Exploring many-body Physics with Recurrent Neural Networks

by

Mohamed Hibat Allah

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Physics

Waterloo, Ontario, Canada, 2023

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Fabien Alet Deputy Director, Laboratoire de Physique Théorique Université Paul Sabatier
Supervisors:	Juan Carrasquilla Faculty Member, Vector Institute Roger Melko Professor, Dept of Physics and Astronomy University of Waterloo
Internal Members:	Lauren Hayward Teaching Faculty, Perimeter Institute for Theoretical Physics Timothy Hsieh Faculty Member, Perimeter Institute for Theoretical Physics Pooya Ronagh Research Assistant Professor, Dept of Physics and Astronomy Institute of Quantum Computing, University of Waterloo
Internal-External Member:	Marcel Nooijen Professor, Dept of Chemistry, University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored. A Statement of Contributions is included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

I am the sole author of Chapters 1, 2, 3, and 9. Other parts of this thesis consist in part of five published manuscripts in collaboration with my co-authors in addition to work that is not published elsewhere. Research conducted in Chapters 4, 5, 6, 7, and 8 has been published in part in several academic journals which are detailed below. A summary of the main contributions is also provided.

Research presented in Chapters 4 and 5: Chapters 4 and 5 are the fruit of different projects. The first one was in collaboration with my supervisors Juan Carrasquilla and Roger Melko as well as with my other collaborators Lauren Hayward and Martin Ganahl, which resulted in a manuscript published in Physical Review Research:

- M. Hibat-Allah, M. Ganahl, L. E. Hayward, R. G. Melko, and J. Carrasquilla, "Recurrent neural network wave functions", Phys. Rev. Research 2, 023358 (2020).

Chapter 5 also includes research in collaboration with Juan Carrasquilla and Roger Melko published in the Machine Learning for Physical Science NeurIPS 2021 workshop:

- M. Hibat-Allah, R. G. Melko, and J. Carrasquilla, "Supplementing recurrent neural network wave functions with symmetry and annealing to improve accuracy", Machine learning and the physical sciences, NeurIPS (2021).

Part of this research is also included in Chapter 6. Additionally, Chapter 5 includes research not published elsewhere on the use of recurrent neural networks for 3D systems and for targeting excited states and low-energy excitation gaps.

Research presented in Chapter 6: Chapter 6 is the fruit of different research projects. The first one is in collaboration with Juan Carrasquilla, Roger Melko, Estelle Inack, and Roeland Wiersema on the development of a novel scheme for simulating a variational version of classical and quantum annealing for solving optimization problems, which resulted in a publication in Nature Machine Intelligence:

- M. Hibat-Allah, E. M. Inack, R. Wiersema, R. G. Melko, and J. Carrasquilla, "Variational neural annealing", Nature Machine Intelligence 3, 952–961 (2021).

The second project was in collaboration with two undergraduate students: Shoummo Khandoker and Jawaril Abedin as a part of their bachelor thesis at BRAC University. This project is a follow-up research on the previous project and which is published in Machine Learning Science and Technology journal:

- S. A. Khandoker, J. M. Abedin, and M. Hibat-Allah, "Supplementing recurrent neural networks with annealing to solve combinatorial optimization problems", Machine Learning: Science and Technology 4, 015026 (2023).

Research presented in Chapter 7: Chapter 7 includes research in collaboration with Juan Carrasquilla and Roger Melko, which is submitted to arXiv, and is currently under review up-to-date:

- M. Hibat-Allah, R. G. Melko, and J. Carrasquilla, "Investigating topological order using recurrent neural networks", arXiv:2303.11207.

Research presented in Chapter 8: Chapter 8 includes research results obtained during my research internship at Zapata Computing, in collaboration with Juan Carrasquilla, Alejandro Perdomo-Ortiz, Luis Serrano, Jing Chen, Atithi Acharya, and John Realpe-Gomez. The outcome of this project has not been published yet.

Summary of research contributions: For all the previous projects, to which I am contributing as a first author, I was responsible for contributing to conceptualizing the study design, carrying out data collection and analysis, and drafting and submitting manuscripts. My collaborators also contributed to the conceptualizing and study design. Additionally, they provided valuable assistance with the tasks I was responsible for, provided guidance during each step of the research, and also delivered helpful feedback on the draft versions of the manuscripts. With respect to the project conducted in collaboration with Shoummo Khandoker and Jawaril Abedin for which I participated as a co-author. I acted as a guide for my collaborators throughout their research and provided assistance with drafting and submitting the manuscript.

Abstract

Originally developed within the natural language processing community, Recurrent neural networks (RNNs) have enabled remarkable progress in speech recognition and machine translation. These architectures belong to the class of autoregressive generative models which allow for exact likelihood estimation and for a perfect sampling of multi-modal complex probability distributions. These desirable features suggest that RNNs may serve as ansätzes wave functions in the context of Variational Monte Carlo (VMC), where ansätzes based on a Markov chain Monte Carlo sampling scheme can be limited by long autocorrelation time. The main vision developed here replaces words with physical degrees of freedom as inputs to the RNN in order to transfer this technology to the context of many-body physics. In this thesis, we develop RNN wave functions in multiple spatial dimensions and with different flavors and symmetry considerations that can suit the need for different variational calculations. We demonstrate the power of RNN wave functions on various prototypical systems in one, two, and three spatial dimensions. We show that our ansatz can compete and outperform state-of-the-art methods such as Density Matrix Renormalization Group (DMRG). We also illustrate how to estimate observables, and entanglement, with which we can study different phases of matter including conventional and topologically ordered states, as well as phase transitions among different phases. We also develop a scheme for simulating a variational version of classical and quantum annealing for the purpose of solving combinatorial optimization problems. We demonstrate that our scheme, tested on various RNNs architectures, shows superior average performances compared to Markov-chain Monte Carlo implementation of classical annealing and quantum annealing on prototypical and real-world combinatorial optimization problems. We also highlight the importance of the annealing scheme in overcoming local minima in a traditional VMC optimization, especially in frustrated systems. We conclude this thesis with examples of exact constructions of traditional probability distributions based on RNNs as a first step toward understanding the promising performances of these architectures. In addition to tensor network and Monte Carlo methods, we believe that RNNs are a valuable toolbox for physicists to help address open questions in classical and quantum many-body physics.

Acknowledgements

In the name of GOD, the most Gracious, the most Merciful. I am very thankful to GOD for giving me the courage and the strength to complete this thesis, for having sustained me in the best times and in the challenging times, for all the countless blessings, and for all the precious knowledge I learned.

This thesis would not be possible without the support and help of many people. To start, I would like to express my sincere and immense gratitude to Juan Carrasquilla, my supervisor, who played the role of a coach, collaborator, mentor, colleague, motivator, teacher, and friend at different stages of my PhD. I am also grateful for his invaluable contributions to research development. I am also thankful to him for giving me the freedom to pursue my research and teaching interests, which contributed significantly to my personal growth since the start of my PhD. I will be always indebted to him for trusting me and for giving me the opportunity to pursue a PhD on a topic that became close to my heart. I am also grateful to my advisor Roger Melko for his guidance, and also for his invaluable feedback on my research at different stages of my PhD. Both Juan's and Roger's kindness, words of encouragement, and good spirit made this PhD journey more pleasant and more enjoyable.

My sincere thanks also go to Lauren Hayward who introduced me to machine learning research and was very supportive since I was a master's student. I would also like to offer my special thanks to her along with Timothy Hsieh for serving on my dissertation committee, for their invaluable feedback, and for sharing their excitement about my PhD research. I would like also to thank Fabien Alet for his valuable and constructive feedback on this thesis.

I am also grateful to Shoummo Khandoker and Jawaril Abedin for sharing their excitement about RNNs research and for interesting discussions. Many thanks also go to my fellow lab mates and researchers in my supervisors' research groups, including Andrew Jreissaty, Roeland Wiersema, Aroosa Ijaz, Matthew Duschenes, Schuyler Moss, Estelle Inack, Ejaaz Merali, and many others, for stimulating discussions and for sharing their excitement about my research. Additionally, my sincere thanks should go to Alejandro Perdomo-Ortiz for providing me with a valuable opportunity to pursue an internship at Zapata Computing during my PhD. I am also thankful to all members of the Quantum AI team at Zapata for their warm welcome during my internship.

I would like also to thank Debbie Guenther, Diana Goncalves, Anabela Bonada, Holly Rutherford, Krista Parsons, and Kayla Sutton for their support with all the administrative duties. Their valuable support was vital during all the stages of my PhD. Many thanks

should also go to Maïté Dupuis, Martin Ganahl, Di Luo, and all other researchers, staff members, and friends who helped me during my PhD journey.

Last but not least, I am very grateful to my dear wife Amal, and my dear parents Khadijah and El Mamoune for their support, care, encouragement, prayers, patience, understanding, and sacrifice. They are beacons of light in my life, and I pray that their light will continue to shine for the rest of my life. I will be always indebted to them, and will never be able to repay back. My appreciation also goes to my brother Hassan and to the rest of my family for their support.

Dedication

This thesis is dedicated to my wife and my parents.

Table of Contents

List of Figures

List of Tables

Nomenclature

- LLM: large language model.
- RNN: recurrent neural network.
- 1D: one-dimensional.
- 2D: two-dimensional.
- 3D: three-dimensional.
- 1DRNN or 1D RNN: one-dimensional recurrent neural network.
- 2DRNN or 2D RNN: two-dimensional recurrent neural network.
- 3DRNN or 3D RNN: three-dimensional recurrent neural network.
- pRNN: positive recurrent neural network.
- cRNN: complex recurrent neural network.
- TRNN: tensorized recurrent neural network.
- pGRU: positive gated recurrent unit.
- cTRNN: complex tensorized recurrent neural network.
- cTGRU: complex tensorized gated recurrent unit.
- VMC: variational Monte Carlo.
- SA: simulated annealing.
- SQA: simulated quantum annealing.

- PIQMC: path-integral quantum Monte Carlo.
- VCA: variational classical annealing.
- VQA: variational quantum annealing.
- RVQA: regularized variational quantum annealing.
- TFIM: transverse-field Ising model.
- EE: entanglement entropy.
- TEE: topological entanglement entropy.
- DMRG: density-matrix renormalization group.
- QMC: quantum Monte Carlo.
- RBM: restricted Boltzmann machine.
- QCBM: quantum circuit Born machine.
- MPS: matrix product state.
- PEPS: pair-entangled project states.
- $\partial_{\boldsymbol{\lambda}}$: partial derivatives with respect to a vector of parameters $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_N)$.

Chapter 1

Introduction

The contemporary advances in our understanding of quantum systems (quantum materials, molecules, and light) are at the origin of many technological revolutions during the last century, such as nanotechnology, medical imaging with MRI machines, lasers, and LED light bulbs. Throughout these advancements, the interplay between theoretical, experimental, and numerical studies is crucial to igniting these revolutions. With theoretical studies alone, we cannot provide a full description of quantum systems with their exotic phenomena as they carry an exponential number of degrees of freedom. There, numerical studies are great tools to bridge this gap. Yet, there is still an open challenge; can we use classical computers to simulate quantum systems that live in an exponential complexity space?

Monte Carlo simulations are one example of many other numerical tools aiming to answer this question. The goal of these methods is to estimate observables by taking into account the most important configurations without having to explore the space of all possible configurations [1–3]. Monte Carlo methods gained significant success as they provided us with a lot of answers in the case of bosonic systems [4]. However, these methods are limited in the fermionic case where the sign problem is often the biggest challenge [5]. Historically, variational methods have played an important role in circumventing the sign problem as in the case of Helium superfluidity [6, 7]. Other variational approaches have also emerged to come to the rescue. One of them is tensor network methods, which proved to be very powerful at estimating ground state energies of 1D and 2D systems [8]. Interestingly, tensor networks are considered the gold standard for studying one-dimensional systems. However, in two spatial dimensions, tensor network methods start to face practical challenges where the computational cost becomes unrealistic compared to the 1D case [8]. Other variational approaches are based on quantum computers where the param-

eters of a quantum circuit can be optimized to find an approximation of a ground state of interest. Yet this approach is still limited in terms of quantum hardware as well as in terms of optimization hurdles [9, 10]. All in all, there is still a lack of efficient numerical methods for studying quantum many-body systems with unknown physics. An interesting example is the case of quantum systems establishing high-temperature superconductivity (electricity conduction without dissipation). Without doubt, devising more efficient numerical methods to shed more light on this phenomenon can help to achieve the long-term goal of building high-temperature superconductors under realistic lab conditions (e.g., at room temperature and close to atmospheric pressure), and thus promoting the next generation of future technologies with promising applications ranging from quantum computing to energy efficiency.

Nowadays, neural networks are making a breakthrough in the artificial intelligence (AI) community [11] since the AlexNet breakthrough in computer vision [12]. These systems can recognize faces and objects in a picture [13], understand human speech [14, 15], drive cars [16], assist people in need [17], write new music symphonies [18], and even win over world champions in very difficult games such as Alpha Go [19]. Their phenomenal success is spreading out to all areas of science as they provide great solutions for many interesting real-world problems. In particular, they are used in chemistry to better understand complicated chemical reactions and to predict new ones [20]. In biology, they proved to be useful in the quest for solving the protein folding problem whose resolution would uncover the reason behind many diseases [21]. In climatology, they can be used to correct the weather prediction errors due to Chaos [22]. In medicine, they can promote discoveries of new drugs [23] and assist doctors in analyzing scans of patients to ensure better disease diagnostics [24]. The power of neural networks has also been harnessed to explore interesting problems in cosmology, material science, quantum chemistry, and statistical physics [25]. Just like physics has inspired the machine learning community to develop more advanced architectures [26, 27], we are now in the era where machine learning research is giving back tremendous benefits to the physics community.

All in all, machine learning with neural networks demonstrated its success at solving tasks that require, in principle, an exponential number of resources. This interesting property is what makes neural networks a great candidate for the study of many-body systems. These architectures have also demonstrated phenomenal success in the context of many-body physics. They have proved useful for a wide array of tasks including the classification of phases of matter [28–31], quantum state tomography [32, 33], finding ground states of quantum systems [34–43], studying open quantum systems [44, 45], and simulating quantum circuits [46–48], among many others [49–52].

After the computer vision revolution in 2012, a new revolution has recently emerged

in natural language processing (NLP) with new algorithmic advances that allow computers to produce coherent language and understand human speech. This revolution has been enabled by the surge of large language models (LLMs) [53–55]. Famous examples are ChatGPT and GPT-4 exhibiting human-level performances on different professional and academic tasks [56]. As language and many-body physics, both share the curse of dimensionality in the space of words and physical degrees of freedom respectively, it is a plausible idea to take advantage of the recent advances of LLMs for applications in many-body physics. This thesis is a preliminary step in this direction where we make use of recurrent neural networks (RNNs) [57–61], a key language model that has been used since the infancy of NLP. We show that RNNs constitute powerful models for studying many-body systems including quantum systems. They are particularly flexible and intuitive; they have interesting correspondence with tensor networks and they can also handle relevant physical symmetries. Furthermore, they can be defined in multiple spatial dimensions as well as in the context where the dimension is not well defined. They also have a low computational cost and are also very competitive with state-of-the-art numerical methods.

In this thesis, we focus most of the time on the ground state problems of classical and quantum many-body systems and we demonstrate that RNNs are establishing competitive results with state-of-the-art numerical methods on various benchmarks in different spatial dimensions. We also show that RNNs can approximate low-energy excited states. Furthermore, we showcase the ability of RNNs to handle complex numbers to target the ground state of non-stosquatic Hamiltonians, such as the Heisenberg model, and the J_1-J_2 models. We also illustrate that RNNs are capable of encoding topological order in quantum matter through the examples of 2D toric code, Bose-Hubbard model on the Kagome lattice. We also provide a real-world use case for using RNNs in investigating topological order on the new platform of analog simulators, namely in Rydberg atom arrays.

Moreover, we also develop a novel annealing framework to find global solutions to classical optimization problems. This framework turns out to be also useful for mitigating local minima in a generic variational calculation aiming to find the ground state of a quantum many-body system. Finally, we provide examples of exact RNN constructions of prototypical probability distributions, which turn out to be competitive in terms of computational cost compared to other classical and quantum models. To highlight and describe our findings while introducing the necessary preliminaries, we divide the main body of the thesis into seven different chapters as follows:

Chapter 2:

We define preliminary concepts in statistical physics and quantum many-body physics to lay the foundation for the definition of the variational principle in the next chapter. On the classical side of many-body physics, we motivate the concepts of free energy, phase transitions as well as critical exponents. On the quantum side, we define the notions of a wave function, observables, and of entanglement entropy. We also motivate the existence of phase transitions in a quantum system. We finally illustrate the difficulty of solving a generic ground state problem as a motivation for using the variational principle in the following chapter.

Chapter 3:

We introduce the framework of variational Monte Carlo that is based on the variational principle. This principle is typically used for estimating the ground state and low-energy excited states of quantum many-body systems. We also demonstrate how this framework can be used to estimate observables and entanglement entropies. We further illustrate the possibility of using this framework to study classical many-body systems.

Chapter 4:

We motivate the use of RNNs as ansätzes wave functions and probability distributions. We first define the process of autoregressive sampling in RNNs to produce perfect and uncorrelated configurations. We then define positive RNN wave function and complex RNN wave functions as a class of ansätzes aimed to target the ground states of stoquastic and non-stoquastic Hamiltonians respectively. We also extend the definition of RNNs to multiple spatial dimensions and also in the case of an undefined spatial dimension. We further highlight the possibility to include tensor structures in RNNs in a similar fashion to tensor networks. We then show how to encode continuous and discrete physical symmetries in an RNN wave function to enhance accuracy in a variational calculation.

Chapter 5:

We demonstrate the ability of RNN to find accurate ground state energies as well as to estimate observables and entanglement entropies. We also showcase their ability to compete with state-of-the-art numerical methods, especially in two spatial dimensions.

We further illustrate that RNNs can target excited states through the use of a built-in $U(1)$ symmetry. We also show that RNNs are valuable tools for studying phase transitions through the example of the 3D transverse-field Ising model. Finally, we provide benchmarks for different hyperparameters of the RNN with their effect on accuracy in a variational calculation.

Chapter 6:

We develop a new framework for solving optimization problems based on the physical principle of annealing. We introduce both classical and quantum versions of our annealing frameworks and we demonstrate their effectiveness at solving combinatorial optimization problems using RNNs. In particular, we show that our variational classical annealing framework is superior on average compared to traditional annealing algorithms when targeting prototypical spin-glass models as well as real-world combinatorial optimization problems. We also highlight the value of annealing in mitigating the effect of local minima when searching for the ground state of frustrated quantum systems.

Chapter 7:

We shift our attention to topological order which is of practical value in the area of topological quantum computing. We showcase the ability of RNNs to encode and detect topological order on the 2D toric code as well as in a Bose-Hubbard model on the Kagome lattice. We also show that RNNs provide a negative signature for the existence of topological order in Rydberg atom arrays on the Kagome lattice.

Chapter 8:

We demonstrate different RNN constructions of prototypical probability distributions, and we compare them to other generative models namely tensor networks, restricted Boltzmann machines, as well as quantum circuit Born machines. We compare the generative models on the bimodal distribution, parity distribution, cardinality distribution, as well as on toric code distributions. We find that RNNs are competitive with the constructions of the other models in terms of computational cost.

Chapter 2

Many-body Physics

The behavior of many-body systems is a topic of interest in a wide variety of areas of science including physics, chemistry, and biology. Exotic properties of a many-body system, at the macroscopic scale, are a consequence of the collective behavior of microscopic degrees of freedom that are interacting under elementary rules. In this chapter, we provide an introduction to many-body physics from the point of view of statistical physics on the classical side (Sec. 2.1), and quantum many-body physics on the quantum side (Sec. 2.2). We discuss different elementary concepts which are aimed at laying the foundation for the concepts discussed in the following chapters.

2.1 Statistical physics

Statistical mechanics is a probabilistic approach to studying the equilibrium properties of a many-body system. This approach circumvents the need for an intractable approach that tracks the evolution of every elementary particle in order to predict the physics of a many-body system of interest. To provide the necessary prerequisites of statistical mechanics for the following chapters, we discuss the concepts of Shannon entropy, Boltzmann probability, free energy, observables, phase transitions, as well as critical exponents in the following sections.

2.1.1 Shannon entropy

Entropy is a quantity that is often associated with a disorder or uncertainty. It has known its first origins in classical thermodynamics where the entropy is assumed to be increasing with the flow of time according to the second law of thermodynamics. In the context of statistical physics, it is associated with a measure of lack of information about a certain system.

This concept has an interesting correspondence with information entropy in the context of information theory. To illustrate this concept, let us take the example of a treasure hidden in one of $\Omega = 2^Q$ seats in a room [62] with an equal probability of being in one of the seats. The uncertainty about the location of the treasure corresponds to the minimal number of questions that can be asked to dismantle the uncertainty about the location of the treasure. An efficient way to do that is by the mean of dichotomy. Thus we would need to ask $Q = \log_2(\Omega)$ questions. In this case, the information entropy is defined as Q. In general, it is given by

$$S = \log_2(\Omega),$$

which is the same expression predicted by Ludwig Boltzmann $S = k_B \log(\Omega)$, up to a constant, for the entropy of a physical system since the nineteen century. The previous formula corresponds to the case when all configurations are equiprobable. In the generic case, this assumption does not generally hold. In this case, each configuration $\boldsymbol{\sigma}$ in our system of interest[1] has a probability $P(\boldsymbol{\sigma})$, then the entropy, also known as Shannon entropy, is given as:

$$S = k_B \sum_{\boldsymbol{\sigma}} P(\boldsymbol{\sigma}) \log(1/P(\boldsymbol{\sigma})) = -k_B \sum_{\boldsymbol{\sigma}} P(\boldsymbol{\sigma}) \log(P(\boldsymbol{\sigma})),$$

where $\log_2(1/P(\boldsymbol{\sigma}))$ can be interpreted as the minimal number of questions to be asked to know the configuration $\boldsymbol{\sigma}$ with certainty.

Hereafter, we assume that the physical constants such as the Boltzmann constants are set to 1 for the sake of simplicity and numerical convenience.

2.1.2 Boltzmann probability

Computing expected values of observables such as magnetization, pressure, or other physical quantities can be done through the use of different statistical ensembles with

[1]Here we focus on the discrete case, where $\boldsymbol{\sigma}$ corresponds to a bitstring $(\sigma_1, \sigma_2, \ldots, \sigma_N)$ that can be used to identify each configuration. This convention is more relevant in the context of spin configurations that are of interest in this thesis.

different assumptions namely the micro-canonical ensemble, canonical ensemble, and the grand canonical ensemble, which are equivalent in the thermodynamic limit [63]. The micro-canonical ensemble assumes a physical system is isolated, i.e., the energy, particle numbers, and volume are conserved, which means that all possible configurations (or micro-states) are equiprobable. In this chapter, we focus on the canonical ensemble, which is the most relevant to the work conducted in this thesis. This ensemble relaxes the conserved energy assumption of the micro-canonical ensemble and assumes that a macroscopic system of interest has a fixed temperature T which is imposed by an external large heat path. For the grand-canonical ensemble, we further relax the conservation of the number of particles while imposing a fixed chemical potential. However, this ensemble is beyond the scope of this thesis.

To derive the expression of the probabilities $P(\boldsymbol{\sigma})$ in the canonical ensemble, we can start from the principle of maximum entropy, where entropy is assumed to be maximized at equilibrium as expected from the second law of classical thermodynamics. This maximization is subject to two constraints. The first one corresponds to the normalization of the probability P to one to maintain a probabilistic interpretation. The second condition forces the measured energy to be equal to the expected energy. This idea comes from the intuition that, the energy of a system at thermal equilibrium should only depend on temperature. This principle allows us to obtain the Boltzmann probability distribution at a given temperature T:

$$P(\boldsymbol{\sigma}) = \frac{\exp(-\beta E(\boldsymbol{\sigma}))}{Z}, \tag{2.1}$$

where $\beta = 1/T$ is the inverse temperature, and Z is the normalization known as the partition function:

$$Z = \sum_{\boldsymbol{\sigma}} \exp(-\beta E(\boldsymbol{\sigma})). \tag{2.2}$$

Based on the Boltzmann probability, we can show that the equilibrium entropy is given as:

$$\begin{aligned} S &= -\sum_{\boldsymbol{\sigma}} P(\boldsymbol{\sigma}) \log(P(\boldsymbol{\sigma})), \\ &= \beta \langle E \rangle + \log(Z). \end{aligned} \tag{2.3}$$

2.1.3 Free energy

Free energy is a state function known in the context of classical thermodynamics as the amount of useful energy (or work) after deducting the energy lost in the form of heat. In

statistical physics, the free energy F can be defined as:

$$F = \langle E \rangle - TS,$$

where TS can be interpreted as the amount of heat that is subtracted from the expected energy of the system. From Eq. (2.3), we can also show that at thermal equilibrium:

$$F = -T \log(Z). \tag{2.4}$$

2.1.4 Classical observables

The derivatives of the free energy contain sufficient information about all other thermodynamic quantities. In particular, the entropy is given by:

$$S = -\frac{\partial F}{\partial T}.$$

Additionally, the expected energy can be computed as:

$$\langle E \rangle = \frac{\partial(\beta F)}{\partial \beta}.$$

Furthermore, the heat capacity is given as:

$$C_v = -\beta^2 \frac{\partial^2(\beta F)}{\partial \beta^2}.$$

A similar analysis can be conducted for other quantities, such as magnetization, magnetic susceptibility, and so on. From the expression (2.4), it is also clear that the partition function Z holds sufficient information about the observables.

2.1.5 Phase transitions

The collective behavior of a large number of microscopic degrees of freedom can result in specific phases of matter, such as the paramagnetic phase which corresponds to a state that has zero magnetization. There is also the example of a ferromagnetic phase where the magnetization is non-zero in such as way that the system can interact with magnets. Under a change in external conditions such as temperature, a system can endure a phase transition.

The non-analyticity of the free energy F for a system at the thermodynamic limit is a clear indicator of a phase transition. This non-analyticity can be in the form of a discontinuity in the derivatives of F. In a first-order phase transition, the first derivative of the free energy becomes discontinuous, which results in an energy discontinuity. A famous real-world example of this transition is the ice-liquid transition in water. Furthermore, in a second-order phase transition (also known as a continuous phase transition), the second derivative of the free energy is discontinuous, which results in quantities like the heat capacity being divergent. An example of this transition is the ferromagnetic-paramagnetic phase transition.

For a finite system size, the free energy is always analytic, meaning there is no phase transition for a finite number of degrees of freedom N. However, when taking the limit N to or close to infinity, such as in the case of the Avogadro number, the non-analytic behavior of the free energy starts to emerge in a phase transition. This observation highlights the importance of conducting finite-size scaling through extrapolation in numerical experiments. This step allows us to derive conclusions about the expected behavior in the thermodynamic limit, as indicated in the next section.

2.1.6 Critical exponents

Critical exponents are quantities that characterize the behavior of observables near a second-order phase transition [63–66]. The values of these exponents are insensitive to the fine details of a many-body system, and they only depend on the general features (or relevant features). In particular, they are conjectured to be universal quantities, that can allow classifying different types of phase transitions based on the values of these critical exponents.

Near a phase transition, correlations are expected to be stronger among the different physical degrees of freedom. For this reason, it is expected to diverge at the phase transition with the following power law:

$$\xi \sim |t|^{-\nu}, \tag{2.5}$$

where $t = (T - T_c)/T_c$ is the reduced temperature and T_c is the critical temperature. This power-law decay is predicted by the phenomenological Landau theory of phase transitions [65]. This theory also predicts the divergence of other quantities. In this thesis, we focus on the magnetization per site, which has the following scaling law:

$$\langle |m| \rangle \sim |t|^{\beta}. \tag{2.6}$$

The scaling laws of other physical quantities such as heat capacity and magnetic susceptibility also follow a power law but with different exponents.

The scaling hypothesis of statistical physics [66] provides a finite-scaling of the physical quantities such as the magnetization as follows:

$$\langle |m| \rangle = |t|^{\beta} f(\xi/L), \tag{2.7}$$

where f is a universal scaling function. For a finite system size, the correlation length ξ is limited by the linear system size L, thus ξ reaches the maximum length L. As a result:

$$L \sim |t|^{-\nu}, \tag{2.8}$$

or equivalently:

$$|t| \sim L^{-1/\nu}, \tag{2.9}$$

which means, that near the transition, the critical temperature is size-dependent and tends the critical temperature T_c in the infinite limit $L \to \infty$. This also means that there is no divergence for finite system sizes. From the previous equation, we can deduce, based on Eq. (2.7), that:

$$\langle |m| \rangle L^{\beta/\nu} = g(tL^{1/\nu}) \tag{2.10}$$

for a universal scaling function g. Here the argument g is independent of the system size L. As a result, we can extract the critical temperature T_c as well as the exponent β and ν by finding the right values that collapse different '$\langle |m| \rangle L^{\beta/\nu}$' versus '$tL^{1/\nu}$' curves. This setup can be also implemented for other quantities to extract other critical exponents. An interesting choice for the finite size scaling study, that is dimensionless, is the Binder cumulant:

$$B = 1 - \frac{\langle m^4 \rangle}{3\langle m^2 \rangle}, \tag{2.11}$$

which follows the finite scaling law:

$$B = l(tL^{1/\nu}), \tag{2.12}$$

for a universal scaling function l, thus allowing for the extraction of the exponent ν independently of other critical exponents. The finite size scaling procedure is numerically illustrated in Sec. 5.4.

2.2 Quantum many-body physics

After introducing the preliminary concepts in statistical physics, we now shift our attention to quantum many-body systems where quantum fluctuations play a key role in addition to the collective behavior addressed in the previous section. By starting from the basic law of quantum mechanics and with simple interaction rules, the collective behavior of many quantum particles can give rise to exotic phases of matter as highlighted by the famous quote 'More is different' by Anderson [67]. The latter can be of interest to real-world applications. To name a few, there is superconductivity that is useful for conducting electricity without resistance and for magnetic levitation. Additionally, some fluids known as superfluids can flow without viscosity. We can also mention topological materials which hold promising potential for the future generation of quantum computers.

In this chapter, we introduce the concept of many-body wave functions in quantum mechanics and we provide key definitions such as observables, entanglement, entanglement entropy as well as quantum phase transitions that are of practical use in this thesis. Finally, we highlight a key problem that we aim to solve throughout this thesis, known as the ground state problem, using recurrent neural networks.

2.2.1 Quantum wave functions

The wave function is the most fundamental object in quantum physics and understanding its properties is at the heart of many areas of science such as condensed matter, high-energy physics, and quantum chemistry. It can have several interpretations based on different contexts. Here we focus on the case of particles with discrete quantum degrees of freedom.

For a quantum particle that has two degrees of freedom 0 and 1 (also known as a qubit), the wave function can be seen here as a two-dimensional complex-valued vector. This wave function can be denoted as $|\mathbf{\Psi}\rangle$ and it can generally correspond to a superposition of the state particle being in '0' (denoted as 'ket' $|0\rangle \equiv (1,0)^t$), and the state of the particle being in '1' (denoted as $|1\rangle \equiv (0,1)^t$). This superposition encodes the quantum mechanical uncertainty of the particle being in one of the possible states. This superposition can be written as:

$$|\mathbf{\Psi}\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{2.13}$$

where α and β are complex numbers. Within the context of quantum mechanics, $|\alpha|^2$ and $|\beta|^2$ can be interpreted as the probabilities of the particle being in states '0' and '1'

respectively. For this reason, the wave function is required to be L_2 normalized to 1, i.e.

$$||\mathbf{\Psi}||_2 = 1.$$

In our example, this property corresponds to $|\alpha|^2 + |\beta|^2 = 1$. Additionally, a probability can be written as a projection. For instance:

$$|\alpha|^2 = |\langle 0|\, \mathbf{\Psi}\rangle|^2,$$

where 'bra' $\langle 0|$ can be interpreted as a co-vector or as a transpose of the 'ket' vector $|0\rangle$.

For N particles with two degrees of freedom, the wave function can be a superposition of all possible states of the N particles, i.e.,

$$|\mathbf{\Psi}\rangle = \sum_{\sigma_1,\sigma_2,\ldots\sigma_N} \Psi(\sigma_1, \sigma_2, \ldots \sigma_N)\, |\sigma_1\rangle \otimes |\sigma_2\rangle \otimes \cdots \otimes |\sigma_N\rangle\,, \tag{2.14}$$

where $\otimes$ is the tensor product operators between the different vectors (or kets). In this case, the wave function can be thought of as a 2^N complex-valued vector. For simplicity of notation, quantum physicists usually prefer to drop the tensor product and use the following:

$$|\mathbf{\Psi}\rangle = \sum_{\sigma_1,\sigma_2,\ldots\sigma_N} \Psi(\sigma_1, \sigma_2, \ldots \sigma_N)\, |\sigma_1, \sigma_2, \ldots, \sigma_N\rangle\,. \tag{2.15}$$

It is worth noting that the space where wave functions live is known as the Hilbert space $\mathcal{H}$, which is our example has a dimensionality of 2^N, and can be constructed as a tensor product of the individual (local) Hilbert spaces of each individual particle.

2.2.2 Operators and observables

One way to act on wave functions is by applying operators $\hat{O}$, which correspond in the example of N qubits to matrices with $2^N \times 2^N$. These operators can be either used to evolve the wave function or to compute observables as we discuss in this section.

A key operator in quantum physics is the quantum Hamiltonian, which is the counterpart of an energy function in statistical physics. This operator typically dictates the interactions in a many-body system of particles. Additionally, its eigenvalues can be interpreted as energy spectra with the lowest being the lowest energy the system can have. The corresponding eigenvectors are wave functions describing the state of the many-body system. A typical example of a many-body Hamiltonian is the following:

$$\hat{H} = -\sum_{i,j=1}^{N} J_{ij}\hat{\sigma}_i^z\hat{\sigma}_j^z - \sum_{i=1}^{N} h_i\hat{\sigma}_i^x,$$

where J_{ij}, h_i are coupling parameters. Furthermore, $\hat{\sigma}_i^{(x,y,z)}$ are Pauli matrices acting on site i, and that are defined as follows:

$$\hat{\sigma}^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \hat{\sigma}^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \hat{\sigma}^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{2.16}$$

The notation $\hat{\sigma}_i^x$ is in reality defined as a tensor product $I_2 \otimes \ldots \otimes \hat{\sigma}_i^x \otimes \ldots \otimes I_2$ with identity operators I_2. The full expression is omitted for the sake of simplicity. The same also holds for the terms $\hat{\sigma}_i^z \hat{\sigma}_j^z$. For instance, if we assume $1 < i < j < N$ then $\hat{\sigma}_i^z \hat{\sigma}_j^z \equiv I_2 \otimes \ldots \otimes \hat{\sigma}_i^x \otimes \ldots \otimes \hat{\sigma}_j^x \otimes \ldots \otimes I_2$.

Importantly, given a ground state or a low energy state $|\mathbf{\Psi}\rangle$ of a many-body Hamiltonian $\hat{H}$, it is of practical use to compute physical observables that we can measure in the lab, such as magnetization or magnetic susceptibility of a spin system. An observable is given by an operator $\hat{O}$ that can be seen, in the case of a system with N spin-1/2 particles, as a matrix with size $2^N \times 2^N$. In this case, the observable expectation value is given by:

$$\langle \hat{O} \rangle = \langle \mathbf{\Psi} | \hat{O} | \mathbf{\Psi} \rangle.$$

The latter is in the form of a vector-matrix-vector contraction, which is very inefficient for a large number of degrees of freedom N. In Sec. 3.7, we present a variational scheme for computing approximations of these observables with a more efficient computational cost. It is important to note that $\hat{O}$ has to be Hermitian in order to obtain real expectation values.

2.2.3 Density matrices and quantum entanglement

A quantum many-body system described by a single wave function $|\mathbf{\Psi}\rangle$ can be referred to as a quantum system in a pure state. In this case, our system is characterized by a 'density matrix' ρ with the following expression:

$$\rho = |\mathbf{\Psi}\rangle \langle \mathbf{\Psi}| .$$

If our quantum many-body system is in a statistical mixture of different quantum states $\{|\mathbf{\Psi}_i\rangle\}_{i=1}^n$ with probabilities $\{p_i\}_{i=1}^n$, then we say that our system is described by a 'mixed state' with the following density matrix:

$$\rho = \sum_{i=1}^{n} p_i |\mathbf{\Psi}_i\rangle \langle \mathbf{\Psi}_i| .$$

If we divide our system with Hibert space $\mathcal{H}$ into two portions A and B with corresponding Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$, then we can define a reduced density matrix as:

$$\rho_A = \mathrm{Tr}_A(\rho),$$

where the trace operation over $\mathcal{H}_A$ can be interpreted as the operation of preserving the B part of our system while discarding the A part of our system. This operation turns out to be very helpful in defining the notion of quantum entanglement, where we call that a quantum system in its pure state with two regions A and B are entangled if its wave function cannot be factorized into a tensor product of two wave functions living on separate Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$. Equivalently, we can say that the wave function of the quantum system cannot be written as a product state.

The characterization of quantum entanglement can be done through the definition of the α-Renyi entropies between region A and B as:

$$S_\alpha(A) = \frac{1}{1-\alpha} \log\left(\mathrm{Tr}(\rho_A^\alpha)\right), \tag{2.17}$$

where $\rho_A = \mathrm{Tr}_B \left|\mathbf{\Psi}\right\rangle \left\langle\mathbf{\Psi}\right|$ and α is an integer [68]. In particular, $S_1(A)$ corresponds to the von Neumann entropy defined as:

$$S_1(A) = -\mathrm{Tr}\left(\rho_A \log(\rho_A)\right),$$

which is the quantum counterpart of the Shannon entropy in Sec. 2.1.1. The estimation of this entanglement is challenging as it is difficult to provide an estimate of the logarithm of a matrix with an exponential size. In Sec. 3.8, we explain how to go around that by estimating the second Renyi entropy with $\alpha = 2$, which carries similar information about entanglement compared to the von Neumann entropy.

To understand why $S_2(A)$ is a good measure of entanglement, we can take the example of the GHZ state [69] for two qubits, which is defined as:

$$\left|\mathbf{\Psi}\right\rangle = \frac{1}{\sqrt{2}}(\left|00\right\rangle + \left|11\right\rangle),$$

where we can see that measuring 0 along the first qubit automatically projects the second qubit to state 0 with the same behavior occurring if we measure state 1 on the first qubit. This property is a clear manifestation of entanglement since the GHZ state cannot be written as a product state. For this state, the reduced density matrix on one qubit can be estimated as:

$$\rho = \frac{1}{2}\left|0\right\rangle\left\langle 0\right| + \frac{1}{2}\left|1\right\rangle\left\langle 1\right|,$$

and which corresponds to $S_2 = \log(2)$. Now in the case where our system is described by a product state:

$$|\mathbf{\Psi}\rangle = |00\rangle\,,$$

which has no entanglement, then the second Renyi entropy is vanishing, i.e., $S_2 = 0$.

Finally, we note that in Sec. 5.1.1, we demonstrate the numerical estimation of the second Renyi entropy on a prototypical 1D quantum many-body model, and in Chap. 7, we show how entanglement can be used as a proxy to detect topological order in a quantum many-body system.

2.2.4 Quantum phase transitions and critical exponents

In Sec. 2.1.5, we indicated that a classical many-body system can endure phase transitions upon crossing a critical temperature. Quantum many-body systems also share the same property, and can also change phases if a coupling term in the Hamiltonian is tuned [70]. A prominent real-world example is the phase transition 'Superconductor-Insulator' transition where the addition of disorder was shown to have a destructive effect on superconductivity [71, 72].

By virtue of the path-integral formulation of quantum mechanics [73, 74], there is a one-to-one mapping of a quantum system in D spatial dimensions to a classical statistical system in $D + 1$ dimensions. Note that there are examples of quantum systems where this mapping fails due to the presence of a Berry phase term [75]. We also note that this mapping shows that quantum phase transitions can have critical exponents characterized by the divergence of different quantum observables in a similar fashion to classical observables in statistical physics as shown in Sec. 2.1.6. More details about the definition of the critical exponents can be found in Sec. 2.1.6. In Sec. 5.4, we demonstrate a numerical approach for investigating phase transitions and estimating critical exponents in three spatial dimensions through the example of a prototypical quantum spin system.

2.2.5 Ground state problem

Understanding quantum materials in real-world settings often boils down to solving a ground state problem, where given a Hamiltonian $\hat{H}$ that dictates the interactions between quantum degrees of freedom in our system, we are interested in finding the lowest eigenvalue of $\hat{H}$ that corresponds to the lowest energy of the system. The corresponding eigenvector

$|\boldsymbol{\Psi_G}\rangle$ is called the ground state. To extract this state $|\boldsymbol{\Psi_G}\rangle$, one can solve the Schrödinger equation

$$\hat{H}\,|\boldsymbol{\Psi_G}\rangle = E_G\,|\boldsymbol{\Psi_G}\rangle\,, \tag{2.18}$$

which is in a form of an eigenvalue equation. To appreciate the difficulty of this problem, we can consider a physical system of N spins, where each spin can be in a superposition of ups and downs. The number of total possibilities a spin configuration can have is 2^N. Thus, we can deduce that the ground state $|\boldsymbol{\Psi_G}\rangle$ will be of size 2^N. Thus from the previous eigenvalue equation, it makes sense to think of the Hamiltonian $\hat{H}$ as a square matrix with size $2^N \times 2^N$. Now, one naive way to solve the Schrödinger equation above is by diagonalizing the Hamiltonian $\hat{H}$ to extract E_G and $|\boldsymbol{\Psi_G}\rangle$. However, one important problem with this approach is that the complexity of the best-known diagonalization algorithm is exponential as $\mathcal{O}(2^{cN})$ where c is a positive constant, where c depends on the nature of the diagonalization algorithm. Thus we would need an exponential number of resources if we were to solve the ground-state problem using this approach. To illustrate the hardness of this task, one can observe that $2^{266} \approx 10^{80}$ is at the order of the number of atoms in our known universe, which is an unpractical complexity to handle with realistic resources. Furthermore, for a system size at the order of $N \sim 50$, this problem becomes impossible to solve even with the best contemporary super-computing powers.

All in all, finding the exact ground state $|\boldsymbol{\Psi_G}\rangle$ of a generic Hamiltonian $\hat{H}$ is a tough problem in general as the Hilbert space scales exponentially with the number of degrees of freedom. A way to go around this issue is to rely on approximate methods to find $|\boldsymbol{\Psi_G}\rangle$. Thankfully, the amount of information needed to capture the relevant degrees of freedom of $|\boldsymbol{\Psi_G}\rangle$ is much smaller than the dimension of the total Hilbert space. This observation is motivated by the ability of physically-motivated ansatz wave functions to target the ground state of systems establishing exotic phenomena, such as low-temperature superconductors [76] and systems with a fractional Hall effect [77]. This idea is also put forward by the conjectured area law en entanglement in low energy sectors [78] instead of a volume law of entanglement for a random quantum state.

Chapter 3

Variational Monte Carlo

In memory of Sandro Sorella (1960 - 2022)

Our main goal in this chapter is to give a brief overview of the framework of Variational Monte Carlo (VMC) with its technical details, which will clarify the relevance of machine learning tools namely Recurrent Neural Networks (RNNs) in the quest of finding ground states as well as approximating Boltzmann probability distributions with an efficient computational cost without sacrificing too much accuracy. In Sec. 3.1, we define the variational principle and the concept of an ansatz, we then focus in Sec. 3.2 on the optimization of ansatz wave functions. In Sec. 3.3, we demonstrate the feasibility of estimating expectation values with a realistic cost up to some statistical noise. Furthermore, in Sec. 3.4, we define the zero-variance principle as a criterion for assessing the convergence of a variational calculation. Additionally, in Sec. 3.5, we demonstrate that the gradients' noise can be reduced for a normalized ansatz wave function. In Sec. 3.6, we highlight the possibility of bounding the distance from the ground state using the energy accuracy of a variational calculation. Moreover, we explain how observables and entanglements entropies can be computed from an optimized wave function in Secs. 3.7, 3.8. We also illustrate how the VMC framework can be used to find excited low-energy states in Sec. 3.9. Finally, in Sec. 3.10, we highlight the possibility of using the VMC framework for targeting Boltzmann probability distributions in the case of classical many-body systems at finite temperature.

3.1 Variational Principle

The variational principle can serve as an alternative way to approximately find a ground state (or an excited state) with a cheaper computational cost compared to exact diagonalization. These methods are called variational since they rely on a trial wave function or an ansatz with a set of parameters that one has to tune, using an optimization procedure, to find an approximation of the ground state as well as the ground state energy with a cheaper cost as motivated in Sec. 2.2.5.

One can first notice that the ground state energy is the minimal energy our quantum system of interest can have, i.e.

$$E_G = \min_{|\Psi\rangle} \frac{\langle \Psi | \hat{H} | \Psi \rangle}{\langle \Psi | \Psi \rangle}.$$

Given an ansatz wave function $|\Psi_{\boldsymbol{\lambda}}\rangle$ defined by a set of parameters $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, ...)$, one can define the variational energy $E_{\boldsymbol{\lambda}}$ as follows:

$$E_{\boldsymbol{\lambda}} \equiv \frac{\langle \Psi_{\boldsymbol{\lambda}} | \hat{H} | \Psi_{\boldsymbol{\lambda}} \rangle}{\langle \Psi_{\boldsymbol{\lambda}} | \Psi_{\boldsymbol{\lambda}} \rangle},$$

where $\hat{H}$ is the Hamiltonian of a system of interest. In this case, since a family of $|\Psi_{\boldsymbol{\lambda}}\rangle$ only occupies a subspace of all possible wave functions (see Fig. 3.1), then:

$$\min_{\boldsymbol{\lambda}} E_{\boldsymbol{\lambda}} \geq E_G.$$

The latter implies that the problem of finding an approximation of the ground state can be mapped to a minimization problem on the parameters $\boldsymbol{\lambda}$. Thus, looking for the optimal parameters $\boldsymbol{\lambda}^*$ may lead to an approximation of the ground state energy E_G, and consequently to an approximation of the true ground state $|\Psi_{\boldsymbol{G}}\rangle$, i.e., $|\Psi_{\boldsymbol{\lambda}^*}\rangle \approx |\Psi_{\boldsymbol{G}}\rangle$, providing that $|\Psi_{\boldsymbol{\lambda}}\rangle$ is a good guess. Fig. 3.1 illustrates that a family of ground states of local and relevant Hamiltonians typically occupies a tiny region in the Hilbert space of all possible states. Thus, a physically informed choice of $|\Psi_{\boldsymbol{\lambda}}\rangle$ can allow us to provide a good approximation to the ground state.

3.2 Optimization

Our ultimate goal now is to optimize the parameters $\boldsymbol{\lambda}$ to minimize the energy $E_{\boldsymbol{\lambda}}$ so that the variational state $|\Psi_{\boldsymbol{\lambda}}\rangle$ is as close as possible to the ground state $|\Psi_{\boldsymbol{G}}\rangle$. The first

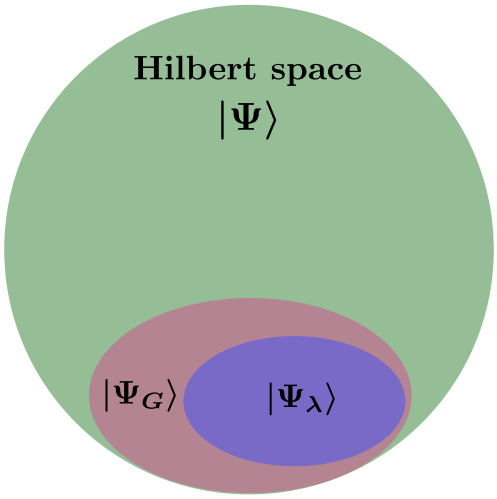

Figure 3.1: Illustration of the Hilbert space with a tiny region occupied by an ensemble of ground states $|\boldsymbol{\Psi_G}\rangle$ for local Hamiltonians of interest. A good choice of the variational wave function $|\boldsymbol{\Psi_\lambda}\rangle$ corresponds to an ensemble of states that can reach most of the relevant space we are interested in.

thing one can think about to find the minimum of $E_{\boldsymbol{\lambda}}$ is to take the partial derivative and set them to zero, i.e.

$$\partial_{\boldsymbol{\lambda}} E_{\boldsymbol{\lambda}} = \mathbf{0}.$$

Solving this equation can be possible in some cases. This approach has led to two Nobel prizes for the understanding of low-temperature superconductivity (BCS theory) [76] and of the fractional Hall effect (Laughlin state) [77]. In other cases, an exact theoretical derivation might not be possible, so we can use other numerical techniques to minimize the variational energy, namely gradient descent.

Gradient descent consists of changing the parameters $\boldsymbol{\lambda}$ in the direction of opposite gradients of $E_{\boldsymbol{\lambda}}$, i.e.

$$\boldsymbol{\lambda} \to \boldsymbol{\lambda'} = \boldsymbol{\lambda} - \eta \frac{\partial E_{\boldsymbol{\lambda}}}{\partial \boldsymbol{\lambda}} \tag{3.1}$$

for η (referred to as the learning rate) positive and small enough so that the new variational energy:

$$E_{\boldsymbol{\lambda'}} = E_{\boldsymbol{\lambda}} + \sum_i \delta\lambda_i \frac{\partial E_{\boldsymbol{\lambda}}}{\partial \lambda_i} + \mathcal{O}((\delta\boldsymbol{\lambda})^2) = E_\lambda - \eta \sum_i \left(\frac{\partial E_{\boldsymbol{\lambda}}}{\partial \lambda_i} \right)^2 + \mathcal{O}(\eta^2) \tag{3.2}$$

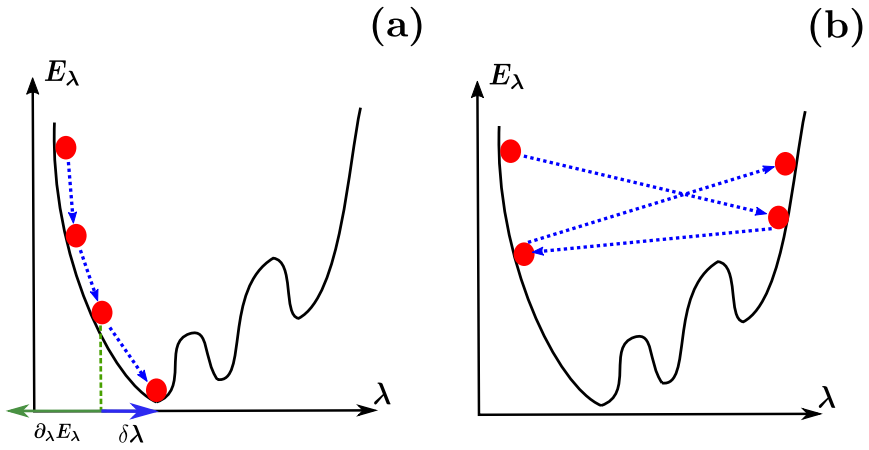

Figure 3.2: (a) Illustration of the gradient descent algorithm applied on the variational energy $E_{\boldsymbol{\lambda}}$ using a reasonably small learning rate. (b) Using large learning rates can lead to unstable training.

is smaller than $E_{\boldsymbol{\lambda}}$. Fig. 3.2(a) illustrates how the gradient descent algorithm and how it can be iterated to reach the minimum of the variational energy. If a large learning rate η is used, the training can be very unstable, as a large magnitude of η violates the first-order Taylor approximation in Eq. (3.2).

In some cases, using plain gradient descent can lead to local minima as illustrated by the variational energy landscape of Fig. 3.2. This problem can be partially mitigated by introducing momentum to vanilla gradient descent [79]. The latter can be done through the following change of parameters at step i:

$$\delta\boldsymbol{\lambda}^{(i)} = -\eta \boldsymbol{v}(i),$$

where

$$\boldsymbol{v}(i) = (1-\beta)\boldsymbol{v}(i-1) + \beta \partial_{\boldsymbol{\lambda}^{(i)}} E_{\boldsymbol{\lambda}^{(i)}},$$

and $\boldsymbol{v}(0) = \mathbf{0}$. Here β is another hyperparameter called momentum. In this case, we can see that:

$$\boldsymbol{v}(i) = (1-\beta)\sum_{j=0}^{i-1} \beta^{i-j} \boldsymbol{v}(j).$$

The last expression shows that by introducing β, we are keeping a memory of the previous gradients. In particular, if the current gradient is zero, the previous gradients are not likely to be vanishing. In this spirit, the use of momentum can circumvent some shallow local

minima in the optimization landscape. The use of momentum also enables a smoother and faster optimization as opposed to plain gradient descent [79]. Nowadays, more sophisticated versions of gradient descent optimizers with momentum have been devised. The most famous and successful one is the so-called Adam optimizer, which we use actively with RNNs in this study [80].

The previous class of gradient descent algorithms is referred to as first-order optimization algorithms. There exists another class of gradient descent algorithms that are second-order and that exploits the curvature of the optimization landscape, namely natural gradient or equivalently stochastic reconfiguration. It has been used both in the context of machine learning [81], as well as in the optimization of variational wave functions [2, 37, 82]. In this study, we do not use those classes of algorithms mainly due to their expensive computational cost. Thankfully, there is more space for exploration in the direction of harnessing the power of approximate natural gradients methods, such as K-FAC [83, 84] and conjugate gradient methods [85, 86] to improve the trainability of RNNs. An example of a prior application of variational wave functions to quantum systems using these second-order methods can be found in Ref. [40].

3.3 Importance Sampling

Variational Monte Carlo (VMC) takes advantage of sampling important configurations to substantially reduce the computation time from an exponential scaling to a more budget-friendly computational cost using the so-called importance sampling[2].

To explain this concept, let us say that our degrees of freedom correspond to spins such as a spin configuration can be represented as $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_N)$ where each $\sigma_i = 0, 1$. Then, by writing the variational wave function in the computational basis $|\boldsymbol{\sigma}\rangle$ as:

$$|\boldsymbol{\Psi_\lambda}\rangle = \sum_{\boldsymbol{\sigma}} \Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}) \, |\boldsymbol{\sigma}\rangle \, , \tag{3.3}$$

and assuming that the variational wave function is normalized (i.e. $\langle \Psi_{\boldsymbol{\lambda}} | \Psi_{\boldsymbol{\lambda}} \rangle = 1$), the variational energy can be expanded as follows:

$$E_{\boldsymbol{\lambda}} = \sum_{\boldsymbol{\sigma'\sigma}} \Psi^*_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}) H_{\boldsymbol{\sigma\sigma'}} \Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma'}). \tag{3.4}$$

We can see that the previous sum runs over an exponential number of configurations $\boldsymbol{\sigma}$, $\boldsymbol{\sigma'}$. One idea contributing toward circumventing this limitation is to multiply and divide

with $\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})$ such that:

$$E_{\boldsymbol{\lambda}} = \sum_{\boldsymbol{\sigma}} |\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})|^2 \sum_{\boldsymbol{\sigma'}} H_{\boldsymbol{\sigma}\boldsymbol{\sigma'}} \frac{\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma'})}{\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})}. \tag{3.5}$$

The latter can be rewritten as:

$$E_{\boldsymbol{\lambda}} = \sum_{\boldsymbol{\sigma}} |\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})|^2 E_{\mathrm{loc}}(\boldsymbol{\sigma}), \tag{3.6}$$

$$= \langle E_{\mathrm{loc}}(\boldsymbol{\sigma}) \rangle \,. \tag{3.7}$$

where

$$E_{\mathrm{loc}}(\boldsymbol{\sigma}) \equiv \sum_{\boldsymbol{\sigma'}} H_{\boldsymbol{\sigma}\boldsymbol{\sigma'}} \frac{\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma'})}{\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})} \tag{3.8}$$

is known as 'local energy'. This quantity can be computed efficiently for a local Hamiltonian $\hat{H}$, since there is only $\mathcal{O}(N)$ non-zero matrix elements $H_{\boldsymbol{\sigma}\boldsymbol{\sigma'}}$ for a fixed $\boldsymbol{\sigma}$. The notation $\langle . \rangle$ stands for an expectation value over the probability distribution $|\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})|^2$. From the previous equation (3.6), we can see that we ended up with another sum over all possible configurations, which is still intractable. Luckily, it is a sum over a probability times another term which can be approximated through the so-called importance sampling. This step can be accomplished by stochastically sampling configurations according to the probability weights $|\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})|^2$.

Several algorithms have been devised to do importance sampling. These algorithms are discussed in detail in Chap. 4. Here let us assume that such a procedure that can generate M important and independent samples $\{\boldsymbol{\sigma}^{(i)}\}_{i=1}^{M}$ exists. In this case, we can approximate the variational energy as follows:

$$E_{\boldsymbol{\lambda}} \approx \frac{1}{M} \sum_{i=1}^{M} E_{\mathrm{loc}}(\boldsymbol{\sigma}^{(i)}), \tag{3.9}$$

where the error bars on the energy estimator are given by

$$\epsilon_o = \sqrt{\frac{\mathrm{Var}(E_{\mathrm{loc}}(\boldsymbol{\sigma}^{(i)}))}{M}}.$$

Here, it is clear that the approximation becomes exact in the limit $M \to \infty$.

To quantify the cost of computing the energy, let us assume that the cost of computing an amplitude $\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})$ is $f(N)$ and that there $\mathcal{O}(N)$ of non-diagonal matrix elements in our Hamiltonian $\hat{H}$, then the cost of estimating the energy is $\mathcal{O}(MNf(N))$.

Now to optimize the variational energy using gradient descent, we can derive the expression of the gradients as follows:

$$\begin{aligned}
\partial_{\boldsymbol{\lambda}} E_{\boldsymbol{\lambda}} &= \sum_{\boldsymbol{\sigma}'\boldsymbol{\sigma}} \partial_{\boldsymbol{\lambda}} \Psi^*_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}) H_{\boldsymbol{\sigma}\boldsymbol{\sigma}'} \Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}') + \sum_{\boldsymbol{\sigma}'\boldsymbol{\sigma}} \Psi^*_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}) H_{\boldsymbol{\sigma}\boldsymbol{\sigma}'} \partial_{\boldsymbol{\lambda}} \Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}'), \\
&= 2\mathfrak{Re}\left(\sum_{\boldsymbol{\sigma}'\boldsymbol{\sigma}} \partial_{\boldsymbol{\lambda}} \Psi^*_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}) H_{\boldsymbol{\sigma}\boldsymbol{\sigma}'} \Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}') \right), \\
&= 2\mathfrak{Re}\left(\sum_{\boldsymbol{\sigma}} \partial_{\boldsymbol{\lambda}} \Psi^*_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}) \sum_{\boldsymbol{\sigma}'} H_{\boldsymbol{\sigma}\boldsymbol{\sigma}'} \Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}') \right), \\
&= 2\mathfrak{Re}\left(\sum_{\boldsymbol{\sigma}} |\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})|^2 \frac{\partial_{\boldsymbol{\lambda}} \Psi^*_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})}{\Psi^*_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})} \sum_{\boldsymbol{\sigma}'} H_{\boldsymbol{\sigma}\boldsymbol{\sigma}'} \frac{\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}')}{\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})} \right), \\
&= 2\mathfrak{Re}\left(\sum_{\boldsymbol{\sigma}} |\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})|^2 \frac{\partial_{\boldsymbol{\lambda}} \Psi^*_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})}{\Psi^*_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})} E_{\mathrm{loc}}(\boldsymbol{\sigma}) \right), \\
&= 2\mathfrak{Re}\left(\sum_{\boldsymbol{\sigma}} |\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})|^2 \partial_{\boldsymbol{\lambda}} \log\left(\Psi^*_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})\right) E_{\mathrm{loc}}(\boldsymbol{\sigma}) \right).
\end{aligned} \tag{3.10}$$

Here $\mathfrak{Re}(z)$ stands for the real part of a complex number z. Similarly to the variational energy, we use a finite number M of important samples to estimate the gradients as:

$$\partial_{\boldsymbol{\lambda}} E_{\boldsymbol{\lambda}} \approx 2\mathfrak{Re}\left(\frac{1}{M} \sum_{i=1}^{M} \partial_{\boldsymbol{\lambda}} \log\left(\Psi^*_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}^{(i)})\right) E_{\mathrm{loc}}(\boldsymbol{\sigma}^{(i)}) \right). \tag{3.11}$$

A standard trick one can use is to define a fake cost function

$$E_{\mathrm{fake}} = \frac{1}{M} \left(\sum_{i=1}^{M} \log\left(\Psi^*_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}^{(i)})\right) E^{\perp}_{\mathrm{loc}}(\boldsymbol{\sigma}^{(i)}) \right),$$

where $E^{\perp}_{\mathrm{loc}}(\boldsymbol{\sigma}^{(i)})$ is considered as a constant when a gradient operation is applied. Thus

$$\partial_{\boldsymbol{\lambda}} E_{\boldsymbol{\lambda}} \approx \partial_{\boldsymbol{\lambda}} E_{\mathrm{fake}}.$$

This idea allows avoiding the computation of the individual gradients of $\log\left(\Psi^*_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}^{(i)})\right)$ and to parallelize the computation through the use of the fake cost function and automatic differentiation [87].

In practice, $M = 100$ samples to 1000 samples is usually enough to achieve convergence during the training if we use a first-order optimization scheme such as Adam with

RNNs [80]. For second-order optimization methods such as stochastic reconfiguration, the number of samples M needs to be much larger compared to the number of parameters of the ansatz to guarantee a stable convergence [2]. The latter can be a serious limitation with current hardware if the ansatz reaches tens of thousands to millions of parameters. Recent work has addressed this limitation for stochastic reconfiguration opening the door for scalability of training ansatz wave functions with a large number of parameters using stochastic reconfiguration [88].

3.4 Zero-variance principle

So far, we have discussed the optimization of a variational ansatz as a strategy to obtain an approximation of the ground state. However, we have not yet covered how to determine whether a variational calculation has converged. An important concept is the zero-variance principle, which can serve as a heuristic to estimate how close we are to the ground state. To understand how this principle works, let us recall the eigenvalue equation of the ground state in Eq. (2.18). Here we can apply $\langle \boldsymbol{\sigma}|$ for a fixed configuration $\boldsymbol{\sigma}$ to obtain the following:

$$\sum_{\boldsymbol{\sigma'}} H_{\boldsymbol{\sigma}\boldsymbol{\sigma'}} \Psi_G(\boldsymbol{\sigma'}) = E_G \Psi_G(\boldsymbol{\sigma}).$$

Thus for a non-zero amplitude $\Psi_G(\boldsymbol{\sigma})$, we can see that:

$$\sum_{\boldsymbol{\sigma'}} H_{\boldsymbol{\sigma}\boldsymbol{\sigma'}} \frac{\Psi_G(\boldsymbol{\sigma'})}{\Psi_G(\boldsymbol{\sigma})} = E_G.$$

In this case, it is clear that if our variational wave function $|\boldsymbol{\Psi_\lambda}\rangle = |\boldsymbol{\Psi_G}\rangle$, then:

$$E_{\text{loc}}(\boldsymbol{\sigma}) = E_G.$$

for all spin configurations $\boldsymbol{\sigma}$ that have a non-zero amplitude. As a consequence, close to convergence, we expect that the local energies will be close to each other, and thus their variance will be closer to zero. A measure of this quantity is the energy variance per spin defined as:

$$\sigma^2 \equiv \frac{\langle \boldsymbol{\Psi_\lambda}|\hat{H}^2|\boldsymbol{\Psi_\lambda}\rangle - \langle \boldsymbol{\Psi_\lambda}|\hat{H}|\boldsymbol{\Psi_\lambda}\rangle^2}{N}, \tag{3.12}$$

where N is the system size. We can show that this quantity is related to the variance of the local energies, i.e.

$$\sigma^2 = \frac{\text{Var}(E_{\text{loc}}(\boldsymbol{\sigma}))}{N},$$

where the variance is taken over the probability distribution $|\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})|^2$. Since the energy variance is always positive, then the closer we are to convergence the closer our energy variance to zero. This makes the variance an efficient measure to monitor the convergence of our variational energy toward the ground state energy, during the VMC training process. The variance measure is also helpful in assessing the quality of variational approximation of the ground state [2, 89, 90], as previously done in the case of matrix product state based techniques [91, 92]. A variant of the energy variance has been also used to quantify the difficulty of finding the ground state of quantum many-body systems as described in Ref. [93]. The energy variance can allow obtaining a more accurate estimate of the ground state through extrapolation if we have a way to systematically improve the variational energy as described in Refs. [94, 95].

One important caveat to note is that the variance of the local energies could be also very small when our variational wave function is close to an excited state. To heuristically go around this limitation, one can take advantage of symmetries and group characters as described in Chap. 4. One could also use annealing techniques to overcome local minima by introducing thermal fluctuations as described in Chap. 6.

3.5 Noise reduction of the energy gradients

The estimation of the energy gradients in Eq. (3.11) is noisy due to the use of a finite number of samples M. This noise can potentially be an obstacle to achieving a smooth convergence to an approximate ground state. To reduce the noise in our estimation of the gradients, we use the following unbiased estimator of the gradient

$$\partial_{\boldsymbol{\lambda}} E_{\boldsymbol{\lambda}} \approx 2\mathfrak{Re}\left(\frac{1}{M}\sum_{i=1}^{M}\partial_{\boldsymbol{\lambda}}\log\left(\Psi^*_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}^{(\boldsymbol{i})})\right)\left(E_{\mathrm{loc}}(\boldsymbol{\sigma}^{(\boldsymbol{i})})-E_{\boldsymbol{\lambda}}\right)\right), \tag{3.13}$$

$$= 2\mathfrak{Re}\left(\frac{1}{M}\sum_{i=1}^{M}\partial_{\boldsymbol{\lambda}}\log\left(\Psi^*_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}^{(\boldsymbol{i})})\right)\overline{E}_{\mathrm{loc}}(\boldsymbol{\sigma}^{(\boldsymbol{i})})\right), \tag{3.14}$$

where $\overline{E}_{\mathrm{loc}}(\boldsymbol{\sigma}^{(\boldsymbol{i})}) \equiv E_{\mathrm{loc}}(\boldsymbol{\sigma}^{(\boldsymbol{i})}) - E_{\boldsymbol{\lambda}} = E_{\mathrm{loc}}(\boldsymbol{\sigma}^{(\boldsymbol{i})}) - \langle E_{\mathrm{loc}}(\boldsymbol{\sigma}^{(\boldsymbol{i})})\rangle$. Here the additional term compared to Eq. (3.11) allows for reducing the variance of the gradients close to convergence, i.e., when $E_{\mathrm{loc}}(\boldsymbol{\sigma}^{(\boldsymbol{i})}) \approx E_{\boldsymbol{\lambda}}$. This idea is very similar in essence to control variate methods in Monte Carlo [96] and to baseline methods in Reinforcement learning [97].

To show that the additional baseline term does not bias the true gradients in Eq. (3.10).

It is sufficient to show that:

$$\mathfrak{Re}\left(\langle \partial_{\boldsymbol{\lambda}} \log\left(\Psi^*_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})\right)\rangle E_{\boldsymbol{\lambda}}\right) = 0. \tag{3.15}$$

Let us first write $\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}) = \sqrt{P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})}\exp(\mathrm{i}\phi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}))$, which leads to the following expression:

$$\log\left(\Psi^*_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})\right) = \frac{1}{2}\log\left(P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})\right) - \mathrm{i}\phi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}).$$

Thus, we have to prove the following:

$$\frac{1}{2}\langle \partial_{\boldsymbol{\lambda}} \log\left(P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})\right)\rangle \mathfrak{Re}(E_{\boldsymbol{\lambda}}) + \langle \partial_{\boldsymbol{\lambda}}\phi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})\rangle \mathfrak{Im}(E_{\boldsymbol{\lambda}}) = \mathbf{0}. \tag{3.16}$$

We can remark that [97, 98]:

$$\begin{aligned}
\langle \partial_{\boldsymbol{\lambda}} \log\left(P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})\right)\rangle &= \sum_{\boldsymbol{\sigma}} P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})\ \partial_{\boldsymbol{\lambda}} \log\left(P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})\right), \\
&= \sum_{\boldsymbol{\sigma}} P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}) \frac{\partial_{\boldsymbol{\lambda}} P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})}{P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})}, \\
&= \partial_{\boldsymbol{\lambda}} \sum_{\boldsymbol{\sigma}} P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}), \\
&= \partial_{\boldsymbol{\lambda}} 1 = \mathbf{0}.
\end{aligned}$$

Hence it is sufficient to show that $\langle \partial_{\boldsymbol{\lambda}}\phi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})\rangle \mathfrak{Im}(E_{\boldsymbol{\lambda}}) = \mathbf{0}$. In fact, since the Hamiltonian $\hat{H}$ is Hermitian, the variational energy $E_{\boldsymbol{\lambda}}$ is real. The latter finishes our proof of Eq. (3.15).

To demonstrate the noise reduction claim more rigorously, let us focus on the variance of the gradient for a parameter λ in the set of the variational parameters $\boldsymbol{\lambda}$, after subtracting the baseline. Here we focus on the case of a positive ansatz wave function $\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}) = \sqrt{P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})}$ that we used in our study. First of all, we define:

$$O_\lambda(\boldsymbol{\sigma}) \equiv \partial_\lambda \log\left(\Psi^*_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})\right) = \frac{1}{2}\partial_\lambda \log(P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})).$$

Thus, the gradient with a baseline can be written as:

$$\partial_\lambda E_{\boldsymbol{\lambda}} = 2\left\langle O_\lambda(\boldsymbol{\sigma})\overline{E}_{\mathrm{loc}}(\boldsymbol{\sigma})\right\rangle,$$

where $\overline{E}_{\mathrm{loc}}(\boldsymbol{\sigma}) \equiv E_{\mathrm{loc}}(\boldsymbol{\sigma}) - E_{\boldsymbol{\lambda}}$ and $\langle . \rangle$ denotes an expectation value over the Born distribution $|\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})|^2$. To estimate the gradients' noise, we look at the variance of the gradient estimator, which can be decomposed as follows:

$$\begin{aligned}
\mathrm{Var}(O_\lambda \overline{E}_{\mathrm{loc}}) = &\ \mathrm{Var}(O_\lambda E_{\mathrm{loc}}) \\
&- 2\mathrm{Cov}(O_\lambda E_{\mathrm{loc}}, O_\lambda E_{\boldsymbol{\lambda}}) + E^2_{\boldsymbol{\lambda}}\mathrm{Var}(O_\lambda).
\end{aligned}$$

Thus the variance reduction R, after subtracting the baseline, is given as:

$$\begin{aligned} R &\equiv \mathrm{Var}(O_\lambda \overline{E}_{\mathrm{loc}}) - \mathrm{Var}(O_\lambda E_{\mathrm{loc}}), \\ &= -2E_{\boldsymbol{\lambda}}\mathrm{Cov}(O_\lambda E_{\mathrm{loc}}, O_\lambda) + E_{\boldsymbol{\lambda}}^2 \mathrm{Var}(O_\lambda). \end{aligned}$$

Since the gradients' magnitude tends to near-zero values close to convergence, statistical errors are more likely to make the VMC optimization more challenging. We focus on this regime for this derivation to show the importance of the baseline in reducing noise. Thus, we assume that $E_{\mathrm{loc}}(\boldsymbol{\sigma}) = E_{\boldsymbol{\lambda}} + \xi(\boldsymbol{\sigma})$, where the supremum of the local energies fluctuations is much smaller compared to the variational energy, i.e., $(\sup_{\boldsymbol{\sigma}} |\xi(\boldsymbol{\sigma})|) \ll E_{\boldsymbol{\lambda}}$. From this assumption, we can deduce that:

$$\begin{aligned} R &= -2E_{\boldsymbol{\lambda}}^2 \mathrm{Cov}(O_\lambda, O_\lambda) && (3.17) \\ &\quad - 2E_{\boldsymbol{\lambda}}\mathrm{Cov}(O_\lambda \xi, O_\lambda) + E_{\boldsymbol{\lambda}}^2 \mathrm{Var}(O_\lambda), && (3.18) \\ &= -E_{\boldsymbol{\lambda}}^2 \mathrm{Var}(O_\lambda) - 2E_{\boldsymbol{\lambda}}\mathrm{Cov}(O_\lambda \xi, O_\lambda). && (3.19) \end{aligned}$$

The second term can be decomposed as follows:

$$\mathrm{Cov}(O_\lambda \xi, O_\lambda)) = \langle O_\lambda^2 \xi \rangle - \langle O_\lambda \xi \rangle \langle O_\lambda \rangle. \tag{3.20}$$

Since $\langle O_\lambda \rangle = \frac{1}{2}\langle \partial_\lambda \log(P_{\boldsymbol{\lambda}}) \rangle = 0$ [97, 98], then we can bound the covariance term from above as:

$$\begin{aligned} \mathrm{Cov}(O_\lambda \xi, O_\lambda) &\leq \left(\sup_{\boldsymbol{\sigma}} |\xi(\boldsymbol{\sigma})| \right) \langle O_\lambda^2 \rangle \\ &= \left(\sup_{\boldsymbol{\sigma}} |\xi(\boldsymbol{\sigma})| \right) \mathrm{Var}(O_\lambda), \\ &\ll E_{\boldsymbol{\lambda}} \mathrm{Var}(O_\lambda). \end{aligned}$$

Thus, we can conclude that the variance reduction R in Eq. (3.19) is negative. This observation highlights the importance of the baseline in reducing the statistical noise of the energy gradients near convergence.

For a complex ansatz wave function $\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}) = \sqrt{P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})} \exp(\mathrm{i}\phi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}))$, the expectation value $\langle O_\lambda \rangle = \frac{1}{2}\langle \partial_\lambda \log(P_{\boldsymbol{\lambda}}) \rangle - \mathrm{i}\langle \partial_\lambda \phi_{\boldsymbol{\lambda}} \rangle = -\mathrm{i}\langle \partial_\lambda \phi_{\boldsymbol{\lambda}} \rangle$ is no longer equal to zero in general. This term is related to the phase variations, which are susceptible to contributing to the variance of the gradients. This observation is also an interesting indication for why optimizing wave functions with a sign is more challenging compared to positive wave functions. We leave the investigation of this point for future studies.

3.6 Distance from the ground state

Getting lower variational energies is always desirable, but knowing how much accuracy is needed to achieve a certain variational calculation is crucial to know when to stop the training and how powerful our ansatz should be. In this section, we discuss how we can bound the distance between the ground state $|\boldsymbol{\Psi_G}\rangle$ and the variational wave function $|\boldsymbol{\Psi_\lambda}\rangle$ in terms of the variational energy accuracy. To do so, let us write our variational wave function as [2]:

$$|\boldsymbol{\Psi_\lambda}\rangle = (1-\epsilon)^{\frac{1}{2}} |\boldsymbol{\Psi_G}\rangle + \epsilon^{\frac{1}{2}} |\boldsymbol{\Psi}_\perp\rangle , \tag{3.21}$$

where $|\boldsymbol{\Psi}_\perp\rangle$ corresponds to a superposition of excited states that are orthogonal to the ground state. $1-\epsilon$ is the overlap between the variational state and the ground state or equivalently the fidelity measure between these two states. Here we assume that the ground state is not degenerate for simplicity but our analysis is similar in the degenerate case. Now, we can show that the variational energy satisfies the following:

$$E_{\boldsymbol{\lambda}} = \langle \boldsymbol{\Psi_\lambda}| \hat{H} |\boldsymbol{\Psi_\lambda}\rangle = (1-\epsilon)E_G + \epsilon \langle \boldsymbol{\Psi}_\perp| \hat{H} |\boldsymbol{\Psi}_\perp\rangle , \tag{3.22}$$

using the orthogonality property between the ground state $|\boldsymbol{\Psi_G}\rangle$ and the perpendicular state $|\boldsymbol{\Psi}_\perp\rangle$ and the observation $\hat{H}|\boldsymbol{\Psi_G}\rangle = E_G |\boldsymbol{\Psi_G}\rangle$. Assuming that the system we are interested in has a gap g between the ground state and the first excited state, then we can show:

$$\langle \boldsymbol{\Psi}_\perp| \hat{H} |\boldsymbol{\Psi}_\perp\rangle \geq E_G + g.$$

Plugging this inequality in Eq. (3.22), allows us to prove that:

$$E_{\boldsymbol{\lambda}} \geq E_G + \epsilon g,$$

or equivalently:

$$\epsilon \leq \frac{E_{\boldsymbol{\lambda}} - E_G}{g}. \tag{3.23}$$

This bound provides us with valuable information about how much accuracy is needed so that the discrepancy ϵ between the variational state and the ground state is as small as possible. More specifically, we need to have a residual energy $E_{\boldsymbol{\lambda}} - E_G$ to be much smaller compared to the system gap g. This requirement can be difficult to fulfill for gapless systems in the thermodynamic limit as the gap typically decreases with the system size. In the worst case, the gap decay can be exponential in the system size. Fortunately, the use of symmetries is useful to construct a variational state that is initially orthogonal to some of the lowest-energy excited states.

It is important to note that it is intractable to compute the gap in general, but the hope is that knowing the scaling of the gap with the system size can provide valuable information on how to scale our numerical resources.

As pointed out in Sec. 3.4, the variance of the local energies is a good heuristic to monitor the variations of ϵ. In fact, one can show that [90]:

$$\begin{aligned} E_{\text{loc}} - E_{\boldsymbol{\lambda}} &= \left(\frac{\langle \boldsymbol{\sigma} | \hat{H} | \boldsymbol{\Psi}_{\boldsymbol{\lambda}} \rangle}{\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})} - E_G \right) - (E_{\boldsymbol{\lambda}} - E_G), \\ &= \frac{\langle \boldsymbol{\sigma} | \hat{H} - E_G | \boldsymbol{\Psi}_{\boldsymbol{\lambda}} - \boldsymbol{\Psi}_{\boldsymbol{G}} \rangle}{\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})} - (E_{\boldsymbol{\lambda}} - E_G). \end{aligned}$$

From Eq. (3.22), we can show that $E_{\boldsymbol{\lambda}} - E_G = \mathcal{O}(\epsilon)$. From Eq. (3.21), we also have $\boldsymbol{\Psi}_{\boldsymbol{\lambda}} - \boldsymbol{\Psi}_{\boldsymbol{G}} = \mathcal{O}(\epsilon^{1/2})$. Thus, we have

$$E_{\text{loc}} - \langle E_{\text{loc}} \rangle = E_{\text{loc}} - E_{\boldsymbol{\lambda}} = \mathcal{O}(\epsilon^{1/2}).$$

As a consequence, the variance of the local energies is given as:

$$\text{Var}(E_{\text{loc}}) = \mathcal{O}(\epsilon).$$

The latter justifies the use of the energy variance for tracking how close our variational wave function is from the ground state.

3.7 Observable estimators

Estimating observables is an important step to extract more information about the ground state once we have obtained an approximation with our wave functions ansatz. Examples of such observables include magnetization and two-point correlations which are commonly used to assess the nature of the phase of the quantum system of our interest.

The expectation value of an observable in our optimized ansatz wave function can be computed as follows:

$$\begin{aligned} \left\langle \hat{O} \right\rangle_{\boldsymbol{\lambda}} &= \sum_{\boldsymbol{\sigma}'\boldsymbol{\sigma}} \Psi^*_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}) O_{\boldsymbol{\sigma}\boldsymbol{\sigma}'} \Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}'), \\ &= \sum_{\boldsymbol{\sigma}} |\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})|^2 \sum_{\boldsymbol{\sigma}'} O_{\boldsymbol{\sigma}\boldsymbol{\sigma}'} \frac{\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}')}{\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})}, \\ &= \left\langle \hat{O}_{\text{loc}}(\boldsymbol{\sigma}) \right\rangle, \end{aligned}$$

where $O_{\text{loc}}(\boldsymbol{\sigma}) = \sum_{\boldsymbol{\sigma'}} O_{\boldsymbol{\sigma}\boldsymbol{\sigma'}} \frac{\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma'})}{\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})}$. In this case, the estimation of the observable expectation value is simple for a diagonal observable in the computational basis $|\boldsymbol{\sigma}\rangle$. In this case, we have:

$$\left\langle \hat{O} \right\rangle_{\boldsymbol{\lambda}} \approx \frac{1}{M_o} \sum_{i=1}^{M_o} O_{\boldsymbol{\sigma}^{(i)}\boldsymbol{\sigma}^{(i)}}.$$

For non-diagonal observables, we need to compute the amplitudes of the off-diagonal elements. Thus, they are computationally more expensive and can be harder to estimate compared to diagonal operators. Note that the number of samples M_o can be much larger compared to the number of samples M used for training our ansatz wave function. Increasing M_o allows to reduce the statistical error on $\left\langle \hat{O} \right\rangle$ and which can be estimated as follows:

$$\epsilon(\langle \hat{O} \rangle) = \sqrt{\frac{\text{Var}(O_{\text{loc}}(\boldsymbol{\sigma}))}{M_o}}.$$

With respect to the accuracy of our estimate compared to the ground state estimate, we can recall the decomposition in Eq.(3.21) and compute the expectation value as

$$\langle \hat{O} \rangle_{\boldsymbol{\lambda}} = (1-\epsilon) \langle \boldsymbol{\Psi_G} | \hat{O} | \boldsymbol{\Psi_G} \rangle + \epsilon \langle \boldsymbol{\Psi}_\perp | \hat{O} | \boldsymbol{\Psi}_\perp \rangle + 2\sqrt{\epsilon(1-\epsilon)} \langle \boldsymbol{\Psi_G} | \hat{O} | \boldsymbol{\Psi}_\perp \rangle .$$

Thus,

$$|\langle \hat{O} \rangle_{\boldsymbol{\lambda}} - \langle \hat{O} \rangle_G| = |\epsilon(\langle \hat{O} \rangle_G + \langle \hat{O} \rangle_\perp) + 2\sqrt{\epsilon(1-\epsilon)} \langle \boldsymbol{\Psi_G} | \hat{O} | \boldsymbol{\Psi}_\perp \rangle |.$$

If the observable $\hat{O}$ commutes with the Hamiltonian $\hat{H}$, or equivalent if $\hat{O}$ corresponds to a conserved quantity that came out of a symmetry of $\hat{H}$, then $\langle \boldsymbol{\Psi_G} | \hat{O} | \boldsymbol{\Psi}_\perp \rangle = 0$, since $\langle \boldsymbol{\Psi_G}|$ would be also an eigenvalue for $\hat{O}$. In this case:

$$|\langle \hat{O} \rangle_{\boldsymbol{\lambda}} - \langle \hat{O} \rangle_G| = \mathcal{O}(\epsilon),$$

which is very desirable in practice, since targeting a good accuracy on the variational energy can provide us with the same level of accuracy for our observable. On the other hand if $[\hat{O}, \hat{H}] \neq 0$, then

$$|\langle \hat{O} \rangle_{\boldsymbol{\lambda}} - \langle \hat{O} \rangle_G| = \mathcal{O}(\epsilon^{\frac{1}{2}}).$$

The latter means that we have only a square root accuracy on observables that do not commute with $\hat{H}$, such as correlation functions in the typical case. In some cases, this accuracy is enough to resolve the physics we are interested in. If more accuracy is needed, one might resort to constructing improved observable estimators using Hellman-Feynman theorem [90] or through variance reduction techniques [96].

3.8 Entanglement entropy estimator

In some applications, it could be useful to go beyond computing physical observables to compute non-local measures that can provide us with information about the entanglement in our system of interest and can also provide us with valuable insights about topological properties [99, 100]. In this section, we will talk about how we can heuristically estimate entanglement in our system after the end of a variational calculation.

Given a quantum system with a spatial bipartition (A, B), one can write the variational wave function $|\boldsymbol{\Psi}_{\boldsymbol{\lambda}}\rangle$ as

$$|\boldsymbol{\Psi}_{\boldsymbol{\lambda}}\rangle = \sum_{\boldsymbol{\sigma}_A, \boldsymbol{\sigma}_B} \Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}_A \boldsymbol{\sigma}_B)\ |\boldsymbol{\sigma}_A \boldsymbol{\sigma}_B\rangle\,,$$

where $\boldsymbol{\sigma}_{A/B}$ denotes the spin configuration that lives in the partition A/B and $\boldsymbol{\sigma}_A\boldsymbol{\sigma}_B$ stands for a concatenation of $\boldsymbol{\sigma}_A$ and $\boldsymbol{\sigma}_B$.

The estimation of entanglement between A and B can be done through the calculation of the Renyi entropies defined in Sec. 2.2.3. In this thesis, we focus on the second Renyi entropy with $\alpha = 2$ (see Sec. 2.2.3). Here we use the so-called replica trick [68], where we consider the action of the Swap_A operator on the two copies of our ansatz wave function, which swaps the spins in the region A between the two copies (as demonstrated in Fig. 3.3) such that

$$\begin{aligned} &\mathrm{Swap}_A\, |\boldsymbol{\Psi}_{\boldsymbol{\lambda}}\rangle \otimes |\boldsymbol{\Psi}_{\boldsymbol{\lambda}}\rangle \\ &= \sum_{\boldsymbol{\sigma}, \tilde{\boldsymbol{\sigma}}} \Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}_A \boldsymbol{\sigma}_B) \Psi_{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\sigma}}_A \tilde{\boldsymbol{\sigma}}_B)\, |\tilde{\boldsymbol{\sigma}}_A \boldsymbol{\sigma}_B\rangle \otimes |\boldsymbol{\sigma}_A \tilde{\boldsymbol{\sigma}}_B\rangle\,. \end{aligned} \tag{3.24}$$

The expectation value of Swap_A in the double copy of our ansatz "$|\boldsymbol{\Psi}_{\boldsymbol{\lambda}}\rangle \otimes |\boldsymbol{\Psi}_{\boldsymbol{\lambda}}\rangle$" is given by [32, 68]

$$\begin{aligned} \langle \mathrm{Swap}_A \rangle &= \sum_{\boldsymbol{\sigma}, \tilde{\boldsymbol{\sigma}}} \Psi^*_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}_A \boldsymbol{\sigma}_B) \Psi^*_{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\sigma}}_A \tilde{\boldsymbol{\sigma}}_B) \Psi_{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\sigma}}_A \boldsymbol{\sigma}_B) \Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}_A \tilde{\boldsymbol{\sigma}}_B) \\ &= \mathrm{Tr}\rho_A^2 = \exp(-S_2(A)). \end{aligned} \tag{3.25}$$

Hence, by calculating the expectation of the value of the Swap operator in the double copy of the variational wave function, we can access the second Rényi entropy. The Rényi entropies S_α have been shown to encode similar properties independently of α, namely topological properties [68, 101].

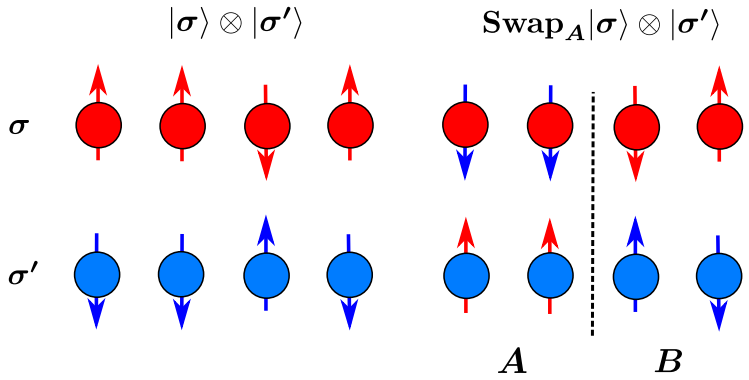

Figure 3.3: The Swap operator acting on the tensor product of two samples $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma'}$.

Although an exact evaluation of Eq. (3.25) is numerically intractable, we can use importance sampling to estimate it as [68]

$$\begin{aligned}\langle \mathrm{Swap}_A \rangle &= \sum_{\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}} |\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}_A\boldsymbol{\sigma}_B)|^2 |\Psi_{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\sigma}}_A\tilde{\boldsymbol{\sigma}}_B)|^2 \frac{\Psi_{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\sigma}}_A\boldsymbol{\sigma}_B)\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}_A\tilde{\boldsymbol{\sigma}}_B)}{\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}_A\boldsymbol{\sigma}_B)\Psi_{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\sigma}}_A\tilde{\boldsymbol{\sigma}}_B)}, \\ &\approx \frac{1}{M_e}\sum_{i=1}^{M_e} \frac{\Psi_{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\sigma}}_A^{(i)}\boldsymbol{\sigma}_B^{(i)})\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}_A^{(i)}\tilde{\boldsymbol{\sigma}}_B^{(i)})}{\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}_A^{(i)}\boldsymbol{\sigma}_B^{(i)})\Psi_{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\sigma}}_A^{(i)}\tilde{\boldsymbol{\sigma}}_B^{(i)})}. \end{aligned} \tag{3.26}$$

Using this trick, it is sufficient to generate two sets of exact samples $\{\boldsymbol{\sigma}^{(i)}\}_{i=1}^{M_e}$ and $\{\tilde{\boldsymbol{\sigma}}^{(i)}\}_{i=1}^{M_e}$ independently from $|\boldsymbol{\Psi}_{\boldsymbol{\lambda}}|^2$. By defining

$$\mathrm{Swap}_A^{(i)} \equiv \frac{\Psi_{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\sigma}}_A^{(i)}\boldsymbol{\sigma}_B^{(i)})\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}_A^{(i)}\tilde{\boldsymbol{\sigma}}_B^{(i)})}{\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}_A^{(i)}\boldsymbol{\sigma}_B^{(i)})\Psi_{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\sigma}}_A^{(i)}\tilde{\boldsymbol{\sigma}}_B^{(i)})},$$

the statistical error on the estimation of the Rényi-2 entropy can be calculated, using error propagation on the log, as

$$\epsilon = \frac{1}{\langle \mathrm{Swap}_A \rangle}\sqrt{\frac{\mathrm{Var}\left(\{\mathrm{Swap}_A^{(i)}\}\right)}{M_e}}.$$

Note that the number of samples M_e can be much larger compared to the number of samples M that is used for training our ansatz, in order to reduce the statistical error ϵ. In some cases, the Renyi entropy becomes very large, i.e, Swap expectation value is very small, especially for systems with a spatial dimension larger or equal to 2. This implies

that the signal-to-noise ratio can become small. To go around these limitations, one might resort to the improved ratio trick [32, 68]. One can also estimate the Renyi entropy with a lower statistical error through conditional sampling as shown in Ref. [102].

3.9 Targeting excited states

So far, we only talked about how the variational principle can be used to target ground states of a Hamiltonian of our interest. In this section, we will briefly talk about how we can target excited states using our variational wave function.

Similarly to DMRG, the first way to target an excited state is to target the ground state first and then aim for the first-excited state by minimizing the loss:

$$\mathcal{L}_{\boldsymbol{\lambda}} = \langle \boldsymbol{\Psi}'_{\boldsymbol{\lambda}} | \hat{H} | \boldsymbol{\Psi}'_{\boldsymbol{\lambda}} \rangle,$$

where $\boldsymbol{\Psi}'_{\boldsymbol{\lambda}} = \boldsymbol{\Psi}_{\boldsymbol{\lambda}} - \alpha \boldsymbol{\Psi}^{(G)}_{\text{approx}}$. Here $|\boldsymbol{\Psi}^{(G)}_{\text{approx}}\rangle$ is an approximation of the ground state that was found in a previous variational calculation. α is a positive Lagrange multiplier that aims to encourage the minimization of the overlap $\langle \boldsymbol{\Psi}^{(G)}_{\text{approx}} | \boldsymbol{\Psi}'_{\boldsymbol{\lambda}} \rangle$. To ensure this overlap is vanishing, we can use $\alpha = \langle \boldsymbol{\Psi}^{(G)}_{\text{approx}} | \boldsymbol{\Psi}_{\boldsymbol{\lambda}} \rangle$ [103]. Targeting other excited states can be done sequentially through the use of additional Lagrange multipliers.

The previous method can be computationally expensive and assume the knowledge of an approximation of the ground state before targeting the excited states. To go around that, one can make use of useful quantum numbers when a quantum system has a certain symmetry. For instance, if a quantum system has $U(1)$ symmetry, i.e. the Hamiltonian $\hat{H}$ and the magnetization $\hat{M}$ commute, then we construct a variational ansatz with a fixed magnetization. In this case, the task of finding an excited state boils down to the optimization of the variational energy in a similar fashion to the ground state.

To illustrate the previous point, let us take the example of the Heisenberg model, here we can use an ansatz with a fixed non-zero integer magnetization. This approach is very beneficial since we can estimate excited state energies without having access to a ground state approximation. The challenging part of this approach is figuring out how to fix a quantum number in our ansatz. We will see in the following chapter how this procedure can be done for the magnetization using our ansatz based on Recurrent Neural Networks. Discrete point group symmetries can be also useful to narrow the search space, through the use of character tables that can allow targeting different eigenstates with different group characters [104].

3.10 Variational principle in statistical physics

Previously, we have talked about how we can define a variational principle to estimate ground states or excited states of a quantum Hamiltonian. Here we will see that it is also possible to apply a similar approach to approximate the Boltzmann distribution of a classical system at a fixed temperature.

Let us consider a classical system, whose physics is governed by a classical Hamiltonian H. Such a system is considered to be at equilibrium in a fixed temperature T if it is described by the Boltzmann probability P. The latter corresponds to the minimum of the free energy F_Q (see Secs. 2.1.2 and 2.1.3), which can be written in the computational basis $\{\boldsymbol{\sigma}\}$ as:

$$F_Q = \sum_{\boldsymbol{\sigma}} Q(\boldsymbol{\sigma}) \left(H(\boldsymbol{\sigma}) + T \log(Q(\boldsymbol{\sigma}))\right) = \langle H \rangle_Q - TS(Q). \tag{3.27}$$

where the first term is the expectation value of the Hamiltonian over a probability distribution Q. T is the temperature and S corresponds to the Shannon entropy which is given by:

$$S(Q) = -\sum_{\boldsymbol{\sigma}} Q(\boldsymbol{\sigma}) \log(Q(\boldsymbol{\sigma})) = \langle -\log(Q) \rangle_Q.$$

Since the Boltzmann distribution P minimizes the free energy then we have:

$$F_Q \geq F_P; \qquad \forall Q. \tag{3.28}$$

The previous inequality is key to understanding the variational principle in statistical physics [105]. Similarly to the variational principle in quantum physics, we can introduce a well-chosen parametrized probability $P_{\boldsymbol{\lambda}}$ with some parameters $\boldsymbol{\lambda}$ and minimize the variational free energy $F_{\boldsymbol{\lambda}}$ which is defined as:

$$F_{\boldsymbol{\lambda}}(T) = \sum_{\boldsymbol{\sigma}} P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}) H(\boldsymbol{\sigma}) - TS(P_{\boldsymbol{\lambda}}), \tag{3.29}$$

where H is a classical Hamiltonian of interest and its expected value is defined over a variational probability distribution $P_{\boldsymbol{\lambda}}$. The more expressive our model $P_{\boldsymbol{\lambda}}$ the closer we expect our variational free energy to be from the minimum free energy. In this case, using the identity on the KL divergence [105]:

$$\mathrm{KL}(P||P_{\boldsymbol{\lambda}}) = T(F_{\boldsymbol{\lambda}} - F_P),$$

we can deduce that when $(F_{\boldsymbol{\lambda}} - F_P)$ gets smaller during training, then $P_{\boldsymbol{\lambda}}$ approaches the true Boltzmann distribution P.

Similarly to the VMC scheme presented in the previous section, we use importance samples to get M samples to compute the variational free energy as follows:

$$F_{\boldsymbol{\lambda}}(T) \approx \frac{1}{M} \sum_{\boldsymbol{\sigma} \sim P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})} F_{\text{loc}}(\boldsymbol{\sigma}),$$

where $F_{\text{loc}}(\boldsymbol{\sigma}) = H(\boldsymbol{\sigma}) + T \log(P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}))$ [105]. Here the ansatz must be capable of computing $P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})$ exactly and efficiently, which is the case for autoregressive models [43, 106]. For approximate likelihood models (Restricted Boltzmann Machine, Variational Autoencoder, ...), it is intractable to estimate $\log(P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}))$ and thus it would not be possible to compute the variational free energy.

Additionally, the gradients of $F_{\boldsymbol{\lambda}}$ can be computed as follows:

$$\partial_{\boldsymbol{\lambda}} F_{\boldsymbol{\lambda}}(T) \approx \frac{1}{M} \sum_{\boldsymbol{\sigma} \sim P_{\boldsymbol{\lambda}}} \partial_{\boldsymbol{\lambda}} \log\left(P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})\right) \left(F_{\text{loc}}(\boldsymbol{\sigma}) - F_{\boldsymbol{\lambda}}(T)\right),$$

where we subtract $F_{\boldsymbol{\lambda}}(T)$ in order to reduce noise in the gradients [43, 105]. For approximate-likelihood models, the computation of the gradients is efficient but as mentioned earlier the exact estimation of the variational free energy is not efficient [107].

Since phases are not necessary within this scheme, we can use a variational wave function ansatz that is positive instead of a complex-valued ansatz. It is worth noting that this variational scheme also enjoys a zero-variance principle similar to VMC, i.e. in the absence of mode collapse, the free energy variance per spin

$$\sigma_F^2 \equiv \frac{\text{var}(\{F_{\text{loc}}(\boldsymbol{\sigma})\})}{N}, \tag{3.30}$$

allows to heuristically characterize the probabilistic distance between the variational probability distribution $P_{\boldsymbol{\lambda}}$ and the Boltzmann factor which minimizes the free energy [105].

For a quantum Hamiltonian $\hat{H}$, we can also define the variational free energy to add thermal-like fluctuations to the VMC scheme to cope with local minima [35, 108]. Here we can add a pseudo-entropy [35, 106, 108] so that the cost function is defined as a pseudo variational free energy $\tilde{F}_{\boldsymbol{\lambda}}$, i.e.

$$\tilde{F}_{\boldsymbol{\lambda}}(T) = \langle \boldsymbol{\Psi}_{\boldsymbol{\lambda}} | \hat{H} | \boldsymbol{\Psi}_{\boldsymbol{\lambda}} \rangle - T S_{\text{classical}}(|\boldsymbol{\Psi}_{\boldsymbol{\lambda}}|^2). \tag{3.31}$$

Here, $\hat{H}$ is a quantum Hamiltonian with non-zero diagonal elements, as opposed to H in the expression of the variational free energy (3.29). Here, we can similarly estimate $\tilde{F}_{\boldsymbol{\lambda}}(T)$ stochastically as follows:

$$\tilde{F}_{\boldsymbol{\lambda}}(T) \approx \frac{1}{M} \sum_{\boldsymbol{\sigma} \sim |\boldsymbol{\Psi}_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})|^2} \tilde{F}_{\text{loc}}(\boldsymbol{\sigma}),$$

where $\tilde{F}_{\text{loc}}(\boldsymbol{\sigma}) \equiv E_{\text{loc}}(\boldsymbol{\sigma}) + T \log(|\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}))|^2$, such that $E_{\text{loc}}(\boldsymbol{\sigma})$ is given by Eq. (3.8). The gradients of $\tilde{F}_{\boldsymbol{\lambda}}(T)$ can be also estimated stochastically using the following expression:

$$\partial_{\lambda_i} \tilde{F}_{\boldsymbol{\lambda}}(T) \approx \frac{1}{M} \sum_{\boldsymbol{\sigma} \sim |\boldsymbol{\Psi}_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})|^2} \partial_{\lambda_i} \log\left(|\Psi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}^{(i)})|^2\right) \left(\tilde{F}_{\text{loc}}(\boldsymbol{\sigma}) - \tilde{F}_{\boldsymbol{\lambda}}(T)\right).$$

This scheme also enjoys a zero-variance principle, where the variance of $\{\tilde{F}_{\text{loc}}\}$ characterizes the distance from a global or a local minimum. We also note that in the limit of zero-temperature $T = 0$, this scheme is nothing but VMC described in the previous sections of this chapter.

In terms of computational complexity, the cost of computing the free energy gradients scales as $\mathcal{O}(Mf(N))$ if $\hat{H}$ is classical Hamiltonian with no off-diagonal elements, where $f(N)$ is the cost of a forward-pass of a one spin configuration. The latter is N times cheaper compared to the case when $\hat{H}$ is a local Hamiltonian that has off-diagonal elements similar to VMC.

Conclusions

To sum up, in this chapter we have explained the VMC scheme, where starting from a variational ansatz $|\boldsymbol{\Psi}_{\boldsymbol{\lambda}}\rangle$, we estimate its energy expectation value, as well as its gradients stochastically to optimize until reaching an approximation of a ground state or an excited state of interest. We demonstrated how to reduce the noise in the stochastic estimation of the gradients and how to monitor the convergence of our optimization through the energy variance. At the end of a variational calculation, we showed how to estimate observables and entanglement entropies. We have also shown that the VMC scheme can be applied to classical many-body systems within the framework of statistical physics. In the next chapter, we take our ansatz wave function as a recurrent neural network that is inspired by the field of natural language processing. We also demonstrate that these architectures have desirable properties, which allow competing with state-of-the-art numerical methods.

Chapter 4

Recurrent Neural Network Wave Functions

This chapter contains material from Refs. [43, 106] in addition to other material not published elsewhere.

4.1 Introduction

After discussing the variational principle framework in statistical and quantum physics in the previous chapter, we shift our focus to a promising ansatz based on recurrent neural networks (RNNs), that is efficient, highly flexible, and with a cheap computational cost. Historically, RNNs have been discovered in 1986 [109] and have been shown to be universal approximators of sequential data [110] as well as simulators of Turing machines [111]. Compared to traditional neural network architectures, RNNs have the flexibility to model inputs with a variable length such as in the case of language modeling [112]. We demonstrate this advantage also for the case of many-body systems where RNNs can be transferred across multiple system sizes. Additionally, RNNs are capable of handling long-range dependencies, unlike Hidden Markov models which require an exponential complexity for modeling such dependencies [58]. In the last few years, these architectures have set the stage for impressive performances in speech recognition and machine translation [53–55, 60]. Additionally, RNNs have proven to be powerful tools within the field of many-body physics [33, 113, 114].

In this chapter, we define RNNs and their extensions as efficient and highly flexible ansätzes for many-body variational calculations. We introduce the autoregressive property of RNNs which allows for uncorrelated sampling as well for the exact and efficient computation of probabilities and amplitudes. We present different flavors of RNNs and we demonstrate their flexibility in studying systems with multiple spatial dimensions and different types of lattices. We also illustrate the possibility to extend their definition in the case where the spatial dimension is not well defined, such as in the case of fully-connected systems. Furthermore, we show the ability of RNNs to handle physical symmetries as well as to tackle disordered and spin-glass models. Finally, we showcase the cheaper computational cost of RNNs as a highly desirable property compared to other autoregressive models.

4.2 Metropolis-Hastings Scheme

Getting the most relevant samples $\{\boldsymbol{\sigma}^{(i)}\}$ with the highest probabilities $P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})$ as in Eq. 3.9 is an important step toward a better estimation of the variational energy $E_{\boldsymbol{\lambda}}$. This task is numerically intractable if one attempts to do it exactly since one needs prior knowledge of the probability $P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})$ over the full Hilbert space. One way to go around this issue is to use a stochastic approach called the Metropolis-Hastings scheme [115].

The main idea of this scheme is to generate a Markov-Chain of the spin configurations $\tilde{\boldsymbol{\sigma}}^{(1)} \rightarrow \tilde{\boldsymbol{\sigma}}^{(2)} ... \rightarrow \tilde{\boldsymbol{\sigma}}^{(N_{\mathrm{MC}})}$, where at each step i a random spin σ of $\tilde{\boldsymbol{\sigma}}^{(i)}$ is flipped according to the acceptance probability:

$$A(\tilde{\boldsymbol{\sigma}}^{(i)} \rightarrow \tilde{\boldsymbol{\sigma}}^{(i+1)}) = \min\left(1, \frac{|\Psi_{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\sigma}}^{(i+1)})|^2}{|\Psi_{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\sigma}}^{(i)})|^2}\right). \tag{4.1}$$

As a rule of thumb, a new sample is taken after N steps where N is the system size. Following this procedure, one can get a set of configurations $(\boldsymbol{\sigma}^{(1)}, \boldsymbol{\sigma}^{(2)}, ..., \boldsymbol{\sigma}^{(M)})$ sampled from the probability distribution $P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})$ obtained from the variational wave function through the Born rule. With this scheme, we perform $N_{\mathrm{MC}} = MN$ metropolis move (4.1).

Although this scheme opened the door for Monte Carlo methods to estimate ground state energies of quantum many-body systems [2, 37], it has a serious downside often referred to as the auto-correlation problem which limits their performances [2]. One can clearly see that if the ratio $|\Psi_{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\sigma}}^{(i+1)})|^2/|\Psi_{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\sigma}}^{(i)})|^2$ is very small in the case where $\tilde{\boldsymbol{\sigma}}^{(i)}$ is highly probable compared to $\tilde{\boldsymbol{\sigma}}^{(i+1)}$. In this case, the move (4.1) is going to be frequently rejected leading to samples that are correlated and hence not faithfully representing the probability distribution $P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})$. Thus limiting the quality of observables' estimates.

4.3 Autoregressive Sampling

To go around the auto-correlation problem of the Metropolis-Hastings sampling scheme, one can make use of an effective form of sampling called autoregressive sampling. This way of sampling has been originally introduced in Sigmoid belief networks [116], later on in feed-forward neural networks [117–119], in convolutional neural networks [120], as well as in (RNNs) [121]. This autoregressive property makes these models more advantageous in terms of sampling compared to energy-based models such as Restricted Boltzmann Machines (RBMs) [122, 123], based on approximate sampling schemes such as Metropolis-Hasting scheme introduced in the previous section.

Autoregressive sampling relies on casting the joint distribution $P_{\boldsymbol{\lambda}}(\sigma_1, \sigma_2, ..., \sigma_N)$ into a product of conditional probabilities [117, 120, 121], i.e.,

$$P_{\boldsymbol{\lambda}}(\sigma_1, \sigma_2, ..., \sigma_N) = P_{\boldsymbol{\lambda}}(\sigma_1) P_{\boldsymbol{\lambda}}(\sigma_2 \,|\, \sigma_1) P_{\boldsymbol{\lambda}}(\sigma_3 \,|\, \sigma_2, \sigma_1) ... P_{\boldsymbol{\lambda}}(\sigma_N \,|\, \sigma_{N-1}, ..., \sigma_2, \sigma_1). \tag{4.2}$$

In this case, we map the problem of sampling from the joint probability to a sequential sampling from the conditional probabilities. In our case, a single configuration consists of a list $\boldsymbol{\sigma} \equiv (\sigma_1, \sigma_2, \ldots, \sigma_N)$ of N variables σ_n, and $\sigma_n \in \{0, 1, \ldots, d_v - 1\}$. Here, the *input dimension* d_v represents the number of possible values that any given variable σ_n can take.

Specifying every conditional probability $P_{\boldsymbol{\lambda}}(\sigma_i|\sigma_{<i})$ gives a full characterization of any possible distribution $P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})$, but in general such a representation grows exponentially with system size N. Typically, real-world distributions are assumed to endow enough structure on the problem to allow for accurate approximate descriptions of $P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})$ that use far fewer resources [124]. This assumption is also applicable in the context of ground state wave functions that arise in physical systems, which we will discuss at length in this thesis. In the following section, we define RNNs as a class of variational wave functions that enjoy the autoregressive sampling property.

4.4 Vanilla RNNs

RNNs form a class of correlated probability distributions of the form (4.2), where the $P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})$ are entirely specified through the conditionals $P_{\boldsymbol{\lambda}}(\sigma_i|\sigma_{<i})$. The elementary building block of an RNN is a *recurrent cell*, that has emerged in different versions in the past [61]. In its simplest form, a recurrent cell is a non-linear function that maps the direct sum (or concatenation) of an incoming *hidden* vector $\boldsymbol{h}_{n-1}$ of dimension d_h and an input vector

$\boldsymbol{\sigma}_{n-1}$ to an output hidden vector $\boldsymbol{h}_n$ of dimension d_h such that

$$\boldsymbol{h}_n = f\left(W[\boldsymbol{h}_{n-1}; \boldsymbol{\sigma}_{n-1}] + \boldsymbol{b}\right), \tag{4.3}$$

where f is a non-linear *activation function*.

The parameters of this simple RNN (vanilla RNN) are given by the weight matrix $W \in \mathbb{R}^{d_h \times (d_h + d_v)}$, the bias vector $\boldsymbol{b} \in \mathbb{R}^{d_h}$, and the states $\boldsymbol{h}_0$ and $\boldsymbol{\sigma}_0$ that initialize the recursion. Here $\boldsymbol{h}_0$ and $\boldsymbol{\sigma}_0$ are initialized to constant values. The standard initialization is a null vector. The vector $\boldsymbol{\sigma}_n$ is a one-hot encoding of the input σ_n such that, e.g., $\boldsymbol{\sigma}_n = (1,0), (0,1)$ for $\sigma_n = 0, 1$ (respectively) when the input dimension is two. The computation of the full probability $P(\boldsymbol{\sigma})$ is carried out by sequentially computing the conditionals, starting with $P(\sigma_1)$, as

$$P_{\boldsymbol{\lambda}}\left(\sigma_n | \sigma_{n-1}, \ldots, \sigma_1\right) = \boldsymbol{y}_n \cdot \boldsymbol{\sigma}_n,$$

where the right-hand side contains the usual scalar product between vectors and

$$\boldsymbol{y}_n \equiv S\left(U\boldsymbol{h}_n + \mathbf{c}\right). \tag{4.4}$$

Here, $U \in \mathbb{R}^{d_v \times d_h}$ and $\boldsymbol{c} \in \mathbb{R}^{d_v}$ are weights and biases of a so-called Softmax layer, and the Softmax activation function S is given by

$$\mathrm{S}(v_n) = \frac{\exp(v_n)}{\sum_i \exp(v_i)}.$$

By setting $U = 0$, our RNN can produce a product state controlled by the biases $\boldsymbol{c}$. This observation can play an important role in choosing a good initialization of the parameters of the RNN before starting a variational calculation. We note that advanced constructions of probability distributions using RNNs can be found in Chap. 8.

In Eq. (4.4) $\boldsymbol{y}_n = (y_n^1, \ldots, y_n^{d_v})$ is a d_v-component vector of positive, real numbers summing up to 1, i.e.,

$$\|\boldsymbol{y}_n\|_1 = 1, \tag{4.5}$$

and thus forms a probability distribution over the states σ_n. Once the vectors $\boldsymbol{y}_n$ have been specified, the full probability $P(\boldsymbol{\sigma})$ is given by

$$P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}) = \prod_{n=1}^{N} \boldsymbol{y}_n \cdot \boldsymbol{\sigma}_n.$$

Note that $P(\boldsymbol{\sigma})$ is already properly normalized to unity such that

$$\|P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})\|_1 = 1, \tag{4.6}$$

thanks to the conditional probability normalization in Eq. (4.5). Sampling from an RNN probability distribution is achieved in a similar sequential fashion. To generate a sample $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_N)$ consisting of a set of N configurations σ_n, one first calculates the hidden state $\boldsymbol{h}_1$ and the probability $\boldsymbol{y}_1$ from the initial vectors $\boldsymbol{h}_0$ and $\boldsymbol{\sigma}_0$. A sample σ_1 from the probability distribution $\boldsymbol{y}_1$ is drawn, which is then fed as a one-hot vector $\boldsymbol{\sigma}_1$ along with $\boldsymbol{h}_1$ back into the recurrent cell to obtain $\boldsymbol{y}_2, \boldsymbol{h}_2$ and then σ_2. The procedure is then iterated until N configurations σ_n have been obtained as illustrated in Fig. 4.1(c).

From Eqs. (4.3) and (4.4), it is evident that the hidden vector $\boldsymbol{h}_n$ encodes information about previous spin configurations $\sigma_{<n}$. For correlated probabilities, the history $\sigma_{<n}$ is relevant to the prediction of the probabilities of the following σ_n. By passing on hidden states in Eq. (4.4) between sites, the RNN is capable of modeling strongly correlated distributions. Hereafter, we shall call the dimension d_h of the hidden state $\boldsymbol{h}_n$ the *number of memory units*. We emphasize that the weights W and U and the biases $\boldsymbol{b}$ and $\boldsymbol{c}$ together comprise the variational parameters of our ansatz wave function of the next section. These parameters are typically shared among the different values of n, giving rise to a highly compact parametrization of the probability distribution. Once the dimension d_h is specified, the number of parameters in the ansatz is independent of the system size N.

By construction, the model allows for efficient estimation of the normalized probability of a given configuration $\boldsymbol{\sigma}$. This construction is unlike energy-based models, which require intractable calculations of the partition function, or likelihood-free models such as Generative Adversarial Networks (GANs) that do not allow for an explicit estimation of probabilities [124, 125]. The sequential process of computing the probability vectors $\boldsymbol{y}_n$ is schematically depicted in Fig. 4.1(a). Deep architectures can be obtained by stacking several RNN cells as shown in Fig. 4.1(b) for a general activation function A (not necessarily Softmax). As illustrated in Fig. 4.1(c), RNNs have the *autoregressive property*, meaning that the conditional probability $P_{\boldsymbol{\lambda}}(\sigma_n|\sigma_{<n})$ depends only on configurations $\sigma_1, \ldots \sigma_{n-1}$. We also note that the computational cost of sampling a configuration $\sigma_1, \ldots \sigma_N$ is linear in the length N of the configuration. Another important property of the normalized RNN probability distribution is that it can be used to produce successive samples $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma'}$ that are independent. Taking advantage of this property, the sampling procedure can be parallelized.

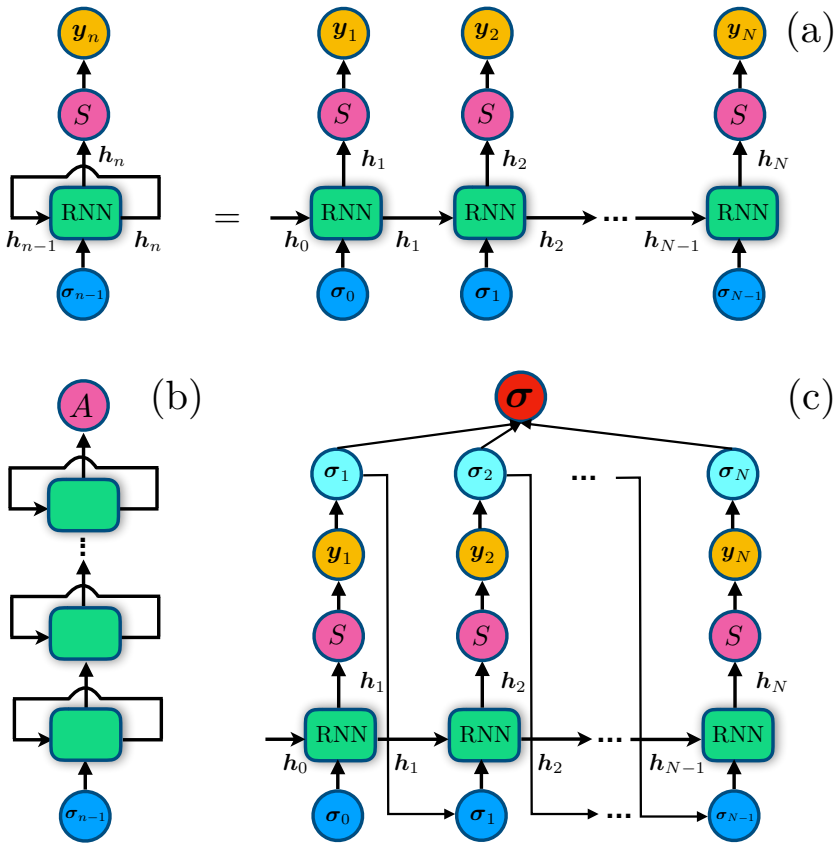

Figure 4.1: (a) Left-hand side: An RNN cell (green box) takes a sequence of inputs $\{\boldsymbol{\sigma}_n\}$, where at each step n the input $\boldsymbol{\sigma}_{n-1}$ and the vector $\boldsymbol{h}_{n-1}$ are fed in the RNN cell which generates a vector $\boldsymbol{h}_n$ called the hidden state of the RNN. $\boldsymbol{h}_n$ is meant to encode the history of the previous inputs $\boldsymbol{\sigma}_{n'<n}$. Moreover, the hidden state $\boldsymbol{h}_n$ is fed to a fully connected layer with Softmax activation S (magenta circles) to compute conditional probabilities. Right-hand side: The unrolled version of the RNN layer on the left-hand side. (b) A deep RNN model with N_l stacked single RNN cells (green blocks) followed by a fully connected layer with activation function A (magenta circle). Each single RNN cell at the ℓ-th layer has its corresponding hidden state $\boldsymbol{h}_n^\ell$, which serves also as an input for the RNN cell at the $(\ell+1)$-th layer. (c) A graphical representation of autoregressive sampling of RNNs.

4.5 Positive and Complex RNNs

The previous section focused exclusively on the efficient parametrization of classical probability distributions $P(\boldsymbol{\sigma})$. In contrast, quantum mechanical wave functions are in general a set of complex-valued amplitudes $\Psi(\boldsymbol{\sigma})$, rather than conventional probabilities. Before discussing how to modify the RNN ansatz to represent complex wave functions, we note that an important class of *stoquastic* many-body Hamiltonians has ground states $|\Psi\rangle$ with real and positive amplitudes in the standard product spin basis [126]. Thus, these ground states have representations in terms of probability distributions,

$$|\Psi\rangle = \sum_{\boldsymbol{\sigma}} \Psi(\boldsymbol{\sigma}) \, |\boldsymbol{\sigma}\rangle = \sum_{\boldsymbol{\sigma}} \sqrt{P(\boldsymbol{\sigma})} \, |\boldsymbol{\sigma}\rangle \, . \tag{4.7}$$

This property has been exploited extensively in wave function representations using generative models such as RBMs [123]. For such wave functions, it is also natural to try to approximate $P(\boldsymbol{\sigma})$ with a conventional RNN, as illustrated in Fig. 4.2(a). For later reference, we call this architecture a *positive recurrent neural network wave function* (pRNN wave function).

The generalization to the complex case starts by splitting the wave function into amplitude and phase $\phi(\boldsymbol{\sigma})$ [32] as

$$|\Psi\rangle = \sum_{\boldsymbol{\sigma}} \sqrt{P(\boldsymbol{\sigma})} \exp(\mathrm{i}\phi(\boldsymbol{\sigma})) \, |\boldsymbol{\sigma}\rangle \, . \tag{4.8}$$

As illustrated in Fig. 4.2(b), we use one RNN cell and a Softmax layer to model the probability, together with a Softsign layer (as defined below) to model the phase. In this parametrization, the first layer uses the Softmax activation function to get conditional probabilities P_n as

$$P_n = \boldsymbol{y}_n^{(1)} \cdot \boldsymbol{\sigma}_n, \tag{4.9}$$

where

$$\boldsymbol{y}_n^{(1)} = \mathrm{S}\left(U^{(1)} \boldsymbol{h}_n + \boldsymbol{c}^{(1)}\right), \tag{4.10}$$

in a similar fashion to Eq. (4.4). The Softsign layer is used to compute the phases as

$$\phi_n = \boldsymbol{y}_n^{(2)} \cdot \boldsymbol{\sigma}_n, \tag{4.11}$$

where

$$\boldsymbol{y}_n^{(2)} = \pi \, \mathrm{Softsign}\left(U^{(2)} \boldsymbol{h}_n + \boldsymbol{c}^{(2)}\right). \tag{4.12}$$

The Softsign function is defined as

$$\text{Softsign}(x) = \frac{x}{1+|x|} \in (-1, 1),$$

so that the conditional phases ϕ_n are between $-\pi$ and π. Finally, the probability $P(\boldsymbol{\sigma})$ is obtained from the N individual contributions P_n as

$$P(\boldsymbol{\sigma}) \equiv \Pi_{n=1}^{N} P_n, \tag{4.13}$$

and, similarly, the phase $\phi(\boldsymbol{\sigma})$ is computed as

$$\phi(\boldsymbol{\sigma}) \equiv \sum_{n=1}^{N} \phi_n. \tag{4.14}$$

Note that sampling from the square of the amplitudes $P(\boldsymbol{\sigma})$ is unaffected by the Softsign layer and is carried out, as described above, using only the Softmax layer as in Fig. 4.1(c). This observation motivates the use of complex numbers with a module-phase decomposition. In fact, with our construction, the change of the phase does not a-prior affect the probabilities, whereas, for a real-valued ansatz, one has to cross zero to change the amplitudes signs. The latter can introduce numerical instabilities (see denominator in Eq. (3.8)) and make the sampling of important configurations highly improbable during the training [127].

For later reference, we call this architecture a *complex recurrent neural network wave function* (cRNN wave function), and hereafter, the term RNN wave function will refer to both pRNN wave functions and cRNN wave functions.

4.6 Vanishing/Exploding Gradient Problem

In practice, training vanilla RNNs can be challenging, since capturing long-distance correlations between the variables σ_n tends to make the gradients either explode or vanish [60, 128–130]. Similar to MPS [131], long-distance correlations in RNNs are suppressed exponentially [132]. To better understand this issue, let us a simple toy model of a vanilla RNN with no activation function and with a hidden dimension $d_h = 1$. Here the RNN recursion relation can be written as:

$$h_n = \alpha h_{n-1} + \beta \sigma_{n-1} + \gamma.$$

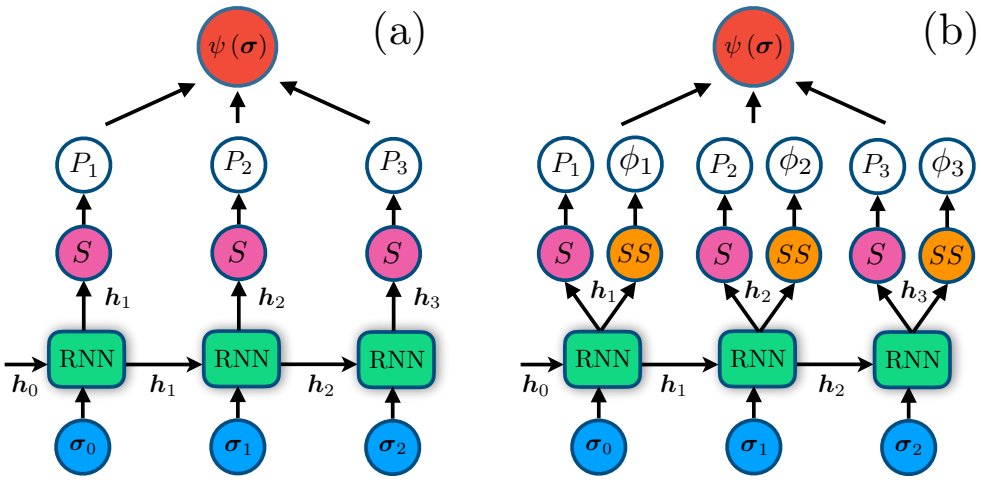

Figure 4.2: (a) pRNN wave function: A graphical representation of the computation of positive amplitudes using one RNN cell along with a Softmax layer (magenta circles) to compute the modulus $|\psi(\boldsymbol{\sigma})|^2 = P(\boldsymbol{\sigma})$. (b) cRNN wave function: A graphical representation of the computation of complex amplitudes using one RNN cell along with a Softmax layer (magenta circles) and a Softsign (SS) layer (orange circles). The first computes the modulus $|\psi(\boldsymbol{\sigma})|^2 = P(\boldsymbol{\sigma})$, the second to computes the phase $\phi(\boldsymbol{\sigma})$ of $\psi(\boldsymbol{\sigma})$.

Thus, we re-iterate this recursion relation to obtain

$$h_n = \alpha^{n-m} h_m + \beta \alpha^{n-m-1} \sigma_m + \dots ,$$

for $m < n$. The dots refer to the rest of the terms that are not relevant to our discussion. From the last expression, we can see that the partial derivative

$$\frac{\partial h_n}{\partial h_m} = \alpha^{n-m}.$$

If $|\alpha| < 1$, then the contributions to the gradient of a loss function decay exponentially with the distances $n - m$. In this case, we are dealing with a vanishing gradient problem. Otherwise, for $|\alpha| > 1$, we obtain very large gradients that can make the gradient descent optimization very unstable. This issue is also known in the literature as the exploding gradient problem. A more detailed discussion about this problem can be found in Ref. [124]. It is also interesting to see that the magnitude of the term connected to σ_m can either grow or decrease exponentially in $n - m$ which signals either the loss or the explosion of correlations propagated from large distances.

Fortunately, to go around some of the limitations of the exploding gradient problem, one can apply gradient clipping techniques [130]. Dealing with the vanishing gradient problem is a much harder task since long-distance information can get lost in the presence of noise in the gradients. One way to help mitigate this problem is through the use of other variants of RNN cells as shown in the following section. Other ways to mitigate these limitations are discussed in Ref. [130].

4.7 Gated RNNs

Extensions of the vanilla RNN have been proposed [57, 133] in order to go around the limitations of the vanishing gradient problem. Two successful examples are the long short-term memory (LSTM) unit [57], and the gated recurrent unit (GRU) [133]. In this thesis, we focus on GRUs as extensions to vanilla RNNs. The use of LSTMs could be a valuable improvement on top of GRUs.

Let us get started with the minimal version of GRUs which consist of replacing the simple recursion relation of a vanilla RNN with the following recursion relations:

$$\begin{aligned} \tilde{\boldsymbol{h}}_n &= \tanh\left(W\left[\boldsymbol{h}_{n-1}; \boldsymbol{\sigma}_{n-1}\right] + \boldsymbol{b}\right), \\ \boldsymbol{u}_n &= \mathrm{sig}\left(W_u\left[\boldsymbol{h}_{n-1}; \boldsymbol{\sigma}_{n-1}\right] + \boldsymbol{b}_u\right), \\ \boldsymbol{h}_n &= (1 - \boldsymbol{u}_n) \odot \boldsymbol{h}_{n-1} + \boldsymbol{u}_n \odot \tilde{\boldsymbol{h}}_n, \end{aligned} \tag{4.15}$$

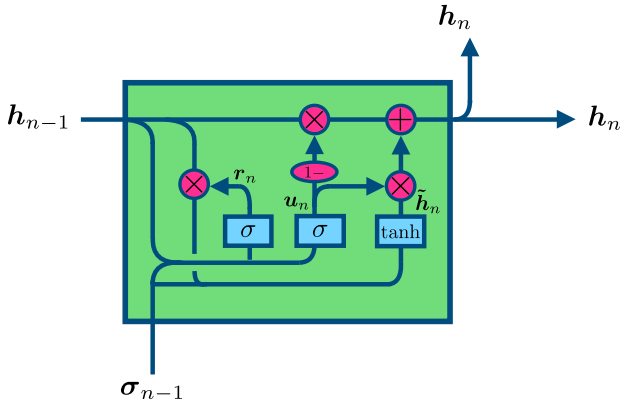

Figure 4.3: Graphical representation of the gated recurrent unit cell described in Eq. (4.16). The magenta circles/ellipses represent point-wise operations such as vector addition or multiplication. The blue rectangles represent neural network layers labeled by the non-linearity we use. Merging lines denote vector concatenation and forking lines denote a copy operation. The sigmoid activation function is represented by σ. Credit: Juan Carrasquilla.

where 'sig' and 'tanh' represent the sigmoid and hyperbolic tangent activation functions, respectively. Thus, the hidden vector $\boldsymbol{h}_n$ is updated through an interpolation between the previous hidden state $\boldsymbol{h}_{n-1}$ and a candidate hidden state $\tilde{\boldsymbol{h}}_n$. The update gate $\boldsymbol{u}_n$ decides to what extent the contents of the hidden state are modified, and depends on how relevant the input σ_{n-1} is to the prediction (Softmax layer). The symbol $\odot$ denotes the pointwise (Hadamard) product. It is important to note that the components of $\boldsymbol{h}_n$ are always between -1 and 1 as we are using 'sig' and 'tanh'.

In addition to the minimal version of GRUs, one can define an advanced GRU cell, as introduced in Ref. [133], which processes the spin configurations $\boldsymbol{\sigma}$ as

$$\begin{aligned}
\boldsymbol{u}_n &= \mathrm{sig}\left(W_u\left[\boldsymbol{h}_{n-1};\boldsymbol{\sigma}_{n-1}\right]+\boldsymbol{b}_u\right),\\
\boldsymbol{r}_n &= \mathrm{sig}\left(W_r\left[\boldsymbol{h}_{n-1};\boldsymbol{\sigma}_{n-1}\right]+\boldsymbol{b}_r\right),\\
\tilde{\boldsymbol{h}}_n &= \tanh\left(W_c\left[\boldsymbol{r}_n\odot\boldsymbol{h}_{n-1};\boldsymbol{\sigma}_{n-1}\right]+\boldsymbol{b}_c\right),\\
\boldsymbol{h}_n &= (1-\boldsymbol{u}_n)\odot\boldsymbol{h}_{n-1}+\boldsymbol{u}_n\odot\tilde{\boldsymbol{h}}_n.
\end{aligned} \tag{4.16}$$

The additional reset gate in this GRU implementation is modeled by the vector $\boldsymbol{r}_n$, such that if the i-th component $\boldsymbol{r}_n$ is close to zero, it cancels out the i-th component of the hidden vector state $\boldsymbol{h}_{n-1}$, effectively making the GRU "forget" part of the sequence that has already been encoded in the state vector $\boldsymbol{h}_{n-1}$. The weights matrices $W_{u,r,c}$ and the bias vectors $\boldsymbol{b}_{u,r,c}$ parametrize the GRU and are optimized using variational energy minimization

as described in Chap. 3. The GRU transformations in Eq. (4.15) are depicted graphically in Fig. 4.3.

To take advantage of the GPU speed up, we use instead the cuDNN variant of GRUs implemented in Tensorflow [134], with

$$\begin{aligned}
\boldsymbol{u}_n &= \mathrm{sig}\left(W_u\left[\boldsymbol{h}_{n-1};\boldsymbol{\sigma}_{n-1}\right]+\boldsymbol{b}_u\right), \\
\boldsymbol{r}_n &= \mathrm{sig}\left(W_r\left[\boldsymbol{h}_{n-1};\boldsymbol{\sigma}_{n-1}\right]+\boldsymbol{b}_r\right), \\
\boldsymbol{h}'_n &= W_c^{(1)}\boldsymbol{h}_{n-1}+\boldsymbol{b}_c^{(1)}, \\
\tilde{\boldsymbol{h}}_n &= \tanh\left(W_c^{(2)}\boldsymbol{\sigma}_{n-1}+\boldsymbol{r}_n\odot\boldsymbol{h}'_n+\boldsymbol{b}_c^{(2)}\right), \\
\boldsymbol{h}_n &= (1-\boldsymbol{u}_n)\odot\boldsymbol{h}_{n-1}+\boldsymbol{u}_n\odot\tilde{\boldsymbol{h}}_n,
\end{aligned} \tag{4.17}$$

which differs slightly from the above implementation of traditional GRU cells [135]. As a final note, to distinguish between the different RNN cells, we denote an RNN ansatz based on GRUs as a GRU wave function to distinguish it from RNN wave functions based on vanilla RNN cells.

4.8 Multi-dimensional RNNs

Standard RNN architectures are typically one-dimensional. However, the most interesting quantum many-body systems live in higher dimensions. By taking inspiration from Refs. [58, 136, 137], we generalize one-dimensional RNNs to multi-dimensional RNN wave functions. In particular, we generalize to two-dimensional vanilla RNNs (2D RNNs) that are more suitable for simulating two-dimensional square lattices than one-dimensional RNNs, which map two-dimensional lattice configurations to one-dimensional configurations and do not necessarily encode spatial information about neighboring sites in a plausible manner.

The main idea behind the implementation of 2D RNNs [58] is to replace the single hidden state that is passed from one site to another with two hidden states, with each one corresponding to the state of a neighboring site (vertical and horizontal) and hence respecting the 2D geometry of the problem. To do so, we change the one-dimensional recursion relation in (4.3) to the two-dimensional recursion relation

$$\boldsymbol{h}_{i,j}=f\left(W^{(h)}[\boldsymbol{h}_{i-1,j};\boldsymbol{\sigma}_{i-1,j}]+W^{(v)}[\boldsymbol{h}_{i,j-1};\boldsymbol{\sigma}_{i,j-1}]+\boldsymbol{b}\right), \tag{4.18}$$

where $\boldsymbol{h}_{i,j}$ is the hidden state at site (i,j), $W^{(v,h)}$ are weight matrices and $\boldsymbol{b}$ is a bias. Here f is a non-linear activation function, which is typically taken in this study to be equal to

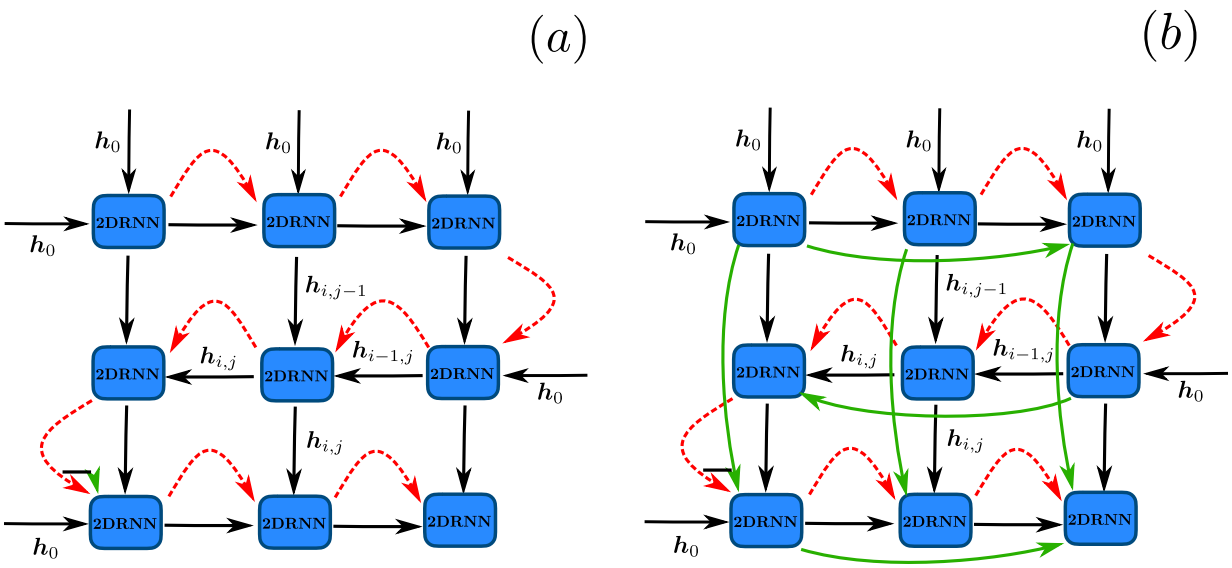

Figure 4.4: (a) A graphical illustration of a 2D RNN. Each RNN cell receives two hidden states $\boldsymbol{h}_{i,j-1}$ and $\boldsymbol{h}_{i-1,j}$, as well as two input vectors $\boldsymbol{\sigma}_{i,j-1}$ and $\boldsymbol{\sigma}_{i-1,j}$ (not shown) as illustrated by the black arrows. The red arrows correspond to the zigzag path we use for 2D autoregressive sampling. The initial memory state $\boldsymbol{h}_0$ of the RNN and the initial inputs $\boldsymbol{\sigma}_0$ (not shown) are null vectors. (b) A 2D periodic RNN that takes periodic boundary conditions into account. RNN cells at the boundary receive an additional $\boldsymbol{h}_{i,\mathrm{mod}(j+1,L)}$ and $\boldsymbol{h}_{\mathrm{mod}(i+1,L),j}$, as well as two input vectors $\boldsymbol{\sigma}_{i,\mathrm{mod}(j+1,L)}$ and $\boldsymbol{\sigma}_{\mathrm{mod}(i+1,L),j}$ (not shown) as demonstrated by the green solid arrows.

the Exponential Linear Unit "ELU" [138] defined as

$$\mathrm{ELU}(x) = \begin{cases} x, & \text{if } x \geq 0\,, \\ \exp(x) - 1, & \text{if } x < 0\,. \end{cases}$$

This activation has shown good results in this study. However, the optimal choice of an activation function is still an open area of research. The cost of computing a new hidden state $h_{i,j}$ is quadratic in the size of the hidden state (number of memory units d_h), and the cost of computing the gradients with respect to the variational parameters of the 2D RNN remains unchanged. This property allows training 2D RNNs with a relatively large d_h. It is also very inexpensive compared to, e.g., the expensive variational optimization of projected entangled pair states (PEPS) [139], which scales as $\chi^2 \tilde{D}^6$ (where $\tilde{D}$ the PEPS bond dimension and χ is the bond dimension of the intermediate MPS) [140].

The scheme for computing positive or complex amplitudes from Sec. 4.5 remains the same. We note that the coordinates $(i-1, j)$ and $(i, j-1)$ are path-dependent, and are given by the zigzag path, illustrated by the black arrows in Fig. 4.4(a). Moreover, to sample configurations from the 2D RNNs, we use the same zigzag path as illustrated by the red dashed arrows in Fig. 4.4(a). Once $\boldsymbol{h}_{i,j}$ is computed, we apply the same scheme as in Sec. 4.4 to sample a spin $\sigma_{i,j}$.

It is important to note that the use of a zigzag path allows for circumventing the use of non-local recurrent connections that are not expected to be efficient compared to local recurrent connections. From the perspective of tensor networks, this intuition is inspired by the efficiency of pair-entangled project states (PEPS) compared to matrix product states (MPS) in terms of the bond dimension size, where non-local bonds tend to carry a large amount of entanglement compared to other local bonds. Overall, a good rule of thumb is to choose a sampling path that avoids non-local recurrent connections. An optimal choice of a sampling path is an interesting research direction to be investigated in a future study.

We note that generalization to higher dimensions, to other lattices, as well as to other types of RNN architectures can be done by taking inspiration from this scheme. For instance, using LSTMs [57], GRUs [133] or Transformers [53] instead of vanilla RNNs in 2D is expected to make a significant improvement. The use of multiplicative interactions [141] is expected to increase the expressiveness of 2D RNNs as compared to the additive interactions in (4.18). This point is illustrated further in the next section where we show how to build a tensorized version of RNNs.

For the sake of concreteness, let us define an RNN in three spatial dimensions. In this case, one can generalize the previous ideas to a 3D RNN, which in its vanilla form is based

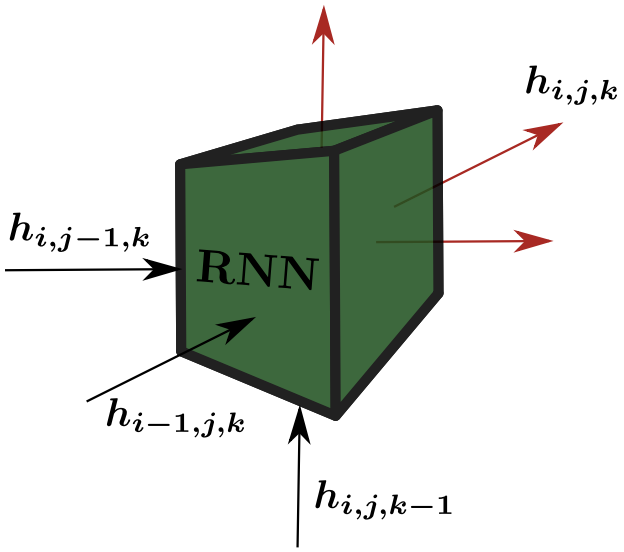

Figure 4.5: A graphical illustration of a 3D RNN. Here the hidden states' indices and the inputs' indices are promoted to 3D. Each RNN cell receives three hidden states $\boldsymbol{h}_{i-1,j,k}$, $\boldsymbol{h}_{i,j-1,k}$, and $\boldsymbol{h}_{i,j,k-1}$, as well as three input vectors $\boldsymbol{\sigma}_{i-1,j,k}$, $\boldsymbol{\sigma}_{i,j-1,k}$, and $\boldsymbol{\sigma}_{i,j,k-1}$ (not shown) as illustrated by the black arrows. At the output, we obtain a new hidden state $\boldsymbol{h}_{i,j,k}$ as indicated by the red arrows.

on the following recursion relation:

$$\boldsymbol{h}_{i,j,k} = f(W[\boldsymbol{\sigma}_{i-1,j,k}; \boldsymbol{h}_{i-1,j,k}; \boldsymbol{\sigma}_{i,j-1,k}; \boldsymbol{h}_{i,j-1,k}; \boldsymbol{\sigma}_{i,j,k-1}; \boldsymbol{h}_{i,j,k-1}] + \boldsymbol{b}).$$

Here, each new hidden state is computed based on the neighboring hidden states and the neighboring inputs as illustrated in Fig. 4.5. An optimal sampling path can be taken as a zigzag path in the 3D lattice in a similar fashion to 2D to avoid passing non-local recurrent connections. Generalizations to a gated version of the 3D RNN can be devised in a similar fashion to the gated RNN in Sec. 4.7.

4.9 Tensorized RNNs

In this section, we show how to build a tensorized version of RNNs, which is inspired by the contraction operations in Tensor Networks [142, 143]. The latter involves multiplicative interactions between spins as opposed to additive interaction in a vanilla RNN. A concrete incentive for using multiplicative interactions is motivated in App. D.1, where we show more

compact constructions of the traditional probability distributions when using multiplicative interactions compared to additive interactions. Our version of tensorized RNNs consists of replacing the concatenation operation in Eq. (4.3) with the operation [144]

$$\boldsymbol{h}_n = F\left(\boldsymbol{\sigma}_{n-1}^{\intercal} T \boldsymbol{h}_{n-1} + \boldsymbol{b}\right), \tag{4.19}$$

where $\boldsymbol{\sigma}^{\intercal}$ is the transpose of $\boldsymbol{\sigma}$, and the variational parameters $\boldsymbol{\lambda}$ are T, U, $\boldsymbol{b}$ and $\boldsymbol{c}_n$. This form of tensorized RNNs increases the expressiveness of our ansatz as we illustrate in Chap. 5.

For two-dimensional systems, we extend our 1D tensorized RNN to a 2D version through the following recursion relation:

$$\boldsymbol{h}_{i,j} = F([\boldsymbol{\sigma}_{i-1,j}; \boldsymbol{\sigma}_{i,j-1}] T [\boldsymbol{h}_{i-1,j}; \boldsymbol{h}_{i,j-1}] + \boldsymbol{b}). \tag{4.20}$$

The tuneable arrays T and $\boldsymbol{b}$ correspond to a tensor and biases, respectively. An interesting combination that one can think about is the incorporation of the gating mechanism in our 2D tensorized RNN cell. We will show in the next chapter that this mechanism allows for improving the accuracy of our variational calculations. A tensorized gated RNN cell can be built through the following recursion relations:

$$\tilde{\boldsymbol{h}}_{i,j} = \tanh([\boldsymbol{\sigma}_{i-1,j}; \boldsymbol{\sigma}_{i,j-1}] T [\boldsymbol{h}_{i-1,j}; \boldsymbol{h}_{i,j-1}] + \boldsymbol{b}), \tag{4.21}$$

$$\boldsymbol{u}_{i,j} = \text{sigmoid}([\boldsymbol{\sigma}_{i-1,j}; \boldsymbol{\sigma}_{i,j-1}] T_g [\boldsymbol{h}_{i-1,j}; \boldsymbol{h}_{i,j-1}] + \boldsymbol{b_g}), \tag{4.22}$$

$$\boldsymbol{h}_{i,j} = \boldsymbol{u}_{i,j} \odot \tilde{\boldsymbol{h}}_{i,j} + (1 - \boldsymbol{u}_{i,j}) \odot (W [\boldsymbol{h}_{i-1,j}; \boldsymbol{h}_{i,j-1}]). \tag{4.23}$$

Here we obtain the state $\boldsymbol{h}_{i,j}$ through a combination of the neighbouring hidden states $\boldsymbol{h}_{i-1,j}, \boldsymbol{h}_{i,j-1}$ and the candidate hidden state $\tilde{\boldsymbol{h}}_{i,j}$. The update gate $\boldsymbol{u}_{i,j}$ decides how much the candidate hidden state $\tilde{\boldsymbol{h}}_{i,j}$ will be modified. The latter combination allows circumventing some of the limitations related to the vanishing gradient problem [132, 145]. Note that the size of the hidden state $\boldsymbol{h}_{i,j}$ that we denote as d_h is a hyperparameter that we choose before optimizing the parameters of our ansatz. Overall, one can generalize each RNN cell in a specific spatial dimension to a tensorized version to improve its expressivity.

Hereafter, we denote an ansatz based on tensorized RNN as a 'TRNN' and an ansatz made of tensorized gated RNN as 'TGRU'. Finally, we would like to note that after the publication of this work in Ref. [108], another approach to implement tensorized RNNs has been suggested in Ref. [146]. This reference also shows that these architectures with a specific design can encode the area law of entanglement in two spatial dimensions.

4.10 Dilated RNNs

To account for the long-distance nature of the correlations induced in a physical system, we use dilated RNNs [147], which are known to alleviate the vanishing gradient problem [128]. Dilated RNNs are multi-layered RNNs that use dilated connections between spins to model long-term dependencies [148], as illustrated in Fig. 4.6. At each layer $1 \leq l \leq L$, the hidden state is computed as

$$\boldsymbol{h}_n^{(l)} = F(W_n^{(l)}[\boldsymbol{h}^{(l)}_{\max(0,n-2^{l-1})}; \boldsymbol{h}_n^{(l-1)}] + \boldsymbol{b}_n^{(l)}).$$

Here $\boldsymbol{h}_n^{(0)} = \boldsymbol{\sigma}_{n-1}$ and the conditional probability is given by

$$P_{\boldsymbol{\lambda}}(\sigma_n|\sigma_{<n}) = \mathrm{Softmax}(U_n \boldsymbol{h}_n^{(L)} + \boldsymbol{c}_n) \cdot \boldsymbol{\sigma}_n.$$

In our work, we choose the size of the hidden states $\boldsymbol{h}_n^{(l)}$, where $l > 0$, as constant and equal to d_h. We also use a number of layers $L = \lceil \log_2(N) \rceil$, where N is the number of spins and $\lceil \ldots \rceil$ is the ceiling function. This means that two spins are connected with a path whose length is bounded by $\mathcal{O}(\log_2(N))$, which follows the spirit of the multi-scale renormalization ansatz (MERA) [149]. Hereafter, we denote this ansatz as 'DRNN'.

4.11 Symmetric RNNs

4.11.1 Imposing discrete symmetries

Inspired by Refs. [105, 150], we propose to implement discrete symmetries in a similar fashion for RNN wave functions without spoiling their autoregressive nature. Assuming that a Hamiltonian $\hat{H}$ has a symmetry under discrete transformations $\mathcal{T}$, its excited states

$$|\boldsymbol{\Psi}_e\rangle = \sum_{\boldsymbol{\sigma}} \Psi_e(\boldsymbol{\sigma}) \, |\boldsymbol{\sigma}\rangle$$

are eigenvectors of the symmetry transformation $\mathcal{T}$. An excited state transforms as

$$\Psi_e(\mathcal{T}\boldsymbol{\sigma}) = \omega_{\mathcal{T}} \Psi_e(\boldsymbol{\sigma}), \tag{4.24}$$

where $\omega_{\mathcal{T}}$ is an eigenvalue with module 1, which is independent of the choice of $\boldsymbol{\sigma}$. This phase verifies the multiplicative law $\omega_{\tilde{\mathcal{T}}}\omega_{\mathcal{T}} = \omega_{\tilde{\mathcal{T}}\mathcal{T}}$ for any two symmetry transformations $\tilde{\mathcal{T}}$ and $\mathcal{T}$. Additionally, the phases $\omega_{\mathcal{T}}$ can be obtained from each symmetry through the

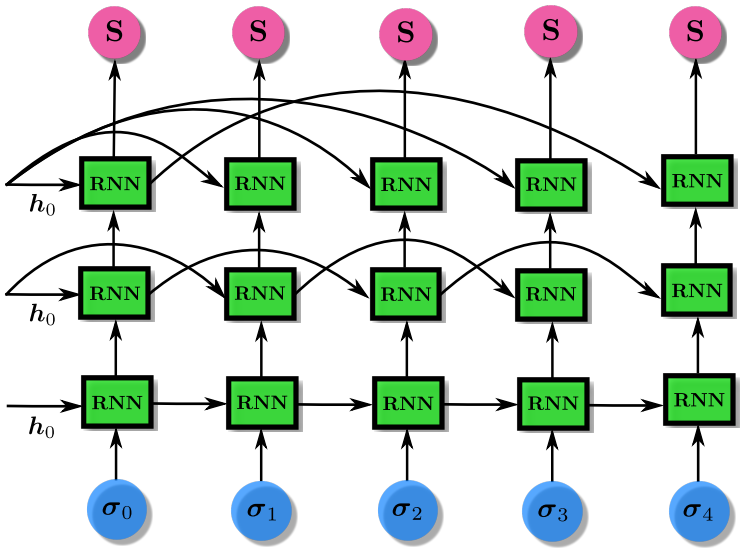

Figure 4.6: An illustration of a dilated RNN, where the distance between each couple of RNN cells grows exponentially with depth to account for long-term dependencies. We choose depth $L = \lceil \log_2(N) \rceil$ where N is the number of spins.

symmetry group character [104]. The transformation law (4.24) implies that the transformation $\mathcal{T}$ changes an eigenstate state with only a global phase term that does not affect the probability distribution. It is thus desirable that our RNN wave function satisfies this symmetry transformation. This approach is very helpful for targeting excited states as discussed in Sec. 3.9.

To enforce a discrete symmetry on an RNN wave function $|\boldsymbol{\Psi}_{\boldsymbol{\lambda}}\rangle$, we implement the following scheme:

- Generate a sample $\boldsymbol{\sigma}$ autoregressively from the RNN wave function.
- Sample with a probability $1/\mathrm{Card}(G)$ a transformation $\mathcal{T}$ from the symmetry transformation group $G = \{I_d, \mathcal{T}_1, ...\}$ that leaves the Hamiltonian $\hat{H}$ invariant, and apply the transformation $\mathcal{T}$ to $\boldsymbol{\sigma}$.
- Assign to the spin configuration $\tilde{\boldsymbol{\sigma}} = \mathcal{T}\boldsymbol{\sigma}$ the amplitude $\Psi'_{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\sigma}}) = \sqrt{P'_{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\sigma}})}\exp(i\phi'_{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\sigma}}))$,

such that

$$P'_{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\sigma}}) = \frac{1}{|G|}\left(\sum_{\tilde{\mathcal{T}}\in G} P_{\boldsymbol{\lambda}}\left(\tilde{\mathcal{T}}\boldsymbol{\sigma}\right)\right), \tag{4.25}$$

$$\phi'_{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\sigma}}) = \mathrm{Arg}\left(\sum_{\tilde{\mathcal{T}}\in G} \omega_{\tilde{\mathcal{T}}}^{-1} \exp\left(\mathrm{i}\phi_{\boldsymbol{\lambda}}\left(\tilde{\mathcal{T}}\boldsymbol{\sigma}\right)\right)\right), \tag{4.26}$$

where $P_{\boldsymbol{\lambda}}(\tilde{\mathcal{T}}\boldsymbol{\sigma})$ is a probability generated by the Softmax layer and $\phi_{\boldsymbol{\lambda}}(\tilde{\mathcal{T}}\boldsymbol{\sigma})$ is a phase generated by the Softsign layer, as explained in Sec. 4.5. The multiplicative law of the phase $\omega_{\mathcal{T}}$ allows showing that our symmetrized complex RNN wave functions satisfy the property (4.24).

For a ground state, the phases $\omega_{\mathcal{T}}$ are typically equal to one for a symmetrical Hamiltonian [151]. In this case, we obtain a simpler scheme for targeting ground states. If the ground state is positive as in stoquastic Hamiltonians [126], we use the same algorithm but only symmetrize the probability $P_{\boldsymbol{\lambda}}$, since there is no need to use the phases $\phi_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})$.

For concreteness, we illustrate the algorithm above with "Symmetric RNNs" that have a built-in parity symmetry and are used in Sec. 5.1.1. Symmetric RNNs can be implemented using the following procedure:

- Sample each configuration $\boldsymbol{\sigma}$.
- Choose to apply or to not apply the parity transformation $\hat{\mathcal{P}}$ on $\boldsymbol{\sigma}$ with a probability $1/2$.
- Assign to $\boldsymbol{\sigma}$ the probability:

$$P'(\boldsymbol{\sigma})_{\boldsymbol{\lambda}} = \frac{\left(P_{\boldsymbol{\lambda}}\left(\boldsymbol{\sigma}\right) + P_{\boldsymbol{\lambda}}\left(\hat{\mathcal{P}}\boldsymbol{\sigma}\right)\right)}{2}.$$

As a final note, the separation between the symmetrization of the probabilities and the phases in Eqs. (4.25), (4.26) allows to conserve the autoregressive property of our RNN wave function. We would like to point out that other symmetrization schemes for applying point group symmetries have been investigated in the literature as shown in Ref. [152]. However, they are shown to spoil the autoregressive property of the RNN [152]. The design of more efficient symmetrization schemes that preserves the autoregressive property of RNNs is of great use for getting accurate variational calculations.

4.11.2 Imposing $U(1)$ symmetry and $SU(2)$ symmetry

Since a large class of ground states has conserved magnetization, i.e., a $U(1)$ symmetry [153, 154], it is helpful to enforce this constraint on our RNN wave functions to get accurate estimations of the ground state energy. An example is the ground state of the quantum Heisenberg model. To do so, we propose an efficient way to generate samples with zero magnetization while maintaining the autoregressive property of the RNN wave function. The procedure effectively applies a projector $\mathcal{P}_{S_z=0}$ to the original state, which restricts the RNN wave function to the subspace of configurations with zero magnetization. This procedure avoids generating a large number of samples and discarding the ones that have non-zero magnetization.

The condition of zero magnetization implies that the number of up spins should be equal to the number of down spins. To satisfy this constraint, we utilize the following algorithm:

- Autogressively sample the first half of the spin configuration $(\sigma_1, \sigma_2, ..., \sigma_{N/2})$
- At each step $i > N/2$:
  - Generate the output of the RNN wave function: $\boldsymbol{y}_i = (\psi_i^{\text{down}}, \psi_i^{\text{up}})$ where ψ_i^{down}, ψ_i^{up} are both non-zero and whose modules squared sum to 1.
  - Define the following amplitudes:

$$a_i = \psi_i^{\text{down}} \times \Xi\left(\frac{N}{2} - N_{\text{down}}(i)\right),$$
$$b_i = \psi_i^{\text{up}} \times \Xi\left(\frac{N}{2} - N_{\text{up}}(i)\right),$$

    where

$$\Xi(x) \equiv \begin{cases} 1, & \text{if } x > 0\,, \\ 0, & \text{if } x \leq 0\,, \end{cases}$$

    and

$$N_{\text{down}}(i) = \text{Card}\left(\{j \ / \sigma_j = 0 \text{ and } j < i\}\right),$$
$$N_{\text{up}}(i) \quad = \text{Card}\left(\{j \ / \sigma_j = 1 \text{ and } j < i\}\right).$$

    In words, $N_{\text{up}}(i)/N_{\text{down}}(i)$ is the number of up/down spins generated before step i.

– Sample σ_i from $|\tilde{\boldsymbol{y}}_i|^2$, where:

$$\tilde{\boldsymbol{y}}_i = \frac{1}{\sqrt{a_i^2 + b_i^2}}(a_i, b_i)$$

which is normalized, i.e. $||\tilde{\boldsymbol{y}}_i||_2 = 1$.

Using this algorithm, it is clear that the RNN wave function will generate a spin configuration that has the same number of up spins and down spins, and hence a zero magnetization. In fact, at each step $i > N/2$, the function Ξ assigns a zero amplitude for the next spin σ_i to be spin up if $N_{\text{up}}(i) = N/2$ or to be spin down if $N_{\text{down}}(i) = N/2$.

Our scheme does not spoil the normalization of the RNN wave function as the new conditional probabilities $|\tilde{\boldsymbol{y}}_i|^2$ are also normalized. We also note that this algorithm preserves the autoregressive property of the original RNN wave function and can also be parallelized. Moreover, this scheme can be easily extended to the generation of samples with a non-zero fixed magnetization, which is useful when considering the problem of finding excited states that live in a non-zero fixed magnetization sector.

In order to apply $SU(2)$ symmetry on the RNN, one can do a change of basis to the j-basis [155] and apply the same trick provided in this section in order to project our model to the total spin quantum number j section [156]. It is not clear up to date whether this approach can improve the accuracy of a variational calculation targeting the ground state. However, we believe that this trick can be beneficial when the goal is to target an excited state with a specific quantum number j.

4.12 RNNs for special lattices

We have seen in Sec. 4.8 how to define an RNN wave function for regular lattices such as the square and the cubic lattices. For other types of lattices, it is challenging to define a customized RNN that can efficiently handle the particular structure of the lattice. One approach is to add recurrent connections to physically mimic the interactions between different degrees of freedom. This approach can be done through the use of graph RNN [157]. Another simpler approach is through the use of an enlarged Hilbert space as described below.

Let us take the example of the configurations of the 2D toric code model [158] that can be mapped to a $L \times L \times 2$ array of spins where L is the number of plaquettes on each side. To take that into account, we enlarge the local Hilbert space dimension in the RNN from

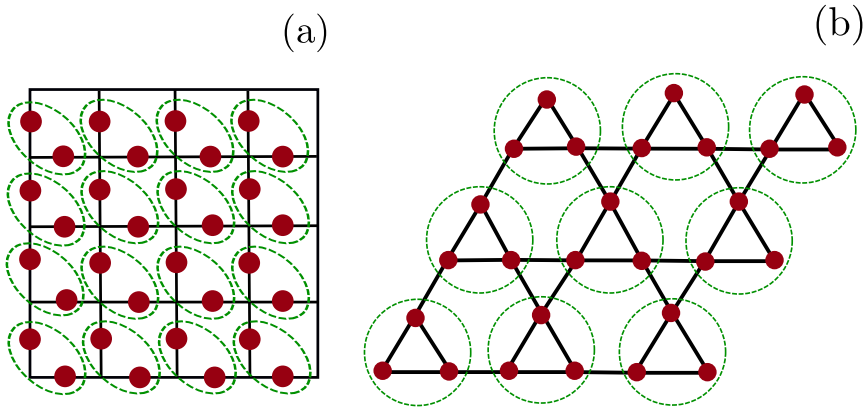

Figure 4.7: Mapping of 2D toric code lattice and Kagome lattice to a square lattice that can be handled by a 2D RNN wave function.

2 to 4 to be able to feed-in/sample two spins to/from the RNN at once. Using this idea, we can use our 2D RNN with an enlarged local Hilbert space to study the 2D toric code, as illustrated in Fig. 4.7(a).

Another example is the spin-1/2 Kagome lattice whose configurations can be mapped to an $L \times L \times 3$ array of binary degrees of freedom where L is the size of each side of the Kagome lattice. Taking this observation into account, we can enlarge the local Hilbert space size from 2 to 8 to be able to feed in/sample 3 spins locally at once to/from the 2D RNN, as illustrated in Fig. 4.7(b). The latter allows us to map our Kagome lattice with a local Hilbert space of 2 to a square lattice with an enlarged Hilbert space of size $2^3 = 8$ and which we can study using our 2D RNN wave function.

We would like to point out that these ideas are also applicable to adapting 1D RNNs to the study of quasi-1D systems, as well as 2D RNNs to the study of quasi-2D systems in a similar manner to how DMRG is used to study quasi-1D systems [159].

As a concluding note, a good rule of thumb is to identify a unit cell in a lattice of interest. This step allows us to transform it into a regular lattice with an enlarged local Hilbert space that can be well-studied with a D-dimensional RNN.

4.13 On weight sharing in RNNs

For physical systems with translation invariance, it is natural to use the same weights for each RNN cell across all sites as illustrated in the recursion relation (4.3) and in the

Softmax layer (4.4). This property promotes the use of trained RNNs on small system sizes as initialization of RNNs targeting a many-body system with larger system sizes as numerically demonstrated in Refs. [35, 108]. A numerical illustration of this idea is given in Sec. 6.7.

For disordered systems such as spin glass models, it is intuitive to forgo the common practice of weight sharing in RNNs [124]. In particular, for a vanilla RNN one can use an extended set of site-dependent variational parameters $\boldsymbol{\lambda}$ comprised of $\{W_n\}_{n=1}^N, \{V_n\}_{n=1}^N$ and $\{U_n\}_{n=1}^N$ and biases $\{\boldsymbol{b}_n\}_{n=1}^N$, $\{\boldsymbol{c}_n\}_{n=1}^N$. The recursion relation (4.3) and the Softmax layer (4.4) are modified to

$$\boldsymbol{h}_n = f(W_n \boldsymbol{h}_{n-1} + V_n \boldsymbol{\sigma}_{n-1} + \boldsymbol{b}_n), \tag{4.27}$$

and

$$P_{\boldsymbol{\lambda}}(\sigma_n|\sigma_{<n}) = \text{Softmax}(U_n \boldsymbol{h}_n + \boldsymbol{c}_n) \cdot \boldsymbol{\sigma}_n, \tag{4.28}$$

respectively. This scheme also applies to other RNNs cells in 1D or higher dimensions. The advantage of using site-dependent parameters for disordered systems is numerically demonstrated in Chap. 6.

4.14 RNNs for periodic boundary conditions

In the previous chapters, we have seen how to use recurrent neural networks as an ansatz wave function for 1D, 2D, and multi-dimensional systems. We have also seen in Sec. 4.6 that information from inputs at long distances can get lost due to the vanishing gradient problem. In particular variables at the boundaries can be weakly correlated in a conventional RNN. In this section, we show how to add extra recurrent connections to our RNNs to go around this limitation thus obtaining a periodic version of RNNs.

Let us take the example of two-dimensional quantum systems with periodic boundary conditions. To study these systems with 2D RNNs, we modify our two-dimensional recursion in Eq. (4.18) as follows:

$$\begin{aligned}\boldsymbol{h}_{i,j} = f\Big(W[&\boldsymbol{\sigma}_{i-1,j}; \boldsymbol{\sigma}_{i,j-1}; \boldsymbol{\sigma}_{\text{mod}(i+1,L),j}; \boldsymbol{\sigma}_{i,\text{mod}(j+1,L)};\\ &\boldsymbol{h}_{i-1,j}; \boldsymbol{h}_{i,j-1}; \boldsymbol{h}_{\text{mod}(i+1,L),j}; \boldsymbol{h}_{i,\text{mod}(j+1,L)}] + \boldsymbol{b}\Big).\end{aligned} \tag{4.29}$$

$\boldsymbol{h}_{i,j}$ is a hidden state with two indices for each spin in the 2D lattice. Furthermore, f is an activation function. After obtaining $\boldsymbol{h}_{i,j}$, the conditional probabilities can be computed as follows

$$p_{\boldsymbol{\theta}}(\sigma_{i,j}|\sigma_{<i,j}) = \text{Softmax}(U\boldsymbol{h}_{i,j} + \boldsymbol{c}) \cdot \boldsymbol{\sigma}_{i,j}. \tag{4.30}$$

The additional spin inputs $\boldsymbol{\sigma}_{\mathrm{mod}(i+1,L),j}$, $\boldsymbol{\sigma}_{i,\mathrm{mod}(j+1,L)}$ and the hidden states $\boldsymbol{h}_{\mathrm{mod}(i+1,L),j}$, $\boldsymbol{h}_{i,\mathrm{mod}(j+1,L)}$ allow to take periodic boundary conditions into account and to introduce extra correlations between variables at the boundaries. This approach has been also suggested and implemented in Ref. [156]. We note that during the process of autoregressive sampling, if one of these additional vectors is not defined, we initialize it to a null vector to preserve the autoregressive nature of our scheme as illustrated in Fig. 4.4(b). Additionally, this figure illustrates the autoregressive sampling path as well as how information is being transferred from one RNN to another.

Furthermore, one can define an advanced version of 2D periodic RNN by incorporating the gating mechanism as it was previously done in Secs. 4.7, 4.9 and Refs. [156, 160]. If we define

$$\begin{aligned}\boldsymbol{h}'_{i,j} &= [\boldsymbol{h}_{i-1,j}; \boldsymbol{h}_{i,j-1}; \boldsymbol{h}_{\mathrm{mod}(i+1,L),j}; \boldsymbol{h}_{i,\mathrm{mod}(j+1,L)}],\\ \boldsymbol{\sigma}'_{i,j} &= [\boldsymbol{\sigma}_{i-1,j}; \boldsymbol{\sigma}_{i,j-1}; \boldsymbol{\sigma}_{\mathrm{mod}(i+1,L),j}; \boldsymbol{\sigma}_{i,\mathrm{mod}(j+1,L)}],\end{aligned}$$

then our gated 2D RNN wave function ansatz is based on the following recursion relations:

$$\begin{aligned}\tilde{\boldsymbol{h}}_{i,j} &= \tanh\Big(W[\boldsymbol{\sigma}'_{i,j}; \boldsymbol{h}'_{i,j}] + \boldsymbol{b}\Big),\\ \boldsymbol{u}_{i,j} &= \mathrm{sigmoid}\Big(W_g[\boldsymbol{\sigma}'_{i,j}; \boldsymbol{h}'_{i,j}] + \boldsymbol{b}_g\Big),\\ \boldsymbol{h}_{i,j} &= \boldsymbol{u}_{i,j} \odot \tilde{\boldsymbol{h}}_{i,j} + (1 - \boldsymbol{u}_{i,j}) \odot (U\boldsymbol{h}'_{i,j}).\end{aligned}$$

A hidden state $\boldsymbol{h}_{i,j}$ can be obtained by combining a candidate state $\tilde{\boldsymbol{h}}_{i,j}$ and the neighbouring hidden states $\boldsymbol{h}_{i-1,j}, \boldsymbol{h}_{i,j-1}, \boldsymbol{h}_{\mathrm{mod}(i+1,L),j}, \boldsymbol{h}_{i,\mathrm{mod}(j+1,L)}$. The update gate $\boldsymbol{u}_{i,j}$ determines how much of the candidate hidden state $\tilde{\boldsymbol{h}}_{i,j}$ will be taken into account and how much of the neighboring states will be considered. With this combination, it is possible to circumvent some limitations of the vanishing gradient problems [132, 145]. The weight matrices W, W_g, U and the biases b, b_g are variational parameters of our RNN ansatz. Note that we choose the size of the hidden state $\boldsymbol{h}_{i,j}$, which we denote as d_h, before optimizing our ansatz's parameters.

4.15 Computational complexity of RNNs

In addition to their flexibility, RNNs are a very suitable class of ansätzes by virtue of their cheaper computational cost compared to other autoregressive models such as conventional transformers [53]. For RNNs defined in this chapter, the cost of computing a

hidden state through any recursion relation that involves local neighbors scales as $\mathcal{O}(d_h^2)$. Thus for a D-dimensional RNN, the cost of a forward pass of a configuration of N inputs scales as $\mathcal{O}(Nd_h^2)$. For a dilated RNN with a logarithmic number of layers, this cost scales as $\mathcal{O}(N\log(N)d_h^2)$. It is also worth noting that the cost of sampling a configuration is the same.

For an autoregressive feed-forward neural network [105], the cost of sampling and forward-pass scales quadratically with the system size N. For a regular Transformer [53], the cost of sampling scales as N^3. Furthermore, the memory footprint of the two previous models is quadratic in N as opposed to RNNs which have a lower memory cost[1]. These crucial differences make RNNs more suitable to reach larger systems and more economical for large-scale experiments with a modest GPU compute budget for each training iteration. Note that there are also other variations of transformers with linear scaling in the system size [161, 162]. There are also different versions for the transformer that rely on dividing the system into multiple chunks to reduce the computational complexity [163]. It would be interesting in future investigations to check how these alternatives compare to the different variations of the RNN both in terms of runtime and accuracy. A recent study [164] is a step forward in this direction that showed promising results in comparison to the 1D RNN.

4.16 Conclusion

In this chapter, we presented RNNs as efficient and flexible autoregressive models. We have shown how they can be defined as efficient ansatz wave functions and trial probability distributions to study physical systems in multiple spatial dimensions and with a wide range of connectivities. We have also shown different ways to boost their performances, namely through the use of symmetries, lattice structures, gating mechanisms, and dilated recurrent connections. We also showed the advantage of RNNs compared to other autoregressive models in terms of computational cost. In Chap. 5, we show the potential of these architectures in achieving accurate variational calculations on prototypical quantum systems in 1D, 2D and 3D. In Chap. 6, we illustrate how to use these architectures for solving combinatorial optimization problems by taking advantage of the physical principle of annealing. In Chap. 7, we show the potential of RNN wave functions in investigating topological properties of quantum many-body systems. Additionally, by virtue of the RNN's flexibility, we demonstrate their use in special lattices such as the Kagome lattice. Lastly in Chap. 8, we show theoretical constructions of traditional distributions using RNNs. Note

[1]Linear memory footprint except for dilated RNNs which have an additional logarithmic factor.

that Tab. 4.16 provides a summary of the desirable features of RNNs presented in this chapter.

RNN properties	Details
Sampling	Perfect autoregressive sampling (see Sec. 4.4)
Normalization	RNN probabilities and amplitudes are normalized to 1 (see Sec. 4.4)
Amplitudes	Positive and complex RNN wave functions (see Sec. 4.5)
RNN spatial dimension	Multiple spatial dimensions (see Sec. 4.8)
Types of RNN cells	Vanilla RNN (Sec. 4.4), Gated recurrent unit (Sec. 4.7), tensorized RNN (Sec. 4.9), and dilated RNNs (Sec. 4.10)
Estimating observables	See Sec. 3.7
Estimating entanglement properties	See Sec. 3.8
Discrete symmetry	Point group symmetries (see Sec. 4.11.1)
Continuous symmetry	U(1) symmetry and SU(2) symmetry (see Sec. 4.11.2)
Special lattices	See Sec. 4.12
Disorder / Translation invariance in the bulk	See Sec. 4.13
Boundary conditions	Open and periodic boundary conditions (see Sec. 4.14)
Complexity of a forward pass and a backward pass	$\mathcal{O}(Nd_h^2)$, except $\mathcal{O}(N\log(N)d_h^2)$ for dilated RNNs (see Sec. 4.15)

Table 4.1: A table summarizing the RNN properties that are discussed in this chapter. Here N is the system size and d_h is the hidden dimension (number of memory units).

Chapter 5

Benchmarking RNNs on prototypical many-body systems

This chapter contains results and material from Refs. [43, 106, 108], in addition to other material not published elsewhere.

In the previous chapter, we focused on the definition of RNN wave functions as an efficient and flexible ansätze that can as a trial wave function for a wide range of quantum many-body systems. We learned how to define a positive and a complex RNN wave function. Additionally, we provided different improvements that could be added to the RNN cells to boost their performance such as symmetry, gating mechanism, RNN tensorization procedure, and dilated recurrent connections. We also motivated their cheap computation budget compared to other autoregressive models. To investigate the efficiency of these wave functions, we put them to the test by targeting the ground state of prototypical quantum many-body systems. In Sec. 5.1.1, we demonstrate the potential of 1D positive RNN wave functions in finding the ground state of the 1D transverse-field Ising model (TFIM). In Sec. 5.1.2, we shift our attention to 1D J_1-J_2 model where we show the potential of complex RNN wave functions. We then demonstrate the value of 2D RNNs in Chap. 5.2 toward finding the ground state of the 2D TFIM, and the 2D Heisenberg model on the square lattice. We also target the excited state of the 2D J_1-J_2 model on a square lattice to showcase the possibility of computing low-energy excitation gaps, which are helpful to infer whether a ground state is gapless or not. Additionally, Sec. 5.4 shows the ability of RNNs to extract the critical phase transition point and critical exponents through a

finite-size scaling study of the 3D TFIM. Finally, in Sec. 5.5 we end this chapter with a benchmark study of the effect of different RNN hyperparameters on the 1D and 2D TFIM.

5.1 One-dimensional systems

5.1.1 1D transverse-field Ising model

In this section, we focus our attention on the ground state properties of the 1D transverse field Ising model (TFIM), with open boundary conditions (OBC), and that has the following Hamiltonian:

$$\hat{H}_{\text{TFIM}} = -\sum_{\langle i,j \rangle} \hat{\sigma}_i^z \hat{\sigma}_j^z - h \sum_i \hat{\sigma}_i^x, \tag{5.1}$$

where $\hat{\sigma}_i^{(x,y,z)}$ are Pauli matrices acting on site i. To demonstrate the power of our proposed method, we use it to target the ground state of a TFIM in 1D with $N = 1000$ spins at the critical point $h = 1$ using a pGRU wave function that has a single-layer GRU with 50 memory units (see Eqs. 4.17 in Sec. 4.7). In Fig. 5.1, we show the evolution of the relative error

$$\epsilon \equiv \frac{|E_{\text{RNN}} - E_{\text{DMRG}}|}{|E_{\text{DMRG}}|}, \tag{5.2}$$

and the energy variance per spin (see Eq. (3.12)) as a function of the training step. E_{DMRG} is the ground state energy as obtained from a density matrix renormalization group (DMRG) calculation [165, 166], and can be considered exact in 1D. We obtain very accurate results with a modest number of parameters ~ 8000. For comparison, the number of parameters of a restricted Boltzmann machine (RBM) [37] with one layer scales as MN with M the number of hidden units and N the number of physical spins. This scaling implies that the pRNN wave function here has the same number of variational parameters as an RBM with only 8 hidden units.

While energies and variances give a quantitative indication of the quality of a variational wave function, correlation functions provide a more comprehensive characterization. Indeed, correlation functions are at the heart of condensed matter theory since many experimental probes in condensed matter physics directly relate to measurements of correlation functions. Examples include inelastic scattering, which probes density-density correlation functions, and Green's functions, out of which important thermodynamic properties of a quantum system can be computed [167]. In Fig. 5.2 we compare the RNN results for the

two-point correlation functions $\langle \hat{S}^x_n \hat{S}^x_m \rangle$ and $\langle \hat{S}^z_n \hat{S}^z_m \rangle$ with DMRG. Here, we see consistency between the RNN and the DMRG results.

Extracting entanglement entropy (EE) from many-body quantum systems is a central theme in condensed matter physics, with EE providing an additional window into the structure of complex quantum states of matter beyond what is seen from correlation functions. Here, we use the *replica trick* [68] to calculate the $\alpha = 2$ Rényi entropy $S_2(\rho)$ for RNN wave functions as described in Sec. 3.8. In Fig. 5.3, we show results for the Rényi entropy $S_2(\rho_\ell)$ for two different system sizes $N = 20, 80$ of 1D TFIM. ρ_ℓ here is the reduced density matrix on the first ℓ sites of the spin chain, obtained by tracing out all sites $n \in [\ell + 1, L]$ such that

$$\rho_\ell = \mathrm{Tr}_{n \in [\ell+1, L]} \left(|\Psi\rangle \langle \Psi| \right). \tag{5.3}$$

Indeed for both system sizes, 5.3 shows excellent agreement between the pRNN wave function estimation and the DMRG result. To improve the overall quality of the quantum state, we have enforced the parity symmetry on our pRNN wave function (see Sec. 4.11.1), denoted by "Symmetric RNN" in Fig. 5.3. We observe that the symmetric pRNN wave function leads to a more accurate estimate of $S_2(\rho_\ell)$ for $N = 80$ sites.

For reproducibility purposes, we are providing the hyperparameters used to produce the results of this section are given in App. A.1.

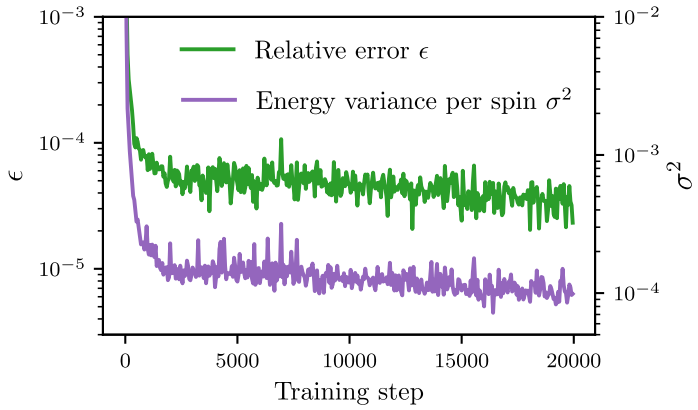

Figure 5.1: The relative error ϵ and the energy variance per spin σ^2 for $N = 1000$ spins. We use only 200 samples per gradient step, which are enough to achieve convergence.

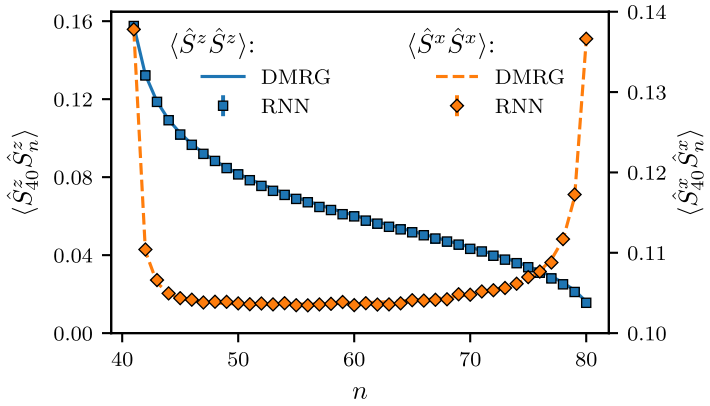

Figure 5.2: The two point correlation function $\langle \hat{S}_{40} \hat{S}_n \rangle$ along the x-axis and z-axis of the optimized pRNN wave function for sites $n > 40$ using 10^6 samples. DMRG results are also shown for comparison. The error bars are smaller than the data points.

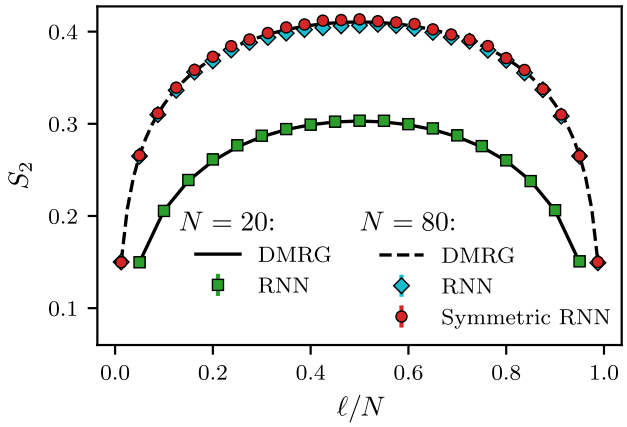

Figure 5.3: The Rényi entropy S_2 against the relative size of subregion A for system sizes $N = 20$ and $N = 80$. The error bars are smaller than the data points.

5.1.2 1D $J_1 - J_2$ model

Moving beyond stoquastic Hamiltonians, we now investigate the performance of RNN wave functions for a Hamiltonian whose ground state has a sign structure in the computational basis, specifically the J_1-J_2 model in one dimension given by

$$\hat{H}_{J_1 J_2} = J_1 \sum_{\langle i,j \rangle} \hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_j + J_2 \sum_{\langle\langle i,j \rangle\rangle} \hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_j. \tag{5.4}$$

where $\mathbf{S}_i$ is a spin-1/2 operator. Here, $\langle i,j \rangle$ and $\langle\langle i,j \rangle\rangle$ denote nearest and next-to-nearest neighbor pairs, respectively. Energies for the J_1-J_2 model are measured in units of $J_1 = 1$ in the results that follow.

We use a variationally optimized deep cRNN wave function with three GRU layers, each with 100 memory units, to approximate the ground state of the J_1-J_2 model. The phase diagram of this model has been studied with DMRG [168], where it was found that the model exhibits a quantum phase transition at $J_2^c = 0.241167 \pm 0.000005$ [169, 170] from a critical Luttinger liquid phase for $J_2 \leq J_2^c$ to a spontaneously dimerized gapped valence bond state phase for $J_2 \geq J_2^c$.

We impose $U(1)$ spin symmetry in the cRNN wave function (see Sec. 4.11.2), and target the ground state at four different points $J_2 = 0.0, 0.2, 0.5, 0.8$. Note that at $J_2 = 0$, the Hamiltonian Eq. (5.4) can be made stoquastic by a local unitary transformation that rotates every other spin by π around the z-axis. The ground state can in this case be decomposed as [153]

$$\psi(\boldsymbol{\sigma}) = (-1)^{M_A(\boldsymbol{\sigma})} \tilde{\psi}(\boldsymbol{\sigma}), \tag{5.5}$$

where $M_A(\boldsymbol{\sigma})$ is given by $M_A(\sigma) = \sum_{i \in A} \sigma_i$ with $\sigma_i \in \{0, 1\}$ [153] and $\tilde{\psi}(\boldsymbol{\sigma})$ is the *positive* amplitude of the wave function. The set A comprises the sites belonging to the sublattice of all even (or all odd) sites in the lattice. The prefactor $(-1)^{M_A(\boldsymbol{\sigma})}$ is known as the Marshall sign of the wave function [153]. For $J_2 \neq 0$, this decomposition is no longer exact, and $\tilde{\psi}(\boldsymbol{\sigma})$ acquires a non-trivial sign structure. For finite J_2 the decomposition in Eq. (5.5) can still be applied with the hope that the sign structure of $\psi(\boldsymbol{\sigma})$ remains close to the Marshall sign [171].

In Fig. 5.4, we compare ground state energies of the cRNN wave function trained on the 1D J_1-J_2 model with $N = 100$ spins with and without applying a Marshall sign. For small values of J_2, we find a considerable improvement of the energies when applying the Marshall sign on top of the cRNN wave function. This observation highlights the importance of considering a prior "sign ansatz" to achieve better results. In the absence of a prior sign, the cRNN wave function can still achieve accurate estimations of the ground state energies,

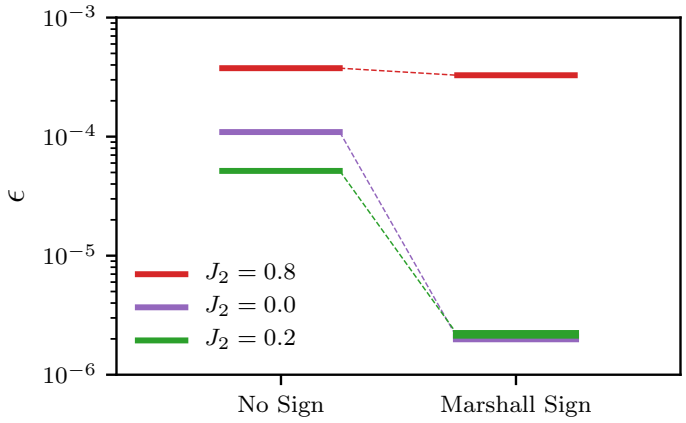

Figure 5.4: The relative error (compared to DMRG) of the cRNN wave function trained on the 1D $J_1 - J_2$ model with $N = 100$ spins for different values J_2, both without a prior sign (represented by "No Sign") and with a prior Marshall sign as in Eq. (5.5) (represented by "Marshall Sign"). We observe that applying a Marshall sign improves accuracy.

showing that cRNN wave functions can recover some of the unknown sign structure of the ground state. For $J_2 = 0.8$, however, the improvement is less pronounced, which is expected due to the emergence of a second sign structure in the limit $J_2 \to \infty$ (when the system decouples into two independent unfrustrated Heisenberg chains) [127, 172], that is widely different from the Marshall sign in Eq. (5.5). We omit from Fig. 5.4 our results at the point $J_2 = 0.5$. In this case, the 1D J_1-J_2 model reduces to the Majumdar-Ghosh model, where the ground state is a product-state of spin singlets, and we find agreement with the exact ground state energy within numerical precision when we apply an initial Marshall sign structure. The hyperparameters used to obtain our results are summarized in App. A.1. We also provide a summary of the cRNN wave function's obtained values in App. A.2.

5.2 Two-dimensional systems

5.2.1 2D transverse-field Ising model

Understanding strongly correlated quantum many-body systems in $D > 1$ spatial dimensions is one of the central problems in condensed matter physics. During the last decade, numerical approaches such as tensor networks [139, 173, 174], quantum Monte

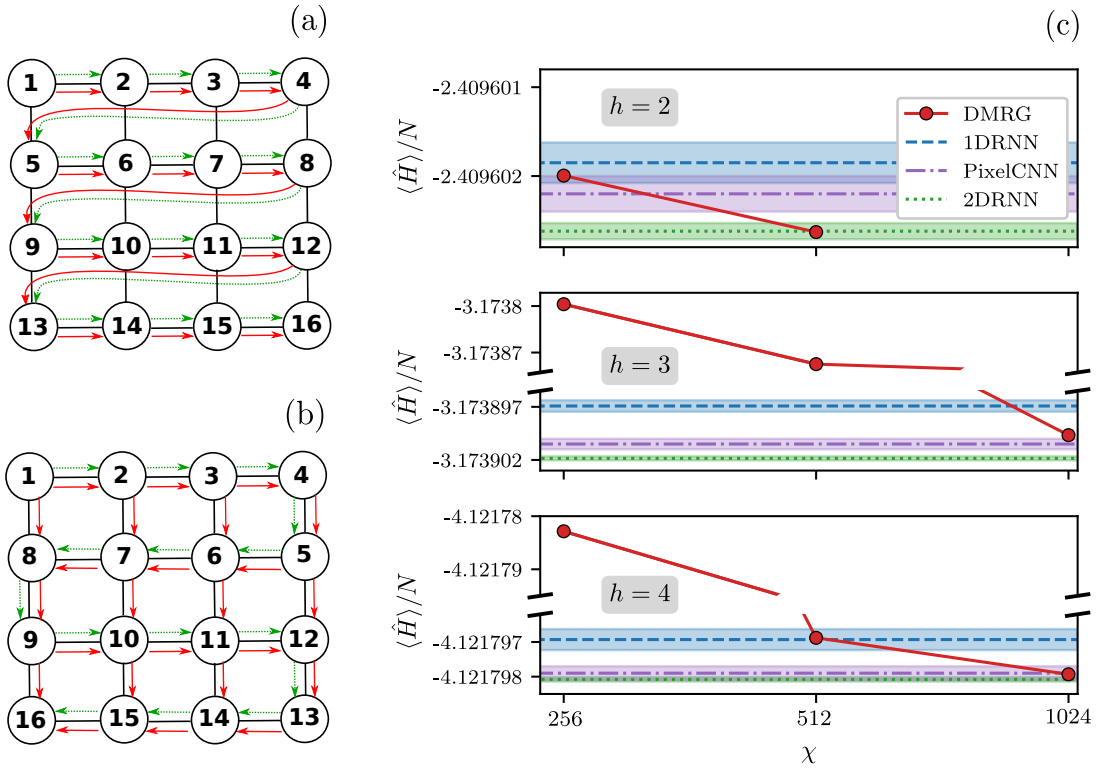

Figure 5.5: **(a)**: Autoregressive sampling path of 2D spin configurations using 1D RNN wave functions. The 2D configurations are generated using a snake path such that in order to generate spin σ_i one has to condition on the spins that are previously generated. **(b)**: Autoregressive sampling path of 2D spin configurations using 2D RNN wave functions through a zigzag path, where each site receives two hidden states from the horizontal and the vertical neighbors that were previously generated. For both Figs. **(a)** and **(b)**, the digits and the green dashed arrows indicate the sampling path, while the red arrows indicate how the hidden states are passed from one site to another. **(c)**: A comparison of the variational energy per spin between a 2D pRNN wave function (labeled as 2DRNN), 1D pRNN wave function (labeled as 1DRNN), PixelCNN wave function [150], and DMRG with bond dimension χ for the 2D TFIM on a system with $L_x \times L_y = 12 \times 12$ spins. The shaded regions represent the error bars of each method. Note the broken y-axis on the plots for $h = 3$ and $h = 4$, denoting a change in scale between the upper and lower portions of the plots. These results show that 2D pRNN wave functions can achieve a performance comparable to PixelCNN wave functions and DMRG with a large bond dimension. Note that the lower the variational energy, the more the estimation of the ground state energy is accurate.

Carlo [2, 175], and neural networks [37] have moved to the forefront of research in this area. Despite tremendous progress, however, solving correlated quantum many-body systems even in 2D remains a challenging problem. We now turn our attention to the application of our RNN wave function approach to the 2D quantum Ising model shown in Eq. (5.1) on a square lattice, a paradigmatic example of a strongly correlated quantum many-body system. This model has a quantum phase transition at a critical magnetic field $h^c \approx 3.044$ that separates a magnetically ordered phase from a paramagnet [176].

The simplest strategy for extending our approach to 2D geometries is to simply treat them as folded 1D chains, similar to the "snaking" approach used in 2D DMRG calculations (see Fig. 5.5(a)). While this approach works quite well, it has the fundamental drawback that neighboring sites on the lattice can become separated in the 1D geometry. As a consequence, local correlations in the 2D lattice are mapped into non-local correlations in the 1D geometry, which can increase the complexity of the problem considerably. For example, 2D DMRG calculations are typically restricted to 2D lattices with small width L_y. This problem has led to the development of more powerful tensor network algorithms for 2D quantum systems such as projected entangled pair states (PEPS) [139].

An advantage of RNN wave functions is their flexibility in how hidden vectors are passed between units. To obtain an RNN wave function more suited to a 2D geometry, we modify the simple 1D approach outlined above by allowing hidden vectors to also be passed vertically, instead of only horizontally. This modification is illustrated by the red arrows in Fig. 5.5(b). We refer to this geometry in the following discussions as a 2D RNN. We optimize the 2D pRNN wave function with a single-layer 2D vanilla RNN cell that has 100 memory units (i.e. with ~ 21000 variational parameters) to approximate the ground state of the 2D quantum Ising model at $h = 2, 3, 4$. The training complexity of the 2D pRNN wave function is only quadratic in the number of memory units d_h (see Sec. 4.8), which is very inexpensive compared to, e.g., the expensive variational optimization of PEPS, which scales as $\chi^2 \tilde{D}^6$ (where $\tilde{D}$ the PEPS bond dimension and χ is the bond dimension of the intermediate MPS) [140].

For comparison, we also optimize a deep 1D pRNN wave function architecture with three layers of stacked GRU cells (see Eqs. 4.17), each with 100 memory units (i.e., with $\sim$152000 variational parameters) for the same values of the magnetic field h. In Fig. 5.5(c) we compare the obtained ground state energies with results from 2D DMRG calculations (run on the same 1D geometry as for the 1D pRNN wave function) and the PixelCNN architecture [177] (with $\sim$800000 variational parameters and results are taken from Ref. [150]. For the magnetic fields shown above and for large bond dimensions, we obtain excellent agreement between all four methods. This agreement is particularly remarkable given that the 2D pRNN wave function uses only about 0.03% of the variational parameters of the

DMRG calculation with bond dimension $\chi = 512$, about 2.6% of the variational parameters of the PixelCNN wave function used in Ref. [150], and about 14% of the parameters used in the 1D pRNN architecture. The hyperparameters used in this section are provided in Appendix A.1. A summary of our results in tabular form can be found in App. A.2.

5.2.2 2D Heisenberg model on the square lattice

We shift our attention to the task of finding the ground state of the two-dimensional anti-ferromagnetic Heisenberg model on the square lattice with open boundary conditions (OBC). The Hamiltonian is given as follows:

$$\hat{H} = \frac{1}{4} \sum_{\langle i,j \rangle} (\hat{\sigma}_i^x \hat{\sigma}_j^x + \hat{\sigma}_i^y \hat{\sigma}_j^y + \hat{\sigma}_i^z \hat{\sigma}_j^z), \tag{5.6}$$

where the sum is over nearest neighbours and $\hat{\sigma}_i^{x,y,z}$ are Pauli matrices. In the square lattice case, $\hat{H}$ has a C_{4v} symmetry[1]. We also remark that the ground state has zero magnetization [154], due to the $U(1)$ symmetry of the Hamiltonian $\hat{H}$. We show that enforcing these symmetries in our ansatz allows for obtaining better accuracy without adding more parameters. More details about how to apply symmetries to the 2D cRNN ansatz can be found in Sec. 4.11.

Additionally, the Hamiltonian $\hat{H}$ for this model can be made stoquastic on the square lattice after applying a Marshall sign transformation [153, 178]. In this case, a 2D pRNN wave function is enough to approximate the ground state. In the following experiments, we use a 2D cRNN wave function to demonstrate the generality of the cRNN ansatz in recovering a constant sign structure. We also use the 2D tensorized RNN version with a gating mechanism as described in Sec. 4.9 and we denote our ansatz as a 2D complex Tensorized GRU (cTGRU) wave function.

First, we train our 2D cTGRU wave function to find the ground state of the 6×6 square lattice with and without applying the symmetries of the Hamiltonian $\hat{H}$. We find in Fig. 5.6(a) that increasing the number of symmetries encoded in the cRNN leads to a more accurate estimation of the ground state energy.

We also use our 2D cTGRU wave function to estimate the ground state of the 10×10 square lattice after applying $U(1)$ and C_{4v} symmetries. The results are shown in Fig. 5.6(b), where we compare our estimates with projected-pair entangled states (PEPS) [179], Pixel CNN wave functions [150], DMRG [146], as well as Quantum Monte Carlo (QMC) [179].

[1] A point group with four rotations and four mirror reflections (vertical, horizontal, and diagonal).

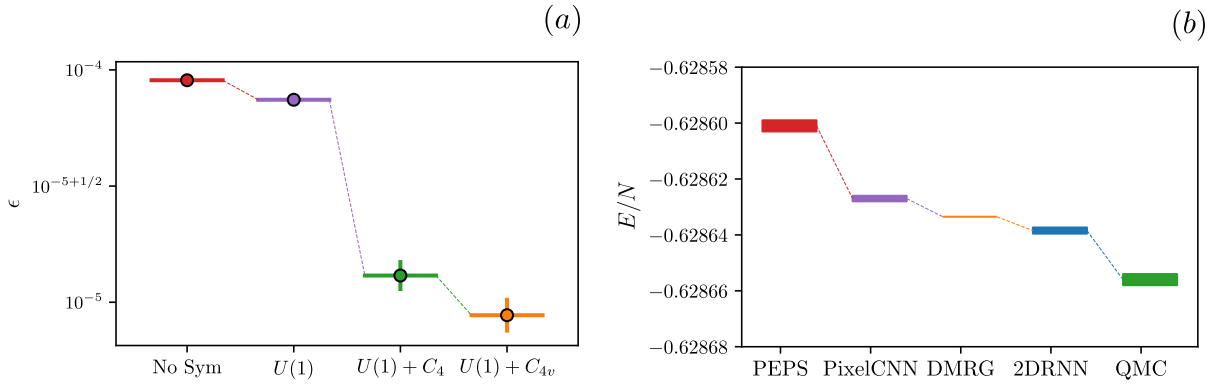

Figure 5.6: (a) A plot of the relative error ϵ after applying different symmetries of the Heisenberg model on the square lattice with size 6×6. The relative error is computed with respect to the DMRG energy [35]. C_4 is the point group of four rotations. (b) A comparison of the energy per site obtained with our 2DRNN ansatz and PEPS [179], PixelCNN [150], DMRG [146] and QMC [179] on the Heisenberg model on the square lattice with size 10×10.

These results show that our ansatz is competitive with PEPS, PixelCNN, and DMRG. We also find that QMC is the best compared to the other variational methods. The hyperparameters used to obtain our results can be found in App. A.1. Furthermore, a comparison with Ref. [146] along with the energy values can be found in App. A.2. In Sec. 6.7, we demonstrate how our 2D RNN ansatz performs on this model embedded on a triangular lattice, after adding the annealing ingredient which turns out to be very useful in overcoming local minima in the VMC optimization landscape (see Chap. 6).

5.3 2D J_1-J_2 model on square lattice

We now focus our attention on the 2D J_1-J_2 model in Eq. (5.4) on the square lattice where open boundary conditions are assumed. Here, we set $J_1 = 1$ for numerical convenience. We note that for $J_2 = 0$, this model corresponds to the Heisenberg model on the square lattice that we studied in the previous section. Since this model is frustrated, this model cannot be made stoquastic with a simple Marshal sign rule [153]. This observation justifies the use of a complex RNN wave function to study this model.

The focus of this section is to estimate the spectral gap of this model at $J_2 = 0.5$ at different system sizes $L \times L$. As outlined in Sec. 3.9, the gap can be computed by running

two different variational calculations. The goal of the first calculation is to estimate the ground state energy and the second one is to target the low-energy excited states of interest.

By virtue of the $U(1)$ of this model, the energy eigenstates have a well-defined magnetization, i.e., a well-defined total spin S_z along the z axis. It is also important to note that the eigenstates are irreducible representations of the symmetry point groups of this model on the square lattice. Thus we can make use of group characters to target specific low-energy excited states [104]. The ground state is known to lie in the singlet sector $S = 0$ [154]. This implies that the ground state has zero magnetization $S_z = 0$. Using the $U(1)$ symmetry construction of the RNN wave function in Sec. 4.11.2, we take advantage of this property and we generalize the construction of $S_z = 0$ to different non-zero S_z to target excited states as motivated in Sec. 3.9. This idea is helpful to construct an RNN wave function ansatz that is orthogonal by construction to the ground state or to other excited state sectors without access to these states, as opposed to the traditional Lagrange multiplier approach highlighted in Sec. 3.9.

Regarding the architecture used in this study, we use a 2D cTGRU with $d_h = 100$ memory units to target the ground state and low-energy excited states[2]. Additionally, we apply the C_4 point group symmetry of the square lattice on our RNN wave function as described in Sec. 4.11.1. This point group has two different group characters A and B. In this study, we target excited states with the A group character.

In Fig. 5.7, we plot two gap excitations versus the inverse length $1/L$ for three different lengths $L = 6, 8, 10$. The first gap corresponds to the difference between the ground state energy and the lowest-excited state energy with $S_z = 1$, and the second gap is the difference between the $S_z = 2$ sector energy and the estimated ground state energy. It is clear from the plot that both gaps follow a power law $\propto 1/L$, where the extrapolation at the thermodynamic limits corresponds to gapless excitations.

A natural future direction is to explore the scaling of the gap for different values of J_2 and to compare the RNN findings with other numerical methods. Another interesting future research direction is to target also singlet excited states which lie in the B group character representation. Importantly, targeting the B group character turns out to be more challenging numerically, as it results in training instabilities, likely because of the interference effects in the symmetrization of the phase provided in Eq. (4.26), which do not exist when targeting group character A states. We empirically found that training instabilities are due to the gradients explosion during the training, and we mitigate their

[2]Here we make use of the principle of annealing by targeting what we expect to be slightly easier point $J_2 = 0.4$, and then gradually increasing the coupling J_2, until we obtain an estimate of the targeted state energy at $J_2 = 0.5$ (see Chap. 6 for more details about annealing).

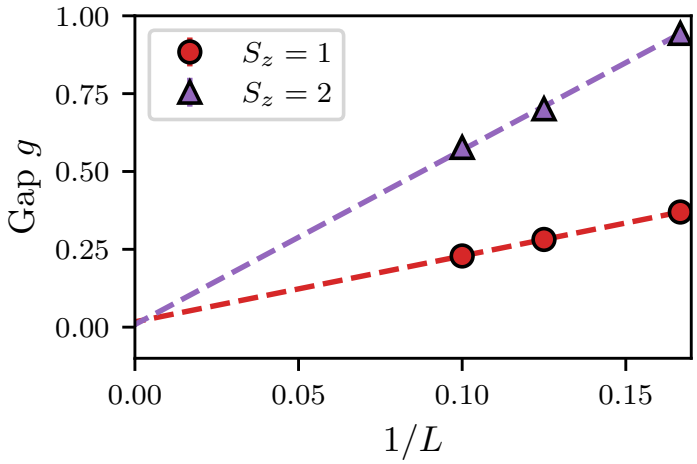

Figure 5.7: A plot of the excitation gaps of the 2D J_1-J_2 model at $J_2 = 0.5$ (in units of J_1) for different system sizes $L = 6, 8, 10$. By extrapolating to the infinite system size limit, we infer that the two gaps are vanishing in the thermodynamic limit.

effects using gradient clipping techniques [130, 180].

5.4 Three-dimensional systems

In this section, we shift our focus to the study of 3D quantum many-body systems using 3D RNNs (see Sec. 4.8). Instead of estimating ground/excited state energies, in this section, we aim to showcase the ability of RNNs in handling phase transitions using a finite size study as motivated in Sec. 2.2.4. The system we target is the 3D TFIM (see Eq. 5.1) with open boundary conditions. Similarly to the 1D and the 2D cases, this model has a second-order phase transition from the ferromagnetic phase to the paramagnetic phase at the estimated critical transverse field $h_c \approx 5.2$ (in units of J), predicted by quantum Monte Carlo (QMC) techniques [181]. By virtue of the quantum-classical correspondence, the critical exponents correspond to the finite temperature transition of the 4D Ising model, which are the mean-field exponents predicted by the phenomenological Landau theory [65].

Here we aim to estimate the critical transverse-field h_c and some of the critical exponents using a 3D pGRU wave function with $d_h = 40$ memory units through a finite-size scaling study with $L = 6, 8, 10, 12$. Here it is important to that note that we reach an order of $N = L^3 \sim 1000$ spins with our 3D RNN ansatz. The extraction of the transition data

can be done by running different variational ground state optimizations at different values of the transverse-field h_c, and then by estimating relevant observables. In particular, we focus on the Binder cumulant B (see Eq. (2.11)) and the expected absolute magnetization per site $\langle|m|\rangle \equiv \langle|\hat{\sigma}^z|\rangle$.

In Fig. 5.8(a), we plot the Binder cumulant for different magnetic fields h, where B tends to zero for large h, whereas for small h we observe a saturation to $2/3$ as expected in the paramagnetic/ferromagnetic phases respectively. The intersection point of the different Binder cumulants allows locating the transition close $h_c \approx 5.1$ by eye inspection. For a more accurate and quantitative estimate, we make use of the property (2.12), where the correct values of the critical field and the critical exponent ν collapse the different finite-size Binder curves. To extract these values, we use Bayesian inference [182], where we apply a feed-forward neural network to extract h_c and ν [183] using the open-source code in Ref. [184] to find the best possible collapse. We run this Bayesian optimization for 500 independent runs to extract error bars. We find that $h_c \approx 5.0708 \pm 0.0001$ which is close to the value predicted by QMC, we also obtain $\nu = 0.5598 \pm 0.0001$ that is also close to the mean-field exponent $\nu = 0.5$. The finite-size curve collapse can be seen clearly in Fig. 5.8(b).

We do a similar Bayesian optimization study on the absolute magnetization $\langle|m|\rangle$ shown in Fig. 5.8(c), where the decay of the magnetization towards close-to-zero values with increasing h is consistent with a transition from a ferromagnetic phase to a paramagnetic phase. With the expected finite-size scaling of $\langle|m|\rangle$ provided in Eq. (2.7), we can extract the critical exponents ν, β as well as the critical magnetic field h_c using the same Bayesian approach highlighted earlier. The optimization resulted in the following estimates $h_c \approx 5.060 \pm 0.001$, $\beta \approx 0.557 \pm 0.002$, and $\nu \approx 0.479 \pm 0.001$ which are consistent with the expected values of h_c and the mean-field critical exponents. These estimates allow for obtaining a very good collapse as shown in Fig. 5.8(d). Deviations from the exact values could be either related to the finite size effects due to open boundary conditions or instead to variational inaccuracies in our calculations.

Note that we can plot $\langle|m|\rangle L^{\beta/\nu}$ versus h to remove the finite size dependence on $\langle|m|\rangle$. With this plot, we can make a similar conclusion to Fig. 5.8(a), where the intersection point of the different curves is a good indicator of the critical point.

Finally, we would like to highlight that in order to extract the critical exponent δ, we need a reliable measure of the quantum magnetic susceptibility $\chi = \frac{\partial\langle|m|\rangle}{\partial h_z}$ with respect to a magnetic field h_z in the z-direction. Possible solutions include a finite difference method to compute derivation which is expected to be noisy. The second possible approach consists of using implicit differentiation as illustrated in Ref. [185]. Another future direction is to

account for the expected logarithmic corrections in 3+1 dimensions, which are very hard to extract within the system sizes explored in our study [186–189].

5.5 Benchmarking RNN hyperparameters

The optimization results of our RNN wave function approach depend on several hyperparameters, including the number of memory units, the number of recurrent layers in deep architectures, and the number of samples used to obtain the gradient during an optimization step (see Chap. 3). Here, we investigate how the energy variance per spin σ^2 (see Eq. (3.12)) depends on these parameters. As shown in Sec. 3.4, the energy variance per spin is an indicator of the quality of the optimized wave function, with exact eigenstates corresponding to $\sigma^2 = 0$.

Since the number of variational parameters is directly related to the number of memory units of the pGRU wave function (see Eq. (4.17) in Chap. 4.7), we study here the scaling of σ^2 with the number of memory units. In Fig. 5.9, we present the dependence of σ^2 on the number of memory units for the 1D and 2D critical TFIMs. Fig. 5.9(a) shows results of σ^2 for a 1D critical TFIM on three system sizes $N = 20, 40$ and 80, and Fig. 5.9(b) for the 2D TFIM on $4 \times 4, 5 \times 5$ and 6×6 square lattices. In all cases, we used a single-layer 1D pGRU wave function and 500 samples during optimization to compute estimates of the gradients. For each system size, we observe a systematic decrease of σ^2 (i.e., an increase in the quality of the wave function) as we increase the number of memory units.

In App. A.3.1, we study the dependence of σ^2 on both the number of samples and the number of layers in the pGRU wave function for a critical 1D TFIM. We observe only a weak dependence on both parameters. The weak dependence on the number of samples suggests that optimizing the RNN wave functions with noisy gradients does not significantly impact the results of the optimization procedure, and yields accurate estimations of the ground state and its energy. From the weak dependence on the number of layers, we conclude that deep architectures do not seem to be beneficial from an accuracy point of view. However, deeper networks could have potential ramifications regarding memory usage and training speed when it comes to training a large number of variational parameters, as shallow RNNs with a large number of memory units are equivalent in terms of the number of parameters to deep RNNs with a smaller number of memory units. We also note that adding residual connections between layers [190] and dilated connections between RNN cells (see Sec. 4.10) to deep RNNs changes our previous conclusions and make deep RNNs more beneficial compared to shallow RNNs. This point is further motivated when we use dilated RNN to study fully-connected spin-glass models in Chap. 6.

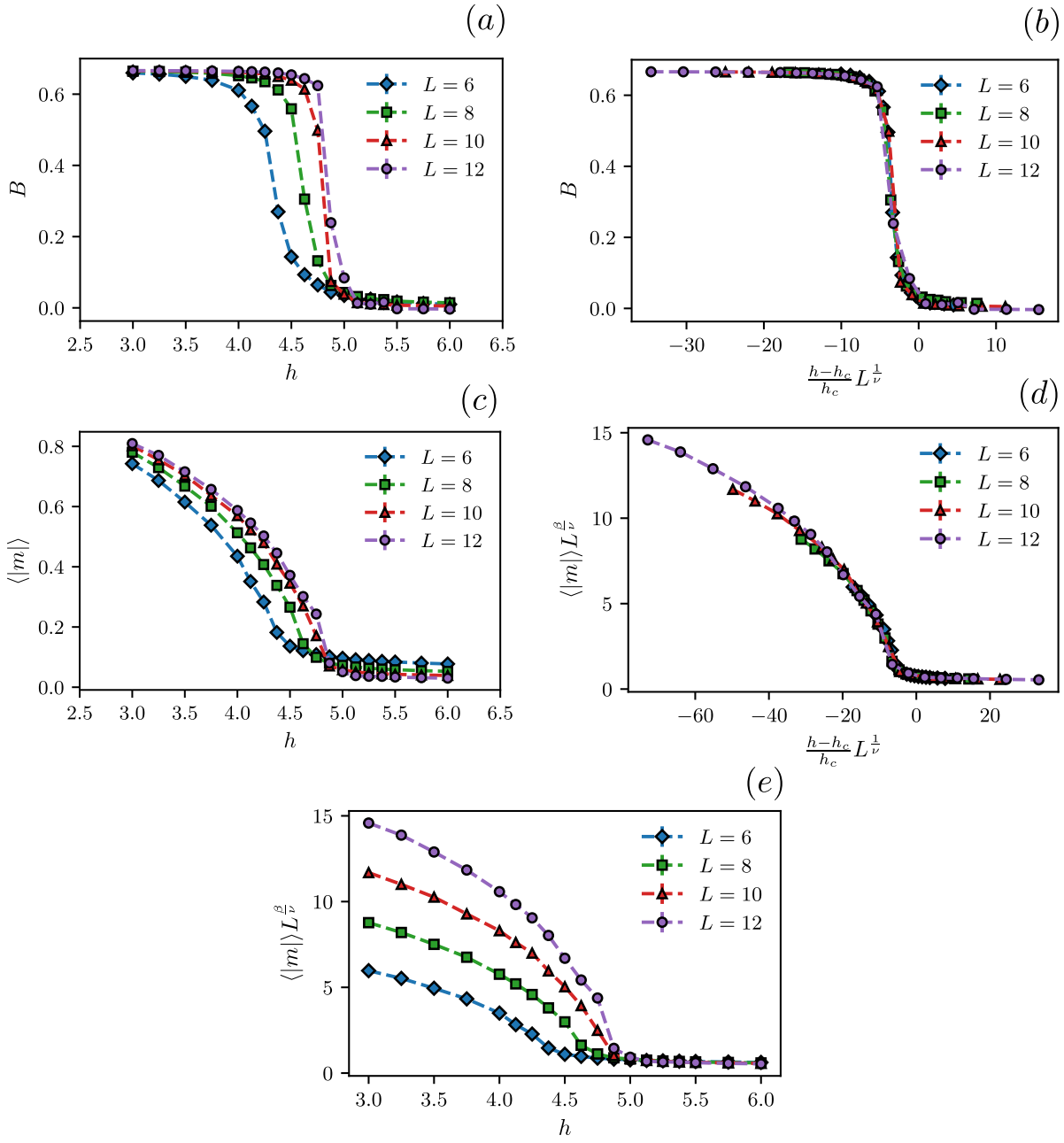

Figure 5.8: Plots of the Binder cumulant B and the absolute magnetization per site $\langle |m| \rangle$ for different values of the magnetic field h. We plot scaled variants of these quantities in order to extract the critical point and the critical exponents. The finite-size scaling study is conducted for different lengths $L = 6, 8, 10, 12$ of the 3D lattice.

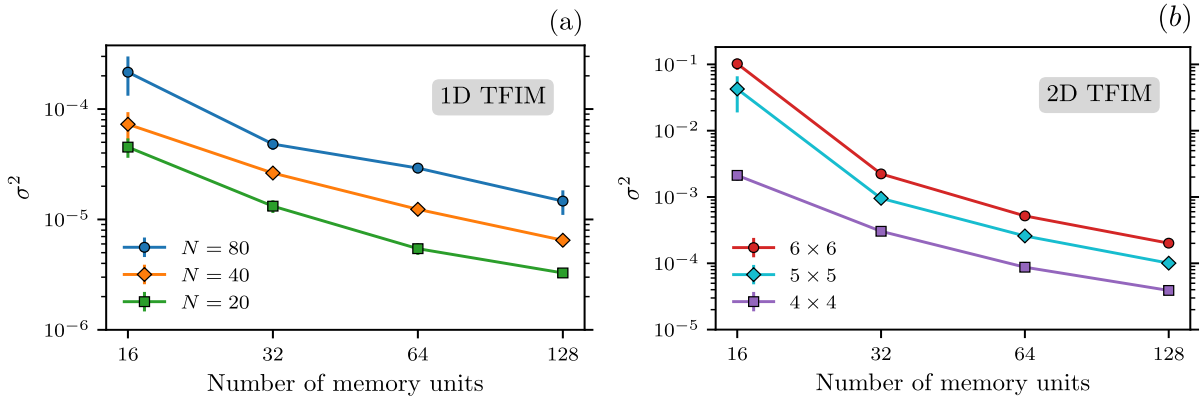

Figure 5.9: The energy variance per spin against the number of memory units of a 1D pRNN wave function trained at the critical point of (a) the 1D TFIM and (b) the 2D TFIM. Both scalings show that we can systematically reduce the bias in the estimation of the ground-state energy.

In App. A.3.2, we study the effect of using RNN cell designs on the accuracy of a variational calculation. In particular, we find that tensorized RNNs are more advantageous compared to vanilla RNNs on the task of finding the ground of the 1D TFIM. We also find that adding a gating mechanism to RNNs allows for a faster and more accurate convergence to the ground state of the 2D Heisenberg model. Furthermore, we found that for disordered systems we obtain better accuracy by abandoning the common practice of using weight sharing (see Sec. 4.13). Finally, we observed that dilated RNNs (see Sec. 4.10) are more accurate compared to a one-layered RNN for a many-body system with all-to-all connectivity.

5.6 Conclusion and Outlooks

In this chapter, we benchmarked RNN wave functions on the task of approximating ground state energies, correlation functions, and entanglement entropies of many-body Hamiltonians of interest to condensed matter physics. We find that RNN wave functions are competitive with state-of-the-art methods such as DMRG and PixelCNN wave functions [150], performing particularly well on the task of finding the ground state of the transverse-field Ising model and the Heisenberg model on square lattices. We have shown furthermore that we can accurately model ground states endowed with a sign structure

using a complex Recurrent Neural Network (cRNN) wave function ansatz. Here, accuracy can be improved by introducing an ansatz sign structure and by enforcing symmetries such as $U(1)$ symmetry and point group symmetries. By increasing the number of memory of units in the RNN, the error in our results can be systematically reduced. The autoregressive nature of RNN wave functions makes it possible to directly generate uncorrelated samples, in contrast to methods based on Markov chain sampling, which are often plagued by long autocorrelation times that affect the optimization and the accurate estimation of correlation functions in a variational ansatz. Thanks to weight sharing among lattice sites, RNN wave functions provide very compact yet expressive representations of quantum states, while retaining the ability to easily train with millions of variational parameters, as opposed to, e.g., restricted Boltzmann machines [37].

In the next chapter, we extend the use of RNNs to the task of finding the ground state of combinatorial optimization problems by harnessing the concept of annealing. We show that we can obtain competitive results compared to traditional optimization algorithms. We further demonstrate that RNNs supplemented with annealing are more equipped compared to traditional RNNs for the study of frustrated systems. In Chap. 7, we also demonstrate the potential of RNNs in detecting topological phases of matter of prototypical models, and in investigating the existence of these phases in real-world quantum systems.

Reproducibility Code

The code we use to produce our results can be found in Ref. [191].

Chapter 6

Variational Neural Annealing

This chapter contains results and material from Refs. [106, 108, 192] and results not published elsewhere.

Combinatorial optimization is widely used in many areas of science such as physics, computer science, and biology. It also has a wide range of applications in the industry including without limitation to supply chain, energy, and transportation. Providing an efficient solution to combinatorial optimization problems can boost scientific progress and provide optimal solutions to a plethora of industry problems. Unfortunately, typical optimization problems are NP-Hard and are challenging to solve with deterministic algorithms in a polynomial time [193].

Various heuristics have been used over the years to find approximate solutions to these NP-hard problems. A notable example is simulated annealing (SA) [194], which mirrors the analogous annealing process in materials science and metallurgy where a solid is heated and then slowly cooled down to its lowest energy and most structurally stable crystal arrangement. In addition to providing a fundamental connection between the thermodynamic behavior of real physical systems and complex optimization problems, simulated annealing has enabled scientific and technological advances with far-reaching implications in areas as diverse as operations research [195], artificial intelligence [196], biology [197], graph theory [198], power systems [199], quantum control [200], circuit design [201] among many others [196]. The paradigm of annealing has been so successful that it has inspired intense research into its quantum extension, which requires quantum hardware to anneal the tunneling amplitude, and can be simulated in an analogous way to SA [202, 203].

The SA algorithm explores an optimization problem's energy landscape via a gradual decrease in thermal fluctuations generated by the Metropolis-Hastings algorithm. The procedure stops when all thermal kinetics are removed from the system, at which point the solution to the optimization problem is expected to be found. While an exact solution to the optimization problem is always attained if the decrease in temperature is arbitrarily slow, a practical implementation of the algorithm must necessarily run on a finite time scale [204]. As a consequence, the annealing algorithm samples a series of effective, quasi-equilibrium distributions close but not exactly equal to the stationary Boltzmann distributions targeted during the annealing [205]. This naturally leads to approximate solutions to the optimization problem, whose quality depends on the interplay between the problem complexity and the rate at which the temperature is decreased.

In this chapter, we offer a promising route to solving optimization problems, called *variational neural annealing*. Here the conventional simulated annealing formulation is substituted with the annealing of a parameterized model. Namely, instead of annealing and approximately sampling the exact Boltzmann distribution, this approach anneals a quasi-equilibrium model, which must be sufficiently expressive and capable of tractable sampling. Fortunately, suitable models have recently been developed [53, 206, 207]. In particular, autoregressive models combined with variational principles have been shown to accurately describe the equilibrium properties of classical and quantum systems [35, 43, 105, 150]. Here, we implement variational neural annealing using RNNs and show that they offer a powerful alternative to conventional SA and its analogous quantum extension, i.e., simulated quantum annealing (SQA) [202].

This chapter is organized as follows: in Sec. 6.1, we talk about a variational emulation of classical annealing, and in Sec. 6.2, we present a quantum version of the variational annealing scheme. We also illustrate the promise of our approach for random Ising chains (6.3), non-stoquastic driving Hamiltonians (6.4), spin-glass models (6.5), real-world combinatorial optimization problems (6.6), as well as for frustrated quantum systems (6.7).

6.1 Variational Classical Annealing

A wide array of complex combinatorial optimization problems can be reformulated as finding the lowest energy configuration of an Ising Hamiltonian of the form [208]:

$$H_{\text{target}} = -\sum_{i<j} J_{ij}\sigma_i\sigma_j - \sum_{i=1}^{N} h_i\sigma_i, \tag{6.1}$$

where $\sigma_i = \pm 1$ are spin variables defined on the N nodes of a graph. The topology of the graph together with the couplings J_{ij} and fields h_i uniquely encode the optimization problem, and the solution to the problem corresponds to the spin configurations σ_i that minimize H_{target}.

We first consider the variational approach to statistical mechanics [105, 209], where a distribution $P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})$ defined by a set of variational parameters $\boldsymbol{\lambda}$ is optimized to reproduce the equilibrium properties of a system with a Hamiltonian H_{target} at temperature T. We dub our first variational neural annealing algorithm *variational classical annealing* (VCA).

The VCA algorithm searches for the ground state of an optimization problem, encoded in a target Hamiltonian H_{target}, by slowly annealing the model's variational free energy

$$F_{\boldsymbol{\lambda}}(t) = \langle H_{\text{target}} \rangle_{\boldsymbol{\lambda}} - T(t) S_{\text{classical}}(P_{\boldsymbol{\lambda}}), \tag{6.2}$$

from a high temperature to a low temperature. The quantity $F_{\boldsymbol{\lambda}}(t)$ provides an upper bound to the true instantaneous free energy and can be used at each annealing stage to update $\boldsymbol{\lambda}$ through gradient-descent techniques. The brackets $\langle ... \rangle_{\boldsymbol{\lambda}}$ denote ensemble averages over $P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})$. The Shannon entropy is given by

$$S_{\text{classical}}(P_{\boldsymbol{\lambda}}) = -\sum_{\boldsymbol{\sigma}} P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}) \log \left(P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}) \right), \tag{6.3}$$

where the sum runs over all possible configurations $\{\boldsymbol{\sigma}\}$. In our setting, the temperature is decreased linearly from T_0 to 0, i.e., $T(t) = T_0(1-t)$, where $t \in [0, 1]$, which follows the traditional implementation of SA.

In order for VCA to succeed, we require parameterized models without a slowdown of their sampling via Markov chain Monte Carlo. Such a slowdown is likely to occur in spin-glass models with a rugged landscape and very small transition probabilities between different modes. These issues preclude un-normalized models such as restricted Boltzmann machines, where sampling relies on Markov chains [206][1]. Instead, we implement VCA using recurrent neural networks (RNNs) [35, 43] as a model for $P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})$, whose autoregressive nature enables statistical averages over exact samples $\boldsymbol{\sigma}$ drawn from the RNN. Since RNNs are normalized by construction, these samples allow the estimation of the entropy in Eq. (6.3). On a technical note, it is sufficient to use the module squared of a pRNN wave function to construct the probability distribution $P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})$. We provide a detailed description of the RNN in Chap. 4 and illustrate the advantage of autoregressive sampling at recovering different modes in App. B.5.

[1] Ref. [107] shows that for an un-normalized model, the estimation of the gradients of the entropy is possible without complete knowledge of the partition function. However, these models might still suffer from the Markov chain Monte Carlo sampling limitations.

The VCA algorithm, summarized in Fig. 6.1(a), performs a warm-up step which brings a randomly initialized distribution $P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})$ to an approximate equilibrium state with free energy $F_{\boldsymbol{\lambda}}(t = 0)$ via N_{warmup} gradient descent steps. At each step t, we reduce the temperature of the system from $T(t)$ to $T(t + \delta t)$, while keeping the model's parameters fixed, and apply N_{train} gradient descent steps to re-equilibrate the model at $T(t + \delta t)$. A critical ingredient to the success of VCA is that the variational parameters optimized at temperature $T(t)$ are reused at temperature $T(t+\delta t)$ to ensure that the model's distribution is near its instantaneous equilibrium state. Repeating the last two steps $N_{\text{annealing}}$ times, we reach temperature $T(1) = 0$, which is the end of the protocol. Here the distribution $P_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})$ is expected to assign a high probability to configurations $\boldsymbol{\sigma}$ that solve the optimization problem. Likewise, the residual entropy Eq. (6.3) at $T(1) = 0$ provides an approach to count the number of solutions to the problem Hamiltonian [105]. Further details about our optimization scheme are provided in Sec. 3.10.

6.2 Variational Quantum Annealing

In quantum annealing [210–213], the search for the ground state of an optimization problem is generally done by promoting the target Hamiltonian, in Eq. (6.1), to a quantum Hamiltonian

$$\hat{H}(t) = \hat{H}_{\text{target}} + f(t)\hat{H}_D, \tag{6.4}$$

where quantum fluctuations are introduced via a driving term $\hat{H}_D$ that does not commute with the target Hamiltonian $\hat{H}_{\text{target}}$. The factor $f(t)$ is a user-defined time-dependent schedule function chosen such that $f(0) = 1$ and $f(1) = 0$. Quantum annealing starts with a dominant driving term $\hat{H}_D \gg \hat{H}_{\text{target}}$ chosen so that the ground state of $\hat{H}(t = 0)$ is easy to prepare. The strength of the driving term is then subsequently reduced –typically adiabatically– using the schedule function f so that at the end of annealing, the system is in the lowest state of the target Hamiltonian. We choose a linear schedule function $f(t) = 1 - t$ with $t \in [0, 1]$.

Here, we leverage the variational principle of quantum mechanics and devise a strategy to simulate quantum annealing that we dub *variational quantum annealing* (VQA). Our framework is based on variational Monte Carlo (VMC), a quantum Monte Carlo method that simulates equilibrium properties of quantum many-body systems at zero-temperature (see Chap. 3). In VMC, the ground-state wave function of a Hamiltonian $\hat{H}$ is modeled via an ansatz $|\Psi_{\boldsymbol{\lambda}}\rangle$ where $\boldsymbol{\lambda}$ are the variational parameters. The variational principle guarantees that the expectation value of the energy over the variational state $\langle\Psi_{\boldsymbol{\lambda}}|\hat{H}|\Psi_{\boldsymbol{\lambda}}\rangle$

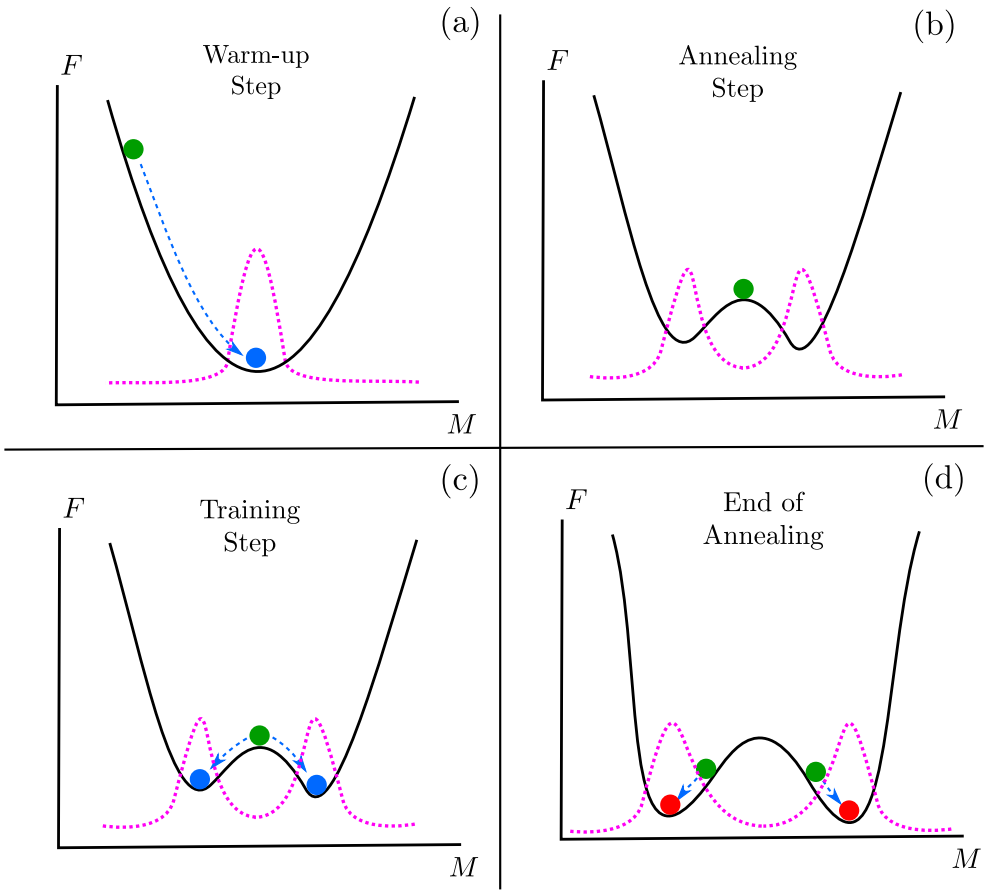

Figure 6.1: An illustration of the variational classical annealing protocol. (a) Initially, we take a warm-up step to bring the initialized variational state (green dot) close to the minimum of the free energy (blue dot) at a given value of the order parameter M of our system of interest. (b) Next, we perform an annealing step by changing the time t by an amount of δt, which, as a consequence, changes the free energy landscape. (c) Here, we perform a training step to bring the variational state back to the new free energy minima. (d) Finally, by repeating the last two steps, we arrive at $t = 1$, where we expect to obtain the minima of the target Hamiltonian H_{target} if the protocol is conducted slowly enough. This figure also represents the corresponding probability distribution that minimizes the free energy, as illustrated by the dashed curves. This illustration corresponds to a continuous phase transition and can also be generalized to first-order transitions.

is an upper bound to the ground state energy of $\hat{H}$. Thus, we use it as a cost function to optimize the parameters $\boldsymbol{\lambda}$. In a similar spirit to VCA, we define a time-dependent cost function as $E(\boldsymbol{\lambda}, t) \equiv \langle \hat{H}(t) \rangle_{\boldsymbol{\lambda}} = \langle \Psi_{\boldsymbol{\lambda}} | \hat{H}(t) | \Psi_{\boldsymbol{\lambda}} \rangle$.

Our VQA setup is implemented via the protocol described in Fig. 6.2. We start by randomly initializing the parameters $\boldsymbol{\lambda}$. Then, we perform a warm-up step to prepare our ansatz close to the ground state of the Hamiltonian $\hat{H}(0)$, as illustrated in Fig. 6.2(a). To do so, we apply N_{warmup} gradient descent steps to minimize the expectation value $E(\boldsymbol{\lambda}, t)$ at a fixed time $t = 0$. The variational energy after this step is $E(\boldsymbol{\lambda}_0, t = 0)$. Next, we set $t = \delta t$, while keeping the parameters $\boldsymbol{\lambda}_0$ of the variational wave function fixed. The variational energy is $E(\boldsymbol{\lambda}_0, t = \delta t)$ as shown in Fig. 6.2(b) (green dot). Next, we take N_{train} gradient descent steps to bring the ansatz closer to the new instantaneous ground state. At the end of the training step, we obtain the energy $E(\boldsymbol{\lambda}_1, t = \delta t)$ as illustrated in Fig. 6.2(c) (cyan dot). Like in VCA, the variational parameters optimized at time step t are used as input at time $t + \delta t$, which ensures that the parameterized wave function is near the instantaneous ground state of $\hat{H}(t)$. This step promotes the adiabaticity of the dynamics induced by the algorithm (see App. B.1). Finally, we repeat the annealing and training steps $N_{\text{annealing}}$ times until $t = 1$, where the system is expected to converge to the ground state of the optimization problem (red dot in Fig. 6.2(d)). Analogously to VCA, we choose RNN wave functions [35, 43] as ansätze to implement the VQA protocol. For the sake of clarity, we provide a flowchart in Fig. 6.3 that illustrates the VCA and the VQA frameworks.

The success of the algorithm, whose ultimate goal is to set the variational wave function as close as possible to the ground state of the target Hamiltonian $\hat{H}(1) = \hat{H}_{\text{target}}$, relies on several key elements. First, the variational annealing evolution should be adiabatic, which is achieved by requiring the annealing time update δt to be small. Additionally, we require an expressive variational wave function capable of accurately capturing all the instantaneous ground states of $\hat{H}(t)$. Assuming that the variational state can be optimized via gradient descent, we also require that the N_{train} number of optimization steps be sufficiently large. To have a theoretical insight on these principles for VQA, we derive a variational version of the adiabatic theorem [211, 214, 215]. We start from a set of assumptions, such as the convexity of the energy landscape of $E(\boldsymbol{\lambda}, t)$ in the warm-up phase and close to convergence during annealing, and the absence of noise in the gradients. This enables us to provide a bound on the total number of gradient descent steps N_{steps}, that is sufficient for the VQA algorithm to remain adiabatic with a success probability of obtaining the ground state $P_{\text{success}} > 1 - \epsilon$. Here, ϵ is an upper bound on the overlap between the variational wave function and the excited states of the Hamiltonian $\hat{H}(t)$, i.e.

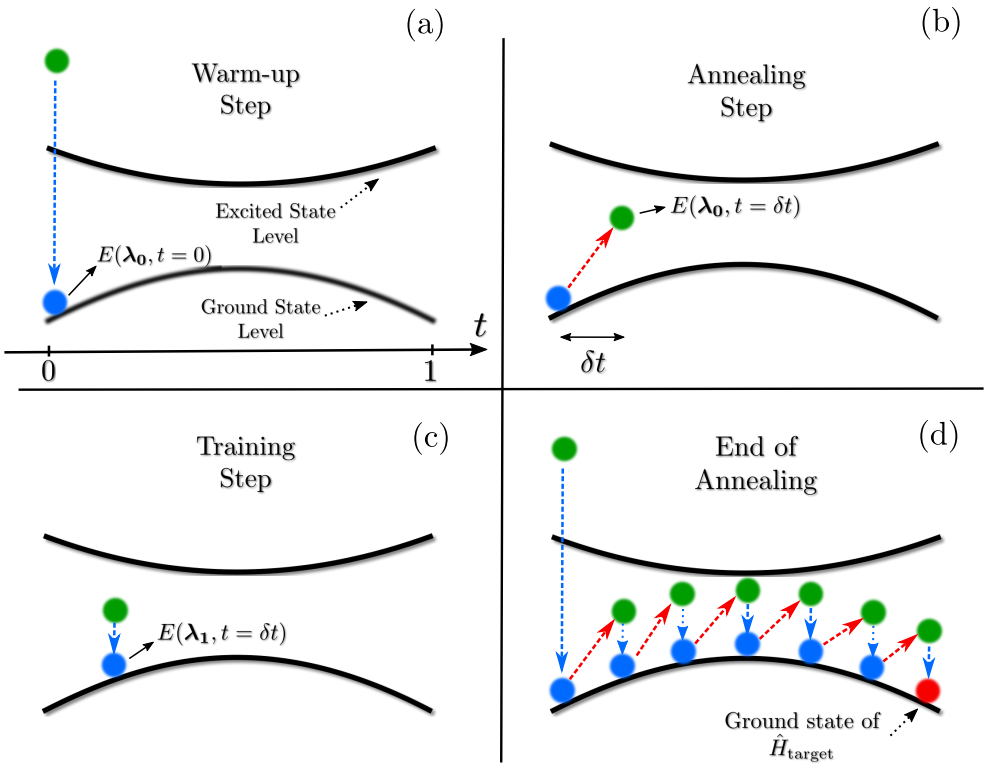

Figure 6.2: Illustration of the variational quantum annealing (VQA). (a) A warm-up step involves taking N_{warmup} gradient descent step to bring the variational wave function (green dot) close to the initial ground state and obtain an estimate $E(\boldsymbol{\lambda_0}, t = 0)$ (cyan dot) of the ground state energy at $t = 0$. (b) Next, we perform an annealing step by changing the time t by an amount of δt while keeping the ansatz's parameters fixed. (c) Next, we perform N_{train} gradient descent steps to bring the variational wave function (green dot) closer to the new ground state energy (cyan dot). (d) We loop over the previous two steps until reaching the target ground state of $\hat{H}_{\text{target}}$ if annealing is performed slowly enough.

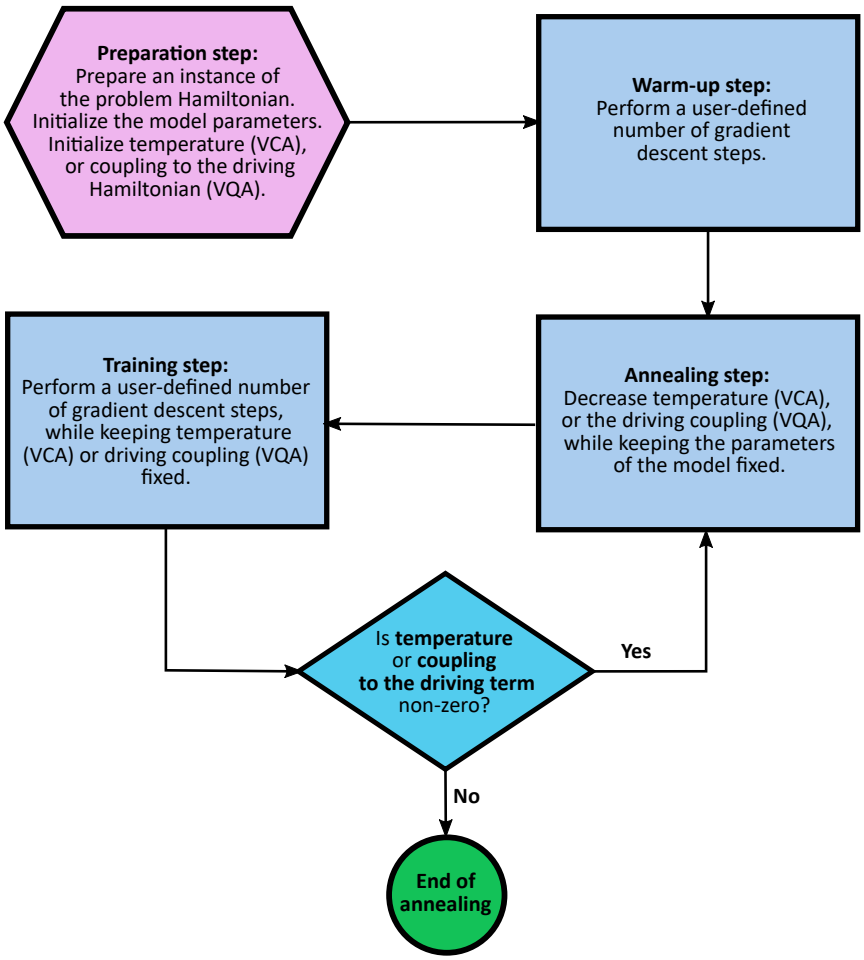

Figure 6.3: A flowchart describing our VCA and VQA implementations.

$|\langle\Psi_\perp(t)|\Psi_{\boldsymbol{\lambda}}\rangle|^2 < \epsilon$. We show in App. B.2 that N_{steps} can be bounded as:

$$\mathcal{O}\left(\frac{\text{poly}(N)}{\epsilon \min\limits_{\{t_n\}}(g(t_n))}\right) \leq N_{\text{steps}} \leq \mathcal{O}\left(\frac{\text{poly}(N)}{\epsilon^2 \min\limits_{\{t_n\}}(g(t_n))^2}\right). \tag{6.5}$$

Here, $g(t)$ is the energy gap between the first excited state and the ground state of the instantaneous Hamiltonian $\hat{H}(t)$ at time t, N is the system size, and the set of times $\{t_n\}$ is defined in App. B.2. Typically, for hard optimization problems, the minimum gap decreases exponentially with system size N, which dominates the computational complexity of the VQA simulations. In the case of a minimum gap that scales as the inverse of a polynomial in N, then the number of steps N_{steps} is bounded by a polynomial in N.

6.3 Application to random Ising chains

As a first benchmark, we consider the one-dimensional Ising model with random couplings $J_{i,i+1}$, whose Hamiltonian is defined as:

$$\hat{H}_{\text{target}} = -\sum_{i=1}^{N-1} J_{i,i+1}\hat{\sigma}_i^z\hat{\sigma}_{i+1}^z, \tag{6.6}$$

where $\sigma_i^{x,y,z}$ are Pauli matrices acting on site i. First, we examine $J_{i,i+1}$ sampled from a uniform distribution in the interval $[0, 1)$. Here, the ground state configuration is given either by all spins up or all spins down, which implies that the ground state energy is $E_G = -\sum_{i=1}^{N-1} J_{i,i+1}$ [216].

To account for the randomness of the model, we use a tensorized RNN without weight sharing for both VCA and VQA (see Secs. 4.13 and 4.9). We consider system sizes $N = 32, 64, 128$ and train the RNNs for $N_{\text{train}} = 5$ per annealing step, which we found to be sufficient to achieve accurate solutions. To quantify the performance of the algorithms, we use the residual energy [202, 217–220],

$$\epsilon_{\text{res}} = \left[\langle\hat{H}_{\text{target}}\rangle_{\text{stat}} - E_{\text{G}}\right]_{\text{dis}}, \tag{6.7}$$

where E_{G} is the exact ground state energy of each instance of $\hat{H}_{\text{target}}$. While we use the arithmetic mean for statistical averages $\langle ...\rangle_{\text{stat}}$, we consider the typical (geometric) mean for averaging over instances of the target Hamiltonian such that, $\left[...\right]_{\text{dis}}^{\text{typ}} = \exp(\left[\ln(...)\right]_{\text{dis}})$.

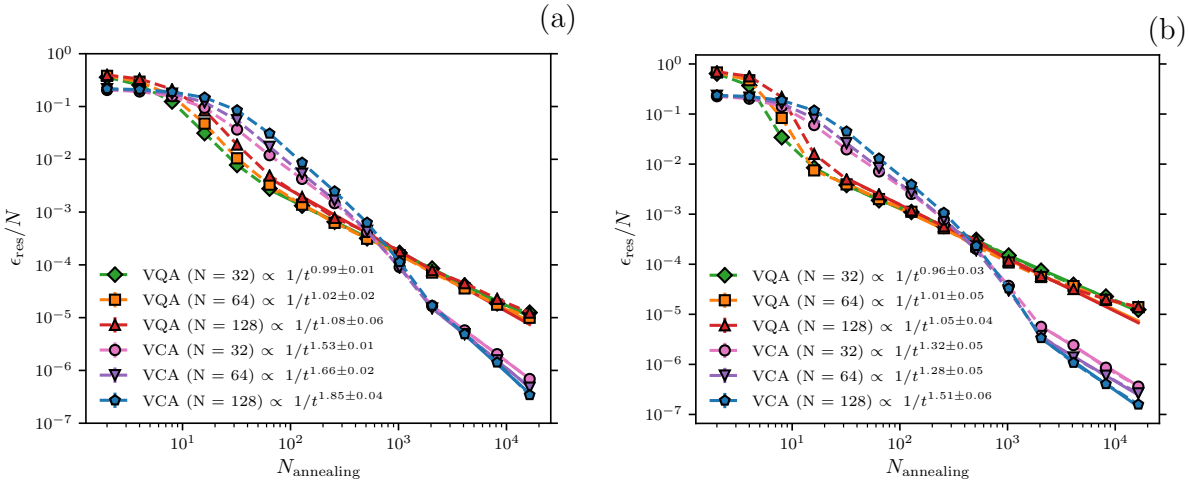

Figure 6.4: The residual energy per site $\epsilon_{\rm res}/N$ vs the number of annealing steps $N_{\rm annealing}$ for both VQA and VCA on the one-dimensional (1D) Ising chain. The system sizes are $N = 32, 64, 128$. (a) random positive couplings $J_{i,i+1} \in [0, 1)$ and (b) random discrete couplings $J_{i,i+1} \in \{-1, 1\}$. For VQA, we use a stoquastic one-body driving term $\hat{H}_D = -\Gamma_0 \sum_{i=1}^{N} \hat{\sigma}_i^x$. The error bars represent the one s.d. statistical uncertainty calculated over different disorder realizations [221].

We take advantage of the autoregressive nature of the RNN and sample 10^6 configurations at the end of the annealing, which allows us to accurately estimate the arithmetic means and post-select low-energy solutions to the optimization problem. The typical mean is taken over 25 instances of $\hat{H}_{\rm target}$.

In Fig. 6.4(a) we report the residual energies per site against the number of annealing steps $N_{\rm annealing}$. As expected for both VQA and VCA, the residual energy is a decreasing function of $N_{\rm annealing}$. The latter observation underlines the importance of adiabaticity in our variational setups. In Fig. 6.4(b) we report similar observations for $J_{i,i+1}$ uniformly sampled from the discrete set $\{-1, +1\}$, where the ground state configuration is disordered and the ground state energy is given by $E_G = -\sum_{i=1}^{N-1} |J_{i,i+1}| = -(N-1)$. To further illustrate the adiabaticity of VCA and VQA, we provide additional benchmarks on small system sizes in App. B.1.

In all of our examples and system sizes, we observe that for sufficiently large annealing steps the decrease of the residual energy of VCA and VQA is consistent with a power law at large annealing steps. The exponent of VCA varies in the interval $1.3-1.9$, whereas VQA's

exponent is about $0.9-1.1$. Both exponent ranges suggest a speed-up compared to SA and coherent quantum annealing, where the residual energies follow a logarithmic law [222–225]. We highlight that contrary to results obtained in Ref. [225] where QA was found superior to SA, our variational emulation of classical annealing outperforms its zero-temperature quantum analog. At long annealing times, VCA finds solutions an order of magnitude more accurate than VQA on average. For VCA and VQA, we note that the variational protocol powered by a stochastic gradient descent appears to surpass the conjecture of thermal annealing dynamics which is expected to follow the Huse-Fisher logarithm scaling law [222, 226], as well as quantum annealing dynamics, which is expected to be hampered by the wide distribution of exponentially closing gap close to criticality [223, 224].

As a final note, the exponents provided above are not expected to be universal and are a priori sensitive to the hyperparameters of the algorithms (e.g., learning rate, number of memory units d_h, number of training steps N_{train}, gradient descent optimizer, number of samples, etc), which may open up avenues to boost the performance of our algorithms. For reproducibility purposes, App. B.7 provides a summary of the hyperparameters used to produce the results shown here.

6.4 Non-stoquastic drivers

Non-stoquastic drivers are efficient at removing problematic first-order quantum phase transitions on certain types of problem Hamiltonians [227]. However, their implementation using typical quantum Monte Carlo methods is hampered by the sign problem. Here, we take advantage of the intrinsic sign-problem-free nature of VMC to set up a VQA scheme that accommodates non-stoquastic driving terms.

As proof of principle, we use a tensorized complex RNN wave function (see Chap. 4) to approximate the anticipated sign structure of the ground states induced by non-stoquastic Hamiltonians. Here, we investigate the random Ising chain with a discrete disorder in the presence of a two-body non-stoquastic driving term as follows

$$\begin{aligned}\hat{H}(t) = &-\sum_{i=1}^{N-1} J_{i,i+1}\hat{\sigma}_i^z\hat{\sigma}_{i+1}^z \\ &\pm \lambda_0(1-t)\sum_{i=1}^{N-1}\hat{\sigma}_i^a\hat{\sigma}_{i+1}^a,\end{aligned} \tag{6.8}$$

where $a = x$ or y. Here, the plus sign corresponds to the non-stoquastic case (for details about the preparation of the initial ground state during the warm-up phase, see App. B.4).

For comparison, we also consider a stoquastic driving term that corresponds to a minus sign in the previous equation. For both cases, the initial value of the driving's coupling strength is $\lambda_0 = 2$. We consider 25 instances of $\hat{H}_{\text{target}}$. The results in Fig. 6.5 confirm that the

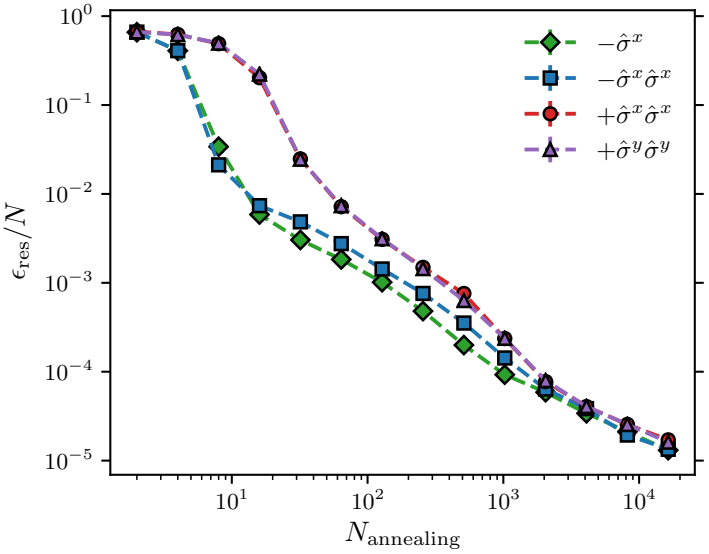

Figure 6.5: The residual energy per site ϵ_{res}/N vs. the number of VQA annealing steps $N_{\text{annealing}}$ on a 1D Ising chain with $N = 64$ spins and random couplings $J_{i,i+1} \in \{-1, 1\}$. Here, we consider four different driving terms. The first two are stoquastic single- and two-body driving terms, which we can simulate using a tensorized positive RNN wave function, and the third and the fourth ones are non-stoquastic two-body driving terms we emulate using a tensorized complex RNN wave function.

residual energy can be systematically reduced by taking more annealing steps. We observe that all the driving terms result in the same asymptotic behavior of the residual energy for a large number of annealing steps. Overall, the single-body driving term performs better than the ones encoding multispin flip dynamics. We note that the stoquastic XX driver performs better than its non-stoquastic version, corroborating the recent finding [228] that de-signing Hamiltonians could be more efficient for quantum annealing. We equally report here the very first simulations of non-stoquastic YY drivers for a system size that is far above exact diagonalization capabilities. This is a very important milestone given that such drivers are currently in experimentation on the newest DWave architectures [229], albeit on very few qubits. Note that here, the XX and YY non-stoquastic drivers give similar results because of the rotational symmetry of the Hamiltonian used. In the absence of

such symmetry, spin reversal transformations are capable of providing some advantage for the YY term, as recently pointed out in Ref. [230]. Most importantly, these results show that VQA setup is capable of emulating a non-stoquastic term despite the presence of a sign structure, and though for this case study it does not show an advantage for non-stoquasticiy, it does provide a platform where it could be studied at moderately large system sizes. We note that in our example, the sign structure corresponds to the Marshall sign [153, 172, 178]. To further demonstrate the possibility of simulating VQA for other Hamiltonians and sign structures, we use a driving term with frustration and show that a tensorized RNN can emulate a non-stoquastic driving term with an unknown sign structure as described in App. B.4.

6.5 Application to spin-glass models

6.5.1 Edwards-Anderson model

The two-dimensional Edwards-Anderson (EA) model is a prototypical spin-glass model where a set of spins are arranged on a square lattice with only nearest neighbor random interactions. The problem of finding ground states of the model has been studied experimentally [203] and numerically [202, 218, 231] from the annealing perspective, as well as theoretically [193] from the computational complexity perspective. In this section, we use the EA model as a benchmark to probe VCA and VQA and compare them against standard heuristics, namely, SA and SQA implemented via discrete-time path-integral Monte Carlo [202, 218]. The EA model is given by

$$\hat{H}_{\mathrm{EA}} = -\sum_{\langle i,j\rangle} J_{ij}\hat{\sigma}_i^z\hat{\sigma}_j^z, \tag{6.9}$$

where the sum runs over nearest neighbors, and the couplings J_{ij} are drawn independently from a uniform distribution in the range $[-1, 1]$. In the absence of a longitudinal field for which solving the EA model is NP-hard, the ground state can be found in polynomial time [193]. For each random realization of the couplings J_{ij}, we use the spin-glass server [232] to obtain the exact ground state energy. This feature makes the EA model a good benchmark for our method, particularly for large system sizes.

To simulate our variational neural annealing protocols, we use a 2D tensorized RNN (see Sec. 4.9) as an ansatz without weight sharing. We implement the methods described in Sec. 6.1 and 6.2 with VQA implemented using a one-body driving term. Fig. 6.6(a)

shows the annealing results obtained on a system size $N = 10 \times 10$ spins. Similarly to the results obtained for the random Ising chains in Sec. 6.3, VCA outperforms VQA and in the adiabatic, long-time annealing regime, VCA produces solutions three orders of magnitude more accurate than VQA. In addition, we investigate the performance of VQA supplemented with a fictitious Shannon information entropy term that induces a thermal-like exploration of the energy landscape during the quantum annealing emulation. The entropy-enhanced protocol, here termed regularized variational quantum annealing (RVQA), is described by a free energy cost function:

$$\tilde{F}_{\boldsymbol{\lambda}}(t) = \langle \hat{H}(t) \rangle_{\boldsymbol{\lambda}} - T(t) S_{\text{classical}}(|\Psi_{\boldsymbol{\lambda}}(t)|^2). \tag{6.10}$$

We emphasize that $S_{\text{classical}}$ corresponds to a pseudo-entropy [35] computed in the computational basis, which is different from the von Neumann entropy of a quantum state at finite temperature. The latter means that $T(t)$ does not act as a physical temperature, but rather as a pseudo-temperature that induces thermal-like fluctuations whose aim is to mimic thermal relaxation effects observed in quantum annealing hardware [233]. Results in Fig. 6.6(a) do show an amelioration of VQA performance, from saturating dynamics at long annealing time to a power-law-like behavior. However, though introducing a pseudo-temperature to avoid subsequent local minima in VQA seems to provide an added advantage, it appears to be insufficient to compete with the VCA scaling. This suggests the superiority of a thermally driven variational emulation of annealing over a pure quantum emulation.

To further scrutinize the relevance of the annealing effects in our variational methods, we now consider a variational method devoid of it. We do this by optimizing the variational parameters through direct minimization of the target Hamiltonian expectation value over the variational ansatz. This idea is known as classical-quantum optimization (CQO) [234–236], and for our setup, it corresponds to implementing VCA with zero thermal fluctuations, i.e., setting $T_0 = 0$. Fig. 6.6(a) shows that CQO takes about 10^3 training steps starting from random parameters initialization to reach close to 1% accuracy. However, the accuracy does not further improve when trained up to 10^5 gradient steps, showing that CQO is prone to get stuck in local minima. In comparison, VCA and VQA offer solutions with orders of magnitude more accurate at long annealing times, thus suggesting the importance of the annealing effect in tackling challenging optimization problems.

Since VCA displays the best performance in the previous benchmarks, we use it to demonstrate its capabilities on a relatively large system with 40×40 spins. For comparison, we use SA as well as SQA with $P = 20$ trotter slices, and take the average energy across all trotter slices, for each realization of randomness (see App. B.3). In addition, we average the energy obtained after 25 annealing runs on every instance of randomness for SA and SQA. To average over Hamiltonian instances, we use the typical mean over 25 different

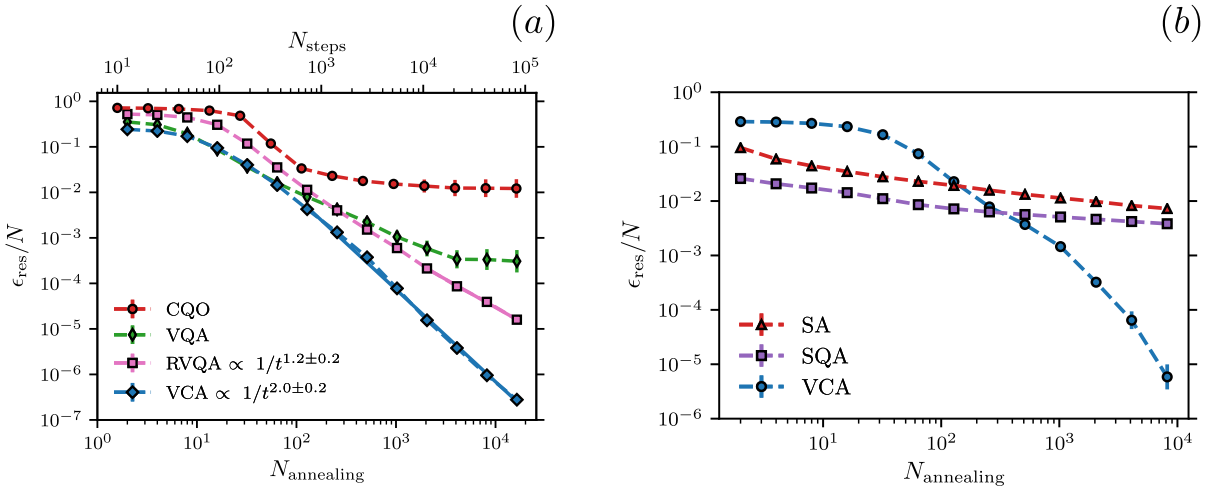

Figure 6.6: (a) A comparison between VCA, VQA, RVQA, and CQO for Edwards-Anderson (EA) on a 10×10 lattice. The residual energy per site vs. $N_{\text{annealing}}$ for VCA, VQA and RVQA. For CQO, we report the residual energy per site vs. the number of optimization steps N_{steps}. (b) Comparison between Simulated Annealing (SA), Path-Integral Quantum Monte Carlo (SQA) with $P = 20$ trotter slices, and VCA using a 2D tensorized pRNN state for the EA model on a 40×40 lattice. We report the residual energy per site as a function of the number of annealing steps $N_{\text{annealing}}$ for SA, VCA, and SQA.

realizations for the three annealing methods. The results are shown in Fig. 6.6(b), where we present the residual energies per site against the number of annealing steps $N_{\text{annealing}}$, which is set so that the speed of annealing is the same for SA, SQA, and VCA. We first note that our results confirm the qualitative behavior of SA and SQA in Refs. [202, 218]. While at short annealing times, SA and SQA produce lower residual energy solutions than VCA, we observe that VCA achieves residual energies for a large annealing time about three orders of magnitude smaller than SQA and SA. Notably, the rate at which the residual energy improves with increasing the annealing time is significantly higher in VCA than SQA and SA even at relatively short annealing. These observations highlight the advantages of solving hard optimization problems in a variational space compared to SA and SQA paradigms. Additional simulations on a system size of 60×60 spins (see App. B.5) corroborate this result.

6.5.2 Sherrington-Kirkpatrick model

We now focus our attention on fully-connected spin glasses [193, 237]. We first consider the Sherrington-Kirkpatrick (SK) model [238], which provides a conceptual framework for the understanding of the role of disorder and frustration in systems ranging from materials to combinatorial optimization and machine learning. The SK Hamiltonian is given by

$$H_{\text{target}} = -\frac{1}{2}\sum_{i\neq j}\frac{J_{ij}}{\sqrt{N}}\sigma_i\sigma_j, \tag{6.11}$$

where $\{J_{ij}\}$ is a symmetric matrix whose elements J_{ij} are sampled from the standard normal distribution.

Since VCA performed best in our previous examples, we use it to find ground states of the SK model for $N = 100$ spins. Here, exact ground states energies of the SK model are calculated using the spin-glass server [232] on a total of 25 instances of disorder. To account for long-distance dependencies between spins in the SK model, we use a dilated RNN ansatz of depth $L = \lceil \log_2(N) \rceil$ structured so that spins are connected to each other with a distance of at most $\mathcal{O}(\log_2(N))$ (see Sec. 4.10). The initial temperature is set to $T_0 = 2$. We compare our results with SQA and SA initialized with $\Gamma_0 = 2$ and $T_0 = 2$, respectively.

For an effective comparison, we first plot the residual energy per site as a function of $N_{\text{annealing}}$ for VCA, SA, and SQA ($P = 100$). Here, the SA and SQA residual energies are obtained by averaging the outcome of 50 independent annealing runs, while for VCA we average the outcome of 10^6 samples from the annealed RNN. For all methods, we consider typical averages over 25 disorder instances. The results are shown in Fig. 6.7(a). As observed in the EA model, we note that SA and SQA produce lower residual energy solutions than VCA for small $N_{\text{annealing}}$, but we emphasize that VCA delivers a lower ϵ_{res} when $N_{\text{annealing}} \gtrapprox 10^3$. Likewise, we observe that the rate at which the residual energy improves with increasing $N_{\text{annealing}}$ is significantly higher for VCA than SQA and SA.

A closer look at the statistical behavior of the methods at large $N_{\text{annealing}}$ can be obtained from the residual energy histograms produced by each method, as shown in Fig. 6.7(d). The histograms contain 1000 residual energies for each of the same 25 disorder realizations. For each instance, we plot results for 1000 SA runs, 1000 samples obtained from the RNN at the end of annealing for VCA, and 10 SQA runs including contribution from each of the $P = 100$ Trotter slices. We observe that VCA is superior to SA and SQA, as it produces a higher density of low-energy configurations. This indicates that, even though VCA typically takes more annealing steps, it results in a higher chance of getting accurate solutions to optimization problems than SA and SQA.

6.5.3 Wishart-Planted Ensemble

We now focus on the Wishart planted ensemble (WPE), which is a class of zero-field Ising models with a first-order phase transition and tunable algorithmic hardness [239]. These problems belong to a special class of hard problem ensembles whose solutions are known a priori, which, together with the tunability of the hardness, makes the WPE model an ideal tool to benchmark heuristic algorithms for optimization problems. The Hamiltonian of the WPE model is given by

$$H_{\text{target}} = -\frac{1}{2}\sum_{i\neq j} J_{ij}^{\alpha}\sigma_i\sigma_j. \tag{6.12}$$

Here J_{ij}^{α} is a symmetric matrix satisfying

$$J^{\alpha} = \tilde{J}^{\alpha} - \text{diag}(\tilde{J})$$

and

$$\tilde{J}^{\alpha} = -\frac{1}{N}W_{\alpha}W_{\alpha}^{\text{T}}.$$

The term W_α is an $N \times \lfloor \alpha N \rfloor$ random matrix satisfying $W_\alpha t_{\text{ferro}} = 0$ where $t_{\text{ferro}} = (+1, +1, ..., +1)$ (see Ref. [239] for details about the generation of W_α). The ground state of the WPE model is known (i.e., it is planted) and corresponds to the states $\pm t_{\text{ferro}}$. α is a tunable parameter of hardness, where for $\alpha < 1$ this model displays a first-order transition, such that near zero temperature the paramagnetic states are meta-stable solutions [239]. This feature makes this model hard to solve with any annealing method, as the paramagnetic states are numerous compared to the two ferromagnetic states and hence act as a trap for a typical annealing method. We benchmark the three methods (SA, SQA, and VCA) for $N = 32$ and $\alpha \in \{0.25, 0.5\}$.

We consider 25 instances of the couplings $\{J_{ij}^{\alpha}\}$ and attempt to solve the model with VCA implemented using a dilated RNN ansatz with $\lceil \log_2(N) \rceil = 5$ layers and $T_0 = 1$. For SQA, we use an initial magnetic field $\Gamma_0 = 1$ and $P = 100$, while for SA we start with $T_0 = 1$.

We first plot the residual energies per site (Figs. 6.7(b)- (c)). Here we note that VCA is superior to SA and SQA for $\alpha = 0.5$ as demonstrated in Fig. 6.7(b). More specifically, VCA is about three orders of magnitude more accurate than SQA and SA for a large $N_{\text{annealing}}$. For $\alpha = 0.25$ (Fig. 6.7(c)), VCA is competitive and performs comparably with SA and SQA on average for a large $N_{\text{annealing}}$. We also represent the residual energies in a histogram form. We observe that for $\alpha = 0.5$ in Fig. 6.7(e), VCA achieves a higher density of configurations

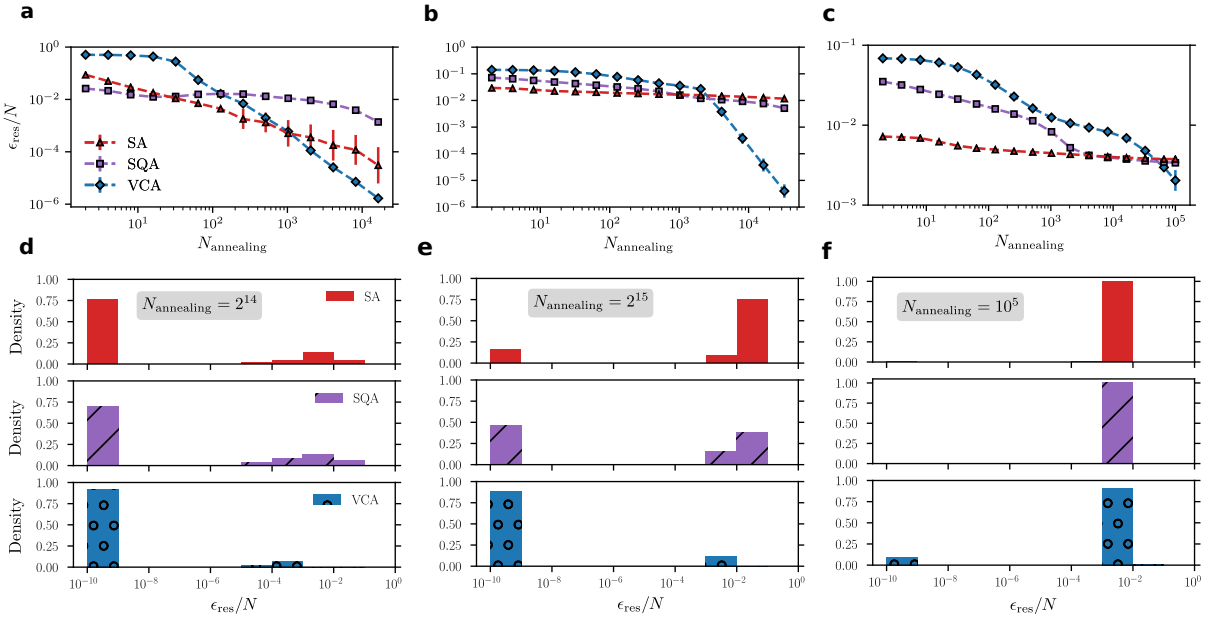

Figure 6.7: Benchmarking SA, SQA ($P = 100$ trotter slices) and VCA on the Sherrington-Kirkpatrick (SK) model and the Wishart planted ensemble (WPE). Panels (a),(b), and (c) display the residual energy per site as a function of $N_{\text{annealing}}$. (a) The SK model with $N = 100$ spins. (b) WPE with $N = 32$ spins and $\alpha = 0.5$. (c) WPE with $N = 32$ spins and $\alpha = 0.25$. Panels (d), (e) and (f) display the residual energy histogram for each of the different techniques and models in panels (a),(b), and (c), respectively. The histograms use 25000 data points for each method. Note that we choose a minimum threshold of 10^{-10} for ϵ_{res}/N, which is within our numerical accuracy.

with $\epsilon_{\text{res}}/N \sim 10^{-9}$-$10^{-10}$ compared to SA and SQA. For $\alpha = 0.25$ in Fig. 6.7(f), VCA leads to a non-negligible density at very low residual energies as opposed to SA and SQA, whose solutions display orders of magnitude higher residual energies. Finally, our WPE simulations support the observation that VCA tends to improve the quality of solutions faster than SQA and SA for a large $N_{\text{annealing}}$. For additional discussion about the WPE and SK results, see App. B.5. The running time estimations for SA, SQA, and VCA are provided in App. B.6.

6.6 Application to real-world optimization problems

This section contains material and results from Ref. [192].

After showing that VCA is superior on average compared to traditional Monte Carlo methods for spin-glass models, we explore VCA's performance in comparison with SA at solving three popular optimization problems: the maximum cut problem (Max-Cut), the nurse scheduling problem (NSP), and the traveling salesman problem (TSP). For all three problems, we find that VCA outperforms SA on average in the asymptotic limit by one or more orders of magnitude in terms of relative error. We reach large system sizes of up to 256 cities for the TSP. We also conclude that in the best-case scenario, VCA can serve as a great alternative when SA fails to find the optimal solution.

6.6.1 The Maximum Cut Problem (Max-Cut)

The first problem we target with VCA is the Max-Cut problem that has been known to be at least NP-hard [240]. The Max-Cut problem is defined as follows: Given an undirected graph $G(V, E)$, we make a cut along the edges of G to get two complementary sets of vertices such that the number of edges between the two sets is maximized. In other words, if $E_{\text{cut}} \subset E$ is the set of edges bridging the two complementary sets of vertices, we wish to maximize its size $|E_{\text{cut}}|$. In the context of our study, we work with unweighted graphs.

To model the partition across the graph, we can set a value of either 1 or 0 to the vertices to denote which side of the partition the vertex belongs to. Therefore, any solution to a graph of N vertices is given by $\boldsymbol{X} = (x_1, \ldots, x_N)$ where $x_i \in \{0, 1\}$. This mapping allows us to use an energy function $H(\boldsymbol{X})$ that computes the negative of the sum of the number of edges belonging in E_{cut}. This step is done by summing the following expression over all the edges in set E:

$$H(\boldsymbol{X}) = -\sum_{(i,j)\in E} (x_i + x_j - 2x_i x_j). \tag{6.13}$$

Taking the negative converts Max-Cut into a minimization problem. An edge connecting x_i and x_j exists in E_{cut} when $x_i \neq x_j$. The latter provides a contribution of -1 to $H(\boldsymbol{X})$. For simplicity, we note that the expression of $H(\boldsymbol{X})$ is equivalent to the following Kronecker

Delta expression

$$H(\boldsymbol{X}) = -\sum_{(i,j)\in E} 1 - \delta_{x_i x_j}.$$

Note that $H(\boldsymbol{X})$ in Eq. (6.13) is quadratic due to the binary nature of variables x_i. Therefore, $H(\boldsymbol{X})$ can be cast into a QUBO form as follows

$$H(\boldsymbol{X}) = \boldsymbol{X}^T \boldsymbol{Q} \boldsymbol{X}, \tag{6.14}$$

where $\boldsymbol{Q} \in \mathbb{R}^{N\times N}$ is the QUBO square matrix with matrix elements derived from Eq. (6.13). The flexibility of the RNN ansatz allows it to handle binary variables in addition to spins in the Ising formulation (6.1). This advantage makes the scaling make our VCA method more favorable in terms of embedding optimization problems compared to Quantum annealers based on Ising formulations such as in D-Wave.

For the numerical implementation, we use unweighted Max-Cut instances of $N = 128$ vertices and edge densities $\rho = 0.12, 0.25, 0.5$ generated by the Rudy graph generator by G. Rinaldi [241]. These graphs were originally generated to benchmark the classical-quantum optimization (CQO) technique [234]. The choice of increasing densities was used to investigate the effects of adding more complexity to the Max-Cut problem while keeping the number of vertices constant. We approximate the ground state of these graphs using the ConicBundle package by C. Helmberg from the Biq Mac Solver server [242].

We use the Vanilla RNN and Dilated RNN with site-dependent parameters for both to run VCA on this problem. As illustrated in Fig. 6.8, the general trend we see for both VCA and SA is that ϵ_{res} decreases with the number of annealing steps. VCA with Dilated RNNs (VCA-Dilated) outperforms SA on average starting from annealing steps $2^{11}, 2^8, 2^{14}$ respectively for the densities $0.12, 0.25, 0.5$. Although SA has better energies on average in the fast regime, VCA provides a better convergence at a larger number of annealing steps. This observation suggests that VCA requires a threshold number of annealing steps before it consistently converges to the ground state.

If we consider the best solutions obtained by SA and VCA (see Tab. 6.1), we observe that for all three densities, SA finds the ground state at $N_{\text{annealing}} = 2^4$. For both VCA variants, it is required to have a longer annealing time to find the ground states. The latter means that SA can find the optimal solution of our Max-Cut instances in fewer annealing steps compared to VCA.

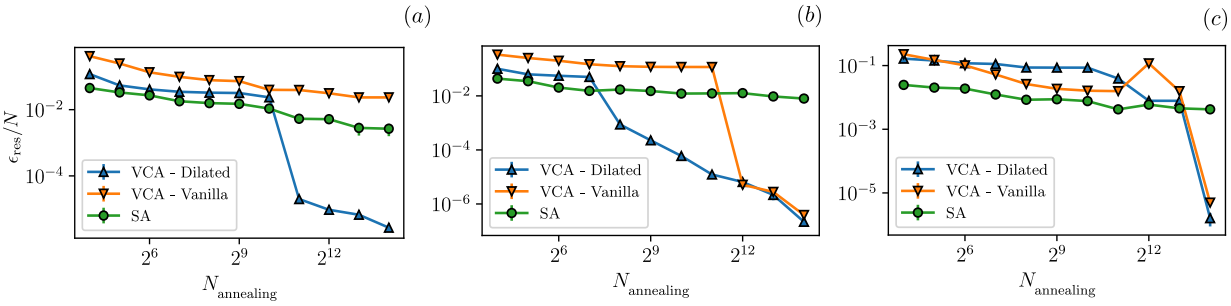

Figure 6.8: **Plot of the residual energy per site ϵ_{res}/N against the number of annealing steps $N_{\text{annealing}}$.** For all the RNN variations of VCA, we compare with SA on a Max-Cut problem with system size $N = 128$ with edge density (a) $\rho = 0.12$, (b) $\rho = 0.25$, and (c) $\rho = 0.5$. We observe that VCA outperforms SA in the limit of large $N_{\text{annealing}}$.

6.6.2 The Nurse Scheduling Problem (NSP)

The second combinatorial optimization problem we explore is the NSP which belongs to the class of scheduling problems in the field of operations research, and it is known to be NP-hard [243]. NSP aims to assign nurses to specific shifts in a hospital under a set of imposed constraints. The imposed constraints make it hard to find satisfactory solutions to the problem. In this work, we apply VCA to the formulation of NSP introduced in Ref. [244].

In this formulation, we have the following constraints: the hard nurse constraint, the hard shift constraint, and the soft nurse constraint. The hardness determines the importance of respecting the constraint. The hard nurse constraint requires that no nurse works for two consecutive shifts as they require sufficient rest after a shift. The hard shift constraint emphasizes the need to deploy enough nurses to handle a given shift. Finally, the soft nurse constraint aims for solutions to come up with an even distribution of nurses assigned to shifts. In this study, the terms 'shift' and 'day' are synonymous.

To better understand NSP, let us take the example of a hospital, where we have N individual nurses given by $n \in \{1, \ldots, N\}$ and D working days given by $d \in \{1, \ldots, D\}$. A solution to the NSP problem could be represented by a matrix

$$\boldsymbol{X_M} = \begin{bmatrix} x_{1,1} & \ldots & x_{1,D} \\ \vdots & \ddots & \vdots \\ x_{N,1} & \ldots & x_{N,D} \end{bmatrix},$$

where $\boldsymbol{X_M} \in \{0, 1\}^{N \times D}$. If nurse n has been assigned to day d, then $x_{n,d} = 1$, otherwise

$x_{n,d} = 0$. We use the flattened vector representation of $\boldsymbol{X_M}$ to represent our solution, denoted by $\boldsymbol{X} \in \{0,1\}^{ND}$. With index manipulation, the matrix elements of $\boldsymbol{X_M}$ map to the elements in $\boldsymbol{X} = [x_{m(1,1)}, \ldots, x_{m(N,D)}]$ using $m(n,d) = D(n-1) + d$.

The three constraints are formulated individually into separate quadratic penalty functions [244]. The hard nurse constraint, which is quantified by

$$H_1(\boldsymbol{X}) = \alpha \sum_{n=1}^{N} \sum_{d=1}^{D} x_{m(n,d)} x_{m(n,d+1)}, \tag{6.15}$$

penalizes a solution that has instances of a nurse n working two days in a row. Every time this violation occurs, a penalty of α is added.

The hard shift constraint has been formulated using the workforce required on a particular day $W(d)$. The constraint function

$$H_2(\boldsymbol{X}) = \sum_{d=1}^{D} \left(\sum_{n=1}^{N} x_{m(n,d)} - W(d) \right)^2, \tag{6.16}$$

aims to equalize the accumulated contribution of the nurses assigned to a working day and the actual amount of workforce required on that day. A penalty is incurred when there are not enough nurses assigned to a day, or conversely, when there is a surplus of nurses.

Lastly, we have the soft nurse constraint which promotes equal distribution of all nurses across the working days. The latter is as follows:

$$H_3(\boldsymbol{X}) = \sum_{n=1}^{N} \left(\sum_{d=1}^{D} x_{m(n,d)} - F(n) \right)^2. \tag{6.17}$$

The soft constraint term $H_3(\boldsymbol{X})$ has been formulated using $F(n)$ that gives us the number of days a nurse n wishes to work. If equal distribution of nurses is to be achieved, then setting at least $F(n) = \lfloor D/N \rfloor$ for all nurses ensures a fair workload when D is a multiple of N.

Now we can sum the three Hamiltonian terms shown in Eq. (6.15), (6.16), (6.17) to get the resultant energy function

$$H(\boldsymbol{X}) = H_1(\boldsymbol{X}) + \lambda H_2(\boldsymbol{X}) + \gamma H_3(\boldsymbol{X}), \tag{6.18}$$

where λ and γ are real coefficients that scale the penalty functions $H_2(\boldsymbol{X})$ and $H_3(\boldsymbol{X})$ respectively. $H(\boldsymbol{X})$ is minimized when, for the given NSP configuration, the least number of constraints are broken. If all the constraints are obeyed then the optimal energy

$H(\boldsymbol{X}^*) = 0$. However, it should be noted that for certain combinations of N and D, it is not always possible for the ground state $\boldsymbol{X}^*$ to respect all the constraints, resulting in $H(\boldsymbol{X}^*) > 0$.

Given the quadratic form of $H(\boldsymbol{X})$ the binary nature of $\boldsymbol{X}$, $H(\boldsymbol{X})$ can be represented as a quadratic unconstrained binary optimization (QUBO) model [244, 245],

$$H(\boldsymbol{X}) = \sum_{i=1}^{ND} \sum_{j=i}^{ND} q_{ij} x_i x_j + c,$$

where coefficient $q_{i,j}$ and constant c are derived from Eq. (6.18). This expression can be made more compact to get

$$H(\boldsymbol{X}) = \boldsymbol{X}^T \boldsymbol{Q} \boldsymbol{X} + c,$$

such that $\boldsymbol{Q} \in \mathbb{R}^{ND \times ND}$ is a square matrix with matrix elements given by q_{ij}.

We use a configuration of 15 days and 7 nurses, giving us a system size of $15 \times 7 = 105$. Taking inspiration from Ref. [244], we use the following NSP parameters described in Sec. 6.6.2: $W(d) = 1, F(n) = \lfloor D/N \rfloor, \alpha = 3.5, \lambda = 1.3, \gamma = 0.3$. Based on the chosen parameters, the ground state energy is $H(\boldsymbol{X}^*) = \gamma$, which corresponds to a configuration with one soft constraint violation.

We use vanilla and dilated RNNs with site-dependent parameters to solve this optimization problem. The results are presented in Fig. 6.9(a). Again, we see that increasing the number of annealing steps allows VCA to reduce the average error ϵ_{res}/N. Both VCA-Dilated and VCA-Vanilla reach lower error margins compared to SA at large $N_{\text{annealing}}$. For a small number of annealing steps, SA outperforms VCA on the average case. Between annealing steps 2^9 and 2^{13}, VCA-Dilated maintains the lowest average error among all the models. For the slowest annealing schedule, $N_{\text{annealing}} = 2^{14}$, VCA-Vanilla reaches the overall lowest average error, whereas VCA-Dilated has a sudden increase in ϵ_{res}/N. A possible reason for this sudden increase can be related to the choice of our hyperparameters, which can be further tuned.

Lastly, by considering the best solutions, SA finds the ground state at $N_{\text{annealing}} = 2^4$ whereas VCA lands at the optimal solution around $N_{\text{annealing}} = 2^6$ as illustrated in Tab. 6.1. The latter means that SA, similarly to Max-Cut, needs fewer $N_{\text{annealing}}$ steps compared to VCA to find optimal solutions.

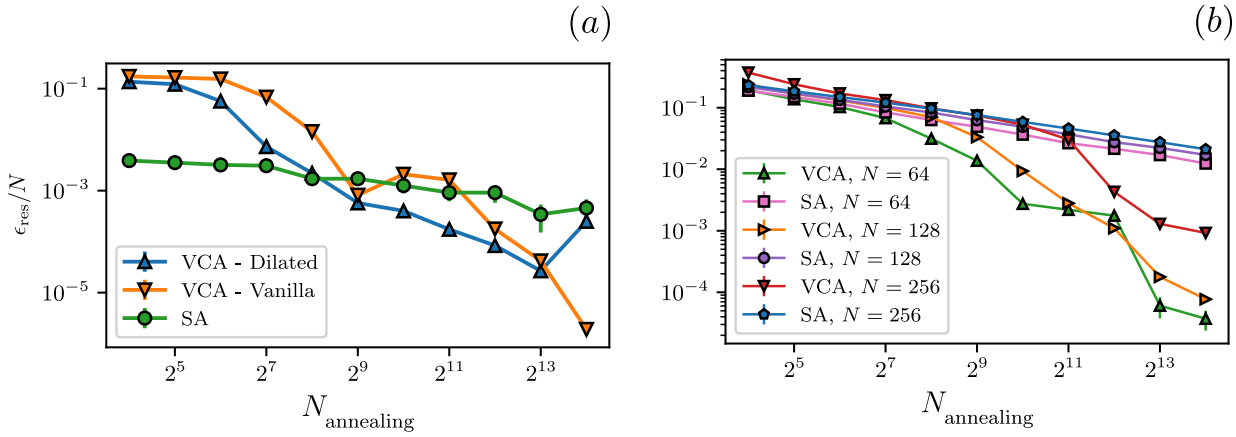

Figure 6.9: **Plots of the residual energy per site ϵ_{res} vs $N_{\text{annealing}}$.** (a) For NSP with 15 days and 7 nurses, both the Vanilla and Dilated RNN variants of VCA are plotted alongside SA. (b) For TSP with system sizes $N = 64, 128, 256$, VCA with Dilated RNNs and with shared parameters is plotted alongside SA. For panels (a) and (b), we see that VCA has a lower ϵ_{res} compared to SA for large $N_{\text{annealing}}$.

6.6.3 The Traveling Salesman Problem (TSP)

TSP is another famous combinatorial optimization problem that we attempt to solve with VCA. The aim of TSP is to visit a set of cities exactly once and return to the starting city in the tour with a specific order that minimizes the total cost of travel. In our study, the cost is the total distance traveled. We are particularly interested in the 2D Euclidean TSP which is known to be NP-complete [246].

Given N cities on a 2D Cartesian plane, a solution tour looks as $\boldsymbol{X} = (\pi(1), \ldots, \pi(N))$ where π is a permutation of N cities. Each city $\pi(i)$ has a tuple of x and y coordinates $(x_{\pi(i)}, y_{\pi(i)})$. An important property of a valid solution that should be noted is that every city on the tour should be unique. We also remark that the permutation order of cities is translation invariant. The latter property is exploited in the construction of our parameterized model. The energy function of the 2D Euclidean TSP is given by the sum of the Euclidean distances between consecutive cities, including the trip from the last city back to the first as follows

$$H(\boldsymbol{X}) = \sum_{i=1}^{N} \sqrt{(x_{\pi(i)} - x_{\pi((i+1)\%N)})^2 + (y_{\pi(i)} - y_{\pi((i+1)\%N)})^2}, \tag{6.19}$$

where $\%$ denotes the modulo operator.

Numerically, we use three instances of TSP with varying system sizes $N = 64, 128, 256$. Each component of the coordinates (x, y) of the cities has been uniformly sampled from the range $(0, 1)$. The approximation of the ground state tours of these TSP configurations is provided by the Concorde algorithm [247] through the NEOS Server [248].

Unlike the previous problems, we use VCA with dilated RNNs that have shared parameters owing to the translation invariance property of TSP. The focus on dilated RNNs for this problem is motivated by the presence of long-range interactions in TSP that can be captured by this architecture as well by the overall advantage of this architecture compared to vanilla RNNs in Max-Cut and TSP. Additionally, the Softmax output probability (see Sec. 4.10) has a size equal to the number of cities N as opposed to Max-Cut and NSP with output size equal to 2 for binary variables.

To avoid revisiting the same cities during the process of autoregressive sampling in VCA, we take inspiration from the U(1) symmetry implementation in Sec. 4.11.2, and in Ref. [249] in the context of TSP. Here, we apply a mask on the visited cities when computing the conditional probabilities $P_{\boldsymbol{\theta}}(c_i|c_{i-1}, \ldots, c_2, c_1)$ at the level of the Softmax layer (see Eq. (4.4)), where $c_i \in \{1, 2, \ldots, N\}$ corresponds to the city to be visited at step i and $\boldsymbol{\theta}$ are the parameters of our model. The masking at step i is done as follows:

- If cities $(c_1, c_2, \ldots c_{i-1})$ were visited, where $c_j \in \{1, 2, \ldots, N\}$, then $P_{\boldsymbol{\theta}}(c_j|c_{i-1}, \ldots, c_2, c_1)$ is set to zero for $1 \leq j \leq i - 1$.
- After the previous step, the conditional probability $P_{\boldsymbol{\theta}}(.|c_{i-1}, \ldots, c_2, c_1)$ is renormalized to 1.

After implementing these steps, our autoregressive model provides us with a permutation π of the cities. We note that this masking trick can be implemented in parallel across the batch size.

The results from our experiments are presented in Fig. 6.9(b). For $N = 64, 128$, VCA reaches a residual energy per site $\epsilon_{\text{res}}/N < 10^{-4}$ whereas for $N = 256$, we reach an average error $\epsilon_{\text{res}}/N \sim 10^{-3}$. We also observe that SA reaches $\epsilon_{\text{res}}/N \sim 10^{-2}$ in the slow annealing regime. Overall, in the range of medium to slow annealing ($N_{\text{annealing}} \geq 2^7$), VCA demonstrates a better average performance compared to SA. Furthermore, for the largest system size, VCA requires a relatively larger $N_{\text{annealing}}$ to noticeably improve over SA highlighting the increase in complexity that arises with larger system sizes.

By considering the best-case scenario, VCA finds the best tours for all three problem sizes. It also finds the exact ground state for $N = 64$, as illustrated in Tab. 6.1. In contrast,

		SA			VCA		
	Parameter	$\epsilon_{\text{res}}^{\text{min}}$	$N^*_{\text{annealing}}$	Time	$\epsilon_{\text{res}}^{\text{min}}$	$N^*_{\text{annealing}}$	Time
NSP	15D, 7N	**0.0**	$\mathbf{2^4}$	**3 s**	D: **0.0**	2^6	4 min 5 s
					V: **0.0**	2^7	1 min 25 s
Max-Cut	$\rho = 0.12$	**0.0**	$\mathbf{2^4}$	**6 s**	D: **0.0**	2^{11}	45 min 32 s
					V: **0.0**	2^{13}	1 hr 23 min
	$\rho = 0.25$	**0.0**	$\mathbf{2^4}$	**6 s**	D: **0.0**	2^8	9 min 14 s
					V: **0.0**	2^{12}	42 min 14 s
	$\rho = 0.50$	**0.0**	$\mathbf{2^4}$	**6 s**	D: **0.0**	2^{14}	5 hrs 35 min
					V: **0.0**	2^{14}	2 hrs 43 min
TSP	$N = 64$	3.13×10^{-3}	2^{14}	1 min 30 s	D: **0.0**	$\mathbf{2^{13}}$	**49 min 15 s**
	$N = 128$	9.24×10^{-3}	2^{14}	6 min 19 s	D: $\mathbf{3.84 \times 10^{-5}}$	$\mathbf{2^{13}}$	**1 hr 24 min**
	$N = 256$	1.75×10^{-2}	$\mathbf{2^{14}}$	25 min 42 s	D: $\mathbf{8.60 \times 10^{-4}}$	$\mathbf{2^{14}}$	**6 hrs 51 min**

Table 6.1: **A summary table of the best performances of VCA and SA on NSP, Max-Cut, and TSP.** Here we define the minimal residual energy per site $\epsilon_{\text{res}}^{\text{min}}/N \equiv (H_{\text{min}} - H(\boldsymbol{X}^*))/N$ where H_{min} is the lowest energy obtained by either SA or VCA across the different samples. 'D' stands for the Dilated RNN results and 'V' stands for the Vanilla RNN results. Values in bold font correspond to the lowest $N^*_{\text{annealing}}$ to find the exact or the lowest approximation to the ground state after comparing SA and VCA. Furthermore, bold values highlight the lowest $\epsilon_{\text{res}}^{\text{min}}$, as well as the lowest estimated time to find the exact or the best solution.

SA finds tours with a higher cost and gets stuck in local minima. This result is different from what we observed for the Max-Cut and the NSP. Thus, we conclude that if a user is interested in finding the best solutions to a particular optimization problem, VCA is a valuable alternative when SA fails to find an optimal solution.

More details about the VCA hyperparameters and the SA implementation can be found in Appendix B.7.

6.7 Application to frustrated systems

After revealing the potential of variational annealing for solving combinatorial optimization problems, we now move to another use case of annealing. Here we investigate the value of annealing in finding the ground state of frustrated systems. As a test bed, we take the 2D Heisenberg model on the triangular lattice. Due to the frustrated, non-bipartite nature of the triangular lattice, the Hamiltonian $\hat{H}$ can no longer be made stoquastic with a simple unitary transformation. Such a Hamiltonian can make the VMC optimization landscape rough and filled with local minima [250]. Here, we use annealing to overcome local minima and to obtain accurate estimates of ground state energies using a fictitious

pseudo-temperature (see Eq. 3.31).

To demonstrate the idea of annealing, we target the ground state of the 4×4 triangular lattice at different numbers of annealing steps $N_{\text{annealing}}$. The results in Fig. 6.10(a) show that starting from a non-zero pseudo-temperature T_0 allows obtaining a lower value of the relative error as opposed to traditional VMC where the initial pseudo-temperature $T_0 = 0$. We also remark that higher accuracy is achieved by increasing the number of annealing steps, which underlines the importance of adiabaticity in our scheme.

We now focus our attention on larger system sizes. Here we train our 2D cTGRU wave function (see Sec. 5.2.2) using annealing for a system size 6×6, while applying a Marshall sign to minimize the sign effect. We then optimize our ansatz at larger system sizes after increasing the lattice side length by 2 until we arrive at size 16×16. This step is done at zero pseudo-temperature and without reinitializing the parameters of the RNN. This idea was already proposed in the literature in Ref. [35]. The latter takes advantage of the translation invariance property encoded in the weight-sharing of the parameters of the RNN. The results in Fig. 6.10(b) show that for system sizes larger than 14×14, the variational energy obtained by our RNN ansatz is more accurate compared to the energy obtained by DMRG. It is important to note that our RNN ansatz uses less than 0.1% of the parameters of DMRG for system sizes larger than 14×14. The hyperparameters used to produce our results can be found in App. A.1. Additionally, we note that a comparison of our approach with Ref. [146] is provided in App. A.2.

Conclusion

In conclusion, we have introduced a strategy to combat the slow sampling dynamics encountered by simulated annealing when an optimization landscape is rough or glassy. Based on annealing the variational parameters of a generalized target distribution, our scheme — which we dub *variational neural annealing* — takes advantage of the power of modern autoregressive models, which can be exactly sampled without slow dynamics even when a rough landscape is encountered. We implement variational neural annealing parameterized by a recurrent neural network, and compare its performance to conventional simulated annealing on prototypical spin glass Hamiltonians and real-world optimization problems known to have landscapes of varying roughness. We find that variational neural annealing produces accurate solutions to all of the optimization problems considered, where our techniques typically reach solutions orders of magnitude more accurately on average than conventional simulated annealing in the limit of a large number of annealing steps.

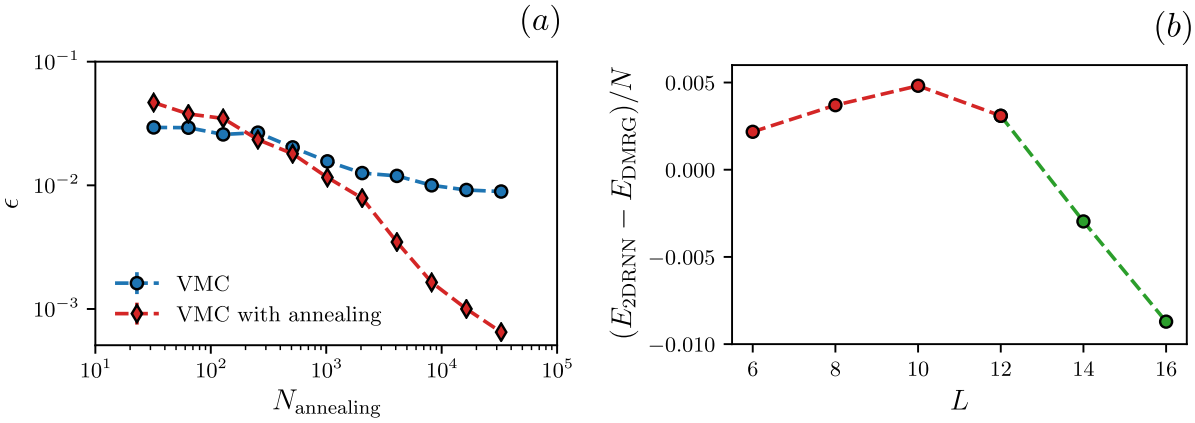

Figure 6.10: (a) A scaling of the relative error ϵ against the number of annealing steps $N_{\text{annealing}}$ for the triangular Heisenberg model with size 4×4, 'VMC' corresponds to an initial pseudo-temperature $T_0 = 0$ whereas for 'VMC with annealing', we start with $T_0 = 1$. (b) A plot of the energy difference per site between the 2DRNN and the DMRG. Negative values show that our ansatz is superior compared to DMRG for system sizes larger than 14×14.

We also find that our framework is applicable to frustrated quantum systems with a rugged optimization landscape.

We emphasize that several hyperparameters, models, hardware, and objective function choices can be explored and may improve our methodologies. We have utilized a simple annealing schedule and we highlight that reinforcement learning can be used to improve our cooling schedules [251]. A critical insight gleaned from our experiments is that certain neural network architectures were more efficient on specific Hamiltonians. Thus, a natural direction is to study the intimate relationship between the model architecture and the problem Hamiltonian, where we envision that symmetries and domain knowledge would guide the design of models and algorithms.

As we witness the unfolding of a new age for optimization powered by deep learning [252], we anticipate rapid adoption of machine learning techniques in the space of combinatorial optimization, as well as anticipate domain-specific applications of our ideas in technological and scientific areas related to physics, biology, health care, economy, transportation, manufacturing, supply chain, hardware design, computing, and information technology, among others.

Reproducibility Code

The code we use to produce our results can be found in Refs. [253, 254].

Chapter 7

Investigating Topological Phases of Matter with RNNs

This chapter contains results and material from Ref. [255], in addition to other material not published elsewhere.

Landau symmetry breaking theory provides a fundamental description of a wide range of phases of matter and their phase transitions through the use of local order parameters [70]. Despite the fact that a great deal of our theoretical and experimental investigations of interacting quantum many-body systems have been developed with the aim of studying local order parameters, it is well-known that the most intriguing strongly correlated phases of matter may not be easily characterized through these observables. Instead, several states of matter seen in modern theoretical and experimental studies are characterized using non-local order parameters that rely on the phases' topological properties [256–258]. Topological order, in particular, refers to a type of order characterized by the emergence of quasi-particle anyonic excitations, topological invariants, and long-range entanglement, which typically do not appear in traditional forms of order. As a result of these properties, topologically ordered phases have been suggested as an important building block for the development of a protected qubit resistant to perturbations and errors [158, 259, 260]. Such qubits have been devised recently at the experimental level [261].

While most manifestations of topological order are dynamical in nature—e.g. anyon statistics, ground state degeneracy, and edge excitations [100]–topological order can also be characterized directly in terms of the ground state wave function and its entanglement. In

particular, a probe for topological order is the topological entanglement entropy (TEE) [99, 100], which offers a characterization of the global entanglement pattern of topological ground states not present in conventionally ordered systems. Notably, the TEE is readily accessible for large classes of topological orders [100, 262], in numerical simulations based on quantum Monte Carlo (QMC) [263–265] and density matrix renormalization group (DMRG) [266, 267], as well as in experimental realizations of topological order based on gate-based quantum computers [258].

In the previous chapter, we have shown the ability of RNNs supplemented with the principle to solve combinatorial optimization problems. In this chapter, we focus our attention on the ability of RNNs to investigate topological order in quantum matter. In particular, we use 2D RNNs with a gating mechanism to investigate topological order in 2D through the estimation of the TEE. We focus on two model Hamiltonians exhibiting topological order, namely Kitaev's toric code [158, 260] and a Bose-Hubbard model on the kagome lattice previously shown to host a gapped quantum spin liquid with non-trivial emergent $\mathbb{Z}_2$ gauge symmetry [263, 265, 268]. We also target Rydberg atom arrays on Kagome lattice where the existence of a spin-liquid is still not clearly known. In our study, we use Kitaev-Preskill constructions [99], Levin-Wen constructions [100], and finite size-scaling analysis of the entanglement entropy to extract the TEE. We find convincing evidence that RNNs are capable of expressing ground states of Hamiltonians displaying topological order. We also find evidence that the RNN wave function is naturally biased toward finding superpositions of minimally entangled states, as reflected in the calculations of entanglement entropy and Wilson loop operators for the toric code. Our RNN ansatz also signals no evidence for the existence of a topological order on the Rydberg atom arrays on the Kagome lattice. Overall, our results indicate that RNNs can represent phases of matter beyond the conventional Landau symmetry-breaking paradigm.

The plan of this chapter is as follows: in Sec. 7.1, we define the concept of the TEE and the different constructions used to extract this quantity. In Secs. 7.2, 7.3 and 7.4, we present our results respectively for the 2D toric code, as well as on hard-core Bose-Hubbard model and the Rydberg atom arrays on Kagome lattices.

7.1 Topological entanglement entropy

A powerful tool to probe topologically ordered states of matter is through the so-called topological entanglement entropy (TEE) [99–101, 262–264, 269–271]. The TEE can be extracted by computing the entanglement entropy of a spatial bipartition of the system into A and B, which together comprise the full system. For many phases of 2D matter,

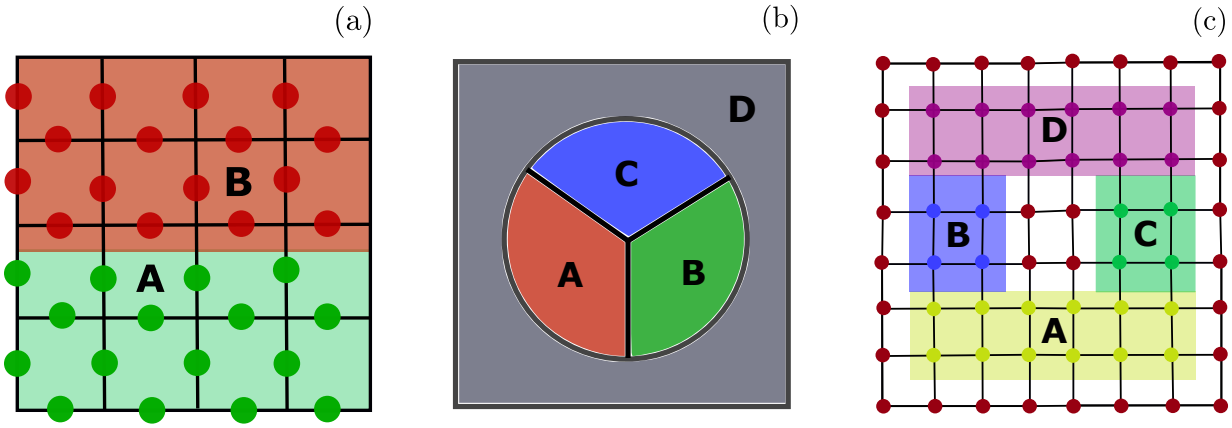

Figure 7.1: In panel (a), we illustrate how we cut the torus geometry of the toric code lattice into two equal cylinders. Additionally, we show a sketch of the parts A, B, and C that we use for Kitaev-Preskill construction to compute the TEE in panel (b), as well as the Levin-Wen construction for a square lattice with size $L = 8$ in panel (c). For the Rydberg atoms Hamiltonian on a Kagome lattice, each dot on the square lattice corresponds to a block of three spins, as shown in Fig. 4.7(b).

the Renyi-n entropy satisfies the area law $S_n(A) = aL - \gamma + \mathcal{O}(L^{-1})$. Here L is the size of boundary between A and B, $S_n(A) \equiv \frac{1}{1-n}\ln(\mathrm{Tr}(\rho_A^n))$, $\rho_A = \mathrm{Tr}_B|\Psi\rangle\langle\Psi|$ is the reduced density matrix of subsystem A, $|\Psi\rangle$ is the state of the system, and γ is the TEE. The TEE detects non-local correlations in the ground state wave function and plays the role of an order parameter for topological phases similar to the notion of a local order parameter in phases displaying long-range order. Since the TEE is shown to be independent of the choice of Renyi index n [101], we can use the swap trick with our RNN wave function ansatz to calculate the second Renyi entropy S_2 as shown in Sec. 3.8 and extract the TEE γ.

To access γ, we can approximate the ground state of the system using an RNN wave function ansatz, i.e. $|\boldsymbol{\Psi}_{\boldsymbol{\theta}}\rangle \approx |\boldsymbol{\Psi}\rangle$ for different system sizes followed by a finite-size scaling analysis of the second Renyi entropy. We can also make use of a TEE construction, e.g., the Kitaev-Preskill construction [99] and the Levin-Wen construction [100] as illustrated in Fig. 7.1.

The Kitaev-Preskill construction prescribes dividing the system into four subregions A,

B, C, and D as illustrated in Fig. 7.1(b). The TEE can be then obtained by computing

$$\gamma = -S_2(A) - S_2(B) - S_2(C) + S_2(AB) \\ + S_2(AC) + S_2(BC) - S_2(ABC),$$

where $S_2(A)$ is the second Renyi entropy of the subsystem A, and AB is the union of A and B and similarly for the other terms. Finite-size effects on γ can be alleviated by increasing the size of the subregions A, B and C [99, 272]. Finally, we highlight the ability of the RNN wave function to study systems with fully periodic boundary conditions as a strategy to mitigate boundary effects, as opposed to cylinders used in DMRG [273, 274], which may potentially introduce edge effects that can affect the values of the TEE [275].

The Levin-Wen construction allows to extract the TEE γ by constructing four different subsystems $A_1 = A \cup B \cup C \cup D, A_2 = A \cup C \cup D, A_3 = A \cup B \cup D$ and $A_4 = A \cup D$ as illustrated in Fig. 7.1(c) such that [263]:

$$\gamma = \frac{-S_2(A_1) + S_2(A_2) + S_2(A_3) - S_2(A_4)}{2}.$$

Note that finite size effects on γ can be eliminated by extrapolating the width and the thickness of A_1, A_2, A_3 and A_4 [263, 272].

Finally, it is important to note that our ability to study quantum systems with fully periodic boundary conditions helps to mitigate boundary effects, as opposed to cylinders used in DMRG [273, 274] that introduces a bias in the value of the TEE [275].

7.2 2D toric code

We now focus our attention on the toric code Hamiltonian which is the simplest model that hosts a Z_2 topological order [158, 269] and has a non-zero TEE equal to $\gamma = \ln(2)$. The Hamiltonian is defined in terms of spin-1/2 degrees of freedom located on the edges of a square lattice (see Fig. 4.7(a)) and is given by

$$\hat{H} = -\sum_p \prod_{i \in p} \hat{\sigma}_i^z - \sum_v \prod_{i \in v} \hat{\sigma}_i^x,$$

where the first summation is on the plaquettes and the second summation is on the vertices [269] of the lattice. Note that the lattice in Fig. 4.7(a) can be seen as a square lattice with a unit cell containing two spins. In our simulations, we use an $L \times L \times 2$ array of

spins where L is the number of plaquettes on each side of the underlying square lattice. It is possible to study the toric code with a two-dimensional RNN defined on a primitive square lattice by merging the two spin degrees of freedom of the unit cell of the toric code into a single "patch" followed by an enlargement of the local Hilbert space dimension in the RNN from 2 to 4. This idea is illustrated in Sec. 4.12. We provide additional details about the mapping in App. C.1.

To extract the TEE from our ansatz, we variationally optimize the 2D RNN wave function targetting the ground state of this model for multiple system sizes on a square lattice with periodic boundary conditions. After the optimization, we compute the TEE using system size extrapolation and using the Kitaev-Preskill scheme provided in Sec. 7.1. More details about the regions chosen for this construction are provided in App. C.2. To avoid local minima during the variational optimization, we perform an initial annealing phase through the use of a fictitious temperature (see Sec. 3.10 and Sec. 6.7). Additional details are provided in App. C.1.

The results shown in Fig. 7.2(a) suggest that our 2D RNN wave function can describe states with an area law scaling in two dimensions. Linearized versions of the RNN wave function have been recently shown to display an entanglement area law [146]. For $L = 10$ (not included in the extrapolations in Fig. 7.2(a)), it is challenging to evaluate S_2 accurately as the expectation value of the swap operator is proportional to $\exp(-S_2)$, which becomes very small and thus hard to resolve accurately via sampling the RNN wave function. The improved ratio trick is an interesting alternative for enhancing the accuracy of our estimates [32, 68]. The use of conditional sampling is also another possibility for enhancing the accuracy of our measurements [102].

The extrapolation also confirms the existence of a non-zero TEE whose value is close to $\gamma' = \ln(2)$ within error bars. Note that the sub-region we have used to compute the TEE is half of the torus, namely a cylinder with two disconnected boundaries[1] (see Fig. 7.1(a)). As shown in Ref. [270], the use of this geometry means that the expected TEE becomes state-dependent and given by

$$\gamma' = 2\gamma + \ln\left(\sum_i \frac{p_i^2}{d_i^2}\right) \tag{7.1}$$

for the second Renyi entropy. Here $d_i \geq 1$ is the quantum dimension of a i-th quasi-particle. For the toric code, we have abelian anyons with $d_i = 1$. Additionally $p_i = |\alpha_i|^2$

[1]Note that this choice allows minimizing the boundary size as opposed to a square region in the bulk. This feature is desirable since the swap operator used to estimate the second Renyi entropy [43] becomes very small, and thus more sensitive to statistical errors when the boundary increases for a quantum system satisfying the area law.

is the overlap of the computed ground state $|\Psi\rangle$ with the i-th minimally entangled state (MES) $|\Xi_i\rangle$ where

$$|\Psi\rangle = \sum_i \alpha_i |\Xi_i\rangle .$$

The observations above and the numerical result $\gamma_{\text{RNN}} \approx \ln(2)$ suggest that the RNN wave functions optimized via gradient descent and annealing find a superposition of MES as opposed to DMRG, which typically collapses to a single MES [264, 266]. The analysis provided in App. C.3 demonstrates that our optimized RNN ansatz finds a uniform superposition of two MES which increases the entanglement in the state with respect to a single MES. Thus using Eq. (7.1), we expect $\gamma' = 2\ln(2) + \ln\left(\frac{1}{4} + \frac{1}{4}\right) = \ln(2)$, which is consistent with our numerical observations.

We note that the exact autoregressive sampling procedure plays a key role in the ability of our RNN ansatz to sample a superposition of different topological sectors when this superposition is encoded in our ansatz. For wave functions representing the ground state of the toric code used in combination with Markov-chain Monte Carlo methods, the probability of sampling different topological sectors of the state is exponentially suppressed even if the exact wave function ansatz encodes different topological sectors. This observation can be illustrated using an exact convolutional neural network construction of the toric code ground state which contains an equal superposition of different topological sectors [28]. Although in principle such representation contains all topological sectors, its form is not amenable to exact sampling and uses Markov chains so that upon sampling with local moves the system chooses a fixed topological sector. The ability of RNNs to recover different modes has been also highlighted for spin-glass models in App. B.5.

To further verify that our 2D RNN wave function can extract the correct TEE of the 2D toric code, we compute the TEE using the Preskill-Kitaev construction, which has contractible surfaces, and for which the TEE does not depend on the topological sector superposition [264, 270] (see App. C.2 for details about the construction). The results reported in Fig. 7.2(b) demonstrate an excellent agreement between the TEE extracted by our RNN and the expected theoretical value for the toric code.

7.3 Bose-Hubbard model

We now turn our attention to a hard-core Bose-Hubbard model on the Kagome lattice, which has been shown to host topological order [263, 265, 268]. The Hamiltonian of this

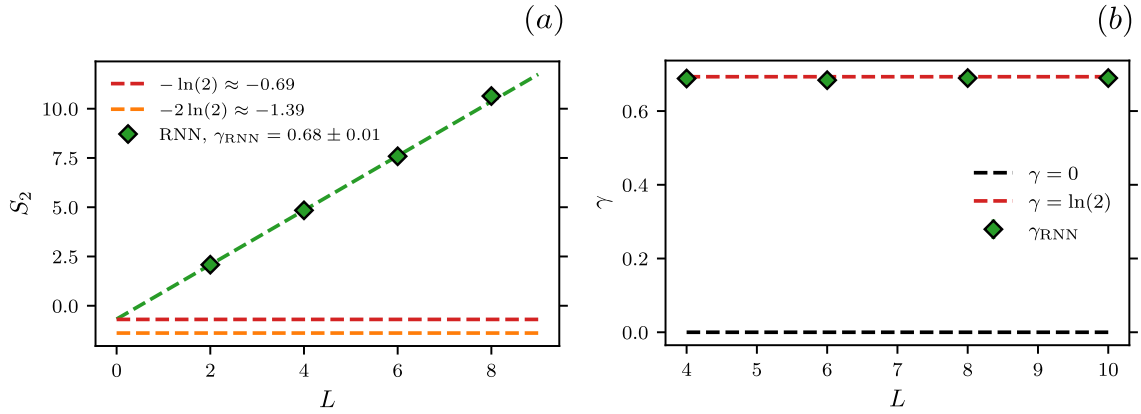

Figure 7.2: **Entanglement properties of the 2D toric code.** (a) Second Renyi entropy scaling computed using our RNN wave function on the 2D toric code. (b) TEE computed with the Kitaev-Preskill construction (see App. C.2) for different system sizes $L \times L \times 2$. The values found by the RNN are very close to $\ln(2)$. Error bars correspond to one standard deviation and are smaller than the symbol size.

model is given by

$$\hat{H} = -t \sum_{\langle i,j \rangle} \left(b_i^\dagger b_j + b_i b_j^\dagger \right) + V \sum_{\hexagon} n_{\hexagon}^2, \tag{7.2}$$

where b_i ($b_i^\dagger$) is the annilihation (creation) operator. Furthermore, t is the kinetic strength, V is a tunable interaction strength and $n_{\hexagon} = \sum_{i \in \hexagon} (n_i - 1/2)$. The first term corresponds to a kinetic term that favors hopping between nearest neighbors, whereas the second term promotes an occupation of three hard-core bosons in each hexagon in the Kagome lattice. In our setup, we choose V in units of the kinetic term strength t.

The atom configurations of this model correspond to an $L \times L \times 3$ array of binary degrees of freedom where L is the size of each side of the Kagome lattice. Following an analogous approach to the toric code, we enlarge the local Hilbert space size from 2 to 8 and gather the 3 spins of the unit cell of the kagome lattice as input to the 2D RNN cell, as illustrated in Sec. 4.12. This allows us to map our Kagome lattice with a local Hilbert space of 2 to a square lattice with an enlarged Hilbert space of size $2^3 = 8$.

The model is known to host a Z_2 spin-liquid phase for $V \gtrsim 7$ [263, 265, 276]. To confirm this finding, we estimate γ for the system sizes $6 \times 6 \times 3$ and $8 \times 8 \times 3$. We use the Kitaev-Preskill construction [99]. The details of the construction of the regions A, B, and C are provided in App. C.2. As the Hamiltonian in Eq. 7.3 has a $U(1)$ symmetry associated with

the conservation of bosons in the system, we impose such symmetry on our RNN wave function [43]. We also supplement the VMC optimization with annealing to overcome local minima as previously done for the 2D toric code (see Sec. 3.10 and Sec. 6.7). For the system size $8 \times 8 \times 3$, the RNN ansatz parameters were initialized using the optimized parameters for the $6 \times 6 \times 3$ (see details about the hyperparameters in App. C.1).

The results are provided in Fig. 7.3. The computed TEEs for $L = 6, 8$ show a saturation of γ_{RNN} for large values of the interaction strength V. We observe that the saturation values of γ_{RNN} are in good agreement with the expected TEE $\gamma = \ln(2)$ of a Z_2 spin-liquid [263]. Additionally, the negative values of γ_{RNN} observed for $V \leq 6$ in the superfluid phase [263] may be related to the presence of Goldstone modes that manifest themselves as corrections to the area law in the entanglement entropy and can be seen as a negative contribution to the TEE [277]. We note that the QMC methods are capable of obtaining a consistent value with the exact TEE for this model at $V = 8$ for very large system sizes [265] using finite-size extrapolation. This observation suggests that our RNN ansatz is still limited by finite-size effects at $V = 8$ (see Fig. 7.3) for which the TEE is not yet saturated to $\ln 2$. Other sources of error in our calculation may be due to inaccuracies in the variational calculations and statistical errors due to the sampling. However, we note that our variational calculation is performed at zero temperature, which makes our calculations insensitive to temperature effects. As a result, the RNN's TEE saturates to the anticipated value of $\ln 2$ using Kitaev-Preskill construction, as opposed to TEE results based on QMC on a similar geometry, which are harder to converge at low temperature and plateau to half of the expected TEE [263].

7.4 Rydberg atom arrays

We now focus our attention on the Rydberg atoms array Hamiltonian on the Kagome lattice. The latter has been extensively studied in the literature and was shown to be experimentally realizable in the lab [278]. This system is also believed to provide a convenient framework to experimentally prepare spin liquids [257, 275, 279]. The Hamiltonian of this model is given by [275, 278]:

$$\begin{aligned}\hat{H} = &\sum_{i=1}^{N} \frac{\Omega}{2} \Big(|g\rangle_i \langle r| + |r\rangle_i \langle g| \Big) - \delta \sum_{i=1}^{N} |r\rangle_i \langle r| \\ &+ \frac{1}{2} \sum_{(i,j)} V(||\boldsymbol{x}_i - \boldsymbol{x}_j||) \, |r\rangle_i \langle r| \otimes |r\rangle_j \langle r| \, .\end{aligned}$$

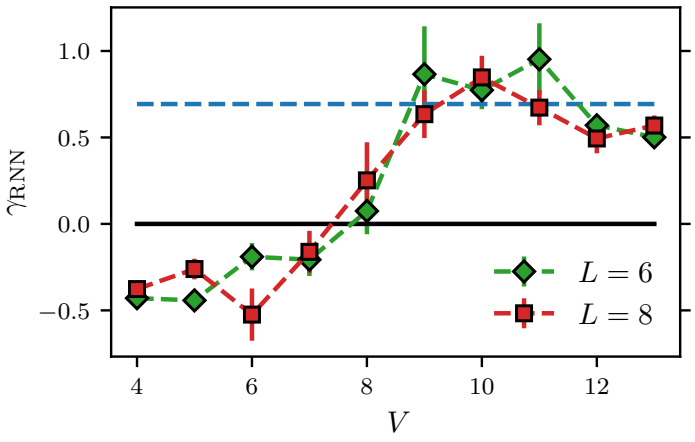

Figure 7.3: A plot of the topological entanglement entropy against the interaction strength V (in units of t) of Hard-core Bose-Hubbard model on Kagome Lattice for system sizes $N = L \times L \times 3$ where $L = 6, 8$. The calculations were performed using the Kitaev-Preskill construction (see App. C.2). The continuous black horizontal line corresponds to a zero TEE, and a dashed blue horizontal line for a ln(2) TEE.

Here $|g\rangle_i, |r\rangle_i$ are respectively the ground and the excited states of the Rydberg atom i. Ω is the Rabi frequency and δ is the laser detuning. $V(R) = C/R^6$ is the repulsive potential due to the dipole-dipole interaction between Rydberg atoms, which is responsible for the Blockade mechanism [278]. In practice, we define a blockade radius R_b such that $V(R_b/a) = \Omega$, where a is the distance between two neighboring Rydberg atoms. Finally, we note that the sum over all possible pairs is truncated to a sum over the neighbors separated by a distance cutoff $R_c = 2$ or $R_c = 4$. The choice $R_c = 2$ is taken mainly to compare with the DMRG [142, 165] results reported in Ref. [275]. For numerical convenience, we set $a = 1 = \Omega$ without loss of generality.

Due to the frustrated nature of this model which induces local minima in our optimization landscape, we supplement our 2D RNN wave function with annealing with an initial pseudo-temperature T_0 and $N_{\mathrm{annealing}}$. Further details about the hyperparameters are provided in App. C.1.

To confirm the correctness of our method, we optimize our ansatz to find the ground state at $R_b = 1.7$ and $\delta = 3.3$ and with $R_c = 2$, which has a nematic order according to the DMRG results [275]. In Fig. 7.4(a), we plot the two point correlations $\langle n_{\mathbf{0}} n_{\boldsymbol{r}} \rangle$. The use of the two-point correlations allows us to obtain a clearer picture of the symmetry-

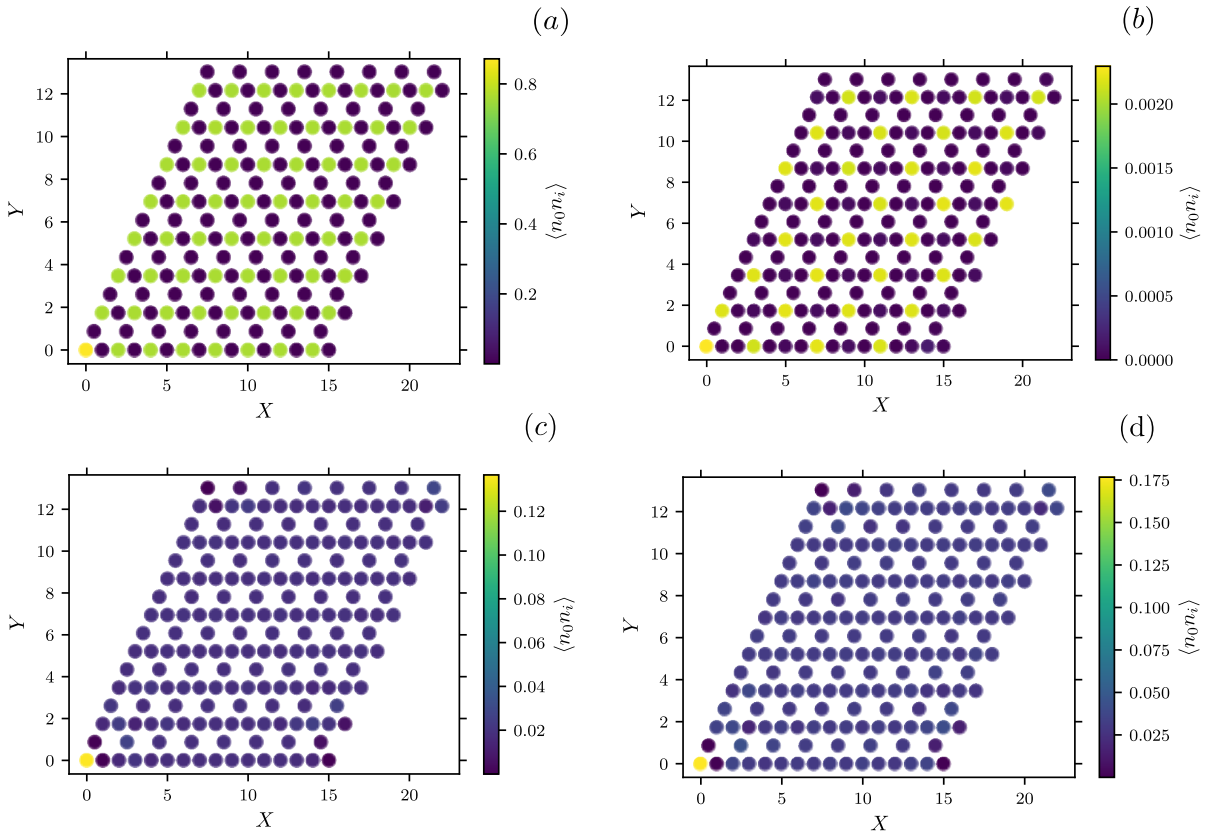

Figure 7.4: Plots of the two-point correlations $\langle n_{\mathbf{0}} n_{\boldsymbol{r}} \rangle$ for $L = 8$ at $\delta = 3.3$ with three different values of blockade radius R_b. The color bars illustrate the value of $\langle n_{\mathbf{0}} n_{\boldsymbol{r}} \rangle$ for each Rydberg atom in the lattice at position $\boldsymbol{r}$. (a) Nematic order: $R_b = 1.7$ and $R_c = 2$. (b) Staggered order: $R_b = 2.1$ and $R_c = 2$. Disordered state: $R_b = 2.1$ and $R_c = 4$ in panel (c), and $R_b = 1.95$ and $R_c = 2$ in panel (d).

breaking phases, as opposed to the density which can show the wrong picture when the RNN finds a superposition of different sectors as observed in some of our experiments for the nematic phase. Our results for the two-point correlation corroborate the DMRG finding. We also find that this state retains a nematic order when using a cutoff radius $R_c = 4$. Furthermore, we optimize our ansatz at the ground state of the point $R_b = 2.1$ and $\delta = 3.3$ with two different values of the cutoff radius $R_c = 2, 4$. For $R_c = 2$, the ground state is known to have a staggered order according to DMRG results. With our ansatz for $R_c = 2$, we find a broken symmetry state with a staggered order as shown in Fig. 7.4(b), such excitations are separated by a minimal distance $\sqrt{7}$ on the bulk. For $R_c = 4$, the ground state becomes disordered with a short-range order as illustrated in Fig. 7.4(c). The latter shows the importance of choosing a large cutoff radius to obtain realistic results, as demonstrated previously with QMC results applied to the Rydberg Hamiltonian on the square lattice [280].

In Ref. [275], the DMRG results suggest the potential existence of a spin liquid phase for this model. Our results suggest that the ground state at $R_b = 1.95$ and $\delta = 3.5$, which is in the expected spin liquid region according to Ref. [275], is rather a disordered state with no topological order. We first plot the correlations $\langle n_{\mathbf{0}} n_{\boldsymbol{r}} \rangle$ in Fig. 7.4(d). The results provide a hint that the extracted state has short-range order with a small correlation length.

To investigate the existence of a spin-liquid in this regime, we calculate the TEE γ using the Levin-Wen construction for a system size $L = 8$ (see Fig. 7.1, and for different values of $\delta \in [2.0, 3.7]$ at $R_b = 1.95$. We also do the same using the Kitaev-Preskill construction [99] in Fig. 7.1 (see App. C.2 for details about the construction). Our results, reported in Fig. 7.5(a), suggest that the TEE extracted by the RNN is consistent with zero and different from $\ln(2)$ within error bars. In a recent QMC study [281], it was suggested that the region, around $R_b = 1.95$ and the values of δ used in our study, contains an emergent spin-glass phase instead of a paramagnetic state. To check this claim, we compute the Edwards-Anderson (EA) order parameters [282, 283], defined as:

$$q_{\text{EA}} = \frac{\sum_{i=1}^{N} \langle n_i - \rho \rangle^2}{N\rho(1-\rho)}, \tag{7.3}$$

where N is the system size, n_i is the occupation number of site i and $\rho = (\sum_{i=1}^{N} n_i)/N$. Deviations of this order parameter from zero values are signals for the existence of a spin-glass phase. In Fig. 7.5(b), we plot this order parameter as a function of δ with $R_c = 2, 4$ and $R_b = 1.95$. We find that the order parameter values are consistent with zero as opposed to the results of QMC in Ref. [281]. This discrepancy in our results could be related to long auto-correlations times affecting the QMC results. Ref. [281] also uses

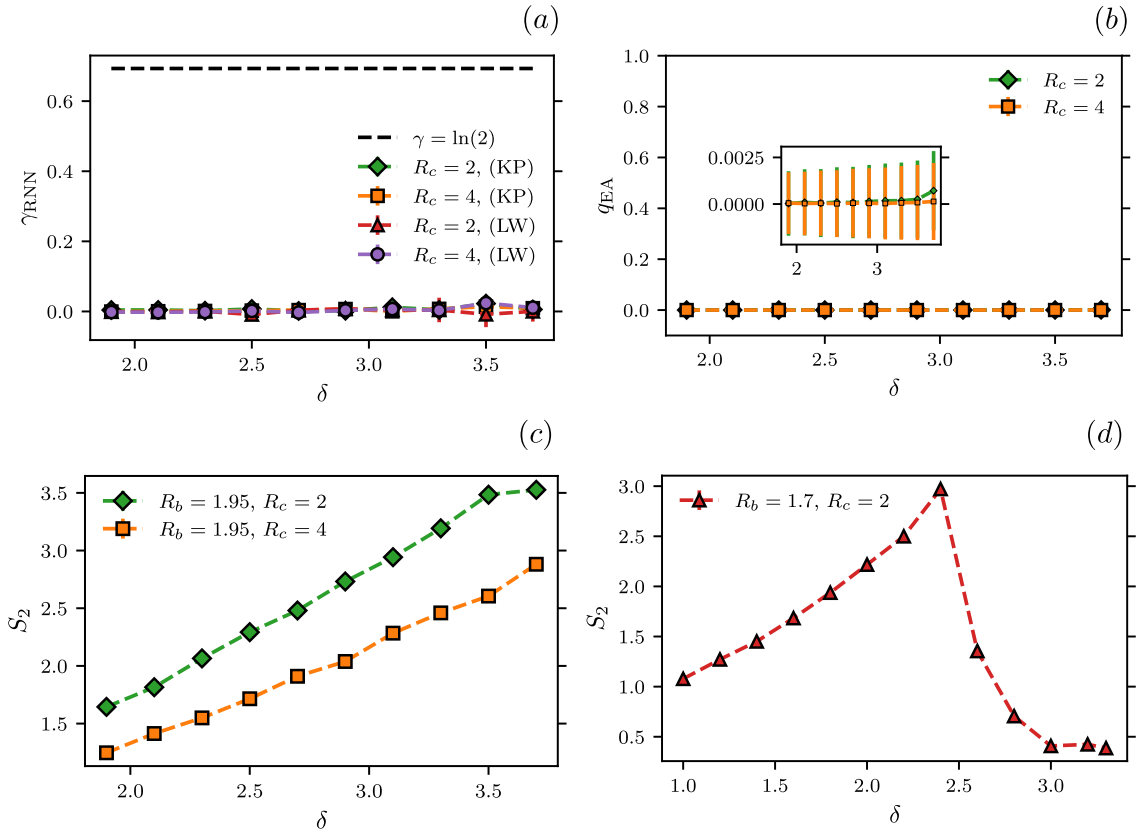

Figure 7.5: **Investigation of the liquid phase in the Rydberg atoms arrays on Kagome lattice**. A plot of the TEE γ_{RNN} and the Edwards-Anderson order parameter q_{EA} versus δ for two different values of the cutoff radius R_c (in the inset of panel (b), we zoom-in close to zero on the y-axis). For the TEE, we use Levin-Wen (LW) construction and the Kitaev-Preskill (KP) construction. For panels (a) and (b), we fix the blockade radius as $R_b = 1.95$. In panels (c) and (d), we report the second Renyi entropies S_2 for different values of R_b, R_c, and δ.

parallel tempering simulations, which are more suitable for spin-glass systems, in addition to traditional QMC. With these simulations, it was found that EA order parameters tend to be zero which corroborates our RNN findings.

All in all, these results suggest the non-existence of a spin liquid within our settings. Instead, our numerical findings corroborate the existence of a disordered state which can be highly entangled with increasing values of δ as illustrated by the second Renyi entropies in Fig. 7.5(c). We highlight that the second Renyi entropy can be used as an efficient measure to detect phase boundaries as suggested in Fig. 7.5(d) with a transition from a highly entangled paramagnetic phase to a nematic phase with lower entanglement at a transition point $\delta \approx 2.4$, which in close agreement with the phase diagram provided by DMRG [275]. We would like also to add that in our experiments, we observe signatures where our RNN finds a superposition of the three nematic sectors with an EE close to $\ln(3)$ at other values of R_b and δ in the nematic phase. This is in contrast to the behavior of DMRG and QMC with local updates which typically collapse to a specific nematic sector [275, 281].

Finally, we note that DMRG's conclusion of a liquid phase for this system in Ref. [275] could be an artifact of using open boundary conditions along the x-direction as opposed to fully periodic boundary conditions, as it was already anticipated in the same Ref. [275]. Future numerical investigations are vital for checking such hypotheses.

Conclusion

In this chapter, we demonstrated a successful application of neural network wave functions to the task of detecting topological order in quantum systems. We revealed their capability of estimating second Renyi entropies using the swap trick, with which we computed TEEs using finite-size scaling, Kitaev-Preskill constructions, and Levin-Wen constructions. Furthermore, the structural flexibility of the RNN offers the possibility to handle a wide variety of geometries including periodic boundary conditions in any spatial dimension which alleviate boundary effects on the TEE.

We have empirically demonstrated that 2D RNN wave functions support the 2D area law and can find a non-zero TEE for the toric code and for the hard-core Bose-Hubbard model on the Kagome lattice. We also find that RNNs favor coherent superpositions of minimally-entangled states over minimally-entangled states themselves. Additionally, we found a negative signal for a Z_2 topological order on the Rydberg atoms array in the Kagome lattice.

The success of our numerical experiments hinges on the combination of the exact sampling strategy used to compute observables, the structural properties of the RNN wave function, and the use of annealing as a strategy to overcome local minima during the optimization procedure.

The accuracy improvement of our findings can be achieved through the use of more advanced versions of RNNs and autoregressive models in general [108, 146], or even a hybrid approach that combines QMC and RNNs [284, 285]. Similarly, the incorporation of lattice symmetries provides a strategy to enhance the accuracy of our calculations [43, 104, 108]. Although our results match the anticipated behavior of the toric code and Bose-Hubbard spin liquid models, we highlight that the RNN wave function may be susceptible to spurious contributions to the TEE [271] and we have not addressed this issue in our work.

Finally, it is worth noting that our methods can be applied to study other systems displaying topological order, such as the Rydberg atom arrays on other lattices [257, 279, 286], either through variational methods or in combination with experimental data. To experimentally study topological order, it is possible to use quantum state tomography with RNNs [33]. This involves using experimental data to reconstruct the state seen in the experiment followed by an estimation of the TEE using the methods outlined in our work. Overall, our findings suggest that RNN wave functions have promising potential for discovering new phases of matter with topological order.

Chapter 8

RNN exact constructions: a comparison with other generative models

In the previous chapters, we numerically confirmed the ability of RNNs to approximate the ground state of prototypical many-body systems with different physical properties. With these promising empirical results, one might ask the question of whether an RNN is capable of providing an exact construction of probability distributions. In particular, those obtained from traditional quantum states through the Born rule. It would be also interesting to know whether RNN constructions can be done with a polynomial or exponential number of resources in terms of the number of degrees of freedom. In this chapter, we address one aspect of this question. More specifically, we aim to compare the RNN exact constructions of prototypical probability distributions with other different paradigms of generative modeling in machine learning. For classical generative modeling, we focus on RNNs and restricted Boltzmann machines (RBMs) [122, 287]. For quantum-inspired generative modeling, we use tensor networks (TNs) [288]. Finally, for quantum generative modeling, we provide the exact constructions using quantum circuit Born machines (QCBMs) [289–291]. These architectures are summarized in Fig. 8.1. This comparative study between different generative models is helpful to highlight the strengths and the weaknesses of each generative model. This chapter also sheds light on the different perspectives a probability distribution can be constructed using different generative models.

As an outline for this chapter, we briefly introduce each generative model in Sec. 8.1 in addition to RNNs that were introduced in Chap. 4. In Sec. 8.2, we present the different

constructions of the bimodal distribution, parity distribution, and cardinality distribution as well as the toric code distribution. We further demonstrate the ability of RNNs to encode the cardinality distribution with a lower number of compute resources compared to RBMs, TNs, and QCBMs.

8.1 Introduction to generative models

In recent years, classical generative modeling has known a stunning success with the development of deep learning models that are making state-of-the-art results in applications ranging from creating images from text [292], generating chat conversation that looks almost realistic [293] to molecular design [294, 295]. In the quantum world, generative modeling has also arisen as an interesting research direction that can showcase a quantum advantage compared to classical generative modeling [296]. In this section, we provide a brief overview of the quantum, quantum-inspired, and classical generative models used in our comparison on top of RNNs introduced in Chap. 4, before presenting our results in Sec. 8.2.

8.1.1 Quantum Circuit Born Machine

Quantum circuit Born machines (QCBMs) are one of the most well-known generative models that are based on quantum circuits. They are known for their expressive power [297], and for their ability to provide uncorrelated and independent samples, as opposed to energy-based models such as RBMs and traditional feed-forward neural networks. A QCBM can be built using a parametrized unitary $U(\boldsymbol{\theta})$ applied on an initial state $|0\rangle^{\otimes N}$ followed by a set of projective measurements as illustrated in Fig. 8.1(a). The QCBM probability distribution is given by the Born rule as:

$$P_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = |\langle \boldsymbol{\sigma}|U(\boldsymbol{\theta})|\mathbf{0}\rangle|^2. \tag{8.1}$$

The unitary $U(\boldsymbol{\theta})$ is traditionally built with one-gate and two-gate layers using different topologies, namely line (see Fig. 8.1(b)), star, and all-to-all topologies [289]. The uniqueness of QCBMs compared to other classical models stems from their ability to describe arbitrary quantum states provided a sufficient number of gates.

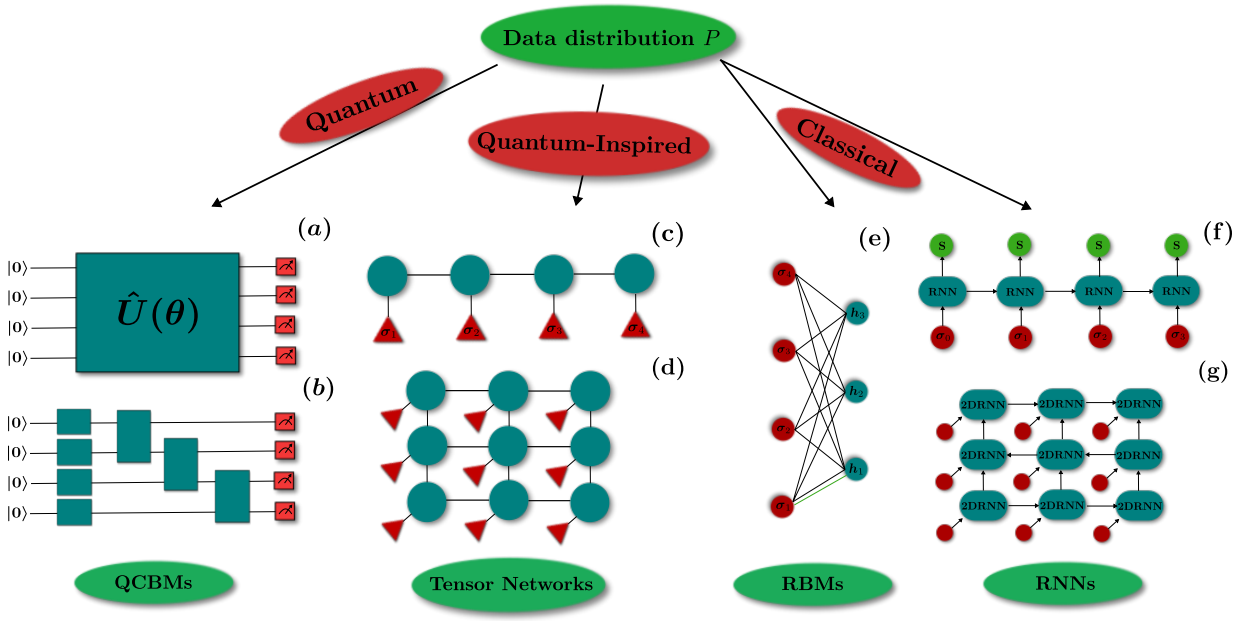

Figure 8.1: **Summary of the generative models used in this chapter to represent a data distribution** P. (a) QCBM illustration. (b) A special case of a QCBM with a line topology and two layers. (c) A diagram of a matrix product state (MPS) that can be used to compute the amplitude of a configuration $(\sigma_1, \sigma_2, \ldots, \sigma_n)$. (d) An illustration of the pair-projected entangled state (PEPS) that can naturally model 2D physical correlations as opposed to MPS. (e) An illustration of the RBM architecture, that has a visible layer with binary variables σ_n that are fully connected to a hidden layer made of binary variables h_i. (f) An illustration of one-dimensional RNN (1DRNN) that is made of a series of recurrent green blocks (RNN). Each block receives an input $\boldsymbol{\sigma}_{n-1}$ and a hidden state $\boldsymbol{h}_{n-1}$ and outputs a new hidden state $\boldsymbol{h}_{n-1}$, that is used as an input to the Softmax layer S. The latter allows computing the conditional probability of getting the next input σ_n. (g) A scheme of the two-dimensional RNN (2DRNN). Here each 2DRNN cell receives two hidden states and two spins from the horizontal and vertical neighboring 2DRNN cells. The autoregressive sampling path is a zigzag path as indicated by the horizontal arrows.

8.1.2 Tensor Networks

Tensor networks (TNs) are a class of models that have been originally developed in the context of condensed matter physics and that are transferred for use cases in Machine Learning. The main idea is to glue different tensors with contraction operations in order to encode a specific quantum state or a probability distribution through the Born rule [298–300]. In recent years, these architectures have been actively used in the context of machine learning as generative models [288].

In one dimension, one can define the so-called Matrix Product States (MPS) that are illustrated in Fig. 8.1(c), where each tensor takes an input $\boldsymbol{\sigma}_i$ that can be a one-hot encoding a spin degree of freedom. In this case, the MPS probability distribution is given by:

$$P_{\text{MPS}}(\boldsymbol{\sigma}) = \left| \frac{A_1^{[\sigma_1]} A_2^{[\sigma_2]} \dots A_N^{[\sigma_N]}}{\mathcal{N}} \right|^2 , \tag{8.2}$$

where $\mathcal{N}$ is the normalization factor that can be computed exactly with an efficient contraction. Each link in Fig. 8.1(c) corresponds to a tensor A_i index. The bulk tensors A_i can have a shape $2 \times \chi \times \chi$ where χ is the so-called bond dimension. The larger this quantity the more entanglement our MPS can store [299, 301].

Since an MPS requires a bond dimension that scales with the system width in two dimensions to efficiently encode the area law of entanglement, a two-dimensional version of TNs has been devised to efficiently encode 2D correlations in a physical system of interest [302]. This class of TNs is called pair-entangled projected states (PEPS) and is illustrated in Fig. 8.1(d). Here the tensors are arranged on a 2D grid and can have up to four external indices in addition to the physical indices index by the red triangles. The challenge that is associated with this architecture is the expensive cost of contracting a PEPS, which makes MPS more frequently used in the literature [299].

8.1.3 Restricted Boltzmann Machines

A restricted Boltzmann machine (RBM) [26] is a stochastic neural network that can be built using an energy function in a similar fashion to Boltzmann distributions in statistical physics. RBMs have been used as variational states to target the ground state of quantum many-body systems [37] and they are known for their ability to model quantum states such as topological states [303].

The RBM is built with two connected layers: a visible layer and a hidden layer as illustrated in Fig. 8.1(e). The RBM distribution is given by:

$$P_{\text{RBM}}(\boldsymbol{\sigma}) = \sum_{\boldsymbol{h}} P(\boldsymbol{\sigma}, \boldsymbol{h}), \tag{8.3}$$

where

$$P(\boldsymbol{\sigma}, \boldsymbol{h}) = \frac{e^{-E(\boldsymbol{\sigma}, \boldsymbol{h})}}{Z}. \tag{8.4}$$

Here, $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_N)$ where $\sigma_i \in \{+1, -1\}$ denotes the visible variables and $\boldsymbol{h} = (h_1, \ldots, h_{N_h})$, with $h_j \in \{+1, -1\}$, for all $j = 1, N_h$, denotes the hidden variables. Furthermore

$$E(\boldsymbol{\sigma}, \boldsymbol{h}) = -\sum_{i=1}^{N}\sum_{j=1}^{N_h} W_{ij}\sigma_i h_j - \sum_{i=1}^{N} a_i\sigma_i - \sum_{j=1}^{N_h} b_j h_j, \tag{8.5}$$

denotes the energy function, where W_{ij} are the couplings between the hidden and visible variables. Additionally, a_i and b_j are the visible and the hidden biases respectively. Finally

$$Z = \sum_{\boldsymbol{\sigma}, \boldsymbol{h}} e^{-E(\boldsymbol{\sigma}, \boldsymbol{h})}, \tag{8.6}$$

is the normalization constant, also known as the partition function.

8.2 Results

We now focus our attention on the comparison between RBMs, RNNs, TNs, and QCBMs in terms of representing different classical probability distributions. To be able to compare the different constructions of the generative models, we use the number of resources as a metric. For an RBM, RNN, and a TN, this metric corresponds to the computational complexity to do a forward pass of a bitstring configuration $\boldsymbol{\sigma}$ to obtain its associated probability. For a parametrized quantum circuit, the number of resources corresponds to the number of local gates used in the quantum circuit.

Bimodal distribution. Let us get started with the *bimodal distribution*, which can be obtained from the GHZ state through the Born rule. It is defined as follows:

$$P_{\text{bimodal}}(\boldsymbol{\sigma}) = \frac{1}{2}\left(\prod_{i=1}^{N}\delta_{\sigma_i, 0} + \prod_{i=1}^{N}\delta_{\sigma_i, 1}\right). \tag{8.7}$$

We report the scaling of resources of the different constructions in Tab. 8.2, with more details about each construction in App. D. For this distribution, all of the generative models need $\mathcal{O}(N)$ resources for an exact construction. In particular, the RBM is able to represent the bimodal distribution with one hidden neuron. Additionally, the TN construction is done with an MPS that has a bond dimension $\chi = 2$, whereas the RNN construction corresponds to a hidden dimension $d_h = 2$. Finally, the QCBM is able to build this distribution using a linear topology of Hadamard and CNOT gates.

Parity (Evens) distribution. We now shift our focus to the *parity distribution* which is defined as:

$$P_{\text{even}}(\boldsymbol{\sigma}) = \begin{cases} \frac{1}{2^{N-1}} & \text{if } \sum_{i=1}^{N} \sigma_i \text{ is even}\,, \\ 0, & \text{otherwise}\,. \end{cases} \tag{8.8}$$

This distribution is related to the parity function which is known for its theoretical role in investigating circuit complexity of Boolean functions [304].

As shown in Tab. 8.2. This construction can be achievable with RBMs, RNNs, TNs, and PQC models with linear scaling in the required resources. For the RBM, complex numbers with one hidden variable h are needed to obtain an optimal construction. It is conjectured that for a real-valued RBM, an exponential number of hidden units is needed to construct this state. See App. D for details. With respect to the MPS and the RNN, they can both construct this distribution using a bond dimension/hidden dimension equal to 2. Finally, our QCBM exact construction of this distribution uses a linear number of XX gates. More details about the constructions can be found in App. D.

Cardinality distribution. This distribution for a hamming weight k is given by the following:

$$P_{\text{card}}(\boldsymbol{\sigma}) = \begin{cases} 1/\binom{N}{k} & \text{if } \sum_{i=1}^{N} \sigma_i = k\,, \\ 0 & \text{otherwise}\,, \end{cases} \tag{8.9}$$

where $k < N$ and N is the total number of bits in $\boldsymbol{\sigma}$. The cardinality distribution can be constructed from Dicke's state through the Born rule [305]. We can typically find this distribution in combinatorial portfolio optimization or in quantum many-body systems with a fixed magnetization sector.

The models' constructions of this distribution are provided in App. D. The exact construction can be achieved in a linear scaling of resources by the RNN. For TNs and QCBMs, it can be obtained with linear scaling in kN. If k is at the order of the system size N, then the previous scaling becomes quadratic in terms of system size which make it less favorable compared to RNNs. For the RBM, we provide construction with a quadratic number of resources in terms of the number of bits. We note that the RBM, TN, and

QCBM construction correspond to our best attempts and might not be optimal, so they are subject to further possible improvements. More details can be found in App. D.

The advantage in scaling for the RNN can be related to the flexibility in choosing a non-linearity and to the ability of RNNs to encode information about the hamming weight k in the biases of the RNN cell in contrast to TNs.

Toric code distribution. As discussed in Sec. 7.2, the toric code state is a type of stabilizer code that allows encoding information about a logical qubit using a two-dimensional array of physical qubits. This state allows for fault-tolerant quantum computing [261, 306]. This state corresponds to the ground state of the 2D Kitaev's toric code with periodic boundary conditions, such that:

$$\hat{H} = -\sum_p \hat{B}_p - \sum_v \hat{A}_v.$$

Here the first summation is on the plaquettes $\hat{B}_p$ and the second summation is on the vertices $\hat{A}_v$. This ground state is an eigenvector of the plaquettes $\hat{B}_p = \Pi_{i\in p}\hat{\sigma}_i^z$ and the vertices $\hat{A}_v = \Pi_{i\in v}\hat{\sigma}_i^x$ with eigenvalues equal to 1. Thus, the toric code ground state can be written using projectors as

$$|\Psi_{\mathrm{TC}}\rangle = \Pi_p \Pi_v \left(\frac{1+\hat{B}_p}{2}\right)\left(\frac{1+\hat{A}_v}{2}\right)|0\rangle^{\otimes N} \tag{8.10}$$

$$= \Pi_p \left(\frac{1+\hat{B}_p}{2}\right)|0\rangle^{\otimes N}, \tag{8.11}$$

$$= \Pi_v \left(\frac{1+\hat{A}_v}{2}\right)|0\rangle^{\otimes N}. \tag{8.12}$$

The second equality shows that it is enough to project each plaquette configuration to $+1$ to construct the ground state. The third equality also shows that it is sufficient to project each vertex configuration to $+1$ to obtain our toric code ground state. In this work, we consider the *toric code distribution* $P_{\mathrm{TC}}(\boldsymbol{\sigma}) = |\langle \boldsymbol{\sigma}|\Psi_{\mathrm{TC}}\rangle|^2$ that is obtained from the toric code state using the Born rule. As shown in Tab. 8.2, the toric code distribution can be built using a 2DRNN and PEPS with the same bond dimension $\chi = 2$/hidden dimension $d_h = 2$. For a PEPS, an exact contraction of the tensors provided in App. D is needed to obtain the probability of a certain bitstring configuration. However, the task of contracting a PEPS is known to be #P [307] as indicated in Tab. 8.2. For a 2DRNN, the computation of the probability of a configuration is still efficient and can be done with a

	RBM	RNN	TN	QCBM
Bimodal	$\mathcal{O}(N)$	$\mathcal{O}(N)$	$\mathcal{O}(N)$	$\mathcal{O}(N)$
Parity	$\mathcal{O}(N)$	$\mathcal{O}(N)$	$\mathcal{O}(N)$	$\mathcal{O}(N)$
Cardinality	$\mathcal{O}(N^2)$	$\boldsymbol{\mathcal{O}(N)}$	$\mathcal{O}(kN)$	$\mathcal{O}(kN)$ [308]
Toric code	$\mathcal{O}(N)$ [303]	$\mathcal{O}(N)$	#P contraction [307]	$\mathcal{O}(N)$ [258]

Table 8.1: A table representing the number of resources needed for each architecture to exactly represent the bimodal, parity, cardinality, and toric code distributions. We note that in our definition of the number of resources, we do not take into account the possibility of parameter repetition since this observation does not necessarily reduce the computational complexity. The citations in the table correspond to the constructions that were found in the literature. The RNN is found to have the best construction (in bold) for the cardinality dataset. For the TN construction of the toric code distribution, we use a PEPS which is known in the literature for the expensive cost of an exact contraction. The latter is known to be an #P problem [307].

linear complexity with the system size N. The ability of RNNs to construct the 2D toric distribution in App. D reinforces our findings in Chap. 7, where we numerically show that RNNs can encode the topological order of the 2D toric code.

Additionally, we also provide an RBM construction with a linear number of hidden variables and local connectivity [303]. The QCBM construction is built using Hadamard and CNOT gates on a 2D quantum circuit [258]. All in all, RNNs, RBMs, and QCBMs achieve linear scaling in terms of the number of required resources to construct this distribution as shown in App. D.

8.3 Conclusion

In this chapter, we provided a survey of the exact constructions of different synthetic probability distributions using four generative models (QCBMs, TNs, RBMs, and RNNs). We observe a similarity in the scaling of resources for two distributions out of four. The latter outlines the potential of QCBMs to compete with quantum-inspired and classical generative models. Additionally, the advantage of RNNs compared to the other generative models on the cardinality distribution outlines the importance of a flexible choice of non-linear activation functions in a generative model. This observation motivates the use of non-linear in quantum circuits as shown in Ref. [309]. Furthermore, we highlight the advantage of RNNs, RBMs, and QCBMs compared to PEPS when shifting our attention

to the toric code distribution. In particular, we demonstrate that 2DRNNs can provide an exact construction for the toric code distribution, with an efficient cost of a forward pass in comparison to the PEPS architecture. We would like also to mention that there is a possible space for optimizing the TN, QCBM, and RBM construction of the cardinality distribution where we see a scaling advantage of RNNs.

Furthermore, we note that the exact constructions provide a valuable tool for the physics community to have different perspectives on constructing a probability distribution. These constructions are also potentially helpful to pre-train a generative model, especially when the target distribution is related to a prototypical distribution that we can exactly construct using our generative model. Additionally, by comparing RNNs and TNs constructions, we note that for three distributions the models require the same bond dimension/hidden dimension. This observation could be a consequence of possible mapping from TNs to RNNs [146, 310, 311]. Furthermore, we would like to outline the possibility to map a TN to a QCBM [312], as well as the mapping between TNs and RBMs [313]. These mappings are very helpful to map a model's construction to another model if it is difficult to construct the distribution directly. This idea is also helpful to pre-train a generative model using a construction from another generative model. Finally, we note that in this work, we do not explore the use of complex RNN wave functions 4.5 and we believe there is still space for exploration in order to find exact constructions of traditional quantum states with sign structure.

Chapter 9

Conclusions and Outlooks

9.1 Conclusions

In this thesis, we have introduced a class of wave functions based on RNNs that were originally developed in the natural language processing community of machine learning. RNNs belong to the class of autoregressive generative models that have the autoregressive property, and which can allow sampling perfect and uncorrelated configurations from complex multi-modal distributions. Importantly, we have demonstrated RNN ansatzes are highly flexible and very efficient for performing variational calculations through the VMC scheme. More precisely, they can be easily extended to multiple spatial dimensions with an efficient cost that is cheaper compared to PEPS in the 2D case. We also used dilated RNNs when the spatial dimension is not well-defined such as in the case of fully-connected spin-glass models. We further generalized traditional RNNs to complex RNN wave functions to target non-stoquastic Hamiltonian with a non-trivial sign structure. Additionally, we showed that we can apply discrete symmetries as well as the $U(1)$ symmetry to improve ground state estimates, as well as to target specific excited state sectors for the purpose of computing low-energy excitation gaps. Note that the weight-sharing feature of RNNs is a good bias for targeting many-body systems with translation invariance in the bulk more efficiently. By abandoning the common practice of using weight-sharing, we can extend the use of RNNs to the study of disordered systems such as spin-glass models. Furthermore, we illustrated the possibility of using RNNs to target exotic lattices such as the Kagome lattice.

While DMRG is the gold standard numerical method in 1D, we have demonstrated that our 2D RNN wave function is very competitive with DMRG and can lead to state-of-the-

art results in 2D. In particular, we demonstrated the ability of 2D RNNs to outperform DMRG on the 2D TFIM on the square lattice as well as on the 2D triangular Heisenberg model, while using orders of magnitude fewer variational parameters compared to DMRG. We also showed that accuracy can be systematically improved by increasing the number of hidden/memory units in the RNN ansatz. We further showcase the possibility to estimate correlation functions and entanglement entropies.

In this work, we also developed a variational scheme to emulate classical and quantum annealing for the purpose of solving classical combinatorial optimization problems as another use case of RNNs. We first develop the variational classical annealing (VCA) scheme for simulating a variational version of classical annealing by adding an entropy term to the energy cost function. We also demonstrate the quantum counterpart that we denote as variational quantum annealing (VQA) for which, we derived a convergence bound of this algorithm with similar properties to the adiabatic theorem of quantum mechanics. We found that our VCA scheme is more advantageous compared to the VQA scheme on the random Ising chain instances as well as on the 2D Edwards-Anderson spin-glass model. While our results suggest a potential VCA advantage, we refer to the possibility of finding a VQA advantage in other instances. By comparing to traditional Monte Carlo implementations of simulating annealing and simulated quantum annealing, we show that our VCA is more advantageous on average. This observation is likely to be related to the autoregressive property of RNNs that allows sampling different modes, as opposed to Markov-chain Monte Carlo methods, which can be stuck in a specific configuration at low temperatures, especially in a spin-glass model. We further extend the use of VCA with RNNs to real-world combinatorial optimization problems namely the maximum-cut problem, nurse scheduling problem, and the traveling salesman problem where we find an advantage of VCA compared to the traditional Monte Carlo implementation of simulated annealing in the average case.

The developed annealing scheme turns out to be also helpful in targeting frustrated systems in order to mitigate the effect of local minima in a VMC calculation, such as in the triangular Heisenberg model. We also make use of the annealing technique to help in the investigation of the topological properties of quantum many-body systems through the estimation of topological entanglement entropies. We demonstrated that RNNs are capable of encoding different topological sectors as well as sampling those sectors by virtue of the autoregressive property. Additionally, we illustrated that RNNs are capable of making predictions about the topological properties of a real-world quantum many-body system with potential topological order, namely the Rydberg atoms arrays. In particular, we have shown that Rydberg atoms array on the Kagome lattice do not establish a topological order within the regime explored by DMRG in a previous study. We highlight that this

investigation is an important step toward the use of RNNs in practical settings where the physics of a certain many-body system is not well-understood.

To further show the potential of RNNs, we have established different exact constructions of traditional probability distributions using specific RNN parameters as a first step toward understanding why RNNs have worked so well at the various tasks described in this thesis. We compared these constructions to other generative models' constructions, and we highlighted the advantage of RNN in terms of compute resources, thanks to the flexible choice of non-linear activation functions in RNN cells.

9.2 Outlooks

An interesting future research direction is to explore the feasibility of using RNNs to study fermionic quantum systems in order to use these tools for helping to solve open questions in the physics of strongly correlated electrons. We further highlight the importance of conducting more RNN benchmarks in order to learn more about the advantages and limitations of RNNs. Additionally, similar to the thermodynamic limit algorithm in DMRG, it would be valuable to search for possibilities where RNNs can generalize to the thermodynamic limit. We would like to highlight that although RNNs have provided promising results in this study, more work is to be done in order to better understand the physics that links the hyperparameters and the design choice of RNNs with the accuracy of a certain variational calculation. We believe that a more comprehensive understanding would be very helpful in developing better RNN cells and in improving the variational results presented in this work. It would be also interesting to incorporate automated hyperparameter search tools [314, 315] into our framework to make it more user-friendly and to speed up the pace of progress in this area of research.

We also envision the promise of using RNNs in studying real-world quantum systems through a hybrid approach, where RNNs can be trained using experimental data, as well as through the VMC scheme as highlighted in Ref. [285]. Additionally, with the recent progress in large language models (LLMs) such as Chat-GPT and GPT-4, there is a strong research potential for using novel NLP tools to explore the unknown corners of many-body physics. We expect that going from the scale of millions of parameters explored in this thesis to billions of parameters, just like LLMs, could lead to major improvements to the results presented in this study. We also envision that these new advances could lead to new discoveries about the unknown physics of strongly correlated systems, that are hard to simulate using state-of-the-art algorithms, especially in two and three spatial dimensions. This development has to be in alignment with ethical considerations of the training cost

of these architectures and its consequences on the environment. We also hope that the RNN-based tools developed in this thesis to be a valuable toolbox to physicists in the numerical condensed matter community and also to other relevant areas of science.

References

[1] S. Suzuki, J.-i. Inoue, and B. K. Chakrabarti, *Quantum Ising Phases and Transitions in Transverse Ising Models*, Vol. 859 (2013).

[2] F. Becca and S. Sorella, *Quantum monte carlo approaches for correlated systems* (Cambridge University Press, 2017).

[3] E. Berg, S. Lederer, Y. Schattner, and S. Trebst, "Monte carlo studies of quantum critical metals", Annual Review of Condensed Matter Physics **10**, 63–84 (2019).

[4] L. Pollet, "A review of monte carlo simulations for the bose–hubbard model with diagonal disorder", Comptes Rendus Physique **14**, 712–724 (2013).

[5] M. Troyer and U.-J. Wiese, "Computational complexity and fundamental limitations to fermionic quantum monte carlo simulations", Physical review letters **94**, 170201 (2005).

[6] R. P. Feynman, "Atomic theory of the λ transition in helium", Physical Review **91**, 1291 (1953).

[7] W. L. McMillan, "Ground state of liquid He^4", Phys. Rev. **138**, A442–A451 (1965).

[8] R. Orús, "Tensor networks for complex quantum systems", Nature Reviews Physics **1**, 538–550 (2019).

[9] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, "A variational eigenvalue solver on a photonic quantum processor", Nature Communications **5**, 1–7 (2014).

[10] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, and J. Tennyson, "The variational quantum eigensolver: a review of methods and best practices", Physics Reports **986**, 1–128 (2022).

[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning", Nature **521**, 436 (2015).

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in Proceedings of the 25th international conference on neural information processing systems - volume 1, NIPS'12 (2012), pp. 1097–1105.

[13] S. Balaban, "Deep learning and face recognition: the state of the art", **9457**, 94570B (2015).

[14] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks", arXiv, arXiv:1303.5778, 1303.5778 (2013).

[15] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing", arXiv, arXiv:1708.02709, 1708.02709 (2017).

[16] C. Badue, R. Guidolini, R. Vivacqua Carneiro, P. Azevedo, V. Brito Cardoso, A. Forechi, L. Ferreira Reis Jesus, R. Ferreira Berriel, T. Meireles Paixão, F. Mutz, T. Oliveira-Santos, and A. Ferreira De Souza, "Self-Driving Cars: A Survey", arXiv, arXiv:1901.04407, 1901.04407 (2019).

[17] D. Novak and R. Riener, "Control strategies and artificial intelligence in rehabilitation robotics", Ai Magazine **36**, 23 (2015).

[18] A. Huang and R. Wu, "Deep learning for music", arXiv prpages arXiv:1606.04930 (2016).

[19] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge", Nature **550**, Article, 354 EP - (2017).

[20] D. Fooshee, A. Mood, E. Gutman, M. Tavakoli, G. Urban, F. Liu, N. Huynh, D. Van Vranken, and P. Baldi, "Deep learning for chemical reaction prediction", Mol. Syst. Des. Eng. **3**, 442–452 (2018).

[21] Y. Li, H. Kang, K. Ye, S. Yin, and X. Li, "Foldingzero: protein folding from scratch in hydrophobic-polar model", arXiv prpages arXiv:1812.00967 (2017).

[22] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, "Model-free prediction of large spatiotemporally chaotic systems from data: a reservoir computing approach", Phys. Rev. Lett. **120**, 024102 (2018).

[23] I. Wallach, M. Dzamba, and A. Heifets, "AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-based Drug Discovery", arXiv, arXiv:1510.02855, 1510.02855 (2015).

[24] A. S. Lundervold and A. Lundervold, "An overview of deep learning in medical imaging focusing on mri", Zeitschrift für Medizinische Physik, https://doi.org/10.1016/j.zemedi.2018.11.002 (2018).

[25] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, "Machine learning and the physical sciences", Reviews of Modern Physics **91**, 10.1103/revmodphys.91.045002 (2019).

[26] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines", Cognitive science **9**, 147–169 (1985).

[27] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", en, Proc Natl Acad Sci U S A **79**, 2554–2558 (1982).

[28] J. Carrasquilla and R. G. Melko, "Machine learning phases of matter", Nature Physics **13**, 431–434 (2017).

[29] P. Broecker, J. Carrasquilla, R. G. Melko, and S. Trebst, "Machine learning quantum phases of matter beyond the fermion sign problem", Scientific Reports **7**, 8823 (2017).

[30] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, "Machine learning phases of strongly correlated fermions", Phys. Rev. X **7**, 031038 (2017).

[31] C. Miles, R. Samajdar, S. Ebadi, T. T. Wang, H. Pichler, S. Sachdev, M. D. Lukin, M. Greiner, K. Q. Weinberger, and E.-A. Kim, "Machine learning discovery of new phases in programmable quantum simulator snapshots", 10.48550/ARXIV.2112.10789 (2021).

[32] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, "Neural-network quantum state tomography", Nature Physics **14**, 447 (2018).

[33] J. Carrasquilla, G. Torlai, R. G. Melko, and L. Aolita, "Reconstructing quantum states with generative models", Nature Machine Intelligence **1**, 155–161 (2019).

[34] J. Androsiuk, L. Kulak, and K. Sienicki, "Neural network solution of the Schrödinger equation for a two-dimensional harmonic oscillator", Chemical Physics **173**, 377–383 (1993).

[35] C. Roth, "Iterative retraining of quantum spin models using recurrent neural networks", 10.48550/ARXIV.2003.06228 (2020).

[36] "Artificial neural network methods in quantum mechanics", Computer Physics Communications **104**, 1–14 (1997).

[37] G. Carleo and M. Troyer, "Solving the quantum many-body problem with artificial neural networks", Science **355**, 602–606 (2017).

[38] Z. Cai and J. Liu, "Approximating quantum many-body wave functions using artificial neural networks", Phys. Rev. B **97**, 035116 (2018).

[39] D. Luo and B. K. Clark, "Backflow transformations via neural networks for quantum many-body wave functions", Phys. Rev. Lett. **122**, 226401 (2019).

[40] D. Pfau, J. S. Spencer, A. G. D. G. Matthews, and W. M. C. Foulkes, "Iab initio/i solution of the many-electron schrödinger equation with deep neural networks", Physical Review Research **2**, 10.1103/physrevresearch.2.033429 (2020).

[41] J. Hermann, Z. Schätzle, and F. Noé, "Deep-neural-network solution of the electronic schrödinger equation", Nature Chemistry **12**, 891–897 (2020).

[42] K. Choo, A. Mezzacapo, and G. Carleo, "Fermionic neural-network states for ab-initio electronic structure", Nature Communications **11**, 2368 (2020).

[43] M. Hibat-Allah, M. Ganahl, L. E. Hayward, R. G. Melko, and J. Carrasquilla, "Recurrent neural network wave functions", Phys. Rev. Research **2**, 023358 (2020).

[44] F. Vicentini, A. Biella, N. Regnault, and C. Ciuti, "Variational neural-network ansatz for steady states in open quantum systems", Phys. Rev. Lett. **122**, 250503 (2019).

[45] D. Luo, Z. Chen, J. Carrasquilla, and B. K. Clark, "Autoregressive neural network for simulating open quantum systems via a probabilistic formulation", Phys. Rev. Lett. **128**, 090501 (2022).

[46] B. Jónsson, B. Bauer, and G. Carleo, "Neural-network states for the classical simulation of quantum computing", 10.48550/ARXIV.1808.05232 (2018).

[47] M. Medvidović and G. Carleo, "Classical variational simulation of the quantum approximate optimization algorithm", npj Quantum Information **7**, 101 (2021).

[48] J. Carrasquilla, D. Luo, F. Pérez, A. Milsted, B. K. Clark, M. Volkovs, and L. Aolita, "Probabilistic simulation of quantum circuits using a deep-learning architecture", Phys. Rev. A **104**, 032610 (2021).

[49] V. Dunjko and H. J. Briegel, "Machine learning & artificial intelligence in the quantum domain: a review of recent progress", Reports on Progress in Physics **81**, 074001 (2018).

[50] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, "Machine learning and the physical sciences", Rev. Mod. Phys. **91**, 045002 (2019).

[51] J. Carrasquilla, "Machine learning for quantum matter", Advances in Physics: X **5**, 1797528 (2020).

[52] A. Dawid, J. Arnold, B. Requena, A. Gresch, M. Płodzień, K. Donatella, K. A. Nicoli, P. Stornati, R. Koch, M. Büttner, R. Okuła, G. Muñoz-Gil, R. A. Vargas-Hernández, A. Cervera-Lierta, J. Carrasquilla, V. Dunjko, M. Gabrié, P. Huembeli, E. van Nieuwenburg, F. Vicentini, L. Wang, S. J. Wetzel, G. Carleo, E. Greplová, R. Krems, F. Marquardt, M. Tomza, M. Lewenstein, and A. Dauphin, "Modern applications of machine learning in quantum sciences", arXiv, 2204.04198 (2022).

[53] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need", arXiv, 1706.03762 (2017).

[54] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: pre-training of deep bidirectional transformers for language understanding", arXiv, 1810.04805 (2018).

[55] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: generalized autoregressive pretraining for language understanding", arXiv, 1906.08237 (2019).

[56] OpenAI, "Gpt-4 technical report", arXiv, 2303.08774 (2023).

[57] S. Hochreiter and J. Schmidhuber, "Long short-term memory", Neural Comput. **9**, 1735–1780 (1997).

[58] A. Graves, S. Fernandez, and J. Schmidhuber, "Multi-dimensional recurrent neural networks", 10.48550/ARXIV.0705.2011 (2007).

[59] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation", arXiv, 1406.1078 (2014).

[60] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling", arXiv, 1412.3555 (2014).

[61] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning", arXiv, 1506.00019 (2015).

[62] D. Rodney, "Statistical physics course", Ecole Normale Superieure de Lyon (2017).

[63] M. Kardar, *Statistical physics of particles* (Cambridge University Press, 2007).

[64] A. W. Sandvik, A. Avella, and F. Mancini, "Computational studies of quantum spin systems", in AIP conference proceedings (2010).

[65] N. Goldenfeld, *Lectures on phase transitions and the renormalization group* (CRC Press, 2018).

[66] J. Cardy, *Scaling and renormalization in statistical physics*, Vol. 5 (Cambridge university press, 1996).

[67] P. W. Anderson, "More is different", Science **177**, 393–396 (1972).

[68] M. B. Hastings, I. González, A. B. Kallin, and R. G. Melko, "Measuring renyi entanglement entropy in quantum monte carlo simulations", Physical Review Letters **104**, 10.1103/physrevlett.104.157201 (2010).

[69] D. M. Greenberger, M. A. Horne, and A. Zeilinger, "Going beyond bell's theorem", in *Bell's theorem, quantum theory and conceptions of the universe*, edited by M. Kafatos (Springer Netherlands, Dordrecht, 1989), pp. 69–72.

[70] S. Sachdev, *Quantum phase transitions*, 2nd ed. (Cambridge University Press, 2011).

[71] P. Anderson, "Theory of dirty superconductors", Journal of Physics and Chemistry of Solids **11**, 26–30 (1959).

[72] M. Ma and P. A. Lee, "Localized superconductors", Phys. Rev. B **32**, 5658–5667 (1985).

[73] R. P. Feynman, "Space-time approach to non-relativistic quantum mechanics", Rev. Mod. Phys. **20**, 367–387 (1948).

[74] R. G. Melko, "Stochastic series expansion quantum monte carlo", in *Strongly correlated systems: numerical methods*, edited by A. Avella and F. Mancini (Springer Berlin Heidelberg, Berlin, Heidelberg, 2013), pp. 185–206.

[75] T. H. Hsieh, "From d-dimensional quantum to d+ 1-dimensional classical systems", Student review,(2) **1** (2016).

[76] J. Bardeen, in *Cooperative phenomena* (Springer, 1973), pp. 63–78.

[77] R. B. Laughlin, "Anomalous quantum hall effect: an incompressible quantum fluid with fractionally charged excitations", Phys. Rev. Lett. **50**, 1395–1398 (1983).

[78] L. Masanes, "Area law for the entropy of low-energy states", Physical Review A **80**, 10.1103/physreva.80.052104 (2009).

[79] S. Ruder, "An overview of gradient descent optimization algorithms", 10.48550/ARXIV.1609.04747 (2016).

[80] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization", 10.48550/ARXIV.1412.6980 (2014).

[81] S.-i. Amari, "Natural Gradient Works Efficiently in Learning", Neural Computation **10**, 251–276 (1998).

[82] S. Sorella, "Green function monte carlo with stochastic reconfiguration", Physical Review Letters **80**, 4558–4561 (1998).

[83] J. Martens and R. Grosse, "Optimizing neural networks with kronecker-factored approximate curvature", 10.48550/ARXIV.1503.05671 (2015).

[84] J. Martens, J. Ba, and M. Johnson, "Kronecker-factored curvature approximations for recurrent neural networks", in Iclr (2018).

[85] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems", Journal of research of the National Bureau of Standards **49**, 409–435 (1952).

[86] F. Vicentini, D. Hofmann, A. Szabó, D. Wu, C. Roth, C. Giuliani, G. Pescia, J. Nys, V. Vargas-Calderón, N. Astrakhantsev, and G. Carleo, "NetKet 3: machine learning toolbox for many-body quantum systems", SciPost Physics Codebases, 10.21468/scipostphyscodeb.7 (2022).

[87] S.-X. Zhang, Z.-Q. Wan, and H. Yao, "Automatic differentiable monte carlo: theory and application", 10.48550/ARXIV.1911.09117 (2019).

[88] A. Chen and M. Heyl, "Efficient optimization of deep neural quantum states toward machine precision", 10.48550/ARXIV.2302.01941 (2023).

[89] C. Gros, "Criterion for a good variational wave function", Phys. Rev. B **42**, 6835–6838 (1990).

[90] R. Assaraf and M. Caffarel, "Zero-variance zero-bias principle for observables in quantum monte carlo: application to forces", The Journal of Chemical Physics **119**, 10536–10552 (2003).

[91] C. Hubig, J. Haegeman, and U. Schollwöck, "Error estimates for extrapolations with matrix-product states", Physical Review B **97**, 10.1103/physrevb.97.045125 (2018).

[92] M. C. Bañuls, D. A. Huse, and J. I. Cirac, "How much entanglement is needed to reduce the energy variance?", arXiv, 1912.07639 (2019).

[93] D. Wu, R. Rossi, F. Vicentini, N. Astrakhantsev, F. Becca, X. Cao, J. Carrasquilla, F. Ferrari, A. Georges, M. Hibat-Allah, M. Imada, A. M. Läuchli, G. Mazzola, A. Mezzacapo, A. Millis, J. R. Moreno, T. Neupert, Y. Nomura, J. Nys, O. Parcollet, R. Pohle, I. Romero, M. Schmid, J. M. Silvester, S. Sorella, L. F. Tocchio, L. Wang, S. R. White, A. Wietek, Q. Yang, Y. Yang, S. Zhang, and G. Carleo, "Variational benchmarks for quantum many-body problems", 10.48550/ARXIV.2302.04919 (2023).

[94] T. Kashima and M. Imada, "Path-integral renormalization group method for numerical study on ground states of strongly correlated electronic systems", Journal of the Physical Society of Japan **70**, 2287–2299 (2001).

[95] S. Sorella, "Generalized lanczos algorithm for variational quantum monte carlo", Physical Review B **64**, 10.1103/physrevb.64.024512 (2001).

[96] R. Assaraf and M. Caffarel, "Zero-variance principle for monte carlo algorithms", Physical Review Letters **83**, 4682–4685 (1999).

[97] S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih, "Monte carlo gradient estimation in machine learning", arXiv, 1906.10652 (2019).

[98] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation", in Advances in neural information processing systems (2000), pp. 1057–1063.

[99] A. Kitaev and J. Preskill, "Topological entanglement entropy", Phys. Rev. Lett. **96**, 110404 (2006).

[100] M. Levin and X.-G. Wen, "Detecting topological order in a ground state wave function", Physical Review Letters **96**, 10.1103/physrevlett.96.110405 (2006).

[101] S. T. Flammia, A. Hamma, T. L. Hughes, and X.-G. Wen, "Topological entanglement rényi entropy and reduced density matrix structure", Phys. Rev. Lett. **103**, 261601 (2009).

[102] Z. Wang and E. J. Davis, "Calculating renyi entropies with neural autoregressive quantum states", arXiv, 2003.01358 (2020).

[103] K. Choo, G. Carleo, N. Regnault, and T. Neupert, "Symmetries and many-body excitations with neural-network quantum states", Phys. Rev. Lett. **121**, 167204 (2018).

[104] Y. Nomura, "Helping restricted boltzmann machines with quantum-state representation by restoring symmetry", Journal of Physics: Condensed Matter **33**, 174003 (2021).

[105] D. Wu, L. Wang, and P. Zhang, "Solving statistical mechanics using variational autoregressive networks", Phys. Rev. Lett. **122**, 080602 (2019).

[106] M. Hibat-Allah, E. M. Inack, R. Wiersema, R. G. Melko, and J. Carrasquilla, "Variational neural annealing", Nature Machine Intelligence **3**, 952–961 (2021).

[107] C. Roth, A. Szabó, and A. MacDonald, "High-accuracy variational monte carlo for frustrated magnets with deep neural networks", 10.48550/ARXIV.2211.07749 (2022).

[108] M. Hibat-Allah, R. G. Melko, and J. Carrasquilla, "Supplementing recurrent neural network wave functions with symmetry and annealing to improve accuracy", 10.48550/ARXIV.2207.14314 (2022).

[109] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors", **323**, 533–536 (1986).

[110] A. M. Schäfer and H. G. Zimmermann, "Recurrent neural networks are universal approximators", in Artificial neural networks – icann 2006, edited by S. D. Kollias, A. Stafylopatis, W. Duch, and E. Oja (2006), pp. 632–640.

[111] G. S. Carmantini, P. b. Graben, M. Desroches, and S. Rodrigues, "Turing computation with recurrent artificial neural networks", 10.48550/ARXIV.1511.01427 (2015).

[112] W. De Mulder, S. Bethard, and M.-F. Moens, "A survey on the application of recurrent neural networks to statistical language modeling", Computer Speech Language **30**, 61–98 (2015).

[113] J. Carrasquilla, "Machine learning for quantum matter", Advances in Physics: X **5**, https://doi.org/10.1080/23746149.2020.1797528 (2020).

[114] A. Dawid, J. Arnold, B. Requena, A. Gresch, M. Płodzień, K. Donatella, K. A. Nicoli, P. Stornati, R. Koch, M. Büttner, R. Okuła, G. Muñoz-Gil, R. A. Vargas-Hernández, A. Cervera-Lierta, J. Carrasquilla, V. Dunjko, M. Gabrié, P. Huembeli, E. van Nieuwenburg, F. Vicentini, L. Wang, S. J. Wetzel, G. Carleo, E. Greplová, R. Krems, F. Marquardt, M. Tomza, M. Lewenstein, and A. Dauphin, "Modern applications of machine learning in quantum sciences", 10.48550/ARXIV.2204.04198 (2022).

[115] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines", The Journal of Chemical Physics **21**, https://doi.org/10.1063/1.1699114 (1953).

[116] R. M. Neal, "Connectionist learning of belief networks", Artificial intelligence **56**, 71–113 (1992).

[117] S. Bengio and Y. Bengio, "Taking on the curse of dimensionality in joint distributions using neural networks", IEEE Transactions on Neural Networks **11**, 550–557 (2000).

[118] M. Germain, K. Gregor, I. Murray, and H. Larochelle, "Made: masked autoencoder for distribution estimation", 10.48550/ARXIV.1502.03509 (2015).

[119] B. Uria, M.-A. Côté, K. Gregor, I. Murray, and H. Larochelle, "Neural autoregressive distribution estimation", 10.48550/ARXIV.1605.02226 (2016).

[120] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio", arXiv, arXiv:1609.03499, 1609.03499 (2016).

[121] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel Recurrent Neural Networks", arXiv, arXiv:1601.06759, 1601.06759 (2016).

[122] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines", Cognitive science **9**, 147–169 (1985).

[123] R. G. Melko, G. Carleo, J. Carrasquilla, and J. I. Cirac, "Restricted Boltzmann machines in quantum physics", Nature Physics, 1 (2019).

[124] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, http://www.deeplearningbook.org (MIT Press, 2016).

[125] I. Goodfellow, "Nips 2016 tutorial: generative adversarial networks", arXiv, 1701.00160 (2016).

[126] S. Bravyi, D. P. Divincenzo, R. Oliveira, and B. M. Terhal, "The complexity of stoquastic local hamiltonian problems", Quantum Info. Comput. **8**, 361–385 (2008).

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

[128] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult", IEEE Transactions on Neural Networks **5**, 157–166 (1994).

[129] J. F. Kolen and S. C. Kremer, "Gradient flow in recurrent nets: the difficulty of learning longterm dependencies", in *A field guide to dynamical recurrent networks* (IEEE, 2001), pp. 237–243.

[130] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks", in International conference on machine learning (2013), pp. 1310–1318.

[131] M. Fannes, B. Nachtergaele, and R. F. Werner, "Finitely correlated states on quantum spin chains", Communications in Mathematical Physics **144**, 443–490 (1992).

[132] H. Shen, "Mutual information scaling and expressive power of sequence models", arXiv, 1905.04271 (2019).

[133] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: encoder–decoder approaches", in Proceedings of SSST-8, eighth workshop on syntax, semantics and structure in statistical translation (Oct. 2014), pp. 103–111.

[134] Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Y. Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, "TensorFlow: large-scale machine learning on heterogeneous systems", Software available from tensorflow.org (2015).

[135] J. Appleyard, T. Kocisky, and P. Blunsom, "Optimizing performance of recurrent neural networks on gpus", arXiv, 1604.01946 (2016).

[136] A. Graves, S. Fernandez, and J. Schmidhuber, "Multi-dimensional recurrent neural networks", arXiv, 0705.2011 (2007).

[137] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks", 1601.06759 (2016).

[138] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)", 10.48550/ARXIV.1511.07289 (2015).

[139] F. Verstraete and J. I. Cirac, "Renormalization algorithms for quantum-many body systems in two and higher dimensions", arXiv, cond–mat/0407066 (2004).

[140] L. Vanderstraeten, J. Haegeman, P. Corboz, and F. Verstraete, "Gradient methods for variational optimization of projected entangled-pair states", Physical Review B **94**, 10.1103/physrevb.94.155123 (2016).

[141] Y. Wu, S. Zhang, Y. Zhang, Y. Bengio, and R. R. Salakhutdinov, "On multiplicative integration with recurrent neural networks", in Advances in neural information processing systems (2016), p. 1606.06630.

[142] U. Schollwöck, "The density-matrix renormalization group in the age of matrix product states", Annals of Physics **326**, January 2011 Special Issue, 96–192 (2011).

[143] Y. Levine, O. Sharir, N. Cohen, and A. Shashua, "Quantum entanglement in deep learning architectures", Phys. Rev. Lett. **122**, 065301 (2019).

[144] R. Kelley, "Sequence modeling with recurrent tensor networks", Open Review (2016).

[145] G.-B. Zhou, J. Wu, C.-L. Zhang, and Z.-H. Zhou, "Minimal gated unit for recurrent neural networks", arXiv, 1603.09420 (2016).

[146] D. Wu, R. Rossi, F. Vicentini, and G. Carleo, "From tensor network quantum states to tensorial recurrent neural networks", 10.48550/ARXIV.2206.12363 (2022).

[147] S. Chang, Y. Zhang, W. Han, M. Yu, X. Guo, W. Tan, X. Cui, M. Witbrock, M. Hasegawa-Johnson, and T. S. Huang, "Dilated recurrent neural networks", arXiv, 1710.02224 (2017).

[148] S. E. Hihi and Y. Bengio, in *Advances in neural information processing systems 8*, edited by D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo (MIT Press, 1996), pp. 493–499.

[149] G. Vidal, "Class of quantum many-body states that can be efficiently simulated", Physical Review Letters **101**, 10.1103/physrevlett.101.110501 (2008).

[150] O. Sharir, Y. Levine, N. Wies, G. Carleo, and A. Shashua, "Deep autoregressive models for the efficient variational simulation of many-body quantum systems", Physical Review Letters **124**, 10.1103/physrevlett.124.020503 (2020).

[151] H.-J. Schmidt and M. Luban, "Classical ground states of symmetric heisenberg spin systems", Journal of Physics A: Mathematical and General **36**, 6351 (2003).

[152] M. Reh, M. Schmitt, and M. Gärttner, "Optimizing design choices for neural quantum states", 10.48550/ARXIV.2301.06788 (2023).

[153] W. Marshall, "Antiferromagnetism", Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences **232**, 48–68 (1955).

[154] E. Lieb and D. Mattis, "Ordering energy levels of interacting spin systems", Journal of Mathematical Physics **3**, 749–751 (1962).

[155] T. Vieijra, C. Casert, J. Nys, W. D. Neve, J. Haegeman, J. Ryckebusch, and F. Verstraete, "Restricted boltzmann machines for quantum states with non-abelian or anyonic symmetries", Physical Review Letters **124**, 10.1103/physrevlett.124.097201 (2020).

[156] D. Luo, G. Carleo, B. K. Clark, and J. Stokes, "Gauge equivariant neural networks for quantum lattice gauge theories", Phys. Rev. Lett. **127**, 276402 (2021).

[157] J. You, R. Ying, X. Ren, W. L. Hamilton, and J. Leskovec, "Graphrnn: generating realistic graphs with deep auto-regressive models", 10.48550/ARXIV.1802.08773 (2018).

[158] A. Kitaev, "Anyons in an exactly solved model and beyond", Annals of Physics, January Special Issue **321**, 2–111 (2006).

[159] A. Milsted, M. Ganahl, S. Leichenauer, J. Hidary, and G. Vidal, "Tensornetwork on tensorflow: a spin chain application using tree tensor networks", 10.48550/ARXIV.1905.01331 (2019).

[160] C. Casert, T. Vieijra, S. Whitelam, and I. Tamblyn, "Dynamical large deviations of two-dimensional kinetically constrained models using a neural-network state ansatz", Phys. Rev. Lett. **127**, 120602 (2021).

[161] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, *Transformers are rnns: fast autoregressive transformers with linear attention*, 2020.

[162] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, *Linformer: self-attention with linear complexity*, 2020.

[163] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, *An image is worth 16x16 words: transformers for image recognition at scale*, 2021.

[164] K. Sprague and S. Czischek, *Variational monte carlo with large patched transformers*, 2023.

[165] S. R. White, "Density matrix formulation for quantum renormalization groups", Phys. Rev. Lett. **69**, 2863–2866 (1992).

[166] C. Roberts, A. Milsted, M. Ganahl, A. Zalcman, B. Fontaine, Y. Zou, J. Hidary, G. Vidal, and S. Leichenauer, "Tensornetwork: a library for physics and machine learning", arXiv, 1905.01330 (2019).

[167] A. A. Abrikosov, I. Dzyaloshinskii, L. P. Gorkov, and R. A. Silverman, *Methods of quantum field theory in statistical physics* (Dover, New York, NY, 1975).

[168] S. R. White and I. Affleck, "Dimerization and incommensurate spiral spin correlations in the zigzag spin chain: analogies to the kondo lattice", Phys. Rev. B **54**, 9862–9869 (1996).

[169] S. Eggert, "Numerical evidence for multiplicative logarithmic corrections from marginal operators", Phys. Rev. B **54**, R9612–R9615 (1996).

[170] F. Becca, L. Capriotti, A. Parola, and S. Sorella, "Variational wave functions for frustrated magnetic models", arXiv, 0905.4854 (2009).

[171] K. Choo, T. Neupert, and G. Carleo, "Two-dimensional frustrated j1j2 model studied with neural network quantum states", Physical Review B **100**, 10.1103/physrevb.100.125124 (2019).

[172] G. Torlai, J. Carrasquilla, M. T. Fishman, R. G. Melko, and M. P. A. Fisher, "Wavefunction positivization via automatic differentiation", arXiv, 1906.04654 (2019).

[173] S. Yan, D. A. Huse, and S. R. White, "Spin-liquid ground state of the s = 1/2 kagome heisenberg antiferromagnet", Science **332**, 1173–1176 (2011).

[174] G. Evenbly and G. Vidal, "Tensor network renormalization", Phys. Rev. Lett. **115**, 180405 (2015).

[175] A. W. Sandvik, "Computational Studies of Quantum Spin Systems", AIP Conference Proceedings **1297**, 135–338 (2010).

[176] H. W. J. Blöte and Y. Deng, "Cluster monte carlo simulation of the transverse ising model", Phys. Rev. E **66**, 066110 (2002).

[177] A. van den Oord, N. Kalchbrenner, L. Espeholt, k. kavukcuoglu koray, O. Vinyals, and A. Graves, in *Advances in neural information processing systems 29*, edited by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Curran Associates, Inc., 2016), pp. 4790–4798.

[178] L. Capriotti, "Quantum effects and broken symmetries in frustrated antiferromagnets", International Journal of Modern Physics B **15**, 1799–1842 (2001).

[179] W.-Y. Liu, S.-J. Dong, Y.-J. Han, G.-C. Guo, and L. He, "Gradient optimization of finite projected entangled pair states", Physical Review B **95**, 10.1103/physrevb.95.195154 (2017).

[180] A. W. Sandvik and G. Vidal, "Variational quantum monte carlo simulations with tensor-network states", Physical Review Letters **99**, 10.1103/physrevlett.99.220602 (2007).

[181] M. Suzuki, "General review of quantum statistical monte carlo methods", in Quantum monte carlo methods in equilibrium and nonequilibrium systems, edited by M. Suzuki (1987), pp. 2–22.

[182] K. Harada, "Bayesian inference in the scaling analysis of critical phenomena", Physical Review E **84**, 10.1103/physreve.84.056704 (2011).

[183] R. Yoneda and K. Harada, "Neural network approach to scaling analysis of critical phenomena", arXiv, 2209.01777 (2023).

[184] https://github.com/KenjiHarada/FSS-tools, "Github repository",

[185] S. Ahmed, N. Killoran, and J. F. C. Álvarez, "Implicit differentiation of variational quantum algorithms", arXiv, 2211.13765 (2022).

[186] E. Sanchez-Velasco, "A finite-size scaling study of the 4d ising model", Journal of Physics A: Mathematical and General **20**, 5033 (1987).

[187] T. F. A. Alves, G. A. Alves, and M. S. Vasconcelos, "Logarithm corrections in the critical behavior of the ising model on a triangular lattice modulated with the fibonacci sequence", arXiv, 1805.05725 (2018).

[188] R. Kenna, D. A. Johnston, and W. Janke, "Scaling relations for logarithmic corrections", Physical Review Letters **96**, 10.1103/physrevlett.96.115701 (2006).

[189] R. Kenna, in *Order, disorder and criticality* (WORLD SCIENTIFIC, Dec. 2012), pp. 1–46.

[190] M. Hermans and B. Schrauwen, in *Advances in neural information processing systems 26*, edited by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Curran Associates, Inc., 2013), pp. 190–198.

[191] http://github.com/mhibatallah/RNNWavefunctions, "Github repository",

[192] S. A. Khandoker, J. M. Abedin, and M. Hibat-Allah, "Supplementing recurrent neural networks with annealing to solve combinatorial optimization problems", Machine Learning: Science and Technology **4**, 015026 (2023).

[193] F. Barahona, "On the computational complexity of ising spin glass models", Journal of Physics A: Mathematical and General **15**, 3241–3253 (1982).

[194] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing", Science **220**, 671–680 (1983).

[195] C. Koulamas, S. Antony, and R. Jaen, "A survey of simulated annealing applications to operations research problems", Omega **22**, 41–56 (1994).

[196] B. Hajek, "A tutorial survey of theory and applications of simulated annealing", in 1985 24th ieee conference on decision and control (Dec. 1985), pp. 755–760.

[197] D. Svergun, "Restoring low resolution structure of biological macromolecules from solution scattering using simulated annealing", Biophysical Journal **76**, 2879–2886 (1999).

[198] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon, "Optimization by simulated annealing: an experimental evaluation; part ii, graph coloring and number partitioning", Operations Research **39**, 378–406 (1991).

[199] M. A. Abido, "Robust design of multimachine power system stabilizers using simulated annealing", IEEE Transactions on Energy Conversion **15**, 297–304 (2000).

[200] T. Karzig, A. Rahmani, F. von Oppen, and G. Refael, "Optimal control of majorana zero modes", Phys. Rev. B **91**, 201404 (2015).

[201] G. Gielen, H. Walscharts, and W. Sansen, "Analog circuit design optimization based on symbolic simulation and simulated annealing", in Esscirc '89: proceedings of the 15th european solid-state circuits conference (Sept. 1989), pp. 252–255.

[202] G. E. Santoro, R. Martoňák, E. Tosatti, and R. Car, "Theory of quantum annealing of an ising spin glass", Science **295**, 2427–2430 (2002).

[203] J. Brooke, D. Bitko, T. F. Rosenbaum, and G. Aeppli, "Quantum annealing of a disordered magnet", Science **284**, 779–781 (1999).

[204] D. Mitra, F. Romeo, and A. Sangiovanni-Vincentelli, "Convergence and finite-time behavior of simulated annealing", Advances in Applied Probability **18**, 747–771 (1986).

[205] D. Delahaye, S. Chaimatanan, and M. Mongeau, "Simulated annealing: from basics to applications", in *Handbook of metaheuristics*, edited by M. Gendreau and J.-Y. Potvin (Springer International Publishing, Cham, 2019), pp. 1–35.

[206] P. M. Long and R. A. Servedio, "Restricted boltzmann machines are hard to approximately evaluate or simulate", in Proceedings of the 27th international conference on international conference on machine learning, ICML'10 (2010), pp. 703–710.

[207] H. Larochelle and I. Murray, "The neural autoregressive distribution estimator", in Proceedings of the fourteenth international conference on artificial intelligence and statistics, Vol. 15, edited by G. Gordon, D. Dunson, and M. Dudík, Proceedings of Machine Learning Research (Nov. 2011), pp. 29–37.

[208] A. Lucas, "Ising formulations of many np problems", Front. Phys. **2**, 5 (2014).

[209] R. Feynman, *Statistical mechanics: A set of lectures*, Advanced Books Classics (Avalon Publishing, 1998).

[210] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse ising model", Physical Review E **58**, 5355–5363 (1998).

[211] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, "Quantum computation by adiabatic evolution", arXiv, quant–ph/0001106 (2000).

[212] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, "A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem", Science **292**, 472–475 (2001).

[213] T. Albash and D. A. Lidar, "Adiabatic quantum computation", Rev. Mod. Phys. **90**, 015002 (2018).

[214] M. Born and V. Fock, "Beweis des adiabatensatzes", Zeitschrift für Physik **51**, 165–180 (1928).

[215] V. Bapst, L. Foini, F. Krzakala, G. Semerjian, and F. Zamponi, "The quantum adiabatic algorithm applied to random optimization problems: the quantum spin glass perspective", Physics Reports **523**, 127–205 (2013).

[216] G. B. Mbeng, L. Privitera, L. Arceci, and G. E. Santoro, "Dynamics of simulated quantum annealing in random ising chains", Phys. Rev. B **99**, 064201 (2019).

[217] S. Morita and H. Nishimori, "Mathematical foundation of quantum annealing", Journal of Mathematical Physics **49**, 125210 (2008).

[218] R. Marto ňák, G. E. Santoro, and E. Tosatti, "Quantum annealing by the path-integral monte carlo method: the two-dimensional random ising model", Phys. Rev. B **66**, 094203 (2002).

[219] G. S. Grest, C. M. Soukoulis, and K. Levin, "Cooling-rate dependence for the spin-glass ground-state energy: implications for optimization by simulated annealing", Phys. Rev. Lett. **56**, 1148–1151 (1986).

[220] P. Ocampo-Alfaro and H. Guo, "Cooling-rate dependence of the ground-state energy using microcanonical simulated annealing", Phys. Rev. E **53**, 1982–1985 (1996).

[221] N. Norris, "The standard errors of the geometric and harmonic means and their application to index numbers", The Annals of Mathematical Statistics **11**, 445–448 (1940).

[222] S. Suzuki, "Cooling dynamics of pure and random ising chains", Journal of Statistical Mechanics: Theory and Experiment **2009**, P03032 (2009).

[223] J. Dziarmaga, "Dynamics of a quantum phase transition in the random ising model: logarithmic dependence of the defect density on the transition rate", Phys. Rev. B **74**, 064416 (2006).

[224] T. Caneva, R. Fazio, and G. E. Santoro, "Adiabatic quantum dynamics of a random ising chain across its quantum critical point", Phys. Rev. B **76**, 144427 (2007).

[225] T. Zanca and G. E. Santoro, "Quantum annealing speedup over simulated annealing on random ising chains", Phys. Rev. B **93**, 224431 (2016).

[226] D. A. Huse and D. S. Fisher, "Residual energies after slow cooling of disordered systems", Phys. Rev. Lett. **57**, 2203–2206 (1986).

[227] H. Nishimori and K. Takada, "Exponential enhancement of the efficiency of quantum annealing by non-stoquastic hamiltonians", Frontiers in ICT **4**, 2 (2017).

[228] E. Crosson, T. Albash, I. Hen, and A. P. Young, "De-signing hamiltonians for quantum adiabatic optimization", Quantum **4**, 334 (2020).

[229] I. Ozfidan, C. Deng, A. Smirnov, T. Lanting, R. Harris, L. Swenson, J. Whittaker, F. Altomare, M. Babcock, C. Baron, A. Berkley, K. Boothby, H. Christiani, P. Bunyk, C. Enderud, B. Evert, M. Hager, A. Hajda, J. Hilton, S. Huang, E. Hoskinson, M. Johnson, K. Jooya, E. Ladizinsky, N. Ladizinsky, R. Li, A. MacDonald, D. Marsden, G. Marsden, T. Medina, R. Molavi, R. Neufeld, M. Nissen, M. Norouzpour, T. Oh, I. Pavlov, I. Perminov, G. Poulin-Lamarre, M. Reis, T. Prescott, C. Rich, Y. Sato, G. Sterling, N. Tsai, M. Volkmann, W. Wilkinson, J. Yao, and M. Amin, "Demonstration of a nonstoquastic hamiltonian in coupled superconducting flux qubits", Phys. Rev. Applied **13**, 034037 (2020).

[230] E. M. Lykiardopoulou, A. Zucca, S. A. Scivier, and M. H. Amin, "Improving nonstoquastic quantum annealing with spin-reversal transformations", arXiv, 2010.00065 (2020).

[231] B. Heim, T. F. Ronnow, S. V. Isakov, and M. Troyer, "Quantum versus classical annealing of ising spin glasses", Science **348**, 215–217 (2015).

[232] "Http://spinglass.uni-bonn.de/",

[233] N. G. Dickson, M. Johnson, M. Amin, R. Harris, F. Altomare, A. Berkley, P. Bunyk, J. Cai, E. Chapple, P. Chavez, et al., "Thermally assisted quantum annealing of a 16-qubit problem", Nature communications **4**, 1–6 (2013).

[234] J. Gomes, K. A. McKiernan, P. Eastman, and V. S. Pande, "Classical quantum optimization with neural network quantum states", arXiv, 1910.10675 (2019).

[235] S. Sinchenko and D. Bazhanov, "The deep learning and statistical physics applications to the problems of combinatorial optimization", arXiv, 1911.10680 (2019).

[236] T. Zhao, G. Carleo, J. Stokes, and S. Veerapaneni, "Natural evolution strategies and quantum approximate optimization", arXiv, 2005.04447 (2020).

[237] M. Mezard, G. Parisi, and M. Virasoro, *Spin glass theory and beyond* (WORLD SCIENTIFIC, 1986), https://www.worldscientific.com/doi/pdf/10.1142/0271.

[238] D. Sherrington and S. Kirkpatrick, "Solvable model of a spin-glass", Phys. Rev. Lett. **35**, 1792–1796 (1975).

[239] F. Hamze, J. Raymond, C. A. Pattison, K. Biswas, and H. G. Katzgraber, "Wishart planted ensemble: a tunably rugged pairwise ising model with a first-order phase transition", Physical Review E **101**, 10.1103/physreve.101.052102 (2020).

[240] R. M. Karp, in *Complexity of computer computations* (Springer, 1972), pp. 85–103.

[241] *Rudy*, http://www-user.tu-chemnitz.de/~helmberg/rudy.tar.gz.

[242] *Biq mac solver*, https://biqmac.aau.at/.

[243] T. Osogami and H. Imai, "Classification of various neighborhood operations for the nurse scheduling problem", in International symposium on algorithms and computation (Springer, 2000), pp. 72–83.

[244] K. Ikeda, Y. Nakamura, and T. S. Humble, "Application of quantum annealing to nurse scheduling problem", Scientific reports **9**, 1–10 (2019).

[245] F. Glover, G. Kochenberger, and Y. Du, "Quantum bridge analytics i: a tutorial on formulating and using qubo models", 4OR **17**, 335–371 (2019).

[246] C. H. Papadimitriou, "The euclidean travelling salesman problem is np-complete", Theoretical computer science **4**, 237–244 (1977).

[247] D. Applegate, R. Bixby, V. Chvátal, and W. Cook, "Tsp cuts which do not conform to the template paradigm", in *Computational combinatorial optimization: optimal or provably near-optimal solutions*, edited by M. Jünger and D. Naddef (Springer Berlin Heidelberg, Berlin, Heidelberg, 2001), pp. 261–303.

[248] *Neoserver*, https://neos-server.org/neos/solvers/co:concorde/TSP.html.

[249] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, "Neural combinatorial optimization with reinforcement learning", 10.48550/ARXIV.1611.09940 (2016).

[250] M. Bukov, M. Schmitt, and M. Dupont, "Learning the ground state of a non-stoquastic quantum hamiltonian in a rugged neural network landscape", SciPost Physics **10**, 10.21468/scipostphys.10.6.147 (2021).

[251] K. Mills, P. Ronagh, and I. Tamblyn, "Finding the ground state of spin hamiltonians with reinforcement learning", Nature Machine Intelligence **2**, 509–517 (2020).

[252] Y. Bengio, A. Lodi, and A. Prouvost, "Machine learning for combinatorial optimization: a methodological tour d'horizon", European Journal of Operational Research, https://doi.org/10.1016/j.ejor.2020.07.063 (2020).

[253] https://github.com/VectorInstitute/VariationalNeuralAnnealing, "Github repository",

[254] https://github.com/RNN-VCA-CO/RNN-VCA-CO, "Github repository",

[255] M. Hibat-Allah, R. G. Melko, and J. Carrasquilla, "Investigating topological order using recurrent neural networks", arXiv, 2303.11207 (2023).

[256] X.-G. Wen, "Topological order: from long-range entangled quantum matter to a unified origin of light and electrons", ISRN Condensed Matter Physics **2013**, 1–20 (2013).

[257] G. Semeghini, H. Levine, A. Keesling, S. Ebadi, T. T. Wang, D. Bluvstein, R. Verresen, H. Pichler, M. Kalinowski, R. Samajdar, A. Omran, S. Sachdev, A. Vishwanath, M. Greiner, V. Vuletić, and M. D. Lukin, "Probing topological spin liquids on a programmable quantum simulator", Science **374**, 1242–1247 (2021).

[258] K. J. Satzinger, Y.-J. Liu, A. Smith, C. Knapp, M. Newman, C. Jones, Z. Chen, C. Quintana, X. Mi, A. Dunsworth, C. Gidney, I. Aleiner, F. Arute, K. Arya, J. Atalaya, R. Babbush, J. C. Bardin, R. Barends, J. Basso, A. Bengtsson, A. Bilmes, M. Broughton, B. B. Buckley, D. A. Buell, B. Burkett, N. Bushnell, B. Chiaro, R. Collins, W. Courtney, S. Demura, A. R. Derk, D. Eppens, C. Erickson, L. Faoro, E. Farhi, A. G. Fowler, B. Foxen, M. Giustina, A. Greene, J. A. Gross, M. P. Harrigan, S. D. Harrington, J. Hilton, S. Hong, T. Huang, W. J. Huggins, L. B. Ioffe, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, T. Khattar, S. Kim, P. V. Klimov, A. N. Korotkov, F. Kostritsa, D. Landhuis, P. Laptev, A. Locharla, E. Lucero, O. Martin, J. R. McClean, M. McEwen, K. C. Miao, M. Mohseni, S. Montazeri, W. Mruczkiewicz, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Y. Niu, T. E. O'Brien, A. Opremcak, B. Pató, A. Petukhov, N. C. Rubin, D. Sank, V. Shvarts, D. Strain, M. Szalay, B. Villalonga, T. C. White, Z. Yao, P. Yeh, J. Yoo, A. Zalcman, H. Neven, S. Boixo, A. Megrant, Y. Chen, J. Kelly, V. Smelyanskiy, A. Kitaev, M. Knap, F. Pollmann, and P. Roushan, "Realizing topologically ordered states on a quantum processor", Science **374**, https://www.science.org/doi/pdf/10.1126/science.abi8378 (2021).

[259] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, "Topological quantum memory", Journal of Mathematical Physics **43**, 4452–4505 (2002).

[260] A. Kitaev and C. Laumann, "Topological phases and quantum computation", arXiv, 0904.2771 (2009).

[261] R. Acharya, I. Aleiner, R. Allen, T. I. Andersen, M. Ansmann, F. Arute, K. Arya, A. Asfaw, J. Atalaya, R. Babbush, D. Bacon, J. C. Bardin, J. Basso, A. Bengtsson, S. Boixo, G. Bortoli, A. Bourassa, J. Bovaird, L. Brill, M. Broughton, B. B. Buckley, D. A. Buell, T. Burger, B. Burkett, N. Bushnell, Y. Chen, Z. Chen, B. Chiaro, J. Cogan, R. Collins, P. Conner, W. Courtney, A. L. Crook, B. Curtin, D. M. Debroy, A. Del Toro Barba, S. Demura, A. Dunsworth, D. Eppens, C. Erickson, L. Faoro, E. Farhi, R. Fatemi, L. Flores Burgos, E. Forati, A. G. Fowler, B. Foxen, W. Giang, C. Gidney, D. Gilboa, M. Giustina, A. Grajales Dau, J. A. Gross, S. Habegger, M. C. Hamilton, M. P. Harrigan, S. D. Harrington, O. Higgott, J. Hilton, M. Hoffmann, S. Hong, T. Huang, A. Huff, W. J. Huggins, L. B. Ioffe, S. V. Isakov, J. Iveland, E. Jeffrey, Z. Jiang, C. Jones, P. Juhas, D. Kafri, K. Kechedzhi, J. Kelly, T. Khattar, M.

Khezri, M. Kieferová, S. Kim, A. Kitaev, P. V. Klimov, A. R. Klots, A. N. Korotkov, F. Kostritsa, J. M. Kreikebaum, D. Landhuis, P. Laptev, K.-M. Lau, L. Laws, J. Lee, K. Lee, B. J. Lester, A. Lill, W. Liu, A. Locharla, E. Lucero, F. D. Malone, J. Marshall, O. Martin, J. R. McClean, T. McCourt, M. McEwen, A. Megrant, B. Meurer Costa, X. Mi, K. C. Miao, M. Mohseni, S. Montazeri, A. Morvan, E. Mount, W. Mruczkiewicz, O. Naaman, M. Neeley, C. Neill, A. Nersisyan, H. Neven, M. Newman, J. H. Ng, A. Nguyen, M. Nguyen, M. Y. Niu, T. E. O'Brien, A. Opremcak, J. Platt, A. Petukhov, R. Potter, L. P. Pryadko, C. Quintana, P. Roushan, N. C. Rubin, N. Saei, D. Sank, K. Sankaragomathi, K. J. Satzinger, H. F. Schurkus, C. Schuster, M. J. Shearn, A. Shorter, V. Shvarts, J. Skruzny, V. Smelyanskiy, W. C. Smith, G. Sterling, D. Strain, M. Szalay, A. Torres, G. Vidal, B. Villalonga, C. Vollgraff Heidweiller, T. White, C. Xing, Z. J. Yao, P. Yeh, J. Yoo, G. Young, A. Zalcman, Y. Zhang, N. Zhu, and G. Q. AI, "Suppressing quantum errors by scaling a surface code logical qubit", Nature **614**, 676–681 (2023).

[262] A. Hamma, R. Ionicioiu, and P. Zanardi, "Bipartite entanglement and entropic boundary law in lattice spin systems", Physical Review A **71**, 10.1103/physreva.71.022315 (2005).

[263] S. V. Isakov, M. B. Hastings, and R. G. Melko, "Topological entanglement entropy of a bose–hubbard spin liquid", Nature Physics **7**, 772–775 (2011).

[264] J. Wildeboer, A. Seidel, and R. G. Melko, "Entanglement entropy and topological order in resonating valence-bond quantum spin liquids", Physical Review B **95**, 10.1103/physrevb.95.100402 (2017).

[265] J. Zhao, B.-B. Chen, Y.-C. Wang, Z. Yan, M. Cheng, and Z. Y. Meng, "Measuring rényi entanglement entropy with high efficiency and precision in quantum monte carlo simulations", npj Quantum Materials **7**, 10.1038/s41535-022-00476-0 (2022).

[266] H.-C. Jiang, Z. Wang, and L. Balents, "Identifying topological order by entanglement entropy", Nature Physics **8**, 902–905 (2012).

[267] H.-C. Jiang and L. Balents, "Collapsing schrödinger cats in the density matrix renormalization group", 10.48550/ARXIV.1309.7438 (2013).

[268] S. V. Isakov, Y. B. Kim, and A. Paramekanti, "Spin-liquid phase in a spin-1/2 quantum magnet on the kagome lattice", Phys. Rev. Lett. **97**, 207204 (2006).

[269] A. Hamma, R. Ionicioiu, and P. Zanardi, "Ground state entanglement and geometric entropy in the kitaev model", Physics Letters A **337**, 22–28 (2005).

[270] Y. Zhang, T. Grover, A. Turner, M. Oshikawa, and A. Vishwanath, "Quasiparticle statistics and braiding from ground-state entanglement", Physical Review B **85**, 10.1103/physrevb.85.235151 (2012).

[271] I. H. Kim, M. Levin, T.-C. Lin, D. Ranard, and B. Shi, "Universal lower bound on topological entanglement entropy", arXiv, 2302.00689 (2023).

[272] S. Furukawa and G. Misguich, "Topological entanglement entropy in the quantum dimer model on the triangular lattice", Physical Review B **75**, 10.1103/physrevb.75.214407 (2007).

[273] E. Stoudenmire and S. R. White, "Studying two-dimensional systems with the density matrix renormalization group", Annual Review of Condensed Matter Physics **3**, 111–128 (2012).

[274] S.-S. Gong, W. Zhu, D. N. Sheng, O. I. Motrunich, and M. P. A. Fisher, "Plaquette ordered phase and quantum phase diagram in the spin-$\frac{1}{2}$ J_1-J_2 square heisenberg model", Phys. Rev. Lett. **113**, 027201 (2014).

[275] R. Samajdar, W. W. Ho, H. Pichler, M. D. Lukin, and S. Sachdev, "Quantum phases of rydberg atoms on a kagome lattice", Proceedings of the National Academy of Sciences **118**, e2015785118 (2021).

[276] Y.-C. Wang, C. Fang, M. Cheng, Y. Qi, and Z. Y. Meng, "Topological Spin Liquid with Symmetry-Protected Edge States", arXiv, 1701.01552 (2017).

[277] B. Kulchytskyy, C. M. Herdman, S. Inglis, and R. G. Melko, "Detecting goldstone modes with entanglement entropy", Physical Review B **92**, 10.1103/physrevb.92.115146 (2015).

[278] A. Browaeys and T. Lahaye, "Many-body physics with individually controlled Rydberg atoms", Nature Physics **16**, 132–142 (2020).

[279] R. Verresen, M. D. Lukin, and A. Vishwanath, "Prediction of toric code topological order from rydberg blockade", Physical Review X **11**, 10.1103/physrevx.11.031005 (2021).

[280] M. Kalinowski, R. Samajdar, R. G. Melko, M. D. Lukin, S. Sachdev, and S. Choi, "Bulk and boundary quantum phase transitions in a square rydberg atom array", Phys. Rev. B **105**, 174417 (2022).

[281] Z. Yan, Y.-C. Wang, R. Samajdar, S. Sachdev, and Z. Y. Meng, "Emergent glassy behavior in a kagome rydberg atom array", 10.48550/ARXIV.2301.07127 (2023).

[282] S. F. Edwards and P. W. Anderson, "Theory of spin glasses", Journal of Physics F: Metal Physics **5**, 965 (1975).

[283] P. M. Richards, "Spin-glass order parameter of the random-field ising model", Phys. Rev. B **30**, 2955–2957 (1984).

[284] E. R. Bennewitz, F. Hopfmueller, B. Kulchytskyy, J. F. Carrasquilla, and P. Ronagh, "Neural error mitigation of near-term quantum simulations", ArXiv, 2105.08086 (2021).

[285] S. Czischek, M. S. Moss, M. Radzihovsky, E. Merali, and R. G. Melko, "Data-enhanced variational monte carlo simulations for rydberg atom arrays", Physical Review B **105**, 10.1103/physrevb.105.205108 (2022).

[286] G. Giudici, M. D. Lukin, and H. Pichler, "Dynamical preparation of quantum spin liquids in rydberg atom arrays", arXiv, 2202.09372 (2022).

[287] G. Montufar, "Restricted boltzmann machines: introduction and review", 10.48550/ARXIV.1806.07066 (2018).

[288] Z.-Y. Han, J. Wang, H. Fan, L. Wang, and P. Zhang, "Unsupervised generative modeling using matrix product states", PRX **8**, 031012 (2018).

[289] M. Benedetti, D. Garcia-Pintos, Y. Nam, and A. Perdomo-Ortiz, "A generative modeling approach for benchmarking and training shallow quantum circuits", npj Quantum Information **5**, 10.1038/s41534-019-0157-8 (2018).

[290] X. Gao, E. R. Anschuetz, S.-T. Wang, J. I. Cirac, and M. D. Lukin, "Enhancing generative models via quantum correlations", Phys. Rev. X **12**, 021037 (2022).

[291] K. Gili, M. Hibat-Allah, M. Mauri, C. Ballance, and A. Perdomo-Ortiz, "Do quantum circuit born machines generalize?", arXiv, 2207.13645 (2022).

[292] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models", 10.48550/ARXIV.2112.10752 (2021).

[293] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback", 10.48550/ARXIV.2203.02155 (2022).

[294] N. Brown, M. Fiscato, M. H. Segler, and A. C. Vaucher, "Guacamol: benchmarking models for de novo molecular design", Journal of Chemical Information and Modeling **59**, 1096–1108 (2019).

[295] J. Li, R. Topaloglu, and S. Ghosh, "Quantum generative models for small molecule drug discovery", arXiv:2101.03438, 10.1109/TQE.2021.3104804 (2021).

[296] K. Gili, M. Mauri, and A. Perdomo-Ortiz, "Evaluating generalization in classical and quantum generative models", arXiv:2201.08770 (2022).

[297] Y. Du, M.-H. Hsieh, T. Liu, and D. Tao, "Expressive power of parametrized quantum circuits", Physical Review Research **2**, 033125 (2018).

[298] R. Orús, "A practical introduction to tensor networks: matrix product states and projected entangled pair states", Annals of physics **349**, 117–158 (2014).

[299] R. Orús, "Tensor networks for complex quantum systems", Nature Reviews Physics **1**, 538–550 (2019).

[300] E. Stoudenmire and D. J. Schwab, "Supervised learning with tensor networks", Advances in Neural Information Processing Systems **29** (2016).

[301] D. Perez-Garcia, F. Verstraete, M. Wolf, and J. Cirac, "Matrix product state representations", Quantum Information & Computation **7**, 401–430 (2007).

[302] F. Verstraete and J. I. Cirac, "Renormalization algorithms for Quantum-Many Body Systems in two and higher dimensions", arXiv, cond-mat/0407066, cond–mat/0407066 (2004).

[303] D.-L. Deng, X. Li, and S. D. Sarma, "Machine learning topological states", Physical Review B **96**, 10.1103/physrevb.96.195145 (2017).

[304] I. Wegener, *Complexity theory: exploring the limits of efficient algorithms* (Springer Science & Business Media, 2005).

[305] R. H. Dicke, "Coherence in spontaneous radiation processes", Phys. Rev. **93**, 99–110 (1954).

[306] A. Kitaev, "Fault-tolerant quantum computation by anyons", Annals of Physics **303**, 2–30 (2003).

[307] J. Haferkamp, D. Hangleiter, J. Eisert, and M. Gluza, "Contracting projected entangled pair states is average-case hard", Physical Review Research **2**, 10.1103/physrevresearch.2.013010 (2020).

[308] S. Aktar, A. Bartschi, A.-H. A. Badawy, and S. Eidenbenz, "A divide-and-conquer approach to dicke state preparation", IEEE Transactions on Quantum Engineering **3**, 1–16 (2022).

[309] K. Gili, M. Sveistrys, and C. Ballance, "Introducing non-linear activations into quantum generative models", 10.48550/ARXIV.2205.14506 (2022).

[310] G. Rabusseau, T. Li, and D. Precup, "Connecting weighted automata and recurrent neural networks through spectral learning", 10.48550/ARXIV.1807.01406 (2018).

[311] T. Li, D. Precup, and G. Rabusseau, "Connecting weighted automata, tensor networks and recurrent neural networks through spectral learning", 10.48550/ARXIV.2010.10029 (2020).

[312] M. S. Rudolph, J. Chen, J. Miller, A. Acharya, and A. Perdomo-Ortiz, "Decomposition of matrix product states into shallow quantum circuits", arXiv prpages arXiv:2209.00595 (2022).

[313] J. Chen, S. Cheng, H. Xie, L. Wang, and T. Xiang, "Equivalence of restricted boltzmann machines and tensor network states", Physical Review B **97**, 10.1103/physrevb.97.085104 (2018).

[314] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: a next-generation hyperparameter optimization framework", arXiv, 1907.10902 (2019).

[315] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox, "Hyperopt: a python library for model selection and hyperparameter optimization", Computational Science Discovery **8**, 014008 (2015).

[316] J. Martens, J. Ba, and M. Johnson, "Kronecker-factored curvature approximations for recurrent neural networks", in International conference on learning representations (2018).

[317] M. Fishman, S. R. White, and E. M. Stoudenmire, "The ITensor software library for tensor network calculations", arXiv, 2007.14822 (2020).

[318] Y. Nesterov, "Smooth convex optimization", in *Lectures on convex optimization* (Springer International Publishing, Cham, 2018), pp. 59–137.

[319] M. Schmidt, N. L. Roux, and F. Bach, "Minimizing finite sums with the stochastic average gradient", arXiv, 1309.2388 (2013).

[320] S. Boixo, T. F. Rønnow, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, and M. Troyer, "Evidence for quantum annealing with more than one hundred qubits", Nat. Phys. **10**, 218–224 (2014).

[321] R. Martoň ák, G. E. Santoro, and E. Tosatti, "Quantum annealing of the traveling-salesman problem", Physical Review E **70**, 10.1103/physreve.70.057701 (2004).

[322] E. Fradkin, *Field theories of condensed matter physics*, 2nd ed. (Cambridge University Press, 2013).

[323] F. Verstraete, M. M. Wolf, D. Perez-Garcia, and J. I. Cirac, "Criticality, the area law, and the computational power of projected entangled pair states", Physical Review Letters **96**, 10.1103/physrevlett.96.220601 (2006).

[324] G. F. Montúfar, J. Rauh, and N. Ay, "Expressive power and approximation errors of restricted boltzmann machines", Advances in neural information processing systems **24** (2011).

[325] G. Montufar, "Mixture models and representational power of rbms, dbns and dbms", in Nips deep learning and unsupervised feature learning workshop (2010).

[326] A. Bärtschi and S. Eidenbenz, in *Fundamentals of computation theory* (Springer International Publishing, 2019), pp. 126–139.

[327] C. S. Mukherjee, S. Maitra, V. Gaurav, and D. Roy, "On actual preparation of dicke state on a quantum computer", 10.48550/ARXIV.2007.01681 (2020).

APPENDICES

Appendix A

Supplementary material of chapter 5

A.1 Hyperparameters

In this appendix, we present the hyperparameters used to train the RNN wave functions in this study. We anticipate that further improvements such as the use of Stochastic Reconfiguration [2] or a computationally cheaper variant such as K-FAC [316] for the optimization could potentially lead to more accurate estimations of the ground state energies as compared to the Adam optimizer [80]. Seeds are listed in the table for reproducibility purposes.

For the transverse-field Ising model results, Tab. A.1 provides the hyperparameters used to produce the results of Secs. 5.1.1 and 5.2.1. Additionally, tab. A.2 summarizes the hyperparameters of Sec. 5.1.2 for the 1D J_1-J_2 model results.

For the Heisenberg model results, provided in Sec. 5.2.2, we summarize the hyperparameters in Tab. A.4. We note that the training of our ansatz was performed using P100 GPUs.

In order to produce the results of Fig. 6.10(b), we use $d_h = 300$ and $M = 100$ samples for training. We have first performed annealing on the system size 6×6 with $N_{\text{warmup}} = 1000$ and $N_{\text{annealing}} = 10000$ and an initial pseudo-temperature $T_0 = 0.25$ while using a fixed learning rate $\eta = 5 \times 10^{-4}$. The number of training steps during each annealing step is taken as $N_{\text{train}} = 5$. In this initial phase, we only apply the $U(1)$ symmetry. In the next phase, we perform an additional 25000 gradient steps at zero pseudo-temperature and add an additional 25000 gradient steps after applying C_{2d} symmetry. During this convergence phase, the learning rate is decayed as $\eta = 5 \times 10^{-5}/(1 + t/2000)$, where t corresponds

to the current number of convergence steps. We then increase the system size to 8×8, while keeping our ansatz parameters fixed, applying C_{2d} symmetry and while using zero pseudo-temperature. In this phase, the learning rate is fixed to $\eta = 10^{-5}$. After this step, we train our RNN ansatz until convergence. We repeat the same procedure for system sizes 10×10, 12×12, 14×14, and 16×16. We note that for 8×8, we add 40000 gradient steps. For 10×10, we continue training with 20000 gradient steps. For 12×12, we converge using 10000 training steps. For 14×14, we continue training with 5000 gradient steps. Finally, for 16×16, we add 2000 convergence steps.

To produce the DMRG results in Fig. 6.10(b), we used a bond dimension $D = 4000$ for sizes 6×6, 8×8 and 10×10. For 12×12, we used $D = 3000$ and for the sizes 14×14 and 16×16, we used $D = 2000$ since we were not able to obtain the DMRG energy at $D = 4000$ with a limit of 100 GB memory allocation. The DMRG calculations were run using ITensor [317].

We finally note that the Marshall sign [153, 178] is applied on top of our cRNN wave function on the square lattice and on the triangular lattice during all our numerical experiments to speed up convergence. For DMRG, we observed that applying the Marshall sign does not affect the accuracy.

A.2 Table of results

In Tab. A.5, we state the variational energies of the cRNN wave function for the 1D J_1-J_2 model and compare them with results from DMRG. We examine two different methods of training. First, we do not impose an initial sign structure while, secondly, we introduce a background Marshall sign. The results suggest that using a Marshall sign improves the results significantly for $J_2 = 0.0, 0.2$ and 0.5 (with $J_1 = 1$ for all cases). We note that our cRNN wave function recovers the sign structure of the ground state if we train it without an initial Marshall sign.

In Tab. A.6, we compare the variational energies per site of the 2D TFIM with a lattice size of 12×12 for different values of the transverse magnetic field h, for a 1D pRNN wave function, a 2D pRNN wave function, a PixelCNN wave function [150] and DMRG.

In Tab. A.7, we provide a comparison between the different methods for the 10×10 square Heisenberg model in Sec. 5.2.2. For Tab. A.8, we provide the energy values of 2DRNN and DMRG on the triangular Heisenberg model in Sec. 6.7. Finally in Tab. A.9, we compare our values with the results of Ref. [146].

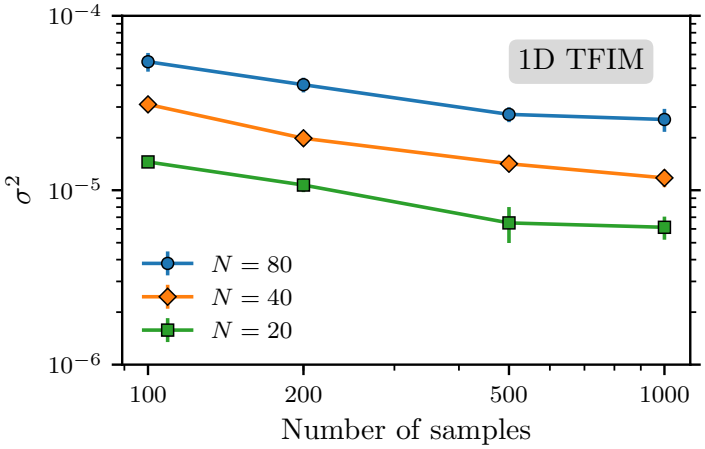

Figure A.1: The energy variance per spin against the number of samples, which suggests that the energy variance saturates and does not improve further by using a larger number of samples for training.

A.3 RNN numerical benchmarks

A.3.1 Benchmarking RNN hyperparameters (continued)

Fig. A.1 shows the dependence of σ^2 on the number of samples used to estimate the gradients of the variational energy (see Chap. 3). We investigate this effect for the case of the 1D TFIM, using 50 memory units in the pRNN wave function. Even though a large number of samples yields higher statistical accuracy of the gradient estimates used in our optimizations, we observe only a weak dependence of σ^2 on the number of samples for all studied system sizes.

In Fig. A.2 we present results for the dependence of σ^2 on the depth of the pRNN wave function architecture for a critical TFIM with $N = 40$ sites. We investigate architectures up to a depth of four layers. The number of memory units is adapted such that we have a similar number of variational parameters ($\sim$31000) for each of the four architectures. We find that σ^2 depends only weakly on the number of layers.

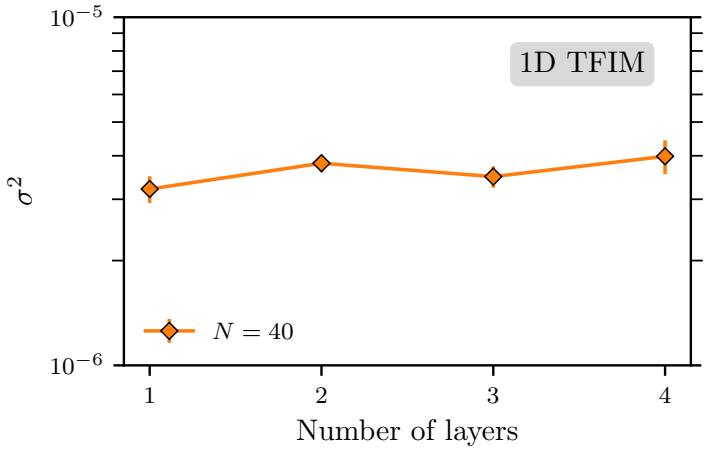

Figure A.2: Scaling study of the energy variance per spin versus the number of layers of a pRNN wave function such that all pRNN wave functions with different layers have the same number of variational parameters. The results show that fixing the number of parameters while changing the number of layers does not affect the energy variance obtained by the pRNN wave function.

A.3.2 Benchmarking RNN cells

To show the advantage of tensorized RNNs over vanilla RNNs, we benchmark these architectures on the task of finding the ground state of the uniform ferromagnetic Ising chain (i.e., $J_{i,i+1} = 1$) with $N = 100$ spins at the critical point (i.e., no annealing is employed). Since the couplings in this model are site-independent, we choose the parameters of the model to be also site-independent. In Fig. A.3, we plot the energy variance per site σ^2 (3.12) against the number of gradient descent steps. The results show that the tensorized RNN wave function can achieve both a lower estimate of the energy variance and a faster convergence.

For the disordered systems studied in this study, we set the RNN parameters to be site-dependent. To demonstrate the benefit of using site-dependent over site-independent parameters when dealing with disordered systems, we benchmark both architectures on the task of finding the ground state of the disordered Ising chain with random discrete couplings $J_{i,i+1} = \pm 1$ at the critical point, i.e., with a transverse field $\Gamma = 1$. We show the results in Fig. A.4 and find that site-dependent parameters lead to a better performance in terms of the energy variance per spin σ^2.

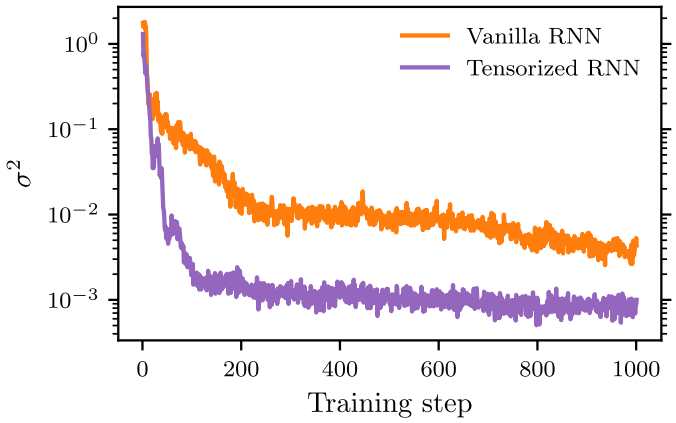

Figure A.3: Energy variance per spin σ^2 vs the number of training steps. Here we compare tensorized and vanilla RNN ansatzes both with weight sharing across sites on the uniform ferromagnetic Ising chain at the critical point with $N = 100$ spins.

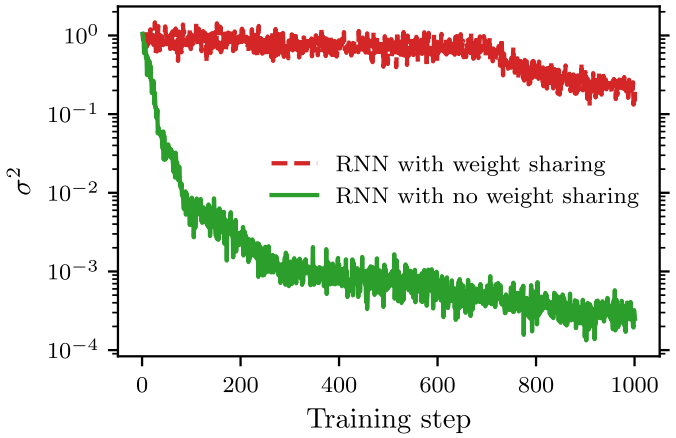

Figure A.4: Comparison between a tensorized RNN with and without weight sharing, trained to find the ground state of the random Ising chain with a discrete disorder ($J_{i,i+1} = \pm 1$) at criticality with $N = 20$ spins.

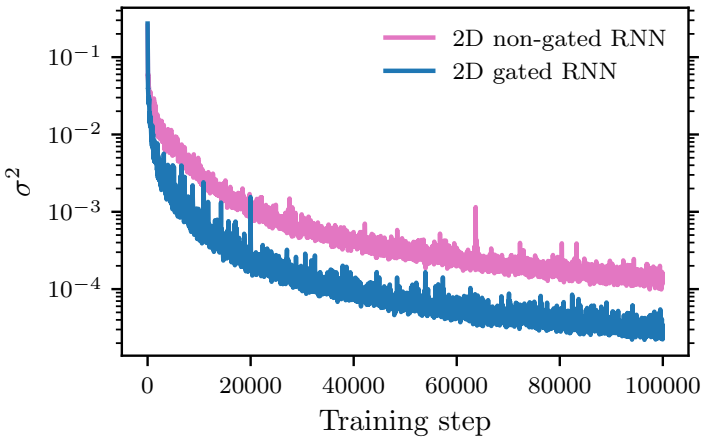

Figure A.5: A plot of the energy variance per spin σ^2 against the number of gradient descent steps, for both a 2D non-gated RNN and a 2D gated RNN. Here we choose the Heisenberg model on a square lattice with size 6×6 as a test bed.

To show the advantage of a 2D TGRU (gated RNN) over the 2D TRNN (non-gated RNN), we train them using the VMC scheme to find the ground state of the Heisenberg model on the square lattice with size 6×6. We find that the gated TRNN allows obtaining more accurate energy as illustrated in Fig. A.5. We observe that the gated RNN can get about an order of magnitude lower σ^2 compared to the non-gated RNN. The latter results demonstrate the advantage of adding the gating mechanism to our wave function ansatz.

Furthermore, we equally show the advantage of a dilated RNN ansatz compared to a tensorized RNN ansatz. We train both of them for the task of finding the minimum of the free energy of the Sherrington-Kirkpatrick model with $N = 20$ spins and at temperature $T = 1$ (see Eq. (6.11)). Both RNNs have a comparable number of parameters (66400 parameters for the tensorized RNN and 59240 parameters for the dilated RNN). In Fig. A.6, we find that the dilated RNN supersedes the tensorized RNN with almost an order of magnitude difference in terms of the free energy variance per spin defined in Eq. (3.30). Indeed, this result suggests that the mechanism of skip connections allows dilated RNNs to capture long-term dependencies more efficiently compared to tensorized RNNs with a single layer.

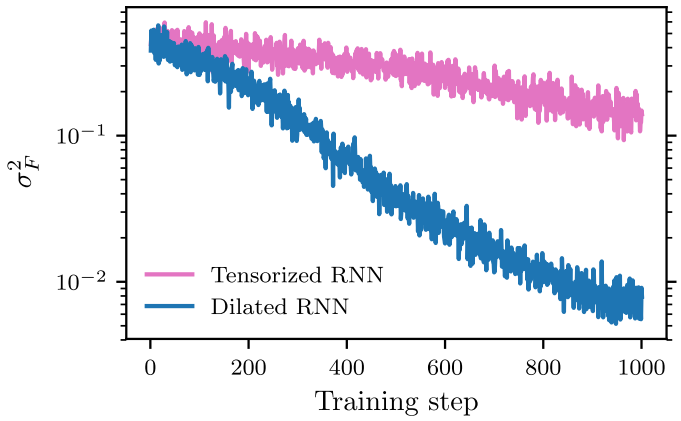

Figure A.6: Free energy variance per spin σ^2 vs the number of training steps. Here we compare a tensorized RNN with one-layer and dilated RNN ansatzes, both with no weight sharing, trained to find the Sherrington-Kirkpatrick model's equilibrium distribution with $N = 20$ spins at temperature $T = 1$.

Figures	Hyperparameter	Value
Figs. 5.1,5.2, 5.3	Architecture	One-layer 1D pRNN wave function with 50 memory units
	Number of samples	$M = 1000$ ($N = 20$), $M = 500$ ($N = 80$), $M = 200$ ($N = 1000$)
	Training iterations	20000
	Learning rate	5×10^{-3}
	Seed	111
Fig. 5.5(c): 1DRNN	Architecture	Three-layer 1D pRNN wave function with 100 memory units
	Number of samples	500
	Training iterations	150000
	Learning rate	$(\eta^{-1} + 0.1t)^{-1}$ with $\eta = 10^{-3}$
	Seed	333
Fig. 5.5(c): 2DRNN	Architecture	One-layer 2D pRNN wave function with 100 memory units
	Number of samples	500
	Training iterations	150000
	Learning rate	$\eta(1 + t/5000)^{-1}$ with $\eta = 5 \times 10^{-3}$
	Seed	111
Fig. 5.9(a)	Architecture	One-layer 1D pRNN wave function
	Number of samples	500
	Training iterations	10000
	Learning rate	10^{-3}
	Seeds	$111, 222, 333, 444, 555$
Fig. 5.9(b)	Architecture	One-layer 1D pRNN wave function
	Number of samples	500
	Training iterations	10000
	Learning rate	$(\eta^{-1} + 0.1t)^{-1}$ with $\eta = 10^{-3}$
	Seeds	$111, 222, 333, 444, 555, 666, 777, 888, 999, 1111$
Fig. A.1	Architecture	One-layer 1D pRNN wave function with 50 memory units
	Training iterations	10000
	Learning rate	10^{-3}
	Seeds	$111, 222, 333, 444, 555$
Fig. A.2	Architecture	1D pRNN wave function
	Number of samples	500
	Training iterations	10000
	Learning rate	5×10^{-3}
	Seeds	$111, 222, 333, 444, 555$

Table A.1: Hyperparameters used to obtain the results reported for the 1D TFIM and the 2D TFIM, as well as to benchmark the RNN cells in App. A.3.2. Note that the number of samples stands for the batch size used to train the RNN wave function. Multiple seeds are used for the scaling of resources study to provide error bars on our results.

Figures	Hyperparameter	Value
Fig. 5.4	Architecture	Three-layer 1D cRNN wave function with 100 memory units
	Number of samples	500
	Training iterations	100000
	Learning rate	$(\eta^{-1} + 0.1t)^{-1}$ with $\eta = 2.5 \times 10^{-4}$
	Seed	111

Table A.2: Hyperparameters used to obtain the results of the 1D J_1-J_2 model reported in Sec. 5.1.2.

Figures	Hyperparameter	Value
Figs. A.3 and A.4	Architecture	RNN wave function
	Number of memory units	$d_h = 50$
	Number of samples	$M = 50$
	Learning rate	$\eta = 10^{-3}$ for Fig. A.3 and $\eta = 5 \times 10^{-4}$ for Fig. A.4
Fig. A.5	Architecture	RNN wave function with no-weight sharing
	Number of memory units of dilated RNN	$d_h = 20$
	Number of memory units of tensorized RNN	$d_h = 40$
	Number of samples	$M = 100$
	Learning rate	$\eta = 10^{-4}$

Table A.3: Hyperparameters used to benchmark the RNN cells in App. A.3.2.

Figures	Hyperparameter	Value
Fig. 5.6(a)	Architecture	2D Tensorized Gated cRNN wave function
	Number of memory units	$d_h = 300$
	Number of samples	$M = 100$
	Learning rate	$\eta = 5 \times 10^{-4} \times (1 + (t/5000))^{-1}$
	Number of training steps	100000
Fig. 5.6(b)	Architecture	2D Tensorized Gated cRNN wave function
	Number of memory units	$d_h = 200$
	Number of samples	$M = 100$
	Learning rate	$\eta = 5 \times 10^{-4} \times (1 + (t/5000))^{-1}$
	Number of training steps	150000
	Applied symmetries	$U(1)$ and C_{4v}
Figs. 6.10(a)	Architecture	2D Tensorized Gated cRNN wave function
	Number of memory units	$d_h = 100$
	Number of samples	$M = 100$
	Learning rate	$\eta = 10^{-4}$
	Number of warmup steps	$N_{\text{warmup}} = 1000$
	Applied symmetries	$U(1), C_{2d}$ and spin parity
Fig. A.5(a)	Number of memory units	$d_h = 300$
	Number of samples	$M = 100$
	Learning rate	$\eta = 5 \times 10^{-4} \times (1 + (t/5000))^{-1}$
	Applied symmetries	$U(1)$ and C_{4v}

Table A.4: Hyperparameters used to obtain the results reported for the Heisenberg model.

J_2	E/N		
	No Sign	Marshall Sign	DMRG
0.0	-0.4412292(2)	-0.4412765(1)	-0.4412773
0.2	-0.4073672(2)	-0.4073873(1)	-0.4073881
0.5	-0.3749996(1)	-0.375	-0.375
0.8	-0.4205425(4)	-0.4205627(5)	-0.4207006

Table A.5: Energy per spin values for the 1D J_1-J_2 model. We consider a cRNN wave function with two different methods of training (with no initial sign structure and with a background Marshall sign) and compare it with results from DMRG. All results correspond to 100 spins and have $J_1 = 1$. We use three GRU layers, where each layer has 100 units. Note that $J_2 = 0.5$ corresponds to the Majumdar-Ghosh model where the ground state is a product state of spin singlets.

h	E/N			
	1DRNN	2DRNN	PixelCNN	DMRG
2	-2.4096018(2)	**-2.40960262(9)**	-2.4096022(2)	**-2.40960263**
3	-3.1738969(5)	**-3.1739018(2)**	-3.1739005(5)	-3.17389966
4	-4.1217969(3)	**-4.12179808(6)**	-4.1217979(2)	-4.12179793

Table A.6: Variational energies per site for a 1D pRNN wave function (3 layers of GRUs with 100 memory units), 2D pRNN wave function (single layer of 2D Vanilla RNN with 100 memory units), PixelCNN wave functions with results taken from Ref. [150] and DMRG (with bond dimension $\chi = 512$ for $h = 2$ and $\chi = 1024$ for both $h = 3, 4$). As a benchmark, we use the 2D TFIM with a lattice size of 12×12 for different values of h where the critical point is at $h \approx 3$. Values in bold font correspond to the lowest variational energies and hence to the most accurate estimations of the ground state energy across all four methods. For the estimation of the variational energy of the trained 1D and 2D pRNN wave functions, we use 2×10^6 samples.

PEPS	PixelCNN	DMRG	2DRNN	QMC
-0.628601(2)	-0.628627(1)	-0.6286335	-0.628638(1)	-0.628656(2)

Table A.7: A between the energies per site for the Heisenberg model on the square lattice 10×10. Here we compare PEPS [179], PixelCNN [150], DMRG [146] and QMC [179]. The trend is illustrated in Fig. 5.6(b).

Method/Size	6×6	8×8	10×10	12×12	14×14	16×16
DMRG	**-0.499048**	**-0.508561**	**-0.514136**	**-0.514419**	-0.508397	-0.505129
2DRNN	-0.4968(1)	-0.5049(1)	-0.5093(1)	-0.5120(1)	**-0.5138(1)**	**-0.5167(1)**

Table A.8: A table representing the comparison between the energies per site for the Heisenberg model on the triangular lattice for different system sizes. For the estimation of the 2DRNN energies, we used 20000 samples. Values in bold correspond to the lowest variational energies.

Model/Method	2DRNN (ours)	1D MPS-RNN	2D MPS-RNN	Tensor RNN
Square (10×10)	**-0.628638(1)**	-0.62587(1)	-0.627697(5)	-0.628528(4)
Triangular (10×10)	-0.5093(1)	-0.48964(2)	-0.50803(1)	**-0.513863(9)**

Table A.9: A comparison between our 2D tensorized RNNs with the tensorial RNN architectures in Ref. [146]. Values in bold correspond to the lowest variational energies.

Appendix B

Supplementary material of chapter 6

B.1 Numerical proof of principle of adiabaticity

As demonstrated in Sec. 6.3, we have shown that both VQA and VCA are effective at finding the classical ground state of disordered spin chains. Here, we provide an intuitive illustration of the adiabaticity of both VQA and VCA. First, we perform VQA on the uniform ferromagnetic Ising chain (i.e. $J_{i,i+1} = 1$) with $N = 20$ spins and open boundary conditions with an initial driving magnetic field $\Gamma_0 = 2$. Here, we use a tensorized pRNN wave function with weight sharing across sites of the chain. We also choose $N_{\text{annealing}} = 1024$. In Fig. B.1(a), we show that the RNN wave function energy matches the exact ground energy throughout the annealing process with high accuracy. Optimizing an RNN wave function from scratch at each new value of the transverse magnetic field is not optimal. This observation underlines the importance of transferring the parameters of our wave function ansatz after each annealing step. Furthermore, we illustrate in Fig. B.1(b) that the RNN wave function's residual energy is much lower compared to the gap throughout the annealing process. Indeed, this shows that VQA can be adiabatic for an annealing time that is long enough.

Similarly, in Fig. B.1(c), we perform VCA with an initial temperature $T_0 = 2$ on the same model, the same system size, the same ansatz, and the same number of annealing steps. We see an excellent agreement between the RNN wave function free energy and the exact free energy, highlighting once again the adiabaticity of our emulation of classical annealing, as well as the importance of transferring the parameters of our ansatz after each annealing step.

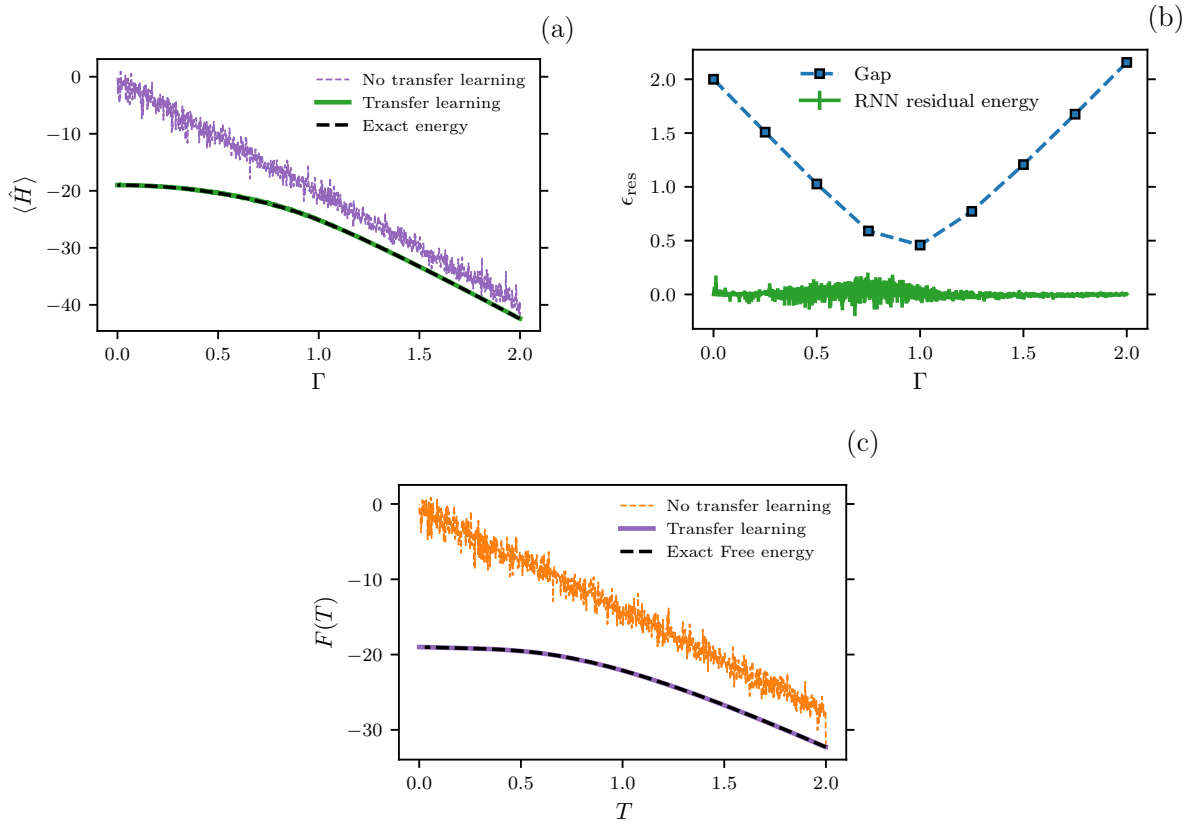

Figure B.1: Proofs of adiabaticity on the uniform Ising chain with $N = 20$ spins for VQA in panels (a) and (b) and VCA in panel (c). (a) A plot of the variational energy of the RNN wave function against the transverse magnetic, without parameter initialization at each annealing step for the green curve and with parameter initialization for the purple curve. We compare both plots with the exact energy obtained from exact diagonalization. (b) we plot the residual energy of the RNN wave function against the transverse magnetic field Γ. We show that throughout annealing with VQA, the residual energy is always much smaller than the gap within the error bars. (c) Similarly to panel (a), we do the same experiment with VCA while plotting the free energy against temperature. All these results support the conclusion that VQA and VCA evolutions can be adiabatic.

B.2 The variational adiabatic theorem

In this section, we derive a sufficient condition for the number of gradient descent steps needed to maintain the variational ansatz close to the instantaneous ground state throughout the VQA simulation. First, consider a variational wave function $|\Psi_{\boldsymbol{\lambda}}\rangle$ and the following the time-dependent Hamiltonian:

$$\hat{H}(t) = \hat{H}_{\text{target}} + f(t)\hat{H}_D,$$

The goal is to find the ground state of the target Hamiltonian $\hat{H}_{\text{target}}$ by introducing quantum fluctuations through a driving Hamiltonian $\hat{H}_D$, where $\hat{H}_D \gg \hat{H}_{\text{target}}$. Here $f(t)$ is a decreasing schedule function such that $f(0) = 1$, $f(1) = 0$ and $t \in [0, 1]$.

Let $E(\boldsymbol{\lambda}, t) = \langle\Psi_{\boldsymbol{\lambda}}| \hat{H}(t) |\Psi_{\boldsymbol{\lambda}}\rangle$, and $E_G(t), E_E(t)$ the instantaneous ground/excited state energy of the Hamiltonian $\hat{H}(t)$, respectively. The instantaneous energy gap is defined as $g(t) \equiv E_E(t) - E_G(t)$.

To simplify our discussion, we consider the case of a target Hamiltonian that has a non-degenerate ground state. Here, we decompose the variational wave function as:

$$|\Psi_{\boldsymbol{\lambda}}\rangle = (1 - a(t))^{\frac{1}{2}} |\Psi_G(t)\rangle + a(t)^{\frac{1}{2}} |\Psi_\perp(t)\rangle , \tag{B.1}$$

where $|\Psi_G(t)\rangle$ is the instantaneous ground state and $|\Psi_\perp(t)\rangle$ is a superposition of all the instantaneous excited states. From the results of Sec. 3.6, one can show that:

$$a(t) \leq \frac{E(\boldsymbol{\lambda}, t) - E_G(t)}{g(t)}. \tag{B.2}$$

As a consequence, in order to satisfy adiabaticity, i.e., $|\langle\Psi_\perp(t)|\Psi_{\boldsymbol{\lambda}}\rangle|^2 \ll 1$ for all times t, then one should have $a(t) < \epsilon \ll 1$ where ϵ is a small upper bound on the overlap between the variational wave function and the excited states. This means that the success probability P_{success} of obtaining the ground state at $t = 1$ is bounded from below by $1 - \epsilon$. From Eq. (B.2), to satisfy $a(t) < \epsilon$, it is sufficient to have:

$$\epsilon_{\text{res}}(\boldsymbol{\lambda}, t) \equiv E(\boldsymbol{\lambda}, t) - E_G(t) < \epsilon g(t). \tag{B.3}$$

To satisfy the latter condition, we require a slightly stronger condition as follows:

$$\epsilon_{\text{res}}(\boldsymbol{\lambda}, t) < \frac{\epsilon g(t)}{2}. \tag{B.4}$$

In our derivation of a sufficient condition on the number of gradient descent steps to satisfy the previous requirement, we use the following set of assumptions:

- **(A1)** $|\partial_t^k E_G(t)|, |\partial_t^k g(t)|, |\partial_t^k f(t)| \leq \mathcal{O}(\text{poly}(N))$, for all $0 \leq t \leq 1$ and for $k \in \{1, 2\}$.
- **(A2)** $|\langle \Psi_{\boldsymbol{\lambda}} | \hat{H}_D | \Psi_{\boldsymbol{\lambda}} \rangle| \leq \mathcal{O}(\text{poly}(N))$ for all possible parameters $\boldsymbol{\lambda}$ of the variational wave function.
- **(A3)** No anti-crossing during annealing, i.e., $g(t) \neq 0$, for all $0 \leq t \leq 1$.
- **(A4)** The gradients $\partial_{\boldsymbol{\lambda}} E(\boldsymbol{\lambda}, t)$ can be calculated exactly, are $L(t)$-Lipschitz with respect to $\boldsymbol{\lambda}$ and $L(t) \leq \mathcal{O}(\text{poly}(N))$ for all $0 \leq t \leq 1$.
- **(A5)** Local convexity, i.e., close to convergence when $\epsilon_{\text{res}}(\boldsymbol{\lambda}, t) < \epsilon g(t)$, the energy landscape of $E(\boldsymbol{\lambda}, t)$ is convex with respect to $\boldsymbol{\lambda}$, for all $0 < t \leq 1$.

  Note that this assumption is ϵ-dependent.
- **(A6)** The parameters vector $\boldsymbol{\lambda}$ is bounded by a polynomial in N. i.e., $||\boldsymbol{\lambda}|| \leq \mathcal{O}(\text{poly}(N))$, where we define "$||.||$" as the euclidean L_2 norm.
- **(A7)** The variational wave function $|\Psi_{\boldsymbol{\lambda}}\rangle$ is expressive enough, i.e.,

$$\min_{\boldsymbol{\lambda}} \epsilon_{\text{res}}(\boldsymbol{\lambda}, t) < \frac{\epsilon g(t)}{4}, \quad \forall t \in [0, 1].$$

  Note that this assumption is also ϵ-dependent.
- **(A8)** At $t = 0$, the energy landscape of $E(\boldsymbol{\lambda}, t = 0)$ is globally convex with respect to $\boldsymbol{\lambda}$.

Theorem Given the assumptions **(A1)** to **(A8)**, a sufficient (but not necessary) number of gradient descent steps N_{steps} to satisfy the condition (B.4) during the VQA protocol, is bounded as:

$$\mathcal{O}\left(\frac{\text{poly}(N)}{\epsilon \min_{\{t_n\}}(g(t_n))}\right) \leq N_{\text{steps}} \leq \mathcal{O}\left(\frac{\text{poly}(N)}{\epsilon^2 \min_{\{t_n\}}(g(t_n))^2}\right),$$

where $(t_1, t_2, t_3, \ldots)$ is an increasing finite sequence of time steps, satisfying $t_1 = 0$ and $t_{n+1} = t_n + \delta t_n$, where

$$\delta t_n = \mathcal{O}\left(\frac{\epsilon g(t_n)}{\text{poly}(N)}\right).$$

Proof: In order to satisfy the condition Eq. (B.4) during the VQA protocol, we follow these steps:

- Step 1 (warm-up step): we prepare our variational wave function at the ground state at $t = 0$ such that Eq. (B.4) is verified at time $t = 0$.
- Step 2 (annealing step): we change time t by an infinitesimal amount δt so that the condition (B.3) is verified at time $t + \delta t$.
- Step 3 (training step): we tune the parameters of the variational wave function, using gradient descent, so that the condition (B.4) is satisfied at time $t + \delta t$.
- Step 4: we loop over steps 2 and 3 until we arrive at $t = 1$, where we expect to obtain the ground state energy of the target Hamiltonian.

Let us first start with step 2 assuming that step 1 is verified. In order to satisfy the requirement of this step at time t, then δt has to be chosen small enough so that

$$\epsilon_{\text{res}}(\boldsymbol{\lambda}_t, t + \delta t) < \epsilon g(t + \delta t) \tag{B.5}$$

is verified given that the condition (B.4) is satisfied at time t. Here, $\boldsymbol{\lambda}_t$ are the parameters of the variational wave function that satisfies the condition (B.4) at time t. To get a sense of how small δt should be, we do a Taylor expansion, while fixing the parameters $\boldsymbol{\lambda}_t$, to get:

$$\begin{aligned}
&\epsilon_{\text{res}}(\boldsymbol{\lambda}_t, t + \delta t) \\
&= \epsilon_{\text{res}}(\boldsymbol{\lambda}_t, t) + \partial_t \epsilon_{\text{res}}(\boldsymbol{\lambda}_t, t)\delta t + \mathcal{O}((\delta t)^2), \\
&< \frac{\epsilon g(t)}{2} + \partial_t \epsilon_{\text{res}}(\boldsymbol{\lambda}_t, t)\delta t + \mathcal{O}((\delta t)^2),
\end{aligned}$$

where we used the condition (B.4) to go from the second line to the third line. Here, $\partial_t \epsilon_{\text{res}}(\boldsymbol{\lambda}_t, t) = \partial_t f(t) \langle \hat{H}_D \rangle - \partial_t E_G(t)$. To satisfy the condition (B.3) at time $t + \delta t$, it is enough to have the right-hand side of the previous inequality to be much smaller than the gap at $t + \delta t$, i.e.,

$$\frac{\epsilon g(t)}{2} + \partial_t \epsilon_{\text{res}}(\boldsymbol{\lambda}_t, t)\delta t + \mathcal{O}((\delta t)^2) < \epsilon g(t + \delta t).$$

By Taylor expanding the gap, we get:

$$\partial_t \epsilon_{\text{res}}(\boldsymbol{\lambda}_t, t)\delta t + \mathcal{O}((\delta t)^2) < \frac{\epsilon g(t)}{2} + \epsilon \partial_t g(t)\delta t + \mathcal{O}((\delta t)^2),$$

hence, it is enough to satisfy the following condition:

$$(\partial_t \epsilon_{\text{res}}(\boldsymbol{\lambda}_t, t) - \epsilon \partial_t g(t))\delta t + \mathcal{O}((\delta t)^2) < \frac{\epsilon g(t)}{2}. \tag{B.6}$$

Using the Taylor-Laplace formula, one can express the Taylor remainder term $\mathcal{O}((\delta t)^2)$ as follows:

$$\mathcal{O}((\delta t)^2) = \int_t^{t+\delta t} (\tau - t) A(\tau) \mathrm{d}\tau,$$

where $A(\tau) = \partial_\tau^2 \epsilon_{\mathrm{res}}(\boldsymbol{\lambda}_t, \tau) - \epsilon \partial_\tau^2 g(\tau) = \partial_\tau^2 f(\tau) \langle \hat{H}_D \rangle - \partial_\tau^2 E_G(\tau) - \epsilon \partial_\tau^2 g(\tau)$ and τ is between t and $t + \delta t$. The last expression can be bounded as follows:

$$\mathcal{O}((\delta t)^2) \leq \int_t^{t+\delta t} (\tau - t) |A(\tau)| \mathrm{d}\tau \leq \frac{(\delta t)^2}{2} \mathrm{sup}(|A|).$$

where "sup(|A|)" is the supremum of $|A|$ over the interval $[0, 1]$. Given assumptions **(A1)** and **(A2)**, then sup(|A|) is bounded from above by a polynomial in N, hence:

$$\mathcal{O}((\delta t)^2) \leq \mathcal{O}(\mathrm{poly}(N))(\delta t)^2 \leq \mathcal{O}(\mathrm{poly}(N))\delta t,$$

where the last inequality holds since $\delta t \leq 1$ as $t \in [0, 1]$, while we note that it is not necessarily tight. Furthermore, since $(\partial_t \epsilon_{\mathrm{res}}(\boldsymbol{\lambda}_t, t) - \epsilon \partial_t g(t))$ is also bounded from above by a polynomial in N (according to assumptions **(A1)** and **(A2)**), then in order to satisfy Eq. (B.6), it is sufficient to require the following condition:

$$\mathcal{O}(\mathrm{poly}(N))\delta t < \frac{\epsilon g(t)}{2}.$$

Thus, it is sufficient to take:

$$\delta t = \mathcal{O}\left(\frac{\epsilon g(t)}{\mathrm{poly}(N)}\right). \tag{B.7}$$

By taking account of assumption (**A3**), δt can be taken non-zero for all time steps t. As a consequence, assuming the condition (B.7) is verified for a non-zero δt and a suitable $\mathcal{O}(1)$ prefactor, then the condition (B.5) is also verified.

We can now move to step 3. Here, we apply a number of gradient descent steps $N_{\mathrm{train}}(t)$ to find a new set of parameters $\boldsymbol{\lambda}_{t+\delta t}$ such that:

$$\epsilon_{\mathrm{res}}(\boldsymbol{\lambda}_{t+\delta t}, t + \delta t) = E(\boldsymbol{\lambda}_{t+\delta t}, t + \delta t) - E_G(t + \delta t) < \frac{\epsilon g(t + \delta t)}{2}, \tag{B.8}$$

To estimate the scaling of the number of gradient descent steps $N_{\mathrm{train}}(t)$ needed to satisfy (B.8), we make use of assumptions **(A4)** and **(A5)**. The assumption **(A5)** is reasonable provided that the variational energy $E(\boldsymbol{\lambda}_t, t + \delta t)$ is very close to the ground state energy $E_G(t + \delta t)$, as given by Eq. (B.5). Using the above assumptions and assuming that the

learning rate $\eta(t) = 1/L(t)$, we can use a well-known result in convex optimization [318](see Sec. 2.1.5), which states the following inequality:

$$E(\tilde{\boldsymbol{\lambda}}_t, t+\delta t) - \min_{\boldsymbol{\lambda}} E(\boldsymbol{\lambda}, t+\delta t) \leq \frac{2L(t)||\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^*_{t+\delta t}||^2}{N_{\text{train}}(t)+4}.$$

Here, $\tilde{\boldsymbol{\lambda}}_t$ are the new variational parameters obtained after applying $N_{\text{train}}(t+\delta t)$ gradient descent steps starting from $\boldsymbol{\lambda}_t$. Furthermore, $\boldsymbol{\lambda}^*_{t+\delta t}$ are the optimal parameters such that:

$$E(\boldsymbol{\lambda}^*_{t+\delta t}, t+\delta t) = \min_{\boldsymbol{\lambda}} E(\boldsymbol{\lambda}, t+\delta t).$$

Since the Lipschitz constant $L(t) \leq \mathcal{O}(\text{poly}(N))$ (assumption **(A4)**) and $||\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^*_{t+\delta t}||^2 \leq \mathcal{O}(\text{poly}(N))$ (assumption **(A6)**), one can take

$$N_{\text{train}}(t+\delta t) = \mathcal{O}\left(\frac{\text{poly}(N)}{\epsilon g(t+\delta t)}\right), \tag{B.9}$$

with a suitable $\mathcal{O}(1)$ prefactor, so that:

$$E(\tilde{\boldsymbol{\lambda}}_t, t+\delta t) - \min_{\boldsymbol{\lambda}} E(\boldsymbol{\lambda}, t+\delta t) < \frac{\epsilon g(t+\delta t)}{4}.$$

Moreover, by assuming that the variational wave function is expressive enough (assumption **(A7)**), i.e.,

$$\min_{\boldsymbol{\lambda}} E(\boldsymbol{\lambda}, t+\delta t) - E_G(t+\delta t) < \frac{\epsilon g(t+\delta t)}{4},$$

we can then deduce, by taking $\boldsymbol{\lambda}_{t+\delta t} \equiv \tilde{\boldsymbol{\lambda}}_t$ and summing the two previous inequalities, that:

$$E(\boldsymbol{\lambda}_{t+\delta t}, t+\delta t) - E_G(t+\delta t) < \frac{\epsilon g(t+\delta t)}{2}.$$

Let us recall that in step 1, we have to initially prepare the variational ansatz to satisfy condition (B.4) at $t = 0$. In fact, we can take advantage of the assumption **(A4)**, where the gradients are $L(0)$-Lipschitz with $L(0) \leq \mathcal{O}(\text{poly}(N))$. We can also use the convexity assumption **(A8)**, and we can show that a sufficient number of gradient descent steps to satisfy condition (B.4) at $t = 0$ is estimated as:

$$N_{\text{warmup}} \equiv N_{\text{train}}(0) = \mathcal{O}\left(\frac{\text{poly}(N)}{\epsilon g(0)}\right).$$

The latter can be obtained in a similar way as in Eq. (B.9).

In conclusion, the total number of gradient steps N_{steps} to evolve the Hamiltonian $\hat{H}(0)$ to the target Hamiltonian $\hat{H}(1)$, while verifying the condition (B.4) is given by:

$$N_{\text{steps}} = \sum_{n=1}^{N_{\text{annealing}}+1} N_{\text{train}}(t_n),$$

where each $N_{\text{train}}(t_n)$ satisfies the requirement (B.9). The annealing times $\{t_n\}_{n=1}^{N_{\text{annealing}}+1}$ are defined such that $t_1 \equiv 0$ and $t_{n+1} \equiv t_n + \delta t_n$. Here, δt_n satisfies

$$\delta t_n = \mathcal{O}\left(\frac{\epsilon g(t_n)}{\text{poly}(N)}\right). \tag{B.10}$$

We also consider $N_{\text{annealing}}$ the smallest integer such that $t_{N_{\text{annealing}}} + \delta t_{N_{\text{annealing}}} \geq 1$, in this case, we define $t_{N_{\text{annealing}}+1} \equiv 1$, indicating the end of annealing. Thus, $N_{\text{annealing}}$ is the total number of annealing steps. Taking this definition into account, then one can show that

$$N_{\text{annealing}} \leq \frac{1}{\min\limits_{\{t_n\}}(\delta t_n)} + 1.$$

Using Eqs. (B.7) and (B.9) and the previous inequality, N_{steps} can be bounded from above as:

$$\begin{aligned} N_{\text{steps}} &\leq (N_{\text{annealing}} + 1) \max_{\{t_n\}} (N_{\text{train}}(t_n)) \\ &\leq \left(\frac{1}{\min\limits_{\{t_n\}}(\delta t_n)} + 2\right) \max_{\{t_n\}} (N_{\text{train}}(t_n)) \\ &\leq \mathcal{O}\left(\frac{\text{poly}(N)}{\epsilon^2 \min\limits_{\{t_n\}}(g(t_n))^2}\right), \end{aligned}$$

where the transition from line 2 to line 3 is valid for a sufficiently small ϵ and $\min_{\{t_n\}}(g(t_n))$. Furthermore, N_{steps} can also be bounded from below as:

$$N_{\text{steps}} \geq \max_{\{t_n\}}(N_{\text{train}}(t_n)) = \mathcal{O}\left(\frac{\text{poly}(N)}{\epsilon \min\limits_{\{t_n\}}(g(t_n))}\right). \tag{B.11}$$

Note that the minimum in the previous two bounds is taken over all the annealing times t_n where $1 \leq n \leq N_{\text{annealing}} + 1$.

In this derivation of the bound on N_{steps}, we have assumed that the ground state of $\hat{H}_{\text{target}}$ is non-degenerate so that the gap does not vanish at the end of annealing (i.e., $t = 1$). In the case of the degeneracy of the target ground state, we can define the gap $g(t)$ by considering the lowest energy level that does not lead to the degenerate ground state.

It is also worth noting that the assumptions of this derivation can be further expanded and improved. In particular, the gradients of $E(\boldsymbol{\lambda}, t)$ are computed stochastically (see Chap. 3), as opposed to our assumption **(A4)** where the gradients are assumed to be known exactly. To account for noisy gradients, it is possible to use convergence bounds of stochastic gradient descent [80, 319] to estimate a bound on the number of gradient descent steps. Second-order optimization methods such as stochastic reconfiguration/natural gradient [2, 81] can potentially show a significant advantage over first-order optimization methods, in terms of scaling with the minimum gap of the time-dependent Hamiltonian $\hat{H}(t)$.

B.3 Simulated Quantum Annealing and Simulated Annealing

Simulated Quantum Annealing is a standard quantum-inspired classical technique that has traditionally been used to benchmark the behavior of quantum annealers [320]. It is usually implemented via the path-integral Monte Carlo method [202], a QMC method that simulates the equilibrium properties of quantum systems at finite temperatures. To illustrate this method, consider a D-dimensional time-dependent quantum Hamiltonian

$$\hat{H}(t) = -\sum_{i,j} J_{ij} \hat{\sigma}_i^z \hat{\sigma}_j^z - \Gamma(t) \sum_{i=1}^{N} \hat{\sigma}_i^x,$$

where $\Gamma(t) = \Gamma_0(1-t)$ controls the strength of the quantum annealing dynamics at a time $t \in [0, 1]$. By applying the Suzuki-Trotter formula to the partition function of the quantum system,

$$Z = \text{Tr} \exp\{-\beta \hat{H}(t)\}, \tag{B.12}$$

with the inverse temperature $\beta = \frac{1}{T}$, we can map the D-dimensional quantum Hamiltonian onto a $(D+1)$ classical system consisting of P coupled replicas (Trotter slices) of the original

system:

$$H_{D+1}(t) = -\sum_{k=1}^{P}\left(\sum_{i,j} J_{ij}\sigma_i^k\sigma_j^k + J_\perp(t)\sum_{i=1}^{N}\sigma_i^k\sigma_i^{k+1}\right), \tag{B.13}$$

where σ_i^k is the classical spin at site i and replica k. The term $J_\perp(t)$ corresponds to uniform coupling between σ_i^k and σ_i^{k+1} for each site i, such that

$$J_\perp(t) = -\frac{PT}{2}\ln\left(\tanh\left(\frac{\Gamma(t)}{PT}\right)\right).$$

We note that periodic boundary conditions $\sigma^{P+1} \equiv \sigma^1$ arise because of the trace in Eq. (B.12).

Interestingly, we can approximate Z with an effective partition function Z_p at temperature PT given by [218]:

$$Z_p \propto \mathrm{Tr}\exp\left\{-\frac{H_{\mathrm{D}+1}(t)}{PT}\right\},$$

which can now be simulated with a standard Metropolis-Hastings Monte Carlo algorithm. A key element to this algorithm is the energy difference induced by a single spin flip at site σ_i^k, which is equal to

$$\Delta_i E_{\mathrm{local}} = 2\sum_j J_{ij}\sigma_i^k\sigma_j^k + 2J_\perp(t)\left(\sigma_i^{k-1}\sigma_i^k + \sigma_i^k\sigma_i^{k+1}\right).$$

Here, the second term encodes the quantum dynamics. In our simulations, we consider single spin-flip (local) moves applied to all sites in all slices. We can also perform a global move [218], which means flipping a spin at location i in every slice k. Clearly, this has no impact on the term dependent on $J_\perp$, because it contains only terms quadratic in the flipped spin, so that

$$\Delta_i E_{\mathrm{global}} = 2\sum_{k=1}^{P}\sum_j J_{ij}\sigma_i^k\sigma_j^k.$$

In summary, a single Monte Carlo step (MCS) consists of first performing a single local move on all sites in each k-th slice and on all slices, followed by a global move for all sites. For the SK model and the WPE model studied in Sec. 6.3, we use $P = 100$, whereas for the EA model, we use $P = 20$ similarly to Ref. [202]. Before starting the quantum annealing schedule, we first thermalize the system by performing SA [218] from a temperature $T_0 = 3$ to a final temperature $1/P$ (so that $PT = 1$). This is done in 60

steps, where at each temperature we perform 100 Metropolis moves on each site. We then perform SQA using a linear schedule that decreases the field from Γ_0 to a final value close to zero $\Gamma(t=1) = 10^{-8}$, where five local and global moves are performed for each value of the magnetic field $\Gamma(t)$ so that it is consistent with the choice of $N_{\text{train}} = 5$ for VCA (see Sec. 6.1 and 6.3). Thus, the number of MCS is equal to five times the number of annealing steps.

For the standalone SA, we decrease the temperature from T_0 to $T(t=1) = 10^{-8}$. Here, a single MCS consists of a Monte Carlo sweep, i.e., attempting a spin-flip for all sites. For each thermal annealing step, we perform five MCS, and hence similar to SQA, the number of MCS is equal to five times the number of annealing steps. Furthermore, we do a warm-up step for SA, by performing N_{warmup} MCS to equilibrate the Markov Chain at the initial temperature T_0 and to provide a consistent choice with VCA (see Sec. 6.1).

B.4 Non-stoquastic Hamiltonians

In Sec. 6.4, we have demonstrated the possibility of variationally emulating the dynamics of a non-stoquastic Hamiltonian using a tensorized complex RNN wave function. In this Appendix, we demonstrate the adiabaticity of VQA with a non-stoquastic term for the uniform Ising chain with $N = 20$ spins, as shown in Figs. B.2(a) and (b), where we see a very good agreement between the complex RNN ansatz energies and the exact ground state energies throughout the annealing process. To further illustrate the feasibility of this approach in the presence of a non-trivial sign structure that is different from the first and the second Marshall sign rules [153], we repeat the same experiment for a disordered Ising chain with discrete couplings $J_{i,i+1} \in \{-1, 1\}$. Here, we use a frustrated driving term $\hat{H}_D = +\lambda_0 \sum_{i=1}^{N-4} \sum_{k=1}^{4} \hat{\sigma}_i^x \hat{\sigma}_{i+k}^x$ for a system size $N = 64$ spins. The result shown in Fig. B.2(c) clearly illustrates that more annealing steps help achieve lower residual errors despite the presence of a non-trivial sign structure.

In our simulation of VQA with a non-stoquastic driving term, we start with an initial value of the coupling $\lambda_0 = 2$ of the non-stoquastic driving term. In the warm-up phase, we use a pseudo-entropy term (see Sec. 6.5) so that the total cost function is given by a pseudo variational free energy as in Eq. (3.31):

$$\begin{aligned} \tilde{F}_{\boldsymbol{\lambda}}(t, t') = \langle \hat{H}_{\text{target}} \rangle_{\boldsymbol{\lambda}} + \lambda_0 (1-t) \langle \hat{H}_D \rangle_{\boldsymbol{\lambda}} \\ - T_0 \, (1-t') \, S_{\text{classical}}(|\Psi_{\boldsymbol{\lambda}}|^2). \end{aligned} \tag{B.14}$$

We found that adding such a term helps to avoid local minima, potentially due to the sign of the ground state. Initially, while $t = t' = 0$, we take an initial temperature $T_0 = 4$ and

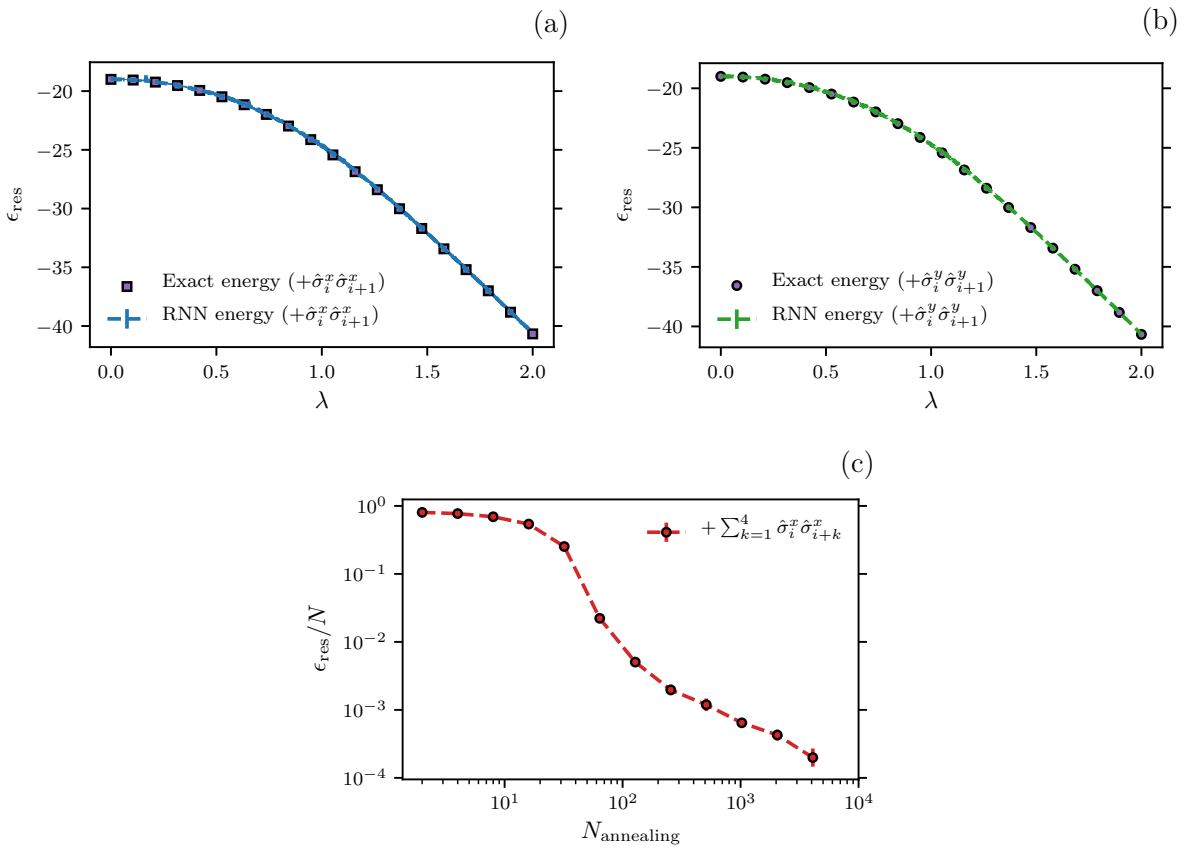

Figure B.2: In panels (a) and (b): we perform VQA with $N_{\text{annealing}} = 1024$ and using a tensorized cRNN wave function on the uniform Ising chain ($J_{i,i+1} = 1$) with $N = 20$ spins for both $+\hat{\sigma}_i^x\hat{\sigma}_{i+1}^x$ and $+\hat{\sigma}_i^y\hat{\sigma}_{i+1}^y$ non-stoquastic driving terms. Here, we see a very good agreement with the exact ground state energy. In panel (c): we perform VQA using a tensorized cRNN wave function on the disordered Ising chain ($J_{i,i+1} = \pm 1$) with $N = 64$ spins and a non-stoquastic driving term, shown in the legend, that has a non-trivial sign structure. Here, we plot the residual energy per site against the number of annealing steps and we see that the complex ansatz allows for systematic improvement upon increasing the number of annealing steps $N_{\text{annealing}}$.

we optimize for 1000 gradient steps, we then use a linear schedule and vary the time t' from 0 to 1 over 3000 annealing steps and one gradient step for each annealing step. Once $t' = 1$, we then optimize the tensorized cRNN wave function for another 1000 gradient steps while $t = 0$. Finally, we choose a different number of annealing steps to perform VQA, by varying $t = 0 \rightarrow 1$ using the non-stoquastic driving terms considered in Sec. 6.4. We note this idea could also be applied to find ground states of quantum Hamiltonians where local minima are serious limitations, as suggested in Sec. 6.7.

B.5 Additional results

In this section, we provide additional results connected with the Edwards-Anderson and the fully connected models in Sec. 6.5.

In Fig. B.3(a), we provide additional evidence that VCA is superior to SA and SQA on a larger system size compared to Fig. 6.6(b) for the EA model. Here, we do the comparison for a system size 60×60. We use a single disorder realization to avoid extra computational resources. In this case, the residual energy ϵ_{res} is defined

$$\epsilon_{\text{res}} = \langle \hat{H} \rangle - E_G, \tag{B.15}$$

where $\langle \ldots \rangle$ is the arithmetic mean over the different runs for SA and SQA and over the samples obtained at the end of annealing from the RNN in the VCA protocol. Our results in Fig. B.3(a) illustrate that VCA is still superior, in terms of the average residual energy, to SA and SQA for the range of $N_{\text{annealing}}$ shown in our plot.

In Fig. B.3(b), we show a comparison between SA, SQA, and VCA on the SK model with $N = 100$ spins. Similarly to Fig. 6.7(a), we do the same comparison, but for an order of magnitude larger $N_{\text{annealing}}$. Here, We use a single instance to avoid using excessive computing resources. We use the same definition of ϵ_{res} in Eq. (B.15). Similarly to the conclusion of Fig. 6.7(a), we still see that VCA provides more accurate solutions on average compared to SA and SQA.

To show the advantage of autoregressive sampling of RNNs, we perform principal component analysis (PCA) on the samples obtained from the RNN at the end of annealing after $N_{\text{annealing}} = 10^5$ steps. We obtain the results in Fig. B.3(c). We observe that the RNN recovers the two ground state configurations $\pm\boldsymbol{\sigma}^*$ as demonstrated by the two violet clusters. Here, we define the distance of a configuration $\boldsymbol{\sigma}$ from the two ground states $\pm\boldsymbol{\sigma}^*$ as

$$D_{\text{res}} = \sqrt{||\boldsymbol{\sigma} - \boldsymbol{\sigma}^*||_1 ||\boldsymbol{\sigma} + \boldsymbol{\sigma}^*||_1}, \tag{B.16}$$

that we represent in the color bar of Fig. B.3(c), where $||.||_1$ is the L_1 norm. These observations show that RNNs are indeed capable of capturing and sampling multiple modes, as opposed to Markov-chain Monte Carlo methods where sampling multiple modes at very low temperatures is often a challenging task when studying spin-glass models.

We finally demonstrate a detailed analysis of the results of Figs. 6.7(d), (e) and (f). Here, we provide the probabilities of success for each instance configuration that we attempt to solve using SA, SQA, and VCA. We note that the probability of success is computed as the ratio of the obtained ground states over the total number of configurations that are obtained for each method.

The results for SK ($N = 100$ spins) are shown in Fig. B.3(d), where it is clear that the RNN provides a very high probability of success compared to SA and SQA on the majority of instances. We observe the same behavior for WPE ($\alpha = 0.5$ and $N = 32$ spins) in Fig. B.3(e). It is worth noting that VCA is not successful at solving a few instances, which could be related to the need to increase $N_{\text{annealing}}$ or to the need to improve the training scheme or representational power of dilated RNNs. Finally, in Fig. B.3(f), we see that WPE ($\alpha = 0.25$ and $N = 32$ spins) is indeed a challenging system for all the methods considered in our work. We also observe that VCA manages to get a very high probability of success on one instance while failing at solving the other instances. Furthermore, we note that SQA was not successful for all the instances, while SA succeeds at finding the ground state for 5 instances with a low probability of success $\sim 10^{-3}$.

B.6 Running time

In this section, we present a summary of the running time estimations for VCA, SA, and SQA, which are shown in Tab. B.1.

B.7 Hyperparameters

In this appendix, we summarize the architectures and the hyperparameters of the simulations performed in Sec. 6.3 to Sec. 6.5 as shown in Tab. B.2. The VCA hyperparameters, used in Sec. 6.6, are detailed in Tab. B.3. Additionally, the hyperparameters in SA of Sec. 6.6 (see Tab. B.4) are chosen to be consistent with VCA hyperparameters.

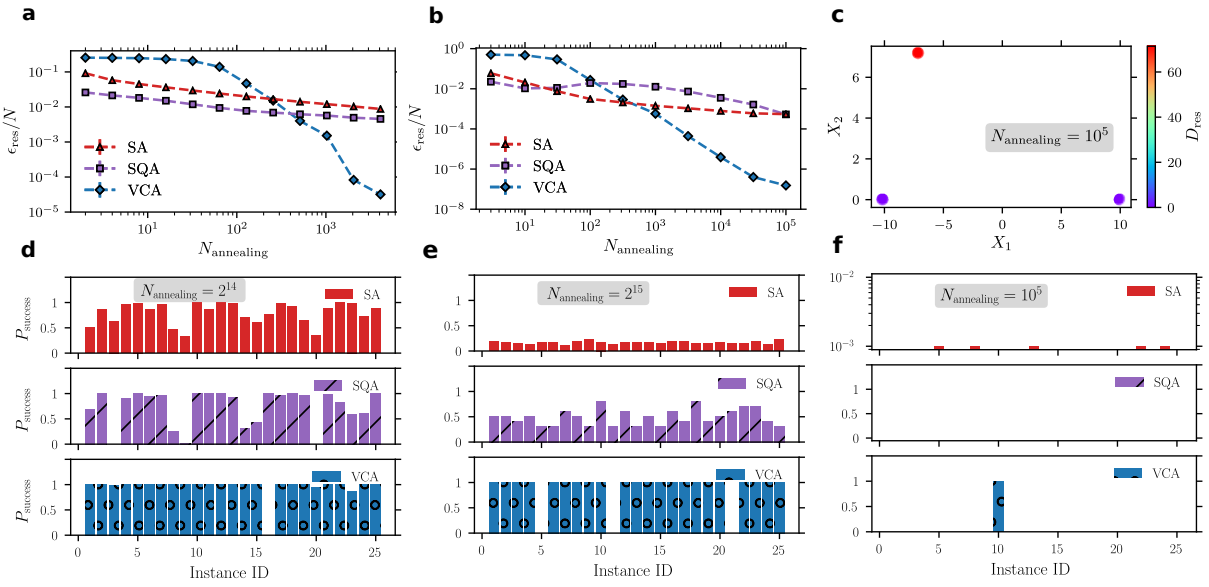

Figure B.3: (a) Comparison between SA, SQA, and VCA on a single instance of the EA model with system size (60×60). 1000 independent runs are performed for SA and, 50 annealing runs for SQA to estimate error bars. For VCA, we estimate error bars using 10^6 configurations obtained from the RNN at the end of annealing. (b) Similar comparison as in Panel (a) on a single realization of the SK model with $N = 100$ spins. (c) A plot of the two principal components after performing PCA on 50000 configurations obtained from the RNN, at the end of VCA protocol when applied to the SK model with $N = 100$ spins as in Panel (b) for $N_{\text{annealing}} = 10^5$. The color bar represents the distance D_{res}, defined in Eq. (B.16), from the two ground state configurations. Panels (d), (e), and (f) display the probability of success on the 25 instances of the SK and WPE models used respectively in Figs. 6.7(d), (e) and (f). Each probability of success P_{success} is computed using 1000 data points.

Model	Method	Number of iterations per second
Edwards-Anderson ($N = 40 \times 40$) (cf. Fig. 6.6(b))	SA (25 annealing runs)	~ 120
	SQA (25 annealing runs)	~ 2.4
	VCA (25 samples)	~ 1.1
SK ($N = 100$) (cf. Fig. 6.7(a))	SA (50 annealing runs)	~ 290
	SQA (50 annealing runs)	~ 1.4
	VCA (50 samples)	~ 5
Wishart ($N = 32$, $\alpha = 0.5$) (cf. Fig. 6.7(b))	SA (50 annealing runs)	~ 1160
	SQA (50 annealing runs)	~ 4.6
	VCA (50 samples)	~ 18.5

Table B.1: A summary of the running times of SA, SQA, and VCA performed in Sec. 6.5. The iteration time for VCA is estimated as the time it takes to estimate the free energy and to compute and apply its gradients to the RNN parameters. For SA and SQA, it is estimated as the time it takes to complete one Monte Carlo step multiplied by the number of annealing runs. The values reported in this table are highly dependent on our numerical implementations, hyperparameters, and the devices we used in our simulations.

Figures	Parameter	Value
Figs. 6.4(a) and 6.4(b)	Architecture	Tensorized RNN wave function with no-weight sharing
	Number of memory units	$d_h = 40$
	Number of samples	$M = 50$
	Initial magnetic field for VQA	$\Gamma_0 = 2$
	Initial temperature for VCA	$T_0 = 1$
	Learning rate	$\eta = 5 \times 10^{-4}$
	Warmup steps	$N_{\text{warmup}} = 1000$
	Number of random instances	$N_{\text{instances}} = 25$
Fig. 6.5	Architecture	Tensorized RNN wave function with no weight-sharing
	Type of RNN wave function	pRNN (stoquastic) and cRNN (non-stoquastic)
	Number of memory units	$d_h = 40$
	Number of samples	$N_s = 50$
	Initial driving term coupling	$\lambda_0 = 2$ and $\Gamma_0 = 2$
	Learning rate	$\eta = 5 \times 10^{-4}$
	Number of warmup steps	$N_{\text{warmup}} = 5000$
	Number of random instances	$N_{\text{instances}} = 25$
Fig. 6.6(a), Fig. B.3(a)	Architecture	2D tensorized RNN wave function with no weight-sharing
	Number of memory units	$d_h = 40$
	Number of samples	$M = 25$
	Initial magnetic field	$\Gamma_0 = 1$ (for SQA, VQA and RVQA)
	Initial temperature	$T_0 = 1$ (for SA, VCA and RVQA)
	Learning rate	$\eta = 10^{-4}$
	Number of warmup steps	$N_{\text{warmup}} = 1000$ for 10×10 $N_{\text{warmup}} = 2000$ for 40×40 $N_{\text{warmup}} = 5000$ for 60×60
	Number of random instances	$N_{\text{instances}} = 25$ for Fig. 6.6(a), $N_{\text{instances}} = 1$ for Fig. B.3(a)
Figs. 6.7(a), (d) and Fig. B.3(b)	Architecture	Dilated RNN wave function with no weight-sharing
	Number of memory units	$d_h = 40$
	Number of samples	$M = 50$
	Initial temperature	$T_0 = 2$ (for SA and VCA)
	Initial magnetic field	$\Gamma_0 = 2$ (for SQA)
	Learning rate	$\eta = 10^{-4}$
	Number of warmup steps	$N_{\text{warmup}} = 2000$
	Number of random instances	$N_{\text{instances}} = 25$ for Figs. 6.7(a), (d), $N_{\text{instances}} = 1$ for Fig. B.3(b)
Figs. 6.7(b), (c), (e), (f)	Architecture	Dilated RNN wave function with no weight-sharing
	Number of memory units	$d_h = 20$
	Number of samples	$M = 50$
	Initial temperature	$T_0 = 1$ (for SA and VCA)
	Initial magnetic field	$\Gamma_0 = 1$ (for SQA)
	Learning rate	$\eta = 10^{-4}$
	Number of warmup steps	$N_{\text{warmup}} = 1000$
	Number of random instances	$N_{\text{instances}} = 25$
Fig. B.1	Architecture	Tensorized RNN wave function with weight sharing
	Number of memory units	$d_h = 20$
	Number of samples	$M = 50$
	Initial temperature	$T_0 = 2$
	Initial magnetic field	$\Gamma_0 = 2$
	Learning rate	$\eta = 10^{-3}$
	Number of warmup steps	$N_{\text{warmup}} = 1000$
Fig. B.2	Architecture	Tensorized cRNN wave function with no weight sharing
	Number of memory units	$d_h = 40$
	Number of samples	$N_s = 50$
	Initial driving term coupling	$\lambda_0 = 2$
	Learning rate	$\eta = 5 \times 10^{-4}$
	Number of warmup steps	$N_{\text{warmup}} = 5000$

Table B.2: Hyperparameters used to obtain the results reported in Sec. 6.3 to Sec. 6.5.

Hyperparameter	Max-Cut (Fig. 6.8)	NSP (Fig. 6.9(a))	TSP (Fig. 6.9(b))
Architecture	Dilated and Vanilla RNN	Dilated and Vanilla RNN	Dilated RNN with weight sharing
M	50 (5×10^5 after annealing)	50 (5×10^5 after annealing)	50 (10^6 after annealing)
N_{train}	5	5	5
N_{warmup}	1000	1000	2000
$N_{\text{annealing}}$	$[2^4, 2^5, ..., 2^{14}]$	$[2^4, 2^5, ..., 2^{14}]$	$[2^4, 2^5, ..., 2^{14}]$
Learning rate	1×10^{-4}	5×10^{-4}	1×10^{-3}
T_0	2.0	2.0	2.0
Number of memory units	40	40	40
Seed	111	111	111

Table B.3: A table of the hyperparameters used for the VCA experiments in Sec. 6.6.

Hyperparameter	Max-Cut (Fig. 6.8)	NSP (Fig. 6.9(a))	TSP (Fig. 6.9(b))
Metropolis move	Bit flips	Bit flips	Permutations (2-Opt) [321]
M	50	50	50
N_{eq}	5	5	5
N_{warmup}	1000	1000	2000
$N_{\text{annealing}}$	$[2^4, 2^5, ..., 2^{14}]$	$[2^4, 2^5, ..., 2^{14}]$	$[2^4, 2^5, ..., 2^{14}]$
T_0	2.0	2.0	2.0
Seed	111	111	111

Table B.4: A table of the hyperparameters used for the SA experiments in Sec. 6.6. M corresponds to the number of configurations we use during SA similar to the batch size in VCA.

Appendix C

Supplementary material of chapter 7

C.1 Hyperparameters

For all models studied in Chap. 7, we note that for each annealing step, we perform $N_{\text{train}} = 5$ gradient steps. Concerning the learning rate η, we choose $\eta = 10^{-3}$ during the warmup phase and the annealing phase and we switch to a learning rate $\eta = 10^{-4}$ in the convergence phase. We finally note that we set the number of convergence steps as $N_{\text{convergence}} = 10000$. In Tab. C.1, we provide further details about the hyperparameters we choose in our study for the different models. The meaning of each hyperparameter related to annealing is discussed in detail in Refs. [106, 108].

We use $M = 2 \times 10^6$ samples for the estimation of the entanglement entropy along with their error bars for the toric code. For the Bose-Hubbard model, we use $M = 10^7$ samples to reduce the error bars on the TEE in Fig. 7.3. To estimate the TEE using Kitaev-Preskill, we use the expression of the standard deviation of the sum of independent random variables to estimate the one standard deviation on γ_{RNN}.

Finally, we note that to avoid fine-tuning the learning rate for each value of V (between 4 and 13) in the Bose-Hubbard model, we target the normalized Hamiltonian

$$\hat{H} = -\frac{1}{V} \sum_{\langle i,j \rangle} \left(b_i^\dagger b_j + b_i b_j^\dagger \right) + \sum_{\hexagon} n_{\hexagon}^2 \tag{C.1}$$

in our experiments.

Figures	Parameter	Value
2D toric code	Number of memory units	$d_h = 60$
	Number of samples	$M = 100$
	Initial pseudo-temperature	$T_0 = 2$
	Number of annealing steps	$N_{\text{annealing}} = 4000$
Bose-Hubbard model (L = 6)	Number of memory units	$d_h = 100$
	Number of samples	$M = 500$
	Initial pseudo-temperature	$T_0 = 1$
	Number of annealing steps	$N_{\text{annealing}} = 10000$
Bose-Hubbard model (L = 8)	Number of memory units	$d_h = 100$
	Number of samples	$M = 500$
	Pseudo-temperature	$T_0 = 0$
	Number of steps	10000
Rydberg atom arrays ($L = 8$, $R_b = 1.7, 2.1$ and $\delta = 3.3$)	Number of memory units	$d_h = 60$
	Number of samples	$M = 500$
	Initial pseudo-temperature	$T_0 = 2$
	Number of annealing steps	$N_{\text{annealing}} = 4000$
Rydberg atom arrays ($L = 6$, $R_b = 1.95$)	Number of memory units	$d_h = 100$
	Number of samples	$M = 500$
	Initial pseudo-temperature	$T_0 = 2$
	Number of annealing steps	$N_{\text{annealing}} = 10000$
Rydberg atom arrays ($L = 8$, $R_b = 1.95$, pre-trained from $L = 6$)	Number of memory units	$d_h = 100$
	Number of samples	$M = 500$
	Initial pseudo-temperature	$T_0 = 0$
	Number of steps	10000

Table C.1: A summary of the hyperparameters used to obtain the results reported in Chap. 7. Note that the number of samples M corresponds to the batch size used during the training phase.

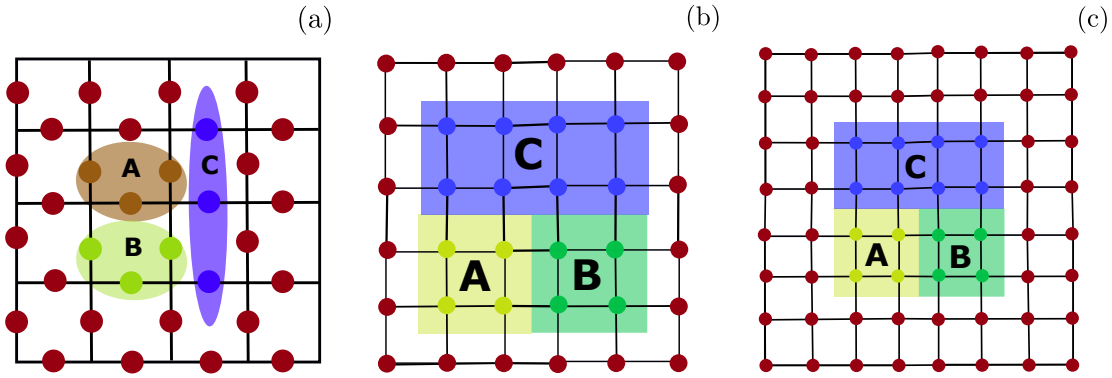

Figure C.1: An illustration of the sub-regions A, B, C chosen for the Kitaev-Preskill construction. (a) For the 2D toric code, each sub-region has 3 spins. For the hard-core Bose-Hubbard model on the Kagome lattice, we have targeted two different system sizes. Panel (b) shows the construction for $L = 6$. Panel (c) provides the construction for $L = 8$. In panels (b) and (c), each site corresponds to a block of three bosons.

C.2 Kitaev-Preskill constructions

In this appendix, we provide details about the subregions used to calculate the TEE using the Kitaev-Preskill construction (see Sec. 7.1). For the 2D toric code, we use three spins for each subregion, and for the Bose-Hubbard model in the Kagome lattice we increase the subregions sizes to mitigate finite size effects [99] as opposed to the 2D toric code that does not suffer from this limitation. The illustrations of these subregions are provided in Fig. C.1.

C.3 RNNs and MES

The results in Sec. 7.2 indicate that the RNN wave function encodes a superposition of minimally entangled states (MES). Here we further investigate this statement by analyzing the expectation values of the average Wilson loop operators and the average 't Hooft loop operators.

We define the average Wilson loop operators as

$$\hat{W}_d^z = \frac{1}{L}\left(\sum_{\mathcal{C}_d}\prod_{\sigma_j\in\mathcal{C}_d}\hat{\sigma}_j^z\right). \tag{C.2}$$

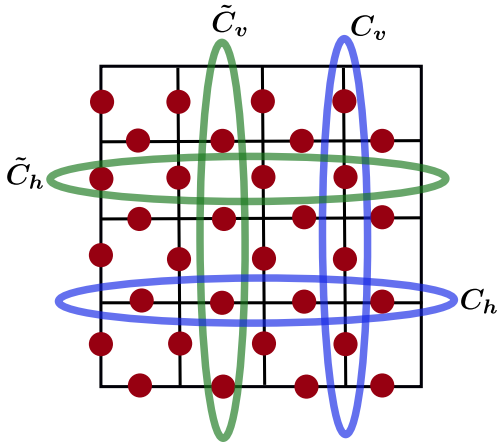

Figure C.2: An illustration of the vertical and the horizontal loops used to compute the Wilson loop operators (see Eq. C.2) and the 't Hooft loop operators (see Eq. C.3).

Here $d = h, v$ and $\mathcal{C}_v, \mathcal{C}_h$ are closed non-contractible loops illustrated in Fig. C.2. A set of degenerate ground states of the toric code are eigenstates of the operators $\hat{W}_h^z$, $\hat{W}_v^z$ with eigenvalues ± 1. Additionally, the two eigenvalues uniquely determine the topological sector of the ground state. In this case, the topological ground states can be labeled as $|\xi_{ab}\rangle$ with $a, b = 0, 1$ [270, 322].

We can also define the average 't Hooft loop operators on non-contractible closed loops [322], such that

$$\hat{W}_d^x = \frac{1}{L}\left(\sum_{\mathcal{C}_d} \prod_{\sigma_j \in \tilde{\mathcal{C}}_d} \hat{\sigma}_j^x\right), \tag{C.3}$$

where $d = h, v$ and $\tilde{\mathcal{C}}_h$ and $\tilde{\mathcal{C}}_v$ correspond to horizontal and vertical loops as illustrated in Fig. C.2. These operators satisfy the anti-commutation relations $\{\hat{W}_h^z, \hat{W}_v^x\} = 0$ and $\{\hat{W}_v^z, \hat{W}_h^x\} = 0$.

From the optimized RNN wave function ($L = 8$), we find $\langle \hat{W}_h^z \rangle = 0.0009(2)$ and $\langle \hat{W}_v^z \rangle = -0.0039(2)$ which are consistent with vanishing expectation values. We also obtain $\langle \hat{W}_h^x \rangle = 0.999846(5)$ and $\langle \hat{W}_v^x \rangle = 0.999785(5)$ for the 't Hooft loop operators, which are consistent with $+1$ expectation values. These results are in part due to the use of a positive RNN wave function which forces the expectation values $\langle \hat{W}_h^x \rangle$ and $\langle \hat{W}_v^x \rangle$ to strictly positive values and rules out the possibility to obtain, e.g., $\langle \hat{W}_h^x \rangle = -1$.

By expanding the optimized RNN wave function in the $|\xi_{ab}\rangle$ basis, where a, b are binary

variables, we obtain

$$|\Psi_{\mathrm{RNN}}\rangle \approx \sum_{ab} c_{ab} |\xi_{ab}\rangle .$$

Here $|\xi_{ab}\rangle$ correspond to the four topological sectors and they are mutually orthogonal, and c_{ab} are also real numbers. The basis states $|\xi_{ab}\rangle$ satisfy

$$\hat{W}_h^z |\xi_{ab}\rangle = (-1)^a |\xi_{ab}\rangle ,$$
$$\hat{W}_v^z |\xi_{ab}\rangle = (-1)^b |\xi_{ab}\rangle .$$

From the anti-commutation relations, we can also show that:

$$\hat{W}_h^x |\xi_{ab}\rangle = |\xi_{a\bar{b}}\rangle ,$$
$$\hat{W}_v^x |\xi_{ab}\rangle = |\xi_{\bar{a}b}\rangle ,$$

where $\bar{a} = 1 - a$ and $\bar{b} = 1 - b$. By plugging the last two equations in the $\hat{W}_h^x$ and the $\hat{W}_v^x$ expectation values of our optimized RNN wave function, we obtain:

$$2c_{00}c_{01} + 2c_{10}c_{11} \approx 1,$$
$$2c_{00}c_{10} + 2c_{01}c_{11} \approx 1.$$

From the normalization constraint $1 = \sum_{ab} c_{ab}^2$, we deduce that:

$$(c_{00} - c_{01})^2 + (c_{10} - c_{11})^2 \approx 0,$$
$$(c_{00} - c_{10})^2 + (c_{01} - c_{11})^2 \approx 0.$$

As a consequence, we conclude that $c_{00} \approx c_{01} \approx c_{10} \approx c_{11}$, which means that the optimized RNN wave function is approximately a uniform superposition of the four topological ground states $|\xi_{ab}\rangle$. This observation is also consistent with vanishing expectation values of the operators $\hat{W}_h^z$, $\hat{W}_v^z$.

Additionally, from Ref. [270] the MES of the toric code are given as follows:

$$|\Xi_1\rangle = \frac{1}{\sqrt{2}} \left(|\xi_{00}\rangle + |\xi_{01}\rangle\right) ,$$
$$|\Xi_2\rangle = \frac{1}{\sqrt{2}} \left(|\xi_{00}\rangle - |\xi_{01}\rangle\right) ,$$
$$|\Xi_3\rangle = \frac{1}{\sqrt{2}} \left(|\xi_{10}\rangle + |\xi_{11}\rangle\right) ,$$
$$|\Xi_4\rangle = \frac{1}{\sqrt{2}} \left(|\xi_{10}\rangle - |\xi_{11}\rangle\right) .$$

Thus, our RNN wave function can be written approximately as a uniform superposition of the MES $|\Xi_1\rangle$ and $|\Xi_3\rangle$, i.e.

$$|\Psi_{\mathrm{RNN}}\rangle \approx \frac{1}{\sqrt{2}} \left(|\Xi_1\rangle + |\Xi_3\rangle \right).$$

Appendix D

Supplementary material of chapter 8

In this appendix, we provide a detailed description of the different exact constructions using RNNs, TNs, RBMs, and QCBMs for the distributions outlined in Chap. 8.

D.1 RNN constructions

Construction of bimodal distribution

To construct this distribution using a vanilla RNN, it is sufficient to take $d_h = 2$. Here we define $W_n = 0$, $V_n = I_2$, $\boldsymbol{b}_n = 0$ and $f = \mathrm{Id}$ in the recursion relation (4.27). For the Softmax layer in Eq. (4.28), one can set $U_n = \alpha I_2$ and $\boldsymbol{c}_n = \mathbf{0}$. When tending α to $+\infty$, we converge to the bimodal distribution. We note that the choice of $W_n = 0$ reflects the Markovian nature of this distribution.

Construction of parity distribution

To exactly parameterize this distribution, we can use the following construction: $W_n = I_2$, $V_n = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$, $\boldsymbol{b}_n = \mathbf{0}$ and the activation $x \to f(x) = \cos^2\left(\frac{\pi}{2}(x+1)\right)$ in Eq. (4.27). The construction is made such that the second component of $\boldsymbol{h}_n$ keeps track of the parity of the sum of the previous set of bits $(\sigma_1, \sigma_2, \ldots \sigma_{n-1})$. The memory state size $d_h = 2$ is similar to the MPS construction with bond dimension $\chi = 2$. For the Softmax layer in

Eq. (4.28), we set $U_n = 0_2$ and $\boldsymbol{c}_n = \mathbf{0}$ for $n < N$. For the last site, we use $U_N = \alpha \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$ and $\boldsymbol{c}_N = \frac{\alpha}{2} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$. In the limit of $\alpha \to +\infty$, we obtain the parity distribution.

One could also think about using a vanilla RNN with multiplicative interactions, with the following recursion relation:

$$\boldsymbol{h}_n = f(A_n \boldsymbol{h}_{n-1} \odot B_n \boldsymbol{\sigma}_{n-1} + \boldsymbol{a}_n),$$

where $\odot$ is the Hadamard product and A_n, B_n, $\boldsymbol{a}_n$ are weights and biases. In this case, we take $A_n = I_2$, $B_n = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ and $\boldsymbol{a}_n = \mathbf{0}$. The activation function f is taken as the identity. For the Softmax, we use $U_n = 0_2$ for $n < N$ and $U_N = \alpha \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, while $\boldsymbol{c}_n = \mathbf{0}$ for all n. By sending α to $+\infty$, we obtain the parity distribution. We note that the initial hidden state for the multiplicative vanilla RNN is set as $\boldsymbol{h}_0 = (1, 1)^t$ instead of $\boldsymbol{h}_0 = (0, 0)^t$ for the ordinary vanilla RNN.

The simplicity of the RNN construction with a multiplicative operation compared to the first one with additive interaction highlights the importance of considering the use of multiplicative interactions in the RNN recursion relation in a generic task of interest.

Construction of cardinality distribution

For the cardinality distribution with hamming weight k, we devise the following construction $W_n = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$, $V_n = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$, $\boldsymbol{b}_n = \begin{pmatrix} 0 \\ -(k-1) \end{pmatrix}$, and the activation function as $f = \mathrm{ReLU}$ (Rectified Linear Unit). Similarly to the previous construction, we use a hidden dimension $d_h = 2$. We also use a site-independent Softmax layer in Eq. (4.28) with $U_n = \alpha \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$ and $\boldsymbol{c}_n = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, for all $1 \leq n \leq N$. By tending $\alpha \to \infty$, the RNN collapse to the cardinality distribution with weight k.

Construction of toric code distribution

For our RNN construction, we pursue the same spirit of projecting each plaquette to the $+1$ state as in the construction of the toric code ground state. Let us assume that

we work on the z-basis so that the plaquette operators B_p are diagonal. Here we use a 2DRNN as shown in Fig. D.1 where the RNN cells are placed on the vertices. Each RNN is labeled with the tuple of indices (i, j). It has two built-in recursion relations to generate the spins $\boldsymbol{\sigma}_{i,j,1}$, $\boldsymbol{\sigma}_{i,j,2}$ respectively labeled as 1 and 2 in Fig. D.1.

The first two-dimensional recursion relation is given by:

$$\boldsymbol{h}_{i,j,1} = \cos^2\left(\frac{\pi}{2}\left(\boldsymbol{\sigma}_{i,j-1,1} + \boldsymbol{\sigma}_{i,j-1,2} + \boldsymbol{\sigma}_{i+1,j-1,2}\right)\right). \tag{D.1}$$

where periodic boundary conditions on the indices are assumed. If a spin $\boldsymbol{\sigma}_{i,j,k}$ is not sampled yet during the raster scan path of the chain rule, it is initialized to $(0, 0)^t$. The sub-indices $k = 1, 2$ of the spins are clarified in Fig. D.1. The use of the $\cos^2$ activation allows us to compute the parity of the number of down spins and the number of up spins in each plaquette. If there is an odd number of spin-up, then $\boldsymbol{h}_{i,j,1} = (0, 1)^t$, and if there is an even number of spins 'up' then $\boldsymbol{h}_{i,j,1} = (1, 0)^t$. In the case where the spins are not reached yet by the autoregressive raster scan path, we obtain $\boldsymbol{h}_{i,j,1} = (0, 0)^t$. Using these values of the hidden state, we can compute the conditional probability using a Softmax layer as follows:

$$P_{\boldsymbol{\theta}}(\sigma_{i,j,1}|\sigma_{<i,j,k}) = \text{Softmax}\left(\alpha_{i,j,1}\boldsymbol{h}_{i,j,1}\right) \cdot \boldsymbol{\sigma}_{i,j,1}. \tag{D.2}$$

This means that if there is an odd number of spin 'up', we need to have a 100% probability of having $\sigma_{i,j,1}$ as a spin 'up', and 0% in the opposite scenario. This construction can be achieved by taking $\alpha_{i,j,1} \to \infty$. If less than three spins are generated in the plaquette $B_{i,j-1}$ (Fig. D.1), then we set $\alpha_{i,j,1} = 0$, such that we have an equal chance to get either a spin 'up' or a spin 'down'. We then compute

$$\boldsymbol{h}_{i,j,2} = \cos^2\left(\frac{\pi}{2}\left(\boldsymbol{\sigma}_{i-1,j,1} + \boldsymbol{\sigma}_{i-1,j,2} + \boldsymbol{\sigma}_{i-1,j+1,1}\right)\right), \tag{D.3}$$

to obtain the conditional probability

$$P_{\boldsymbol{\theta}}(\sigma_{i,j,2}|\sigma_{i,j,1}, \sigma_{<i,j,k}) = \text{Softmax}\left(\alpha_{i,j,2}\boldsymbol{h}_{i,j,2}\right) \cdot \boldsymbol{\sigma}_{i,j,2}. \tag{D.4}$$

Note also that if there are less than three spins generated in the plaquette $B_{i-1,j}$ (see Fig. D.1), then we set $\alpha_{i,j,2} = 0$ to obtain a uniform conditional probability. Otherwise, $\alpha_{i,j,2} \to +\infty$. Through the use of the two previous recursion relations, we impose $L^2 - 1$ plaquette constraint. The constraint on the plaquette $B_{L,L}$ follows from the product of all plaquettes being equal to 1.

After going through the raster-scan path and by taking the product of the conditionals, we obtain the toric code distribution construction. We note that we use a hidden dimension $d_h = 2$ in a similar fashion to the PEPS construction with bond dimension $\chi = 2$.

As a final note, we would like to point out that it is possible to build another RNN construction by focusing on the star operators while working on the x-basis in a similar fashion to the PEPS construction illustrated in Fig. D.2(c).

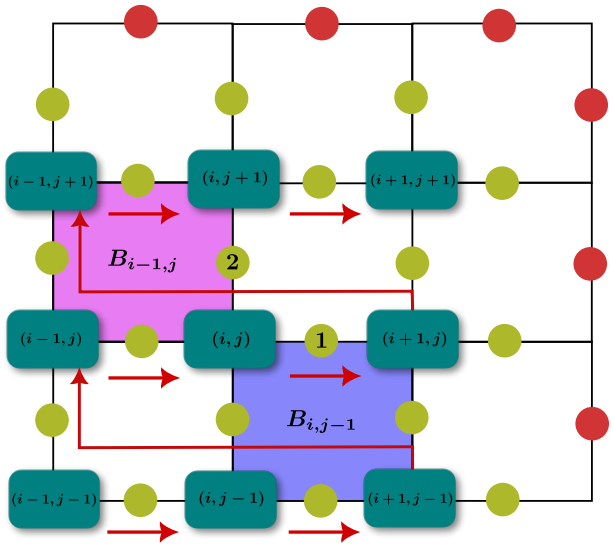

Figure D.1: **Toric code construction with the 2D RNN.** The red arrows indicate the raster-scan sampling path. The green blocks stand for the RNN cells and the yellow dots correspond to the spins of the toric code. The red dots correspond to spins replicated from the opposite sides to impose periodic boundary conditions. The plaquettes $B_{i,j}$ are labeled with the indices (i, j).

D.2 Tensor Network constructions

Construction of bimodal distribution

The bimodal distribution can be built using an MPS with bond dimension $\chi = 2$, as illustrated in Fig. D.2(a). The left tensor L and the right tensor R can be both defined as the identity matrix. Let us denote $A^{[s]}_{mn}$ an element of the tensor A where s is the physical index and m, n are the bond indices. For the bimodal distribution, it can be defined as $A^{[s]}_{mn} = \delta_{sn}\delta_{sm}$. Thus $A^{[s=0]}_{mn} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ and $A^{[s=1]}_{mn} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$. The Kronecker symbol in the tensor elements of A imposes the condition $s = m = n$ so that all the physical binary

variables across the MPS are the same. From this construction, we obtain the GHZ state and consequently the bimodal distribution through the Born rule.

Construction of parity distribution

For the evens/parity distribution, we also use a bond dimension $\chi = 2$ as shown in Fig. D.2(a). Similarly to the bimodal distribution's construction, L and R are defined as the identity. The tensor elements are defined as

$$A^{[s]}_{mn} = \begin{cases} 1; \text{ if } \mathrm{mod}(m+n+s, 2) = 0, \\ 0; \text{ otherwise.} \end{cases}$$

These assignments allow imposing that the sum of the physical variables is an even number.

Construction of cardinality distribution.

The cardinality distribution with Hamming weight k can be constructed using a bond dimension $\chi = k+1$ as illustrated in Fig. D.2(b). Note that we can assume $k \leq \frac{N}{2}$, since for $k > N-k$, we can flip each bit and apply the same construction described here. The bond dimension $\chi = k+1$ is optimal, since the half-size entanglement entropy of the cardinality state (or Dicke's state) is $\log(k+1)$, while the largest entanglement entropy that can be encoded with an MPS is $\log(\chi)$.

In our construction, we have boundary vectors. In particular, the left vector $\boldsymbol{L} = (0, \ldots 0, 1)^t$ and the right vector $\boldsymbol{R} = (1, \ldots 0, 0)^t$ have a size $k+1$. Additionally, the bulk tensors A are given by

$$A^{[\sigma]}_{\alpha\beta} = \begin{cases} 1, \text{ if } \alpha = \sigma + \beta, \\ 0, \text{ otherwise,} \end{cases}$$

such that α is the left bond index, β is the right bond index, and σ is the physical index. The main idea of this construction is to move '1' in the vector $\boldsymbol{L}$ upward each time a physical bit $\sigma = 1$. If there are exactly k ones in a bitstring $(\sigma_1, \sigma_2, \ldots, \sigma_N)$ then we will get an overlap $= 1$ with the right vector $\boldsymbol{R}$. Otherwise, the MPS will output a zero amplitude and thus a zero probability.

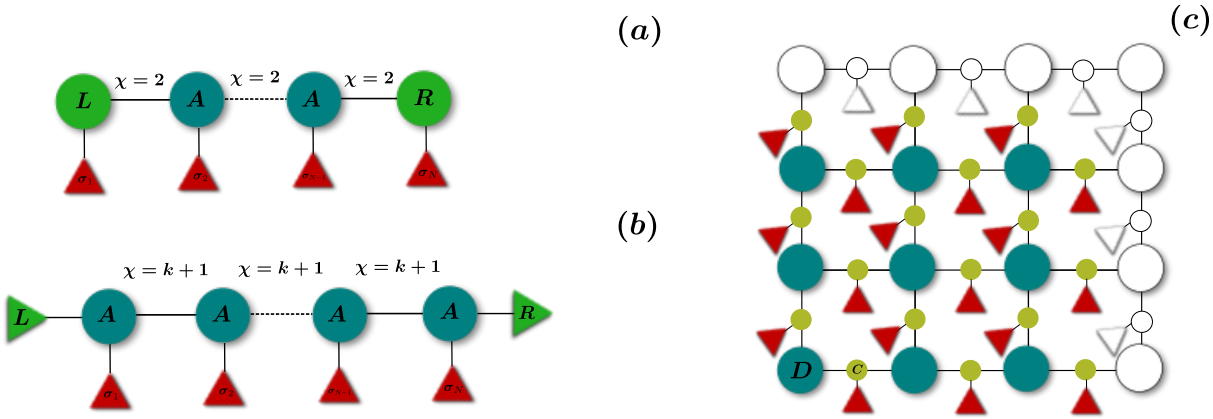

Figure D.2: **Tensor Network constructions**. The circles correspond to tensors with two, three, or four indices. The triangles correspond to a vector that matches the one-hot encoding of the local spins. Additionally, dashed lines correspond to multiple similar contraction operations that depend on the indices of the two tensors connected with a dashed line. Panel (a) shows the MPS construction of the bimodal/evens distribution. In panel (b) we illustrate the MPS construction for the cardinality distribution. Finally, in panel (c) we show the PEPS construction of the toric code. The tensors at the boundary (in black and white) are replicated from the other boundary to encode periodic boundary conditions.

Construction of toric code distribution.

To construct the toric code distribution using tensor networks, we use the PEPS architecture as shown in Ref. [323]. This construction is illustrated in Fig. D.2(c), where we use the x-basis as a computational (diagonal) basis. The yellow dots correspond to a tensor C with dimension $2 \times 2 \times 2$, such that $C^{[\sigma]}_{\alpha\beta} = \delta_{\alpha\sigma}\delta_{\beta\sigma}$ where $\sigma, \alpha, \delta = \pm 1$. This construction allows setting auxiliary indices α and β to the physical index σ. The green tensors D enforce the star operator constraints using the following construction:

$$D_{\alpha\beta\gamma\delta} = \begin{cases} 1, & \text{if } \alpha\beta\gamma\delta = 1 \\ 0, & \text{otherwise,} \end{cases}$$

where $\alpha, \beta, \gamma, \delta = \pm 1$. Since it is sufficient to satisfy the star operator constraints, then our construction allows obtaining the toric code distribution through the Born rule.

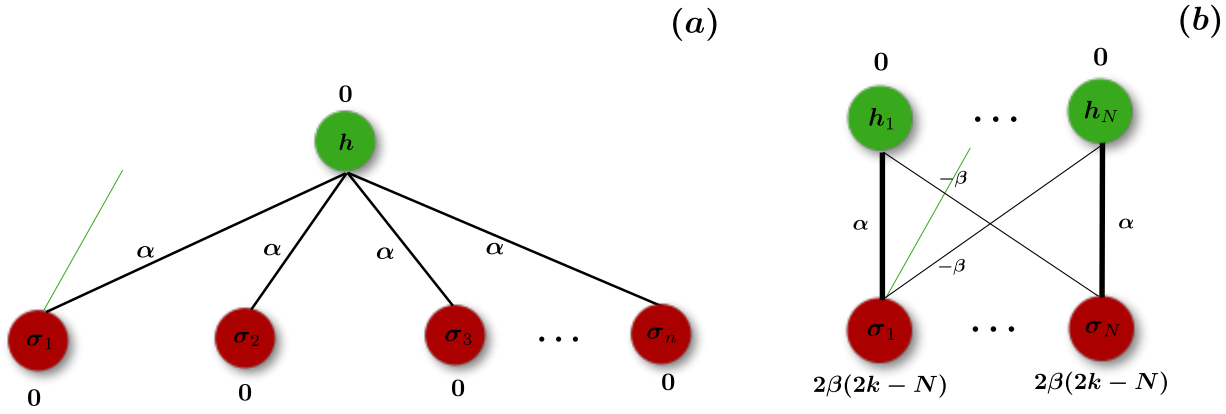

Figure D.3: **RBM exact construction** for (a) the bimodal and the evens distributions using one hidden node and without visible/hidden biases, and for (b) the cardinality distribution using N hidden nodes and with hidden biases.

D.3 RBM constructions

Construction of bimodal distribution

This distribution can be represented with an RBM with a single hidden variable and infinite couplings and with zero biases as shown in Fig. D.3(a). In this case, the probability distribution $P_{\text{RBM}}(\boldsymbol{\sigma})$ is given by:

$$P_{\text{RBM}}(\boldsymbol{\sigma}) = \frac{2\cosh\left(\sum_{i=1}^{N} W_i \sigma_i\right)}{Z}, \tag{D.5}$$

where here $\sigma_i = \pm 1$. In the limit when $W_i = \alpha$ goes to infinity, $p(\boldsymbol{\sigma})$ goes to $1/2$ when all σ_i are equal. Another way of looking at this result is from the energy perspective, where $E(\boldsymbol{\sigma}, h)$ corresponds to the lowest possible energy when all $\sigma_i = h$ for all $1 \leq i \leq N$ given a positive α. Thus our RBM probability distribution after taking the α limit is given by:

$$P_{\text{RBM}}(\boldsymbol{\sigma}) = \frac{1}{2}\left(\prod_{i=1}^{N} \delta_{\sigma_i,+1} + \prod_{i=1}^{N} \delta_{\sigma_i,-1}\right), \tag{D.6}$$

which is equivalent to Eq. (8.7) after mapping the $\pm$ variables into the 0-1 variables.

Construction of parity distribution

Similarly to the bimodal distribution, the parity distribution can be also constructed using one hidden as illustrated in Fig. D.3(a), but this time using complex-valued parameters. Here, we use the amplitude interpretation:

$$\Psi(\boldsymbol{\sigma}) = \frac{2\cosh\left(\sum_{i=1}^{N} W_i \sigma_i\right)}{\sqrt{Z}}, \tag{D.7}$$

instead of the probability interpretation in Eq. (D.5). We also assume that $\sigma_i = 0, 1$ and assumed that $h_i = -1, 1$ to obtain the cosh in Eq. (D.7). Here we take $W_i = \mathrm{i}\frac{\pi}{2}$. The use of an imaginary number allows obtaining the cos function as follows:

$$\Psi(\boldsymbol{\sigma}) = \frac{2\cos\left(\frac{\pi}{2}\sum_{i=1}^{N} \sigma_i\right)}{\sqrt{Z}}.$$

If $\sum_{i=1}^{N} \sigma_i$ is an even number then $\cos(\frac{\pi}{2}\sum_{i=1}^{N} \sigma_i) = \pm 1$, otherwise $\cos(\frac{\pi}{2}\sum_{i=1}^{N} \sigma_i) = 0$. Thus by taking the RBM distribution as the module squared of $\Psi(\boldsymbol{\sigma})$, we obtain the exact construction of the parity distribution.

Using real-valued parameters, it is not known up to date whether RBMs need an exponential number of resources in order to exactly parametrize the parity/evens distribution. It is conjectured to be the case. What is known to the best of our knowledge is that an RBM with $2^N - 1$ hidden units is a universal approximator, i.e., it can exactly parameterize all possible probability distributions including the parity distribution [324, 325].

Construction of the cardinality distribution

The cardinality distribution with system size N and with weight k can be constructed with an RBM that has N hidden units as illustrated in Fig. D.3(b).

In our construction, we take the diagonal weights as α where α is a large number. We also set the off diagonal weights to $-\beta$, the hidden biases b_i to 0, and the visible biases a_j to $2\beta(2k - N)$. By tending α to infinity, we are enforcing the constraint $\sigma_i = h_i$. In this case, we get the following RBM energy:

$$E(\boldsymbol{\sigma}) = -N\alpha + \beta \sum_{i \neq j} \sigma_i \sigma_j - 2\beta(2k - N) \sum_{i=1}^{N} \sigma_i.$$

If there are k' ones in $\boldsymbol{\sigma}$, then we obtain:

$$\begin{aligned} E(\boldsymbol{\sigma}) &= -N\alpha + \beta\big[k'(k'-1) + (N-k')(N-k'-1) \\ &\quad - 2k'(N-k')\big] - 2\beta(2k-N)(2k'-N), \\ &= -N(\alpha+\beta) + \beta\big[(2k'-N)^2 - 2(2k-N)(2k'-N)\big]. \end{aligned}$$

The latter energy has a minimum at $k'=k$. Thus, by sending β to infinity while $\beta/\alpha \to 0$, $P_{\text{RBM}}(\boldsymbol{\sigma})$ becomes uniformly peaked around the configurations with minimal energy $E(\boldsymbol{\sigma})$. In this case, $P_{\text{RBM}}(\boldsymbol{\sigma})$ is the desired cardinality distribution with weight k.

Construction of the toric code distribution

For the 2D toric code, RBMs are capable of constructing this state in addition to other topological states [303]. To prove the latter, let us take the RBM energy as:

$$E(\boldsymbol{\sigma}, \boldsymbol{h}) = -i\frac{\pi}{2}\sum_{B}\sum_{\sigma_i \in B}\sigma_i h_B,$$

where in the second sum, we go over all the plaquettes B in our system. We also assume that h_B and σ_i are binary variables. It is clear from this energy that we are using a linear number of hidden neurons with local connectivity. Starting from the energy, we compute the RBM amplitude as:

$$\Psi(\boldsymbol{\sigma}) = \frac{\Pi_B \left(2\cos(\frac{\pi}{2}\sum_{\sigma_i \in B}\sigma_i)\right)}{\sqrt{Z}}.$$

If the product of the spins in the plaquette B is $+1$, we can verify that $\cos(\frac{\pi}{2}\sum_{\sigma_i \in B}\sigma_i) = \pm 1$. In the other case, where the product of spins in the plaquette is -1, the cosine outputs 0. In the final step, the Born rule $P(\boldsymbol{\sigma}) = |\Psi(\boldsymbol{\sigma})|^2$ can be used to map the ± 1 amplitudes to $+1$ to obtain the toric code distribution.

D.4 QCBM constructions

The GHZ state [69] can be built with a QCBM using a Hardamard gate and CNOT gates as illustrated in Fig. D.4(a). For the parity distribution, we use a line topology with a series of XX gates that have an angle parameter $\frac{\pi}{2}$ as shown in Fig. D.4(b).

Furthermore, there have been multiple attempts in the literature to construct the k-Dicke's state using quantum circuits. Prominent examples include Refs. [308, 326, 327],

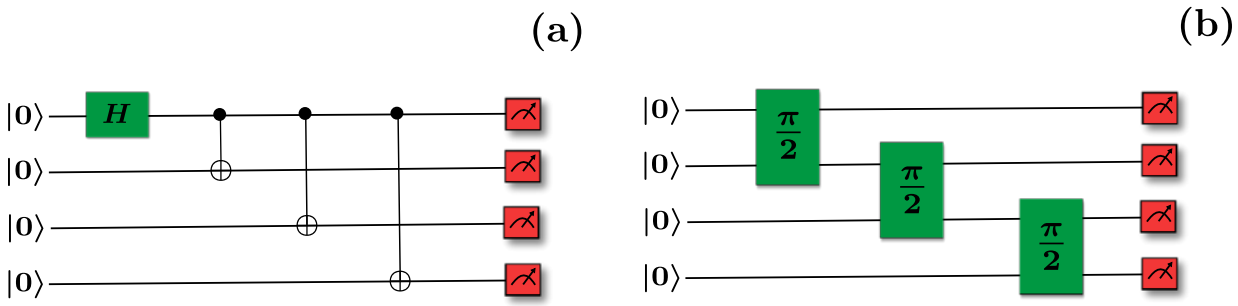

Figure D.4: **Exact QCBM constructions.** (a) An illustration of the construction of the bimodal distribution (GHZ state) using a Hadamart gate and CNOT gates. (b) An illustration of the QCBM construction of the parity distribution using XX gates with a uniform parameter $\pi/2$.

which showed that this state can be constructed with $\mathcal{O}(kN)$ gates. Additionally, the construction of the toric code on a surface instead of a torus has been shown in Ref. [258] using Hadamard and CNOT gates, with $\mathcal{O}(N)$ gates, as illustrated in Ref. [258]. In this reference, the authors showed that a quantum circuit is capable of hosting topological features such as the topological entanglement entropy.