. A. Searles, B. T. Kirby, and R. T. Glasser. "On the Experimental Feasibility of Quantum State Reconstruction via Machine Learning". In: *IEEE Transactions on Quantum Engineering* 2 (2021), pp. 1–10. ISSN: 2689-1808. DOI: 10.1109/tqe.2021.3106958 (cit. on p. 66).

[254] O. Danaci, S. Lohani, B. T. Kirby, and R. T. Glasser. "Machine learning pipeline for quantum state estimation with incomplete measurements". In: *Machine Learning: Science and Technology* 2.3 (2021), p. 035014. DOI: 10.1088/2632-2153/abe5f5 (cit. on p. 66).

[255] S. Lohani, J. M. Lukens, R. T. Glasser, T. A. Searles, and B. T. Kirby. "Data-centric machine learning in quantum information science". In: *Machine Learning: Science and Technology* 3.4 (2022), 04LT01. DOI: 10.1088/2632-2153/ac9036 (cit. on p. 66).

[256] S. Lohani, J. M. Lukens, D. E. Jones, T. A. Searles, R. T. Glasser, and B. T. Kirby. "Improving application performance with biased distributions of quantum states". In: *Physical Review Research* 3.4 (2021), p. 043145. DOI: 10.1103/physrevresearch.3.043145 (cit. on p. 66).

[257] P. Bhatt, N. Singh, and S. Chaudhary. "Machine Learning for Advanced Functional Materials". In: (2023), pp. 75–94. DOI: 10.1007/978-981-99-0393-1_4 (cit. on p. 71).

[258] Y. Levine, O. Sharir, N. Cohen, and A. Shashua. "Quantum Entanglement in Deep Learning Architectures". In: *Physical Review Letters* 122.6 (2019), p. 065301. ISSN: 0031-9007. DOI: 10.1103/physrevlett.122.065301 (cit. on p. 72).

[259] K. Choo, T. Neupert, and G. Carleo. "Two-dimensional frustrated J1-J2 model studied with neural network quantum states". In: *Physical Review B* 100.12 (2019), p. 125124. ISSN: 2469-9950. DOI: 10.1103/physrevb.100.125124 (cit. on p. 72).

[260] M. Krenn, M. Erhard, and A. Zeilinger. "Computer-inspired quantum experiments". In: *Nature Reviews Physics* 2.11 (2020), pp. 649–661. DOI: 10.1038/s42254-020-0230-4 (cit. on p. 72).

[261] X. Gao, M. Erhard, A. Zeilinger, and M. Krenn. "Computer-Inspired Concept for High-Dimensional Multipartite Quantum Gates". In: *Physical Review Letters* 125.5 (2020), p. 050501. ISSN: 0031-9007. DOI: 10.1103/physrevlett.125.050501 (cit. on p. 72).

[262] M. Krenn and A. Zeilinger. "Predicting research trends with semantic and neural networks with an application in quantum physics". In: *Proceedings of the National Academy of Sciences* 117.4 (2020), pp. 1910–1916. ISSN: 0027-8424. DOI: 10.1073/pnas.1914370116 (cit. on p. 72).

[263] M. Krenn, R. Pollice, S. Y. Guo, M. Aldeghi, A. Cervera-Lierta, P. Friederich, G. d. P. Gomes, F. Häse, A. Jinich, A. Nigam, Z. Yao, and A. Aspuru-Guzik. "On scientific understanding with artificial intelligence". In: *Nature Reviews Physics* 4.12 (2022), pp. 761–769. DOI: 10.1038/s42254-022-00518-3 (cit. on p. 72).

[264] P. Friederich, M. Krenn, I. Tamblyn, and A. Aspuru-Guzik. "Scientific intuition inspired by machine learning-generated hypotheses". In: *Machine Learning: Science and Technology* 2.2 (2021), p. 025027. DOI: 10.1088/2632-2153/abda08 (cit. on p. 72).

[265] N. Yoshikawa, M. Skreta, K. Darvish, S. Arellano-Rubach, Z. Ji, L. B. Kristensen, A. Z. Li, Y. Zhao, H. Xu, A. Kuramshin, A. Aspuru-Guzik, F. Shkurti, and A. Garg. "Large language models for chemistry robotics". In: *Autonomous Robots* 47.8 (2023), pp. 1057–1086. ISSN: 0929-5593. DOI: 10.1007/s10514-023-10136-2 (cit. on p. 72).

[266] D. A. Boiko, R. MacKnight, B. Kline, and G. Gomes. "Autonomous chemical research with large language models". In: *Nature* 624.7992 (2023), pp. 570–578. ISSN: 0028-0836. DOI: 10.1038/s41586-023-06792-0 (cit. on p. 72).

Glossary

AE autoencoder 5, 16, 17, 24, 25, 28–30, 33, 46, 48, 51, 53–55, 57, 58, 76, 79

AI artificial intelligence 5–7, 14, 27, 72

ARNN autoregressive neural network 61, 63, 70

BEC Bose-Einstein condensate 1

BFM bose-fermi-mixture 33, 40, 41, 48

BZ Brillouin zone 37, 43–45

CAFCA common analysis framework for cold atoms 4

CCE categorical cross entropy 10

CNN convolutional neural network 6, 7, 15, 16, 45, 46, 50, 51, 55–57, 79

CoM center-of-mass 48–50, 73

DL deep learning 5–7, 9, 13, 19, 72

DNN deep neural network 5–7, 9–13, 15, 16, 20, 21, 26, 27, 29, 62

ED exact diagonalization 68

ELBO evidence lower bound 17, 18

LOO leave one out 20

LSTM long short-term memory 15

MCMC Markov chain Monte Carlo 61

ML machine learning 1, 3–13, 21, 23, 27, 29, 33, 40, 44, 48, 51, 57, 58, 60, 62, 64–66, 71, 72

MLP multilayer perceptron 7

MSE mean square error 10, 17, 53

NN neural network 3, 14–16, 19, 24–26, 30, 50, 60–62, 65, 66

NQS neural quantum state 4, 15, 59–70, 72

PCA principle component analysis 4, 6, 7, 9, 24, 28, 74, 110

PEPS projected entangled pair states 72

QPT quantum phase transition 23

RBM restricted boltzmann machine 61, 62, 66, 68–70

ReLU rectified linear unit 15

RNN recurrent neural network 15

SGD stochastic gradient descent 11, 19, 63, 66, 67, 69

SVD singular value decomposition 25

SVM support vector machnine 6, 7, 9

t-SNE t-distributed stochastic neighbour embedding 28

t-VMC time-dependent variational Monte Carlo 63, 68

TNS tensor network state 61, 72

ToF time of flight 8, 23, 40, 42, 43, 47–52, 71, 75

VAE variational autoencoder 16–18, 48–51, 58, 71, 77, 78

WCSS within-cluster sum of squares 28

List of Figures

List of Algorithms

Danksagung

Ich möchte diese Gelegenheit Nutzen um mich bei denen zu bedanken, die diese Arbeit ermöglichten und mich bei der Erstellung unterstützt haben. Zunächst danke ich Prof. Klaus Sengstock für die Ermöglichung, Betreuung und das Verfassen des Gutachtens der Arbeit. Durch seine Ideen hat er maßgeblich zur thematischen Gestaltung dieser Arbeit beigetragen. Auch sein Vertrauen in meine Arbeit und die damit verbundene Finanzierung vieler Möglichkeiten wie der Auslandsaufenthalt am ICFO sind nicht selbstverständlich. Dank gilt auch Prof. Ludwig Mathey, welcher die Erstellung des zweiten Gutachtens dieser Arbeit übernommen hat.

Besonderer Dank gilt Christof Weitenberg, der mich die meiste Zeit während dieser Arbeit begleitet hat und durch seinen Rat zu jeder Zeit bekräftigt aber auch hinterfragt hat. Sein breit aufgestellter Wissensschatz ist etwas ganz besonderes, von dem nicht nur ich sondern auch alle anderen Teammitglieder immer profitieren konnten. Nicht vergessen werde ich auch die gemeinsame Organisation der internationalen Konferenz zu Machine Learning in Natural Sciences. Diese Erfahrung hat mich in vielerlei Hinsicht weitergebracht und wäre ohne seine Unterstützung nicht möglich gewesen. Auch danke ich ihm für die Hinweise und Anmerkungen zur Verbesserung dieser Arbeit.

Furthermore, I have to thank my second scientific family at ICFO. Prof. Maciej Lewenstein who always welcomed me at his group at ICFO and supported me in every step of the thesis. He always provided helpful hints and ideas for all of the projects. Alexandre Dauphin is a living oracle regarding all scientific questions. His stamina working on dozens of projects at the same time and keeping track everywhere is outstanding. Paolo Stornati who is my first phone joker for every question regarding DMRG. And of course my collaborators Anna Dawid and Korbinian Kottmann. Thank you for the work we did together and all the motivating video calls we made during COVID Lockdowns. Thank you for this international opportunity.

Bei meinem Kollegen Martin Stieben möchte ich mich für die tolle Erfahrung bedanken eine eigene Lehrveranstaltung zu gestalten, zu organisiern und durchzuführen. Es war sehr anstregend, aber ich hatte eine Menge Spaß und konnte viel lernen. Großer Dank gilt ihm auch allein deshalb, da er alle Spezialwünsche, die für die Berechnung der Ergebnisse dieser Arbeit notwendig waren, im Rechnzentrum umgesetzt hat.

Für das positive Arbeitsklima in der Gruppe Sengstock möchte ich mich bei allen Mitdoktoranden, insbesondere Luca Asteria, Marcel Kosch, Henrik Zahn, Justus Brüggenjürgen und Mathis Fischer bedanken. Ihr seid alle eine große mentale Unterstützung gewesen und habt den Arbeitsalltag deutlich erleichtert. Besonderer Dank gilt auch Nils Mueller, der als HiWi wahnsinnigen Einsatz gezeigt hat und viele Programmieraufgaben, auch in von mir betreuten Projekten die es nicht in diese Arbeit geschafft haben, übernommen hat. Auch bedanken möchte ich mich bei den Bachelor und Masterstudierenden - Nora Bidzinski (BSc), Moritz Sträter (BSc+MSc), Felix Herbort (BSc), Abdul Raman-Rasul (BSc), Anna Lena Hauschild (BSc), Corinna Menz (MSc), Tom Blöcker (BSc), Mazen Adwan (MSc) - die ich betreuen durfte. Von euch habe ich viel gelernt und ihr seid eine Bereicherung für meine tägliche Arbeit gewesen.

Nicht zuletzt möchte ich mich bei meiner Familie für die Unterstützung während des Studiums und auch während der Promotion bedanken. Ohne die Ermöglichung eines Studiums wäre der gesamte Weg bis hier her nicht möglich gewesen.