Data-driven and Hamiltonian-driven Quantum Simulations with Generative Models

by

Isaac John Stuart De Vlugt

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Science
in
Physics

Waterloo, Ontario, Canada, 2021

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions 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

Research Presented in Chapter 2

Sec. 2.4 contains research based off of work published in Ref. [1] wherein I was a co-author. In this work, I wrote all of the content and code pertaining to training restricted Boltzmann machines, and helped the first-listed author, Stewart Morawetz, in his analysis for the main results with recurrent neural networks. Roger G. Melko and Juan Carrasquilla played an advisory/supervisory role in the entire writing process. Stewart Morawetz has given me permission to reproduce some published results for the purpose of writing this thesis.

Sec. 2.5 contains research based off of work published in Ref. [2] wherein I was the main author. In this work, I wrote all of the content and code pertaining to training restricted Boltzmann machines. Dmitri Iouchtchenko produced training datasets, and greatly helped with providing results and theory pertaining to the sign problem. I have reproduced some figures for the purpose of writing this thesis. Some of these results are included in Appendix A of this thesis. Ejaaz Merali, Roger G. Melko, and P.-N. Roy played an advisory/supervisory role in the entire writing process.

Research Presented in Chapter 3

The entirety of this chapter is dedicated to explaining content published in Ref. [3] wherein I was a co-author. In this work, I wrote the theory sections on finite temperature and ground state stochastic series expansion formalisms, performed all calculations and analyses for one-dimensional systems, and provided an original derivation for a ground state energy estimator. I also was a vital contributor to the research and development stage of this work. Ejaaz Merali, the first-listed author, is credited, among other vital contributions, with having developed a code base written in the Julia programming language that implemented the entire algorithm and analysis scripts. Given the difficulty in implementing algorithms akin to the one presented in this work, it was unanimously decided to have Ejaaz Merali be the first-listed author. Roger G. Melko played an advisory/supervisory role in the entire writing process.

Research Presented in Chapter 4

This chapter is based off of unpublished, original work of my own.

Abstract

In the era of noisy intermediate-scale quantum devices (NISQ), a paradigm shift is occurring in the numerical physics community to transition towards developing algorithms that facilitate experimental realizations of quantum many-body systems. However, traditional numerical simulations may still be required to reach system sizes beyond that of current experiments and/or to expedite testing newly-developed algorithms intended for use in experiments. In this thesis, two machine learning algorithms that have been successfully employed for quantum state reconstruction, the restricted Boltzmann machine (RBM) and the recurrent neural network (RNN), are assessed based on their ability to reconstruct ground states of experimentally-relevant quantum many-body Hamiltonians, the XY model and nanomolecular assemblies (NMAs) of endofullerenes, using synthetic data. We show that training RBMs can be unstable compared to RNNs for the given learning tasks and that RNNs learn the underlying symmetries more accurately compared RBMs.

For a closer connection to today's experiments, arrays of optically-controlled neutral atoms – Rydberg atoms – are subsequently investigated given their current prominence in experimental quantum computing. To generate synthetic Rydberg atom data for quantum state reconstruction, a Stochastic Series Expansion quantum Monte Carlo (QMC) algorithm is developed for ground states of Rydberg atoms on arbitrary lattices. By generating measurement data from this QMC algorithm across the checkerboard transition that exists on a square lattice, we compare the reconstruction accuracies of the RBM and RNN by estimating relevant ground state observables across the transition. We then propose using the RBM and RNN for estimating the second Renyi entropy, which is seemingly out of reach for the developed QMC algorithm and is highly non-trivial to obtain experimentally. There, we show that the RBM and RNN learn different aspects of the physics governing the checkerboard transition.

Acknowledgements

My time at the University of Waterloo studying physics started with a group of people that eventually ended up being my roommates throughout my undergraduate degree: Aiden Mauti, Bryn Tucker, Joshua Sheridan, Ramtin Yousefgorji, Jake Marion, Jacob Santos, and Brenden Yip. From the induced hysteria from Mastering Physics quizzes to the late nights with several extra-large pizzas to the spirited banter we had over anything and everything, they kept my sanity in check during the tough undergraduate years. For that, I want to thank them with a loud-and-proud "50."

Although he does not have direct ties to the content of this thesis, Scott Hopkins in the Department of Chemistry at the University of Waterloo was the first person to invest in me as a university student. Beyond being a top-notch physical chemist that guided me through an eight-month co-op term and my first published research paper, Scott is the salt of the earth. I learned a lot about being a researcher from him both from a scientific and *human* perspective, and I carry a lot of that with me every day.

Pierre-Nicholas Roy, or P.-N., is the next person to have taken a chance on me when he hired me for a four-month co-op term in the Theoretical Chemistry group at the University of Waterloo. Every time I interact with P.-N. he is smiling from ear to ear. He truly *loves* what he does. He has provided invaluable suggestions and guidance in my research. Like Scott, though, P.-N. was never all business. I have many fond memories of jam sessions and nerdy discussions about music gear. I want to thank P.-N. for taking a chance on me.

During my time with P.-N., I met his then PhD student Dmitri Iouchtchenko. Dmitri was always with me in the trenches helping me figure things out, and even though he was busy with his own research, he *always* found the time to lend a hand, and he still does. He goes above and beyond, works as efficiently as humanly possible, and is an invaluable source of knowledge. His influence on my research is literally everywhere.

My supervisor, Roger Melko, is next in line. I'm a little speechless when it comes to finding the words to express how glad I am to have had Roger as my supervisor. His internal compass for where to go with research is unmatched and it has benefited me immensely. He always has time to talk about research, personal, or administrative matters, and I always walk away from a conversation with Roger having fewer worries. Roger also fostered a fun research environment with his zest for turning anything into a punny acronym and his anecdotes of close encounters with dangerous wildlife. For all of those things along with his humanity, understanding, and empathy, I want to extend my utmost gratitude to Roger for being my mentor for the last few years. I also want to thank Giacomo Torlai, Bohdan Kulchytskyy, Ejaaz Merali, Roger Luo, and Sebastian Wetzel for all of the great times

and discussions we had in the last two years in Roger's group, along with my committee members, Juan Carrasquilla and Rajibul Islam, for helping me refine my work.

Dedication

I dedicate this thesis to my partner in life, best friend, and my rock, Selina Neary. She has stuck with me during my highs and lows and during the sharp turn in my career leading up to this thesis. Although I am finally taking steps in the right direction to become a better person, none of this would have been possible without her support.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Preamble

Without question, a dream of condensed matter physicists is to determine a general method for solving the famous Schrödinger equation

$$\hat{H}\,|\lambda\rangle = E\,|\lambda\rangle\,, \tag{1.1}$$

where $\hat{H}$ is a quantum many-body Hamiltonian describing the microscopic properties of how relevant degrees of freedom interact with each other and their environment, $|\lambda\rangle$ is an eigenstate of $\hat{H}$, and E is the corresponding eigenvalue (energy). The state $|\lambda\rangle$ with the lowest energy – the *ground* state, $|\lambda_0\rangle$ – is of particular interest since it governs the properties of a system at zero temperature. Unfortunately, directly solving the Schrödinger equation for the ground state is generally impossible due to the infamous "curse of dimensionality:" the required computational effort to calculate and store the ground state in memory scales exponentially with the number of degrees of freedom.

Given this exponential wall, various algorithms and tricks have been developed over decades to approximately solve Eq. (1.1) [4–9]. For the purpose of this thesis, we may cynically categorize these algorithms into two categories.

1. **Optimization**: Given an orthonormal computational basis $\{|\boldsymbol{\sigma}\rangle\}$, we make an assumption/guess that the coefficients $\lambda_0(\boldsymbol{\sigma}) \equiv \langle\boldsymbol{\sigma}|\lambda_0\rangle$ take a specific functional form $f(\boldsymbol{\sigma}, \boldsymbol{\theta}) = \big\langle\boldsymbol{\sigma}\big|f(\boldsymbol{\theta})\big\rangle$, where $\boldsymbol{\theta}$ represents a set of tunable parameters. One may then propose an *optimization* routine that, given certain theoretical criteria, determines the optimal set of parameters $\boldsymbol{\theta}^*$ such that $f(\boldsymbol{\sigma}, \boldsymbol{\theta}^*)$ best approximates $\lambda_0(\boldsymbol{\sigma})$.

2. **Markovian**: Given an instance of a configuration describing the system of interest, we devise a stochastic process, or *Markov chain*, that proposes updates to the configuration. The proposal is accepted or rejected based on the *statistical* likelihood that the configuration should be in one state or the other. The algorithm must rely on the foundations of statistical mechanics to inform the dynamics of the stochastic process (the proposals and updates) and to guarantee that we will come to a statistical equilibrium where the corresponding configurations are indicative of the ground state.

Arguably the most widely-known optimization algorithm for solving for ground states is the variational Monte Carlo (VMC) method, where the energy of a parameterized representation,

$$E(\boldsymbol{\theta}) = \left\langle f(\boldsymbol{\theta}) \middle| \hat{H} \middle| f(\boldsymbol{\theta}) \right\rangle, \tag{1.2}$$

is minimized with respect to the parameters $\boldsymbol{\theta}$ using a user-specified optimization routine. Conveniently, it is guaranteed that $E(\boldsymbol{\theta})$ is lower-bounded by the actual ground state energy E_0: $E(\boldsymbol{\theta}) \geq E_0$. Although the optimization routine itself is quite important, what has arguably been more critical to the development of VMC is the choice of representation f, such as matrix product states and the density matrix renormalization group algorithm pioneered by White [6, 7, 10], quantum circuits and the variational quantum eigensolving routine [9, 11], and machine learning algorithms like the recurrent neural network (RNN) [12]. Instead of optimizing the energy in Eq.(1.2), one can perform a data-driven optimization like quantum state reconstruction, where we aim to optimize the parameterized distribution $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = |\left\langle \boldsymbol{\sigma} \middle| f(\boldsymbol{\theta}) \right\rangle|^2$ to best approximate a probability distribution $q(\boldsymbol{\sigma}) = |\left\langle \boldsymbol{\sigma} | \lambda_0 \right\rangle|^2$. The optimization here is a minimization of a *loss function* with respect to the parameters $\boldsymbol{\theta}$, where the loss function measures the difference between $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ and $q(\boldsymbol{\sigma})$. Regardless of the choice of f and the optimization method, a degree of bias is introduced in assuming that the true ground state $\lambda_0(\boldsymbol{\sigma})$ is encapsulated in the functional form of f.

Markovian methods for simulating quantum many-body systems are undoubtedly dominated by quantum Monte Carlo (QMC) methods. Although the landscape of available QMC methods is vast [13–15], they share a common trait in that quantum systems in D spatial dimensions are mapped to a classical statistical mechanical problem in $D+1$ dimensions, where the $+1$ dimension takes the form of imaginary time. The successful application of a QMC algorithm on a particular quantum many-body Hamiltonian is reliant on the absence of the infamous *sign problem* [16–18]. When free of the sign problem, the stochastic process in charge of updating $D+1$ QMC configurations varies drastically from

one QMC flavour to another, but it is precisely developments in efficient update schemes that anchors the use cases of the given QMC flavour.

Of course, not all algorithms fall perfectly into one of the two categories mentioned. A notable exception is the restricted Boltzmann machine (RBM), which is a neural network that assumes its own representational form $f(\boldsymbol{\sigma}, \boldsymbol{\theta})$. However, to sample its parameterized probability distribution $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ requires a Markov chain process called *Gibbs sampling*. More on this can be found in Sec. 2.1.

Lastly, one may wonder why we as physicists are trying to *simulate* quantum many-body systems instead of *physically realizing* them in a laboratory setting. Surely, if we had access to real quantum degrees of freedom that interacted with each other and their environment as prescribed by the Hamiltonian $\hat{H}$ of interest, devising complex and non-trivial algorithms, be they Markovian in nature or representationally based, would be wasted effort. Today, we find ourselves at this inflection point, where experiments are accelerating at a staggering pace [19–27], yet numerical simulations are still required to either verify an experiment's accuracy or to reach system sizes that are not yet possible with today's hardware. Such real-world devices that comprise of $\mathcal{O}(10^2)$ quantum degrees of freedom but are not yet stable nor advanced enough to reap the large-scale benefits of quantum computing are called *noisy intermediate-scale quantum* (NISQ) devices [28].

In tandem with the NISQ era revolution, machine learning algorithms have been a welcome addition to the computational physicist's list of tools [29]. In particular, unsupervised learning algorithms called *generative models* have been employed to reconstruct and verify quantum states from measurement data [1, 2, 30–37]. These machine learning algorithms also generally outperform traditionally-used quantum state reconstruction techniques, such as the maximum-likelihood technique [38]. A focus for this thesis will be to empirically determine which commonly-used generative models, namely the RBM and RNN, best reconstruct a quantum state of interest with the emphasis being placed on quantum states that are, or can foreseeably be, used as a platform for quantum hardware.

1.2 Quantum state reconstruction

The goal of quantum state reconstruction is to devise a protocol for which an unknown quantum state $\hat{\rho}$ can be reconstructed based on a limited set of accessible information that can be extracted from it, like measurement data. For example, consider a state that describes N spin-1/2 particles (qubits). A matrix representation of $\hat{\rho}$ comprises of $2^N \times 2^N$ complex numbers. The well-known Pauli operators $\hat{\sigma}^{(1)} \equiv \hat{\sigma}^x$, $\hat{\sigma}^{(2)} \equiv \hat{\sigma}^y$, $\hat{\sigma}^{(3)} \equiv \hat{\sigma}^z$, and

the identity operator $\hat{\sigma}^{(0)} \equiv \mathbb{I}$ form a complete basis for $2^N \times 2^N$ matrices, so we may write

$$\hat{\rho} = \frac{1}{2^N} \sum_{i_1, i_2, \cdots, i_N = 0}^{3} c_{i_1, i_2, \cdots, i_N} \hat{\sigma}^{(i_1)} \otimes \hat{\sigma}^{(i_2)} \otimes \cdots \otimes \hat{\sigma}^{(i_N)} \tag{1.3a}$$

where the coefficients $c_{i_1, i_2, \cdots, i_N} \in \mathbb{R}$ are traditional expectation values,

$$c_{i_1, i_2, \cdots, i_N} = \mathrm{Tr}\Big(\hat{\sigma}^{(i_1)} \otimes \hat{\sigma}^{(i_2)} \otimes \cdots \otimes \hat{\sigma}^{(i_N)} \hat{\rho} \Big). \tag{1.3b}$$

Even though $\hat{\rho}$ is unknown, the 4^N coefficients $c_{i_1, i_2, \cdots, i_N}$ can be obtained through *statistical inference*.[1] Consider the eigenbasis of the $\hat{\sigma}^z$ operator defined by the eigenequations

$$\hat{\sigma}^z \left| 0 \right\rangle = \left| 0 \right\rangle \tag{1.4a}$$

and

$$\hat{\sigma}^z \left| 1 \right\rangle = - \left| 1 \right\rangle . \tag{1.4b}$$

One can show that the set of single-qubit projective measurements

$$\mathrm{Tr}\Big(\hat{\mu}^{(i_1)} \otimes \hat{\mu}^{(i_2)} \otimes \cdots \otimes \hat{\mu}^{(i_N)} \hat{\rho} \Big)$$

defined by the operators

$$\hat{\mu}^{(0)} = \left| 0 \right\rangle\!\left\langle 0 \right| , \tag{1.5a}$$

$$\hat{\mu}^{(1)} = \left| 1 \right\rangle\!\left\langle 1 \right| , \tag{1.5b}$$

$$\hat{\mu}^{(2)} = \frac{1}{\sqrt{2}} \Big(\left| 0 \right\rangle + \left| 1 \right\rangle \Big) \frac{1}{\sqrt{2}} \Big(\left\langle 0 \right| + \left\langle 1 \right| \Big) \equiv \left| + \right\rangle\!\left\langle + \right| , \tag{1.5c}$$

and

$$\hat{\mu}^{(3)} = \frac{1}{\sqrt{2}} \Big(\left| 0 \right\rangle - i \left| 1 \right\rangle \Big) \frac{1}{\sqrt{2}} \Big(\left\langle 0 \right| + i \left\langle 1 \right| \Big) \equiv \left| - i \right\rangle\!\left\langle - i \right| , \tag{1.5d}$$

suffice to statistically infer the coefficients $c_{i_1, i_2, \cdots, i_N}$ [39].[2] In performing measurements in all of the 4^N possible bases (all possible combinations of the four operators $\hat{\mu}^{(1-4)}$ on

[1] The coefficient $c_{0,0,\cdots,0}$ must be equal to one to ensure that $\hat{\rho}$ is trace-one. So, really, there are $4^N - 1$ coefficients to determine.

[2] This set of projective measurements to infer the coefficients $c_{i_1, i_2, \cdots, i_N}$ is not unique.

N degrees of freedom) many times to obtain statistical inference estimates on the coefficients $c_{i_1,i_2,\cdots,i_N}$, one can therefore reconstruct $\hat{\rho}$ up to the accuracy of the inference. This naive protocol is unfeasible, of course. A more elegant protocol is the *maximum likelihood* technique, where one takes into account that a general state $\hat{\rho}$ is positive semi-definite, hermitian, and trace-one [39]. However, it turns out that exponential bottlenecks persist. Extremely non-trivial reconstruction protocols exist that lighten the exponential load of maximum likelihood and statistical inference techniques [40–43], but it is clear that for general quantum many-body ground states, more flexible and generalizable protocols are desirable.

As was mentioned in Sec. 1.1, ground states of Hamiltonians are of particular interest to physicists. When the state $\hat{\rho}$ is pure, it takes a simplified form

$$\hat{\rho}_{\text{pure}} = |\psi\rangle\langle\psi| \,, \tag{1.6a}$$

where $|\psi\rangle$ can be written as

$$|\psi\rangle = \sum_{\{\boldsymbol{\sigma}\}} c(\boldsymbol{\sigma}) \, |\boldsymbol{\sigma}\rangle \,, \tag{1.6b}$$

where $\{|\boldsymbol{\sigma}\rangle\}$ is an orthonormal basis with cardinality $|\{|\boldsymbol{\sigma}\rangle\}| = \mathcal{D}^N$, the coefficients $c(\boldsymbol{\sigma})$ are generally complex numbers, and $\mathcal{D}$ is the local Hilbert space size of each of the N degrees of freedom.[3] A non-negligible number of relevant Hamiltonians' ground states conveniently fall under the provisions of the well-known Perron-Frobenius theorem [44], which guarantees that a Hamiltonian's ground state can be taken to have strictly *positive* coefficients, modulo an arbitrary global phase, if its off-diagonal matrix elements are negative. Hamiltonian matrices that satisfy this theorem are referred to as *stoquastic* [45].

Assuming that the Hamiltonian of interest is stoquastic, we may write its ground state as

$$|\lambda_0\rangle = \sum_{\{\boldsymbol{\sigma}\}} c(\boldsymbol{\sigma}) \, |\boldsymbol{\sigma}\rangle \,, \qquad c(\boldsymbol{\sigma}) \geq 0. \tag{1.7}$$

This property of ground states greatly facilitates reconstruction protocols in that the measurement data required to reconstruct the coefficients $c(\boldsymbol{\sigma})$ only relies on projective measurements in the $\{|\boldsymbol{\sigma}\rangle\}$ basis. To summarize, instead of an exponential number of measurement bases in the case of a general state $\hat{\rho}$, for ground states of stoquastic Hamiltonians we only require *one* measurement basis. For the remainder of this thesis, we will strictly deal with pure states with positive coefficients. As such, the satisfaction of the Perron-Frobenius theorem will be shown when necessary.

[3] For example, for a spin-S system, $\mathcal{D} = 2S + 1$.

The question remains of how many measurements one requires to reconstruct the target ground state up to a certain desired accuracy. Not only this, but for experimentally-prepared states, one might also be limited to a certain maximum number of measurements as determined by the experiment itself. As such, devising a protocol that can handle smaller measurement datasets is preferred. Machine learning algorithms, namely generative models, have been proven to live up to this task with relatively limited resources [2, 3, 32, 34, 36, 37, 46]. In essence, a generative model describes a parameterized representation of a probability distribution $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$. We may *train* the model's parameters to best approximate a target probability distribution $q(\boldsymbol{\sigma})$ by minimizing a loss function that in some way quantifies the difference between both distributions given a dataset $\mathbf{D}$ sampled from $q(\boldsymbol{\sigma})$. In the context of quantum state reconstruction for ground states with positive coefficients,

$$\lambda_0(\boldsymbol{\sigma}) = \langle \boldsymbol{\sigma} | \lambda_0 \rangle = \sqrt{q(\boldsymbol{\sigma})} \tag{1.8a}$$

and

$$\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = f(\boldsymbol{\sigma}, \boldsymbol{\theta}) = \big\langle \boldsymbol{\sigma} \big| f(\boldsymbol{\theta}) \big\rangle = \sqrt{p_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}. \tag{1.8b}$$

The Universal Approximation theorem for neural networks provides some degree of certainty that, given a large enough number of parameters $\boldsymbol{\theta}$ and dataset $\mathbf{D}$, any generative model *should* be able to learn an underlying target distribution $q(\boldsymbol{\sigma})$. That being said, not all generative models are created equally, and the question of *expressivity* remains: do some generative models require fewer parameters, and therefore less computational effort, to obtain a better reconstruction than other generative models? Not only this, but are the optimal parameters $\boldsymbol{\theta}^*$ that yield the best reconstruction feasibly obtainable? In other words, are some generative models easier *to train* than others? Undoubtedly, answering these questions will to an extent rely on empirical case studies where different generative models can be put to the same test. Answers to these questions are proceeding in earnest [1, 2, 46]. In Ch. 2 of this thesis, two widely-used generative models called the restricted Boltzmann machine (RBM) and recurrent neural network (RNN) are outlined in detail and applied to two different reconstruction tasks. In this way, light is shed on what can be expected from RBMs and RNNs from a *trainability* and *expressivity* perspective when reconstructing ground states of many-body Hamiltonians.

1.3 Monte Carlo fundamentals

Quantum or classical in nature, the goal of Markovian Monte Carlo algorithms is to develop a stochastic process, or Markov chain, for which a normalization constant

$$Z = \sum_{\mu} W(\mu) \tag{1.9}$$

can be *importance sampled* for the purpose of estimating certain quantities of interest in an *unbiased* fashion, where μ represents a label for an abstract configuration and $W(\mu)$ is a real-valued weight. To interpret each $W(\mu)$ as unnormalized probabilities, we require that $W(\mu) \geq 0$, otherwise the infamous *sign problem* halts our progress in its footsteps. Typically, the sum over all possible configurations μ is out of reach for modern computers. Therefore, an efficient sampling routine[4] is required to estimate the average value of quantities as

$$\langle O \rangle = \frac{1}{Z} \sum_{\mu} O(\mu) W(\mu) \approx \frac{1}{N_{\mathbf{D}}} \sum_{\mu \in \mathbf{D}} O(\mu), \tag{1.10}$$

where $\mathbf{D}$ represents a set of instances of the configuration space generated according to $W(\mu)$ and $N_{\mathbf{D}} = |\mathbf{D}|$. This sum over $\mu \in \mathbf{D}$ should circumvent the sum over all possible configurations, and so long as the configurations $\mu \in \mathbf{D}$ are i.i.d. (identically and independently distributed), the Central Limit theorem provides statistical guarantees that, if $N_{\mathbf{D}}$ is large enough, we are indeed providing an accurate estimate of the exact expectation value.

The Markov chain is the mechanism for which a given initial/trial configuration μ is updated according to the unnormalized distribution $W(\mu)$. Crucially, we rely on the Markov chain to be able to feasibly explore the entire configuration space. Not only this, but to generate i.i.d. samples according to $W(\mu)$, the Markov chain must be able to reach a new configuration ν that is statistically uncorrelated from μ in a reasonable number of updates. This property is called *ergodicity* [47]. The mathematics underlying an ergodic Markov chain is the condition of balance,

$$\sum_{\nu} W(\mu) T(\mu \to \nu) = \sum_{\nu} W(\nu) T(\nu \to \mu), \tag{1.11}$$

where $T(\mu \to \nu)$ represents the transition probability that the "current" configuration μ is updated to ν. An obvious solution to this equation is

$$W(\mu) T(\mu \to \nu) = W(\nu) T(\nu \to \mu), \tag{1.12}$$

[4]The procedure for which these samples are generated is what we mean by "importance sampling."

which is referred to as the *detailed balance* condition [47]. That being said, just because our algorithm satisfies the detailed balance condition doesn't guarantee that our algorithm is ergodic. The determination of whether an algorithm is ergodic or not is somewhat subjective. If detailed balance is satisfied, it could still be that the number of updates that our Markov chain requires to generate an uncorrelated configuration – the *correlation time* – is considered too long. If the correlation time is too long, this requires the user to rethink how the transition probabilities T can be modified to lower it.

Typically, it is helpful to break up the transition probabilities into two parts: a *selection* probability g and an *acceptance* probability A,

$$T(\mu \to \nu) = g(\mu \to \nu)A(\mu \to \nu). \tag{1.13}$$

The selection probability $g(\mu \to \nu)$ is the probability for which the algorithm generates a new configuration ν given the current configuration μ. The acceptance probability $A(\mu \to \nu)$ governs whether or not our algorithm should accept the newly-generated configuration ν. Clearly, fine-tuning both g and A will have drastic effects on the correlation time. Consider plugging in Eq. (1.13) into Eq. (1.12),

$$\frac{A(\mu \to \nu)}{A(\nu \to \mu)} = \frac{W(\nu)g(\nu \to \mu)}{W(\mu)g(\mu \to \nu)}, \tag{1.14}$$

where the ratio on the left-hand side is called the *acceptance ratio*. A famous choice for the acceptance probability $A(\mu \to \nu)$ was made by Nicholas Metropolis [48],

$$A(\mu \to \nu) = \min\left(1, \frac{W(\nu)g(\nu \to \mu)}{W(\mu)g(\mu \to \nu)}\right), \tag{1.15}$$

where this choice is made in order to maximize the acceptance ratio in Eq. (1.14).[5] To maximize Eq. (1.14), we may set the larger of $A(\mu \to \nu)$ and $A(\nu \to \mu)$ to 1 and then adjust the complementary acceptance probability to satisfy Eq. (1.14).

1.4 Best of both worlds

Optimization and Markovian methods both have their pros and cons for solving Eq. (1.1). Optimization methods are quite nice since the result is a compact representation of the

[5] Maximizing the acceptance ratio is a good thing! We explore the configuration space "faster" and more efficiently in this way.

ground state wavefunction from which general observables can be calculated with ease. Not only this, but often the wavefunction representation $f(\boldsymbol{\sigma}, \boldsymbol{\theta})$ can be sampled autoregressively; samples $\boldsymbol{\sigma}$ comprising of N degrees of freedom can be drawn for each degree of freedom conditioned on the value of the "previous" degrees of freedom.[6] Samples are therefore inherently i.i.d. for this subclass of wavefunction representations. On the other hand, unwanted bias was introduced into the algorithm in the assumption that the true ground state takes the form of the user-chosen representation. Markov chain Monte Carlo (MCMC) methods improve on this by giving unbiased statistical estimates. However, to generate i.i.d. samples to estimate quantities in an unbiased fashion we must get through correlation times. Not only this, but estimating some observables can be non-trivial. Specifically, the exact form of an observable O in the language of the stochastic process's configuration space may not be obvious.

Correlation times and biased representations are necessarily unavoidable. However, some of the cons of one algorithm might be (somewhat) saved by another. For example, in terms of not being able to evaluate observables in a particular Monte Carlo algorithm, if i.i.d. samples can be generated and used to train a neural network in a quantum state reconstruction context, a wavefunction representation of the Monte Carlo algorithm is therefore available. Having a wavefunction representation then allows for the computation of general observables (see Sec. 2.3) that would otherwise be impossible in a Monte Carlo algorithm. This will also be a central theme in this thesis.

1.5 Thesis outline

Ch. 2 begins with outlining the architectures of two commonly-used generative models in physics: the restricted Boltzmann machine (RBM) and the recurrent neural network (RNN). These two generative models will then be tasked with two different quantum state reconstruction tasks that are ground states of interesting quantum many-body Hamiltonians: the XY model and nanomolecular assemblies of endofullerenes. The goal of this chapter is to inform us of which generative model might give a better reconstruction quality for reconstructing quantum many-body ground states.

Ch. 3 showcases a quantum Monte Carlo (QMC) method developed for Rydberg atoms with the eventual goal in mind to use it for generating datasets for training RBMs and RNNs. It turns out that a very important quantity for physicists to quantify, the entanglement entropy, is out of reach for this QMC method, justifying training generative models

[6] One assumes a "time order" for this sequence of conditional probabilities.

on a correspondingly QMC-generated dataset. This reconstruction task is demonstrated in Ch. 4, where the necessary background on the observable of interest is first demonstrated. Finally, concluding remarks and future considerations are given in Ch. 5.

Chapter 2

Generative models for quantum state reconstruction

2.1 Restricted Boltzmann machines

The restricted Boltzmann machine (RBM) is a widely-used unsupervised machine learning algorithm that has been detailed in many other works that span throughout applications in the computer science community [49–52], and in the physics community, most notably for quantum state reconstruction [1, 2, 30, 31, 33, 34, 36, 37, 46]. For self-consistency, a view of the theoretical landscape of restricted Boltzmann machines is offered in this section of this thesis.

RBMs are stochastic neural networks with a *visible* layer $\boldsymbol{\sigma}$ and a *hidden* layer $\boldsymbol{h}$. For a physical system comprising of N degrees of freedom, $n_v = N$ defines the length of the visible layer (also referred to as the *number of visible units* in literature): $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \cdots, \sigma_{n_v})$ with $\sigma_i \in [0, 1, \cdots, \mathcal{D} - 1]$, where $\mathcal{D}$ is the dimension local Hilbert space of each degree of freedom. For numerical implementation, a one-hot encoding must be performed on each visible unit if $\mathcal{D} > 2$; each $\sigma_i = d \in [0, 1, \cdots, \mathcal{D} - 1]$ is instead represented by a vector of length $\mathcal{D}$ whose d^{th} entry is 1. Therefore, one may think of the visible layer as a rank-2 tensor, $\boldsymbol{\sigma} = \sigma_{id}$, with dimensions $n_v \times \mathcal{D}$. In the $\mathcal{D} = 2$ (binary) case, a one-hot encoding is unnecessary and $\boldsymbol{\sigma}$ is simply a vector of length n_v. The hidden layer, $\boldsymbol{h} = (h_1, h_2, \cdots, h_{n_h})$, acts as a feature detector for the visible layer, where the number of hidden units n_h is a user-specified hyperparameter. It is most common to use binary hidden units $h_i \in [0, 1]$, which is what is employed throughout this thesis.

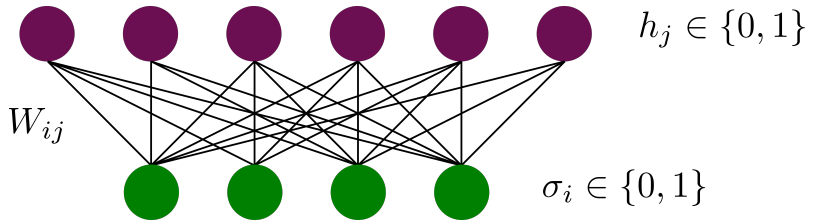

Figure 2.1: A graphical representation of the RBM architecture for $\mathcal{D} = 2$, $n_v = 4$, and $n_h = 6$. Each visible unit σ_i is connected to each hidden unit h_j by a matrix of weights W_{ij}. The external bias fields $\boldsymbol{c}$ and $\boldsymbol{b}$ are omitted from this diagram.

The RBM contains tunable parameters $\boldsymbol{\theta} = (\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{c})$, where $\boldsymbol{W}$ is a weight tensor connecting each degree of freedom in the visible layer to each degree of freedom in the hidden layer, and $\boldsymbol{b}$ and $\boldsymbol{c}$ are external bias fields for the visible and hidden layer, respectively. These parameters have the following size/dimensions:

- $\boldsymbol{W}$: If $\mathcal{D} = 2$, $\boldsymbol{W}$ is a $n_v \times n_h$ matrix. If $\mathcal{D} > 2$, $\boldsymbol{W}$ is a rank-3 tensor with entries $W_{ijd} \in \mathbb{R}$, where $i \in [1, 2, \cdots, n_v]$, $j \in [1, 2, \cdots, n_h]$, and $d \in [1, 2, \cdots, \mathcal{D}]$.
- $\boldsymbol{b}$: If $\mathcal{D} = 2$, $\boldsymbol{b}$ is a vector of length n_v with entries $b_i \in \mathbb{R}$. If $\mathcal{D} > 2$, $\boldsymbol{b}$ is a rank-2 tensor with entries $b_{id} \in \mathbb{R}$.
- $\boldsymbol{c}$: Since the hidden layer will only comprise of binary units in this thesis, $\boldsymbol{c}$ is always a vector of length n_h with entries $c_j \in \mathbb{R}$.

This architecture is depicted in Fig. 2.1 in the case of $\mathcal{D} = 2$.

The probability distribution intrinsic to the RBM's architecture is

$$p_{\boldsymbol{\theta}}(\boldsymbol{\sigma}, \boldsymbol{h}) = \frac{e^{-E_{\boldsymbol{\theta}}(\boldsymbol{\sigma}, \boldsymbol{h})}}{Z_{\boldsymbol{\theta}}}, \tag{2.1}$$

where, using conventional Einstein summation notation,

$$E_{\boldsymbol{\theta}}(\boldsymbol{\sigma}, \boldsymbol{h}) = \begin{cases} -W_{ij} h_j \sigma_i - \sigma_i b_i - h_j c_j & \mathcal{D} = 2 \\ -W_{ijd} h_j \sigma_{id} - \sigma_{id} b_{id} - h_j c_j & \mathcal{D} > 2 \end{cases}, \tag{2.2}$$

and the normalization is

$$Z_{\boldsymbol{\theta}} = \sum_{\boldsymbol{\sigma}, \boldsymbol{h}} e^{-E_{\boldsymbol{\theta}}(\boldsymbol{\sigma}, \boldsymbol{h})}. \tag{2.3}$$

Since the hidden layer $\boldsymbol{h}$ acts as a feature detector between the visible units, the marginalized distribution $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ is more physically relevant. In tracing out $\boldsymbol{h}$ in Eq. (2.1), one obtains

$$p_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = \sum_{\boldsymbol{h}} p_{\boldsymbol{\theta}}(\boldsymbol{\sigma}, \boldsymbol{h}) = \frac{e^{-\mathcal{E}_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}}{Z_{\boldsymbol{\theta}}}, \tag{2.4}$$

where

$$\mathcal{E}_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = \begin{cases} -b_i\sigma_i - \sum_{j=1}^{n_\mathrm{h}} \ln\Big(1 + \exp\left(c_j + W_{ij}\sigma_i\right)\Big) & \mathcal{D} = 2 \\ -b_{id}\sigma_{id} - \sum_{j=1}^{n_\mathrm{h}} \ln\Big(1 + \exp\left(c_j + W_{ijd}\sigma_{id}\right)\Big) & \mathcal{D} > 2 \end{cases}, \tag{2.5}$$

and

$$Z_{\boldsymbol{\theta}} = \sum_{\boldsymbol{\sigma}} e^{-\mathcal{E}_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}. \tag{2.6}$$

To tune the parameters $\boldsymbol{\theta}$ in such a way as to demand that $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ best approximates a target distribution $q(\boldsymbol{\sigma})$, one may minimize the Kullback-Leibler (KL) divergence between $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ and $q(\boldsymbol{\sigma})$,

$$\mathrm{KL}_{\boldsymbol{\theta}} = \sum_{\boldsymbol{\sigma}} q(\boldsymbol{\sigma}) \ln\left(\frac{q(\boldsymbol{\sigma})}{p_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}\right), \tag{2.7}$$

with respect to the parameters $\boldsymbol{\theta}$. In most cases, the target distribution is unknown and one only has access to a dataset $\mathbf{D}$ comprising of samples from $q(\boldsymbol{\sigma})$. Therefore, using the following approximations,

$$q(\boldsymbol{\sigma}) \approx \frac{1}{|\mathbf{D}|} \sum_{\boldsymbol{\sigma}' \in \mathbf{D}} \delta(\boldsymbol{\sigma} - \boldsymbol{\sigma}'), \tag{2.8a}$$

and

$$p_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) \approx \frac{1}{|\boldsymbol{\Gamma}|} \sum_{\boldsymbol{\sigma}' \in \boldsymbol{\Gamma}} \delta(\boldsymbol{\sigma} - \boldsymbol{\sigma}'), \tag{2.8b}$$

the gradient of the KL divergence with respect to the RBM parameters $\boldsymbol{\theta}$ is

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} \mathrm{KL}_{\boldsymbol{\theta}} &\approx \frac{1}{|\mathbf{D}|} \sum_{\boldsymbol{\sigma} \in \mathcal{D}} \nabla_{\boldsymbol{\theta}} \mathcal{E}_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) - \frac{1}{|\boldsymbol{\Gamma}|} \sum_{\boldsymbol{\sigma} \in \boldsymbol{\Gamma}} \nabla_{\boldsymbol{\theta}} \mathcal{E}_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) \\ &= \left\langle \nabla_{\boldsymbol{\theta}} \mathcal{E}_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) \right\rangle_{\mathbf{D}} - \left\langle \nabla_{\boldsymbol{\theta}} \mathcal{E}_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) \right\rangle_{\boldsymbol{\Gamma}}. \end{aligned} \tag{2.9}$$

For stochastic gradient descent optimization methods, $\mathbf{D}$ is randomly divided into *mini batches* $\mathbf{P}$, where $|\mathbf{P}|$ and $|\boldsymbol{\Gamma}|$ are hyperparameters called the *positive* and *negative* batch

sizes, respectively. Here, $\mathbf{\Gamma}$ denotes a dataset generated from sampling $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$. Typically, $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ is sampled by performing k steps of a Markov chain method called contrastive divergence (CD_k) block-Gibbs sampling [53]. The CD_k block-Gibbs sampling procedure is as follows.

1. Initialize the Markov chain with an initial configuration $\boldsymbol{\sigma}^{(0)}$ from the target's dataset $\mathbf{D}$.

2. Sample a hidden layer configuration $\boldsymbol{h}^{(0)}$ given $\boldsymbol{\sigma}^{(0)}$ using

$$p_{\boldsymbol{\theta}}(h_j = 1|\boldsymbol{\sigma}) = \begin{cases} \text{sigmoid}\left(c_j + W_{ij}\sigma_i\right) & \mathcal{D} = 2 \\ \text{sigmoid}\left(c_j + W_{ijd}\sigma_{id}\right) & \mathcal{D} > 2 \end{cases}, \tag{2.10}$$

   where $\text{sigmoid}(\cdots)$ denotes the sigmoid function.

3. Given $\boldsymbol{h}^{(0)}$, sample a new visible layer $\boldsymbol{\sigma}^{(1)}$ via

$$p_{\boldsymbol{\theta}}(\sigma_i = 1|\boldsymbol{h}) = \text{sigmoid}\left(b_i + W_{ij}h_j\right) \qquad \mathcal{D} = 2, \tag{2.11a}$$

   or

$$p_{\boldsymbol{\theta}}(\sigma_{id} = 1|\boldsymbol{h}) = S\left(b_{id} + W_{ijd}h_j\right) \qquad \mathcal{D} > 2, \tag{2.11b}$$

   where $S(\cdots)$ denotes the softmax function.

4. Iterate this procedure k times in total.

Finally, the mathematical form of $\nabla_\theta \mathcal{E}_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = (\frac{\partial}{\partial \mathbf{W}}, \frac{\partial}{\partial \mathbf{c}}, \frac{\partial}{\partial \mathbf{b}})\mathcal{E}_{\boldsymbol{\theta}}$ is

$$\nabla_\theta \mathcal{E}_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = \begin{cases} \frac{\partial \mathcal{E}_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}{\partial W_{ij}} = -p_{\boldsymbol{\theta}}\left(h_j = 1|\boldsymbol{\sigma}\right)\sigma_i \\ \frac{\partial \mathcal{E}_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}{\partial c_j} = -p_{\boldsymbol{\theta}}\left(h_j = 1|\boldsymbol{\sigma}\right) \qquad \mathcal{D} = 2 \\ \frac{\partial \mathcal{E}_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}{\partial b_i} = -\sigma_i \\ \\ \frac{\partial \mathcal{E}_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}{\partial W_{ijd}} = -p_{\boldsymbol{\theta}}\left(h_j = 1|\boldsymbol{\sigma}\right)\sigma_{id} \\ \frac{\partial \mathcal{E}_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}{\partial c_j} = -p_{\boldsymbol{\theta}}\left(h_j = 1|\boldsymbol{\sigma}\right) \qquad \mathcal{D} > 2 \\ \frac{\partial \mathcal{E}_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}{\partial b_{id}} = -\sigma_{id} \end{cases}. \tag{2.12}$$

2.2 Recurrent neural networks

The recurrent neural network (RNN) defines an autoregressive model whose intrinsic probability distribution $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ is obtained via

$$p_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = p_{\boldsymbol{\theta}}(\sigma_1)p_{\boldsymbol{\theta}}(\sigma_2|\sigma_1)\cdots p_{\boldsymbol{\theta}}(\sigma_N|\sigma_{N-1},\cdots\sigma_1), \tag{2.13}$$

where $\boldsymbol{\theta}$ are the tunable parameters within the RNN architecture, and N is the length of the input vector $\boldsymbol{\sigma}$ which coincides with a physical system's degrees of freedom. As with the RBM's visible layer, each element of $\boldsymbol{\sigma}$ has a local Hilbert space dimension $\mathcal{D}$, and a one-hot encoding must be performed for numerical purposes.[1] Therefore, $\boldsymbol{\sigma} = \sigma_{id}$ has dimensions $N \times \mathcal{D}$.

The elementary block of the RNN architecture that calculates each $p_{\boldsymbol{\theta}}(\sigma_i|\boldsymbol{\sigma}_{<i})$, $\boldsymbol{\sigma}_{<i} = (\sigma_1, \sigma_2, \cdots, \sigma_{i-1})$, in Eq. (2.13) is the *recurrent cell*. A standard recurrent cell is simply a user-defined nonlinear function f (e.g. $f = \tanh$) that maps an input vector σ_{i-1} along with a hidden vector $\boldsymbol{h}_{i-1}$ of length n_h and to a new hidden vector $\boldsymbol{h}_i$:

$$\boldsymbol{h}_i = f\left(\boldsymbol{W}\boldsymbol{\sigma}_{i-1} + \boldsymbol{U}\boldsymbol{h}_{i-1} + \boldsymbol{c}\right), \tag{2.14}$$

where $i \in [1, 2, \cdots, N]$, and the tunable parameters $\boldsymbol{W}$, $\boldsymbol{U}$, and $\boldsymbol{c}$ that are a part of the RNN parameters $\boldsymbol{\theta}$ have dimension/size $n_h \times \mathcal{D}$, $n_h \times n_h$, and n_h, respectively. Note that a bolded $\boldsymbol{\sigma}_i$ denotes the one-hotted representation of σ_i. Typically, $\boldsymbol{h}_0$ and $\boldsymbol{\sigma}_0$ are chosen to be null vectors. After the recurrent cell has calculated an output $\boldsymbol{h}_i$, one then maps $\boldsymbol{h}_i$ to an output vector $\boldsymbol{y}_i$ with size $\mathcal{D}$, where

$$\boldsymbol{y}_i = S\left(\boldsymbol{V}\boldsymbol{h}_i + \boldsymbol{b}\right), \tag{2.15}$$

$S(\cdots)$ is the softmax function, and $\boldsymbol{V}$ (dimension $n_h \times \mathcal{D}$) and $\boldsymbol{b}$ (size $\mathcal{D}$) are also a part of $\boldsymbol{\theta}$. A visualization of the functionality of the recurrent cell in tandem with the softmax layer is shown in Fig. 2.2.

The necessity of the softmax layer that yields an output $\boldsymbol{y}_i$ lies in the desire for multi-label classification. One would like to interpret the output of a recurrent cell as a conditional probability vector whose entries correspond to predicting the i^{th} state of the input vector $\boldsymbol{\sigma}$ *given* the previous input states $\boldsymbol{\sigma}_{<i}$. Specifically,

$$\boldsymbol{y}_i = \left[p_{\boldsymbol{\theta}}(\sigma_i = 0|\boldsymbol{\sigma}_{<i}), p_{\boldsymbol{\theta}}(\sigma_i = 1|\boldsymbol{\sigma}_{<i}), \cdots, p_{\boldsymbol{\theta}}(\sigma_i = \mathcal{D}-1|\boldsymbol{\sigma}_{<i})\right]. \tag{2.16}$$

[1] Even in the $\mathcal{D} = 2$ case it makes sense to perform a one-hot encoding.

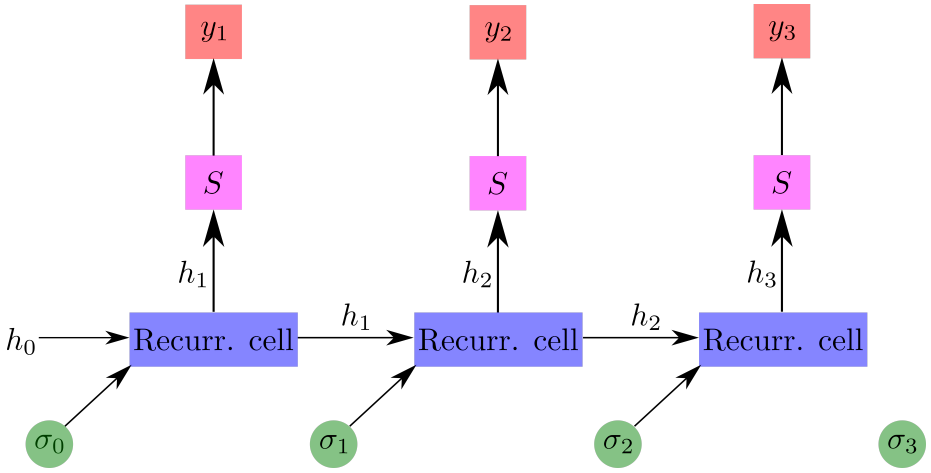

Figure 2.2: An example of the RNN architecture for $N = 3$. At the i^{th} step of the process, σ_{i-1} and $\boldsymbol{h}_0$ are fed into the recurrent cell resulting in a new hidden vector $\boldsymbol{h}_i$. $\boldsymbol{h}_i$ is mapped via Eq. (2.15) to the vector $\boldsymbol{y}_i$ whose entries are given by Eq. (2.16).

Repeating this process of calculating all model predictions $\boldsymbol{y}_1$, $\boldsymbol{y}_2$, $\cdots$, and $\boldsymbol{y}_N$, the full probability distribution given by Eq. (2.13) is

$$p_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = \prod_{i=1}^{N} \boldsymbol{y}_i \cdot \boldsymbol{\sigma}_i, \tag{2.17a}$$

where

$$p_{\boldsymbol{\theta}}(\sigma_i | \boldsymbol{\sigma}_{<i}) = \boldsymbol{y}_i \cdot \boldsymbol{\sigma}_i \tag{2.17b}$$

and the conditional dependence on the previous states $\boldsymbol{\sigma}_{<i}$ is contained within the output vector $\boldsymbol{y}_i$.

As with the RBM, to tune the RNN parameters $\boldsymbol{\theta}$ such that $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ best approximates a target distribution $q(\boldsymbol{\sigma})$, we require optimizing a suitable cost function that measures the "similarity" between the two distributions. It ended up being mathematically convenient to employ the KL divergence in the RBM case, but in the RNN case one has the freedom to choose a cost function from a plethora of options while leaving the calculation of gradients to backpropagation/automatic differentiation software packages [54–59]. Specific cost functions used to optimize RNN parameters $\boldsymbol{\theta}$ throughout this thesis will be shown when required.

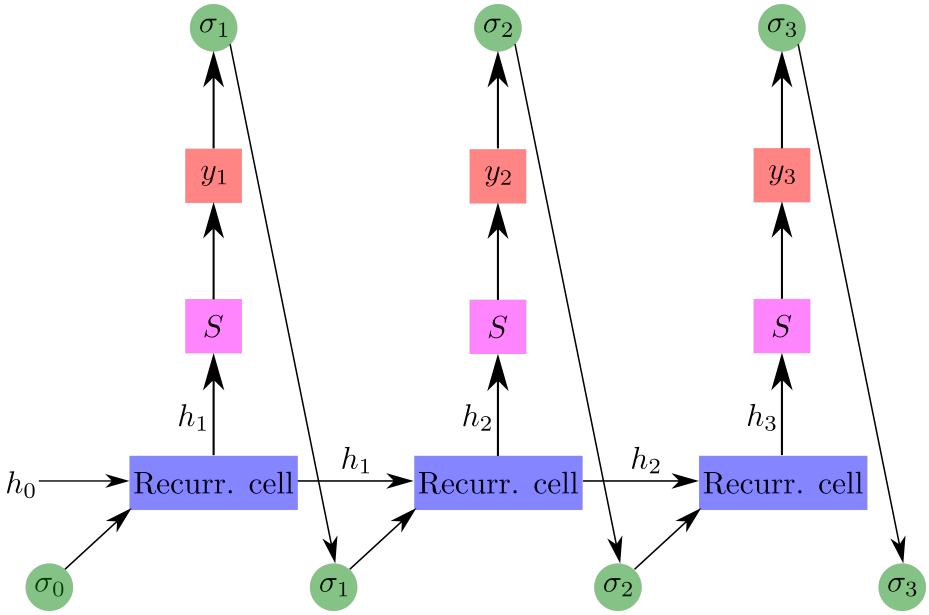

Figure 2.3: The autoregressive RNN sampling algorithm for $N = 3$ visible units. After the RNN output $\boldsymbol{y}_i$ is calculated, it is sampled to give a model-predicted result $\boldsymbol{\sigma}_i$. The model prediction is then used as the proceeding visible unit required to calculate $\boldsymbol{h}_{i+1}$. This process is repeated N times to give a sample of $N = 3$ visible units $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \sigma_3)$.

Evaluating cost functions requires that samples be generated from $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$. Not only this, but for calculating observables using the RNN, we require samples from $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$. Given the RNN's autoregressive architecture, sampling from $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ may be done exactly and without correlations between successively drawn samples. Fig. 2.3 depicts the algorithm for drawing samples from $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ autoregressively. At the i^{th} step in the recurrent process, the output $\boldsymbol{y}_i$ is sampled (recall Eq. 2.16) to give a result $\boldsymbol{\sigma}_i$, which is then fed in as an input to the next recurrent step to calculate $\boldsymbol{y}_{i+1}$. Repeating this process N times yields an RNN-generated sample $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \cdots, \sigma_N)$.

2.2.1 Beyond standard recurrent cells

For an RBM, we are stuck with its fundamental architecture. On the other hand, the RNN's recurrent cell is highly customizable. Other recurrent cells have been formulated that drastically outperform the standard recurrent cell given by Eq. 2.14 in learning longer sequences of data, as gradients of the parameters $\boldsymbol{\theta}$ tend to explode or vanish [60]. Such

commonly-used recurrent cells are the *long short term memory* (LSTM) cell [61], and the *gated recurrent unit* (GRU) cell [62]. A detailed explanation of the LSTM and GRU cell architectures is out of the scope of this thesis. The reader is encouraged to refer to Refs. [12, 37, 61, 62] for more information.

Given the extreme flexibility for the internals of a recurrent cell, one may also try to incorporate some level of physical dimensionality in the recurrent cell and autoregressive architecture. Specifically, Hibat-Allah *et al.* proposed a two-dimensional RNN algorithm for rectangular lattices in Ref. [12] whose hidden vectors $\boldsymbol{h}_i$ are calculated using *horizontal and vertical* information,

$$\boldsymbol{h}_{i,j} = f\left(\boldsymbol{W}^{(\mathrm{h})}\boldsymbol{\sigma}_{i-1,j} + \boldsymbol{W}^{(\mathrm{v})}\boldsymbol{\sigma}_{i,j-1} + \boldsymbol{U}^{(\mathrm{h})}\boldsymbol{h}_{i-1,j} + \boldsymbol{U}^{(\mathrm{v})}\boldsymbol{h}_{i,j-1} + \boldsymbol{c}\right), \tag{2.18}$$

where the superscripts (h) and (v) denote *horizontal* and *vertical*, respectively. As before with the standard RNN architecture, $\boldsymbol{\sigma}_{0,j}$, $\boldsymbol{\sigma}_{i,0}$, $\boldsymbol{h}_{0,j}$, and $\boldsymbol{h}_{i,0}$ are usually null vectors. The method for estimating $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ and drawing samples from it proceed similarly as before, with the added complexity that hidden and visible units from previous rows are taken into account as prescribed by Eq. (2.18). Fig. 2.4 demonstrates the autoregressive algorithm for the two-dimensional RNN. Although this algorithm hinges on open boundary conditions given the recurrence on the boundaries of the architecture (i.e. inputting null vectors), Hibat-Allah *et al.* demonstrated in a variational Monte Carlo setting that this 2D RNN outperforms many state-of-the-art algorithms [12].

2.3 Calculating observables

Monitoring the loss function during the training process of any generative model provides valuable insight into how the actual learning task is progressing. Hopefully, as training progresses, the loss function gets closer and closer to a value indicative of when $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = q(\boldsymbol{\sigma})$. That being said, as physicists we tend to gravitate, no pun intended, towards metrics for the training process that are physical, no pun intended. For instance, for proof-of-concept quantum state reconstruction examples, the target wavefunction is known ahead of time to verify the performance of training the generative model. Therefore, one can calculate the quantum fidelity

$$\mathcal{F} = \left|\langle \lambda_0 | \psi_{\boldsymbol{\theta}} \rangle\right|^2 = \left|\sum_{\boldsymbol{\sigma}} \lambda_0(\boldsymbol{\sigma}) \psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})\right|^2 \tag{2.19}$$

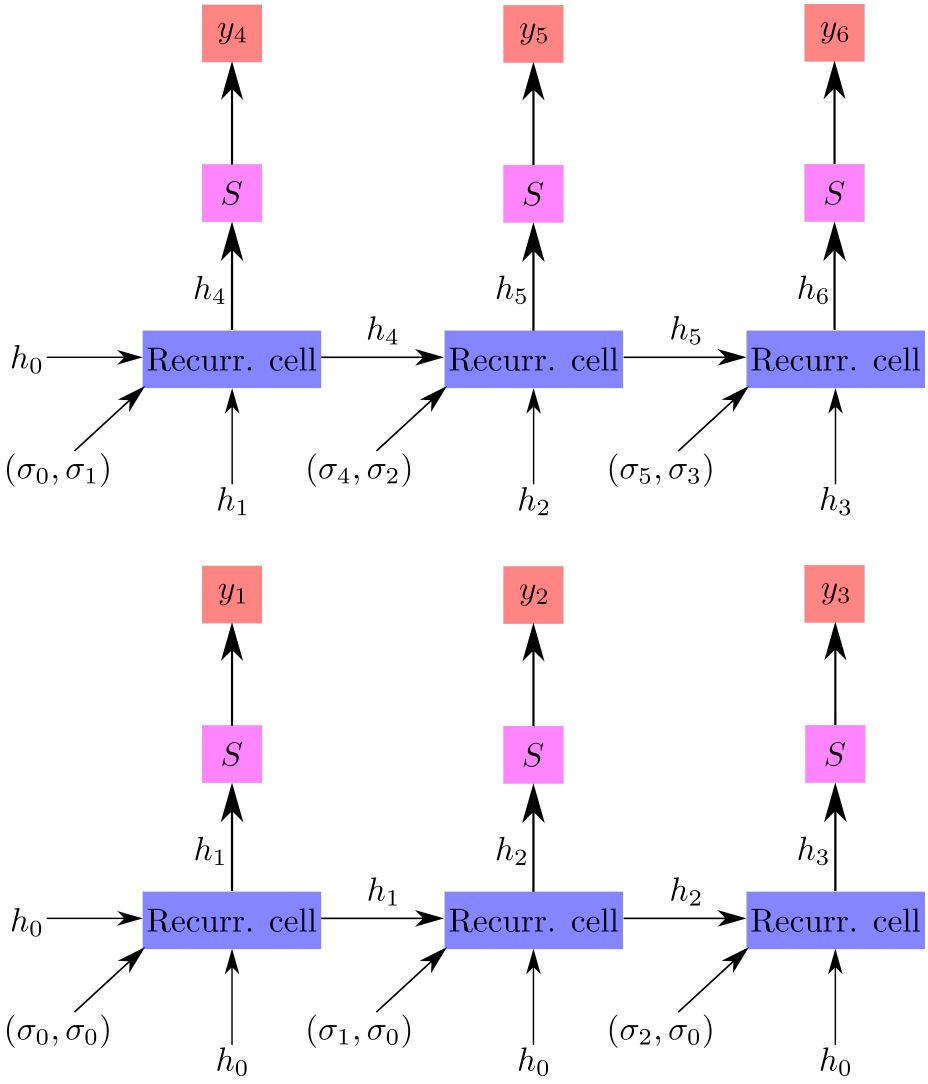

Figure 2.4: An example of the 2D RNN architecture for an $N = 3 \times 2$ square lattice. At lattice site (i, j), neighbouring (down and left) vectors $\boldsymbol{\sigma}_{i,j-1}$, $\boldsymbol{\sigma}_{i-1,j}$, $\boldsymbol{h}_{i,j-1}$, and $\boldsymbol{h}_{i-1,j}$ are fed into the recurrent cell in Eq. (2.18) resulting in a new hidden vector $\boldsymbol{h}_i$. $\boldsymbol{h}_i$ is mapped via Eq. (2.15) to the vector $\boldsymbol{y}_i$ whose entries are given by Eq. (2.16).

between the target quantum state (the ground state of some Hamiltonian in our case) and the neural network quantum state (see Eq. (1.8)).[2]

[2]In Eq. (2.19), we've assumed that $\lambda_0(\boldsymbol{\sigma})$ and $\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ are real and positive: $\lambda_0(\boldsymbol{\sigma}), \psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) \geq 0$.

It may also be useful to monitor an observable during training, such as the energy or the magnetization. For a diagonal observable $\hat{\mathcal{O}}$, its expectation value with respect to the normalized state $\psi_{\boldsymbol{\theta}}$ is

$$\langle \hat{\mathcal{O}} \rangle_{\psi_{\boldsymbol{\theta}}} = \sum_{\boldsymbol{\sigma}} \mathcal{O}(\boldsymbol{\sigma}) |\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})|^2, \tag{2.20}$$

where $\mathcal{O}(\boldsymbol{\sigma}) = \langle \boldsymbol{\sigma} | \hat{\mathcal{O}} | \boldsymbol{\sigma} \rangle$. However, the sum over states $\boldsymbol{\sigma}$ is still exponential in the number of degrees of freedom ($\mathcal{D}^N$). Given the sampling routines for the RBM and RNN, one can generate a set of independent measurements $\boldsymbol{\Gamma}$ in the computational basis to calculate the desired observable on. In this case,

$$\langle \hat{\mathcal{O}} \rangle_{\psi_{\boldsymbol{\theta}}} \approx \langle \hat{\mathcal{O}} \rangle_{\boldsymbol{\Gamma}} = \frac{1}{|\boldsymbol{\Gamma}|} \sum_{\boldsymbol{\sigma} \in \boldsymbol{\Gamma}} \mathcal{O}(\boldsymbol{\sigma}). \tag{2.21}$$

When $\hat{\mathcal{O}}$ is an off-diagonal observable,

$$\langle \hat{\mathcal{O}} \rangle_{\psi_{\boldsymbol{\theta}}} = \sum_{\boldsymbol{\sigma}\boldsymbol{\sigma}'} \psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma}') \psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) \mathcal{O}(\boldsymbol{\sigma}, \boldsymbol{\sigma}'), \tag{2.22}$$

where $\mathcal{O}(\boldsymbol{\sigma}, \boldsymbol{\sigma}') = \langle \boldsymbol{\sigma} | \hat{\mathcal{O}} | \boldsymbol{\sigma}' \rangle$. Again, we have an exponential sum that can be replaced by calculating $\langle \hat{\mathcal{O}} \rangle$ over a dataset $\boldsymbol{\Gamma}$. Consider the following.

$$\begin{aligned}
\langle \hat{\mathcal{O}} \rangle_{\psi_{\boldsymbol{\theta}}} &= \sum_{\boldsymbol{\sigma}\boldsymbol{\sigma}'} \psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) \psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma}') \mathcal{O}(\boldsymbol{\sigma}, \boldsymbol{\sigma}') \\
&= \sum_{\boldsymbol{\sigma}} \psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) \sum_{\boldsymbol{\sigma}'} \psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma}') \times \frac{\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}{\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})} \mathcal{O}(\boldsymbol{\sigma}, \boldsymbol{\sigma}') \\
&= \sum_{\boldsymbol{\sigma}} |\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})|^2 \sum_{\boldsymbol{\sigma}'} \frac{\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma}')}{\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})} \mathcal{O}(\boldsymbol{\sigma}, \boldsymbol{\sigma}')
\end{aligned}$$

Letting

$$\mathcal{O}^{[L]}(\boldsymbol{\sigma}) = \sum_{\boldsymbol{\sigma}'} \frac{\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma}')}{\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})} \mathcal{O}(\boldsymbol{\sigma}, \boldsymbol{\sigma}'), \tag{2.23}$$

we may write

$$\langle \hat{\mathcal{O}} \rangle_{\psi_{\boldsymbol{\theta}}} = \sum_{\boldsymbol{\sigma}} |\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})|^2 \mathcal{O}^{[L]}(\boldsymbol{\sigma}).$$

Now,

$$\langle \hat{\mathcal{O}} \rangle_{\psi_{\boldsymbol{\theta}}} \approx \langle \hat{\mathcal{O}} \rangle_{\boldsymbol{\Gamma}} = \frac{1}{|\boldsymbol{\Gamma}|} \sum_{\boldsymbol{\sigma} \in \boldsymbol{\Gamma}} \mathcal{O}^{[L]}(\boldsymbol{\sigma}). \tag{2.24}$$

If $\hat{\mathcal{O}}$ is sufficiently sparse, the sum over $\boldsymbol{\sigma}'$ in the definition of $\mathcal{O}^{[L]}(\boldsymbol{\sigma})$ can be done efficiently. Notice also that, since we are taking the ratio of wavefunction coefficients in Eq. (2.23), the state $\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ need not be normalized. This is crucial for when $\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ is parameterized by an RBM, as the normalization $Z_{\boldsymbol{\theta}}$ requires an exponential sum over the visible units to evaluate (see Eq. (2.6)).

Given Eq. (2.20) and Eq. (2.24), we are now free to monitor any observable during the training process given a network-generated dataset $\boldsymbol{\Gamma}$. We will now turn to comparing the RBM and RNN in the context of two different learning tasks.

2.4 The XY Model

Hamiltonians, and therefore ground states, that are invariant under a symmetry group come to mind when searching for reconstruction tasks that are unique to quantum many-body systems. Such a ground state may have coefficients that are zero due to the presence of the symmetry, which poses an interesting reconstruction task. The well-known XY Hamiltonian,

$$\hat{H} = -J \sum_{\langle ij \rangle} (\hat{S}_i^x \hat{S}_j^x + \hat{S}_i^y \hat{S}_j^y), \tag{2.25}$$

is invariant under U(1) symmetry operations and it clearly satisfies the Perron-Frobenius theorem when $J > 0$. The U(1) symmetry leads to a conservation of the total S^z magnetization, $S^z = \sum_i S_i^z = 0$, in the ground state. Not only this, but this Hamiltonian has also recently been engineered in trapped ion experiments [63–65]. Given the experimental motivation of this thesis, the XY Hamiltonian ticks all of the boxes for a good reconstruction task. Unsurprisingly, our computational basis for this reconstruction task will be the eigenstates of the $\hat{S}^z = \frac{\hbar}{2}\hat{\sigma}^z$ operator, $\hat{S}^z \left|\uparrow / \downarrow\right\rangle = \pm\frac{\hbar}{2} \left|\uparrow / \downarrow\right\rangle$, which has the trivial binary (0 and 1) representation via the linear transformation $\frac{1}{2}(\hat{\sigma}^z + 1)$. In this representation, 0 and 1 correspond to $\uparrow$ and $\downarrow$, respectively.

2.4.1 Enforcing U(1)-symmetry with RNNs

Although the autoregressive property of RNNs is usually cited as the reason why they are preferred over RBMs, both generative models do not obey any global constraints. For instance, in the case of total S^z magnetization conservation, samples drawn from either generative model will, in general, contain instances that fall outside the $S^z = \sum_i S_i^z = 0$ sector. For the block Gibbs sampling algorithm with RBMs (see Eqs. (2.10) and (2.11)),

implementing this symmetry is not possible. The entire visible layer $\boldsymbol{\sigma}^{(k)}$ is generated in one step by simultaneously sampling each conditional distribution $p(\sigma_i = 1|\boldsymbol{h})$, Eq. (2.11), at the k^{th} step in the block Gibbs sampling procedure. The resulting value of each visible unit σ_i is not conditioned on any other visible unit values, prohibiting a $S^z = 0$ symmetry enforcement.

On the other hand, Hibat-Allah *et al.* showed that the autoregressive nature of the RNN can be taken advantage of for symmetry purposes [12]. In particular, S^z magnetization conservation can be enforced in the autoregressive architecture as follows. Firstly, for even N, a sample drawn from a state that conserves $S^z = \sum_i S_i^z = 0$ will have an equal number of qubits in the $\downarrow$ and $\uparrow$ states. Therefore, in the sequential autoregressive algorithm that we deploy for generating samples one spin at a time, if at any point during the procedure we find that half of the spins are in the $\uparrow$ ($\downarrow$) state, the remainder of the spins in the sequence *must* be in the $\downarrow$ ($\uparrow$) state.

Mathematically, given the original output of one recurrent step,[3]

$$\boldsymbol{y}_i = [p_{\boldsymbol{\theta}}(\sigma_i = 0|\boldsymbol{\sigma}_{<i}), p_{\boldsymbol{\theta}}(\sigma_i = 1|\boldsymbol{\sigma}_{<i})],$$

each entry is modified to

$$\tilde{p}_{\boldsymbol{\theta}}(\sigma_i = 0|\boldsymbol{\sigma}_{<i}) = p_{\boldsymbol{\theta}}(\sigma_i = 0|\boldsymbol{\sigma}_{<i}) \times \Theta\left(\frac{N}{2} - N_{\uparrow}\right) \tag{2.26a}$$

and

$$\tilde{p}_{\boldsymbol{\theta}}(\sigma_i = 1|\boldsymbol{\sigma}_{<i}) = p_{\boldsymbol{\theta}}(\sigma_i = 1|\boldsymbol{\sigma}_{<i}) \times \Theta\left(\frac{N}{2} - N_{\downarrow}\right), \tag{2.26b}$$

where

$$\Theta(x) = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases}, \tag{2.27}$$

is the Heaviside step function and the new *symmetry-enforced* output is

$$\tilde{\boldsymbol{y}}_i = \frac{[\tilde{p}_{\boldsymbol{\theta}}(\sigma_i = 0|\boldsymbol{\sigma}_{<i}), \tilde{p}_{\boldsymbol{\theta}}(\sigma_i = 1|\boldsymbol{\sigma}_{<i})]}{\tilde{p}_{\boldsymbol{\theta}}(\sigma_i = 0|\boldsymbol{\sigma}_{<i}) + \tilde{p}_{\boldsymbol{\theta}}(\sigma_i = 1|\boldsymbol{\sigma}_{<i})}. \tag{2.28}$$

To clarify, in modifying the RNN sampling algorithm to yield samples that are guaranteed to satisfy $S^z = \sum_i S_i^z = 0$, the RNN itself is still *not* inherently U(1)-symmetric

[3] $\mathcal{D} = 2$ here.

in that its probability distribution $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ does not have zero amplitudes for configurations outside of the $S^z = 0$ sector. However, in replacing outputs $\boldsymbol{y}_i$ with $\tilde{\boldsymbol{y}}_i$, we are instead sampling from a different symmetry-enforced distribution $\pi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$. This has rippling effects when training an RNN using this modified distribution. For example, the loss function that will be employed in the next section for training RNNs is the the negative log-likelihood,

$$\text{NLL} = -\frac{1}{|\mathbf{D}|}\sum_{\boldsymbol{\sigma}\in\mathbf{D}} \log P_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = -\frac{1}{|\mathbf{D}|}\sum_{\boldsymbol{\sigma}\in\mathbf{D}}\sum_{i=1}^{N} \log P_{\boldsymbol{\theta}}(\sigma_i|\boldsymbol{\sigma}_{<i}) \tag{2.29}$$

where $P = p$ or π and the summand factorizes given Eq. (2.13). When $P = \pi$, spins where the symmetry enforcement is applied do not contribute to the NLL. If the j^{th} spin in the sequence $1, 2, \cdots, N$ is where the symmetry is applied, each of the spins from j to N will have $\pi_{\boldsymbol{\theta}}(\sigma_j|\boldsymbol{\sigma}_{<j}) = 1$ from Eq. (2.26), making $\log \pi_{\boldsymbol{\theta}}(\sigma_j|\boldsymbol{\sigma}_{<j}) = 0$. Given the sum-of-products structure of the NLL, the effects of these terms being zero might alleviate potential vanishing and exploding gradients, therefore stabilizing the training process [66]. The effective "shortening" of the sequence at the j^{th} site where the symmetry enforcement occurs may provide improvements akin to those from the truncated backpropagation (TB) algorithm [67].

2.4.2 Reconstructing the ground state

The numerical results in this section are designed to demonstrate the differences between training RNNs – with and without symmetry enforcement – and RBMs on data from the ground state of Eq. (2.25). To obtain measurement data required to train these generative models, one can employ the density matrix renormalization group (DMRG) algorithm [6,7]. DMRG is widely considered to be among the top numerical methods available for simulating ground states of 1D systems with nearest-neighbour interactions [10]. Having done a DMRG simulation to generate a training dataset, one also has access to a host of other observables, like the energy, to reference during the reconstruction procedure. Not only this, but the exact sampling algorithm outlined by Ferris and Vidal in Ref. [68] also provides efficient means for sampling the output ground state from DMRG calculations. Therefore, using a synthetically produced training dataset compared to a real-world training dataset is justified.

Using the open-source library called ITensor [69], various 1D system sizes up to $N = 50$ spins were simulated and the resulting DMRG ground states were sampled. Each training dataset comprised of $|\mathbf{D}| = 2 \times 10^5$ independent samples from the ground state. In preliminary calculations, datasets with $|\mathbf{D}| > 2 \times 10^5$ negligibly affected the training

outcomes. Having performed DMRG simulations for each system size of interest, the generative model fidelity (Eq. (2.19)) – for $N \leq 10$ – and the energy difference,

$$\varepsilon = \frac{|E_{\mathrm{DMRG}} - \langle \hat{H} \rangle_{\psi_{\boldsymbol{\theta}}}|}{N}, \tag{2.30}$$

between the DMRG ground state and the generative model were monitored during the training process. Unless otherwise stated, the number of samples that were generated from each generative model to estimate the energy $\langle \hat{H} \rangle_{\psi_{\boldsymbol{\theta}}}$ using Eq. (2.24) was 10^4. Lastly, the hyperparameters used to train the RBMs and RNNs are given in Tables 2.1 and 2.2, respectively. Hyperparameter choices were made based on laborious preliminary calculations that yielded the best results (closest values to the DMRG values) with minimal fine-tuning to best demonstrate the real-world use of both generative models.

Table 2.1: Hyperparameters used for training RBMs.

Hyperparameter	Value / Type
Optimizer	SGD
Learning rate	0.01×0.9998^t, where t is epoch number
Hidden units	50, 100, 500
Positive batch size	100
Negative batch size	200
CD_k steps (k)	100

Table 2.2: Hyperparameters used for training RNNs.

Hyperparameter	Value / Type
Loss function	NLL (Eq. (2.29))
RNN recurrent cell	GRU
Optimizer	SGD
Hidden units	100
Random seed	1
Learning rate	0.001
Batch size	50

In this spirit, determining hyperparameters for training RNNs was relatively simple compared to those of RBMs. For instance, hidden layer sizes in RNNs could be reasonably

increased with sensible diminishing returns without changing the optimizer, learning rate, or batch size. For RBMs, this was not the case. Figs. 2.5 to 2.7 show various metrics for systems sizes ranging from $N = 10$ to 50 as a function of the training epoch and number of RBM hidden units n_h. Not only does the increase in hidden layer size not correspond to a general monotonic increase in the accuracy of the training metric, but larger hidden layer sizes can also lead to volatile instability. Other optimizers besides SGD with an exponentially decaying learning rate were tested in preliminary calculations, but this alone did not fix the instability issues and, in all cases, SGD still prevailed. On top of this, several different RBM parameter initializations were used to determine if bad parameter initialization was the culprit for the observed instability, all of which indicated that the given hyperparameter choices were not adequate. Evidently, the search for hyperparameters required to stabilize larger RBM network sizes is non-trivial and requires non-negligible preliminary manual effort.

In most cases, the smallest hidden layer size $n_h = 50$ yielded the best training accuracy compared to $n_h = 100$ and 500. In hopes of using larger hidden layer sizes to increase the RBM's expressivity while also addressing the highlighted trainability issues, a slightly more severe learning rate decay was employed: 0.01×0.999^t, where t is epoch number. Fig. 2.8 shows the infidelity $1 - \mathcal{F}$ and energy difference ε (Eq. (2.30)) for system sizes $N = 4$ and 10 using RBMs and RNNs with and without symmetry enforcement. In these plots, the RBM clearly does not reach as high of a training accuracy compared to the RNNs. Symmetry-enforcement aside, one can argue that the RNNs can reach a higher reconstruction accuracy than the RBMs due to the parameter number disparity between both networks. For example, an RNN using a GRU recurrent unit with 100 hidden units and a final softmax layer has approximately 3.1×10^5 tunable parameters. On the other hand, an RBM with $N = 10$ and $n_h = 100$ has approximately 10^3. However, as was demonstrated in Fig. 2.5, larger hidden layer sizes can lead to instability. The distinction between RBM expressivity and trainability is important.

The RBM's trainability in this context is clearly problematic. However, this seems to not be the case for the RNNs. Even with orders of magnitude more parameters, training RNNs with or without symmetry enforcement is a stable stochastic process. Comparatively, though, the symmetry-enforced RNN, U(1)-RNN, seems to reach higher training accuracies in fewer training epochs than the standard RNN without symmetry enforcement. Crucially, the standard RNN can reach the same training accuracy when given enough time (enough training epochs), except for the infidelity plot for $N = 4$ in Fig. 2.8. To assess how this learning acceleration scales with system size, Fig. 2.9 illustrates the energy difference ε for larger systems $N = 30$, 40, and 50 using both RNNs and U(1)-RNNs.[4] It is also clear

[4]RBMs with $n_h = 100$ and the more severe learning rate decay of 0.01×0.999^t were also trained for

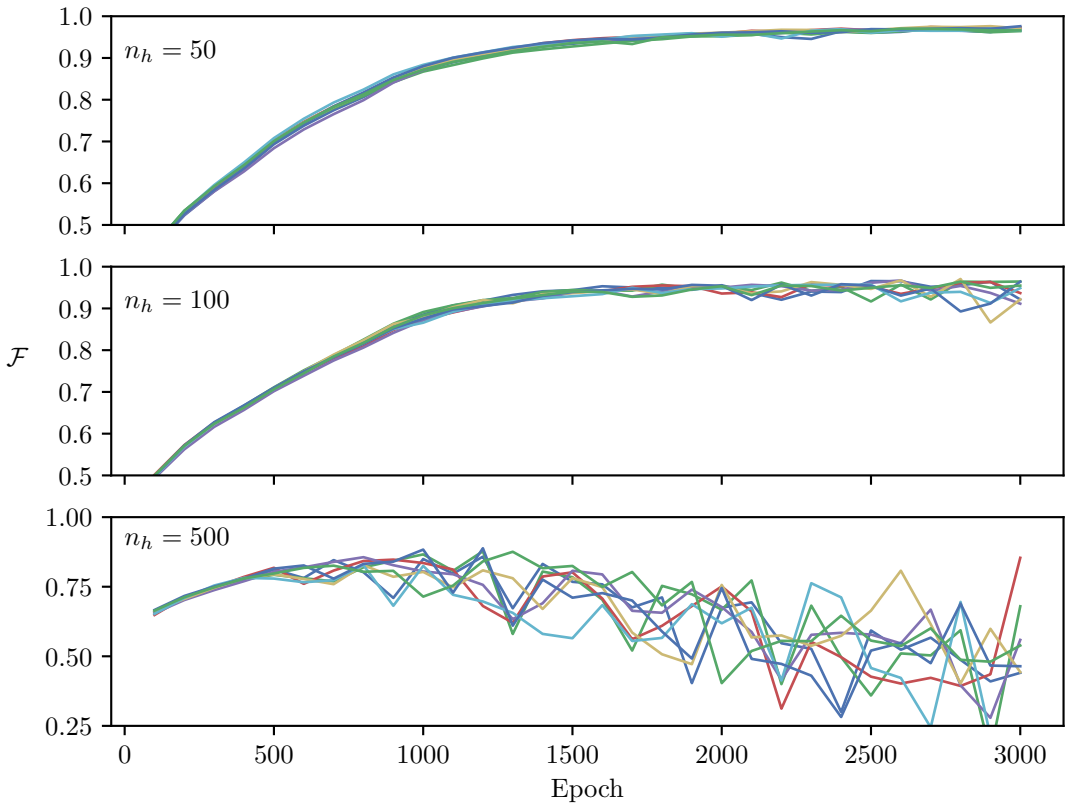

Figure 2.5: The monitored quantum fidelity (Eq. (2.19)) between the RBM wavefunction using three different hidden layer sizes and the ground state of Eq. (2.25) for $N = 10$ spins. Lines on the same plot represent different starting parameter initializations (random seeds), for which there were 8. The RBM hyperparameters used are given in Table 2.1.

here that the U(1)-RNN reaches a better training accuracy faster than the standard RNN. As was the case with the smaller system sizes in Fig. 2.8, the standard RNN is still able to reach reconstruction accuracies on par with the U(1)-RNN about 500 training epochs later.

these system sizes, but the corresponding estimated values of ε were far above of the plot limits in Fig. 2.9.

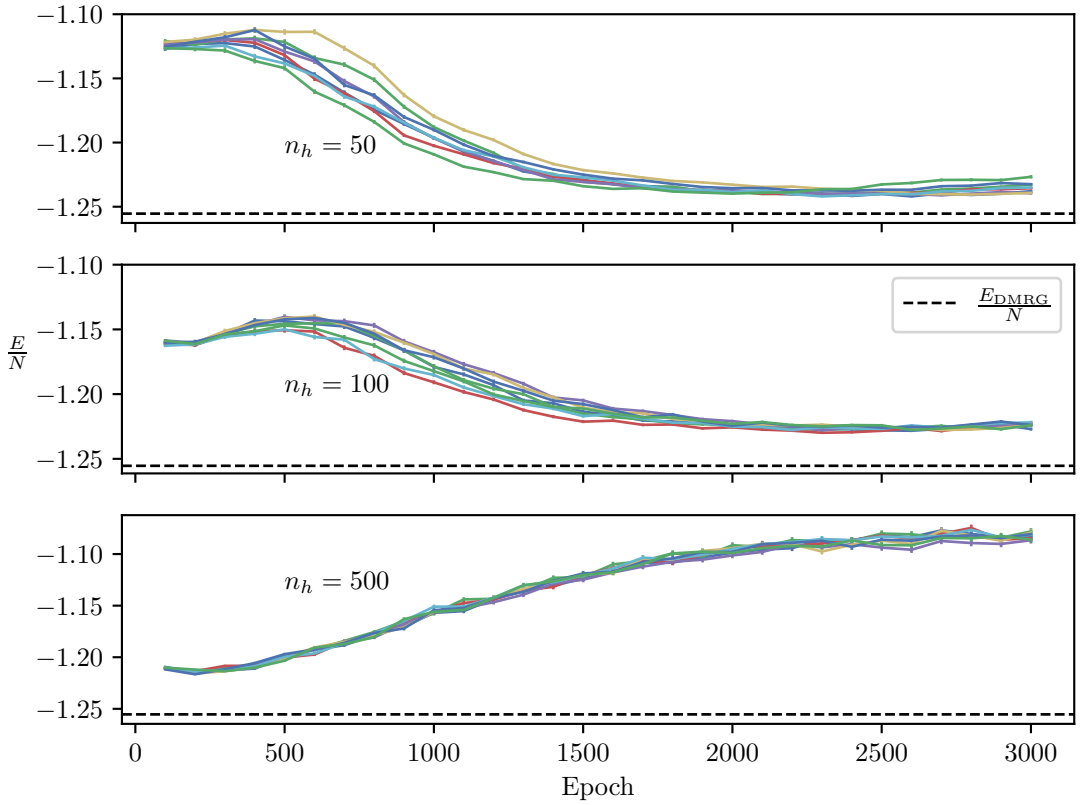

Figure 2.6: The monitored RBM energy $\langle \hat{H} \rangle_{\psi_{\boldsymbol{\theta}}}$ calculated using Eq. (2.24) using three different hidden layer sizes for $N = 40$ spins. Lines on the same plot represent different starting parameter initializations (random seeds), for which there were 8. The RBM hyperparameters used are given in Table 2.1, and 10^4 samples were generated to calculate $\langle \hat{H} \rangle_{\psi_{\boldsymbol{\theta}}}$.

2.5 Nanomolecular assemblies of endofullerenes

Given the experimental motivation interwoven throughout this thesis, the majority of models studied understandably comprise of qubit degrees of freedom. However, the binary nature associated with reconstructing such states is rather restrictive when it comes to machine learning applications. Many machine learning models that were first developed by the computer and data science communities were intended for applications like natural language processing or coloured imaged recognition. Whether the degrees of freedom are

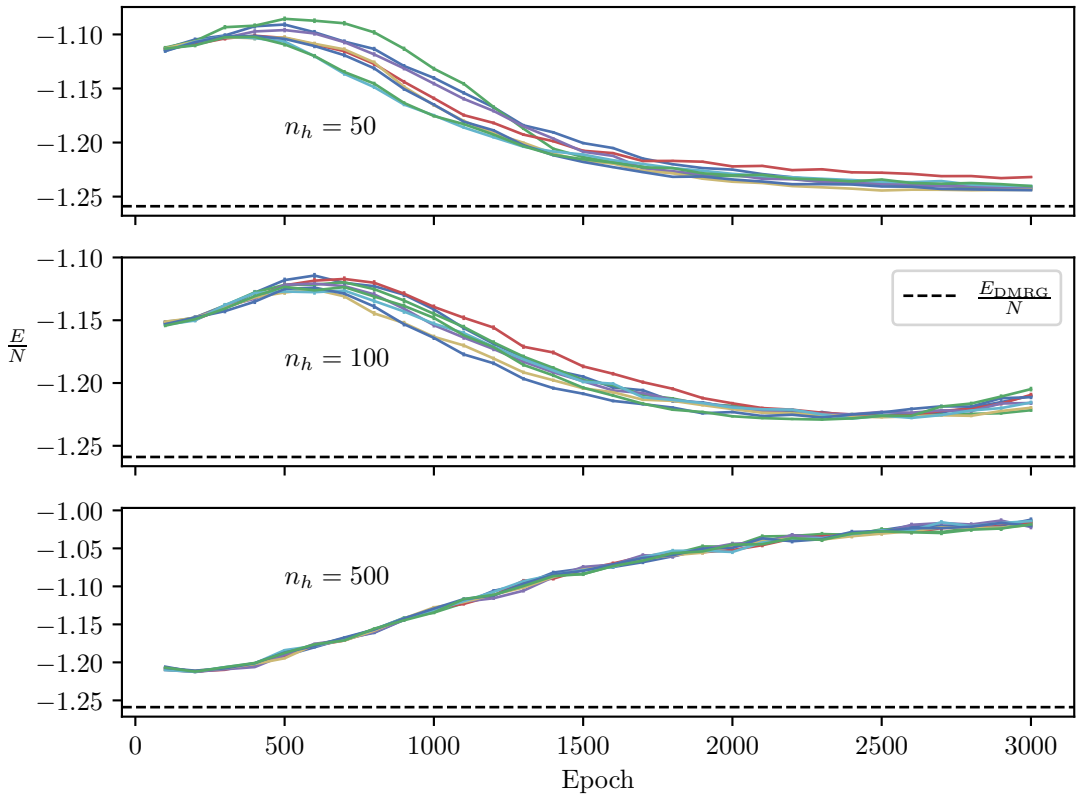

Figure 2.7: The monitored RBM energy $\langle \hat{H} \rangle_{\psi_{\boldsymbol{\theta}}}$ calculated using Eq. (2.24) using three different hidden layer sizes for $N = 50$ spins. Lines on the same plot represent different starting parameter initializations (random seeds), for which there were 8. The RBM hyperparameters used are given in Table 2.1, and 10^4 samples were generated to calculate $\langle \hat{H} \rangle_{\psi_{\boldsymbol{\theta}}}$.

words in a dictionary or pixels with limited colour options, it is clear that binary learning tasks do not paint the full picture of machine learning algorithm performance. Therefore, choosing a quantum many-body Hamiltonian that comprises of non-binary degrees of freedom while also having some experimental ties or future hardware use cases will also be an informative reconstruction task.

Higher-dimensional degrees of freedom that come to mind from a theoretical point of view are the rotational states of a rigid body, which have recently been proposed as a platform for quantum information protocols [70, 71]. A promising platform that provides access to such states is nanomolecular assemblies (NMAs) of endofullerenes. Endofullerenes

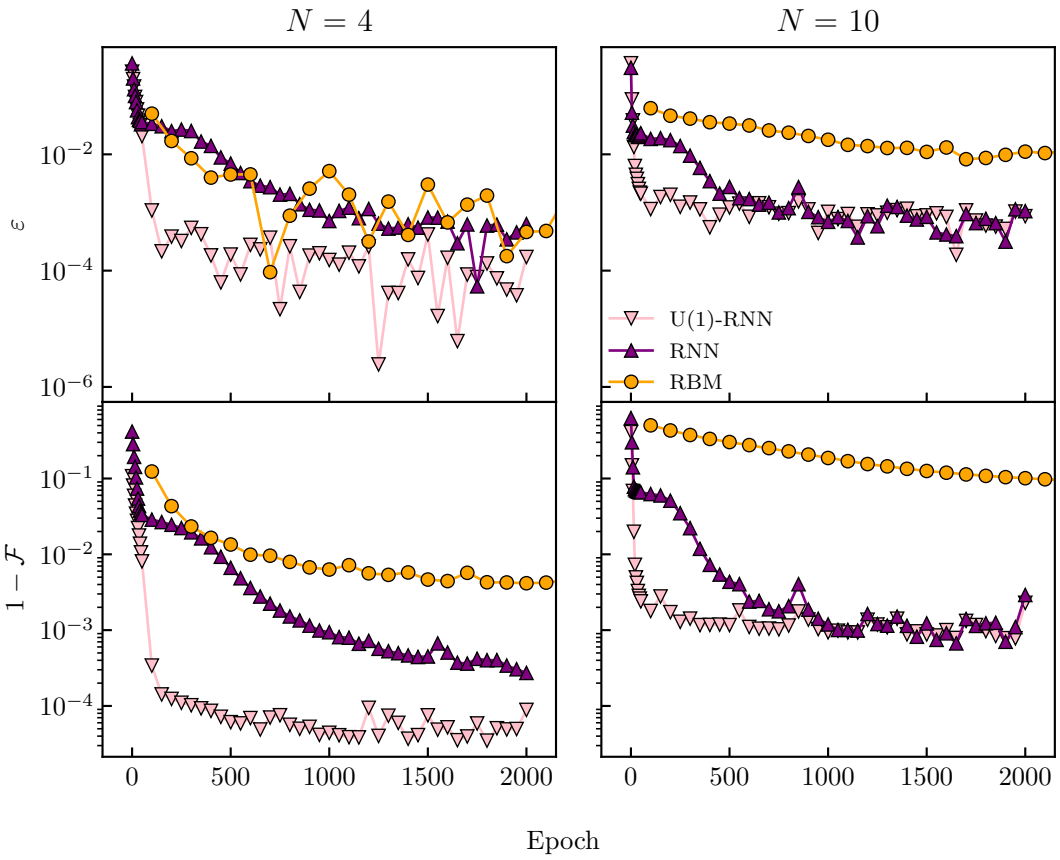

Figure 2.8: Comparing the XY model ground state training accuracy of RBMs (orange) and RNNs with (pink) and without (purple) U(1) symmetry enforcement for $N = 4$ and 10. The RBM and RNN hyperparameters used are given in Tables 2.1 and 2.2, respectively, and 10^4 samples were generated to calculate $\langle \hat{H} \rangle_{\psi_{\boldsymbol{\theta}}}$. However, the number of hidden units used for the $N = 4$ and 10 systems was $n_h = 50$ and 100, respectively, and the learning rate used was 0.01×0.999^t, where t is epoch number. Error bars for ε are smaller than the plot markers.

are entities consisting of a dipolar molecule encaged in a carbon fullerene (e.g. buckminsterfullerene, C_{60}: nature's "carbon soccer ball"). Luckily, linear chains of endofullerenes can be synthesized using nanotechnology tools [72] and molecular surgery techniques [73]. Their unique proposition for quantum information processing lies in their ability to retain quantum properties at higher temperatures compared to some hardware devices comprised

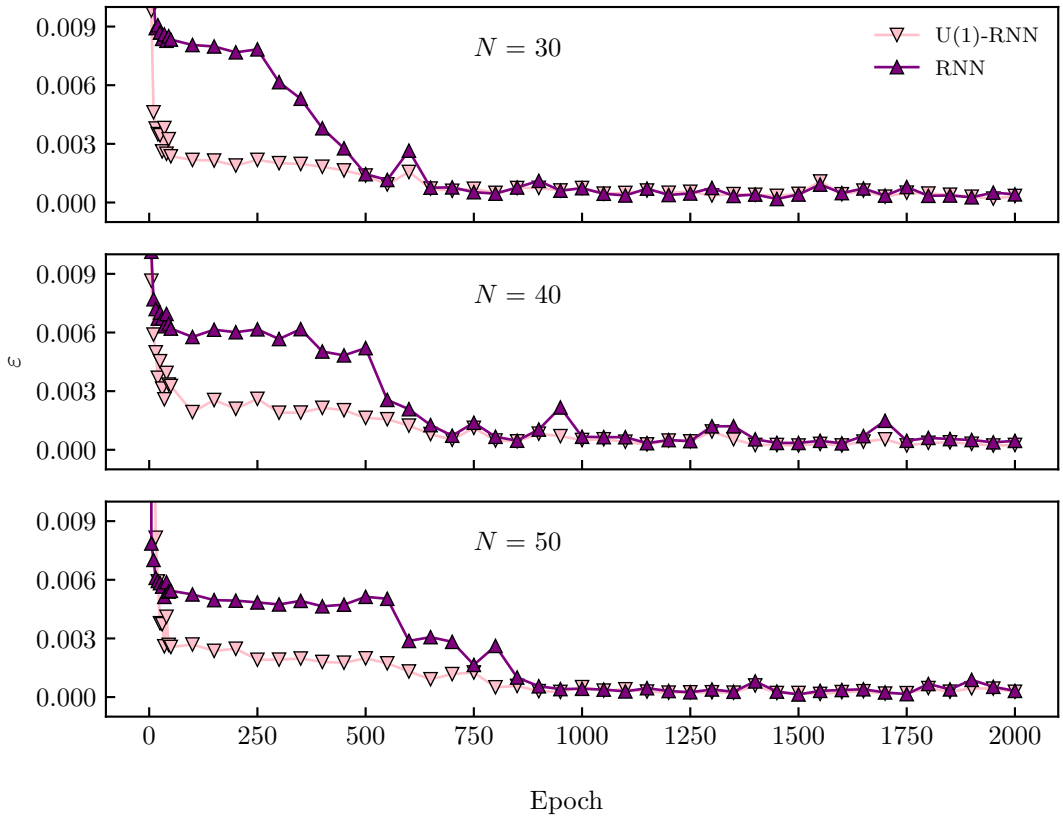

Figure 2.9: Comparing the XY model ground state training accuracy of RNNs with (pink) and without (purple) U(1) symmetry enforcement for $N = 30$, 40, and 50. The RNN hyperparameters used are given in Table 2.2 and 10^4 samples were generated to calculate $\langle \hat{H} \rangle_{\psi_{\boldsymbol{\theta}}}$. Error bars for ε are smaller than the plot markers.

of qubits [74]. Not only this, but linear chains of endofullerenes also provide interesting quantum many-body features that originate from the presence of dipole-dipole interactions [75–78]. In all, NMAs of endofullerenes are a viable candidate for an informative reconstruction task.

2.5.1 Ode to the model

To model a linear chain of N endofullerenes, sensible approximations to describe the effects of the fullerene cage are in order. However, first consider free dipolar molecules, also com-

monly referred to as *rotors*, equally spaced a distance r apart. A Hamiltonian describing a system of free rotors includes the rotational energy of each molecule and their Coulombic dipole-dipole interactions,

$$\hat{H} = \frac{B}{\hbar^2} \sum_{i=1}^{N} \hat{\ell}_i^2 + \frac{\mu^2}{4\pi\epsilon_0 r^3} \sum_{i<j} \hat{V}_{ij}, \tag{2.31}$$

where B is the rotational constant and μ is the dipole moment of each molecule. The operator $\hat{\ell}_i$ has eigenequations

$$\hat{\ell}_i^2 \left|\ell_j m_j\right\rangle = \hbar^2 \ell_i(\ell_i+1)\delta_{ij} \left|\ell_i m_i\right\rangle, \tag{2.32}$$

where the angular momentum of a single rotor is $\ell_i \in \mathbb{N}$ and its projection on a fixed spatial axis is $m_i \in [-\ell_i, \ell_i]$. Lastly, the Coulombic dipole-dipole interaction takes the form

$$\hat{V}_{ij} = \frac{\hat{x}_i\hat{x}_j + \hat{y}_i\hat{y}_j - 2\hat{z}_i\hat{z}_j}{|i-j|^3}, \tag{2.33}$$

where $\hat{x}_i$, $\hat{y}_i$, and $\hat{z}_i$ are the conventional position operators. It is useful to write the Hamiltonian in a dimensionless form

$$\frac{\hat{H}}{B} = \hat{K} + \frac{1}{R^3}\hat{V} = \sum_{i=1}^{N} \frac{\hat{\ell}_i^2}{\hbar^2} + \frac{1}{R^3} \sum_{i<j} \hat{V}_{ij}, \tag{2.34}$$

where

$$R = r\left(\frac{4\pi\epsilon_0 B}{\mu^2}\right)^{1/3} \tag{2.35}$$

is dimensionless and conveniently contains all of the Hamiltonian parameters.

As with the XY Hamiltonian in Eq. (2.25), the Hamiltonian in Eq. (2.34) also possesses symmetries that originate from commuting with two operators: the z-projection of the total angular momentum, $\sum_{i=1}^{N} \hat{\ell}_{i,z}$, and the parity operator $\bigotimes_{i=1}^{N} \hat{\pi}_i$ which reflects the z_i coordinate in the $x-y$ plane in the position representation ($\hat{\pi}_i \left|x_i, y_i, z_i\right\rangle = \left|x_i, y_i, -z_i\right\rangle$). As a result, the two respective conserved quantities are the total m value $m = \sum_{i=1}^{N} m_i$ and the total ℓ parity $\ell_\mathrm{p} \equiv \sum_{i=1}^{N} \ell_i \pmod 2$.

Enter the fullerene cage, where its presence can be safely modelled by a *screening* effect on the interaction strength between molecules [77, 78]. Effectively, the fullerene cage's presence around each degree of freedom reduces the dipole moment μ. In a real-world example, Ref. [74] shows that the dipole moment of the HF molecule is reduced from 1.83

Debye in the gas phase to an effective 0.45 Debye when encapsulated in C_{60}. Regarding other kinetic energy terms (e.g. vibration and translation), *ab initio* studies on HF trapped in C_{60} show that the ground state exhibits small rotational-translation coupling [79], which is in agreement with experiments [74]. Vibrational excitations, on the other hand, are much higher in energy than that of rotations. Therefore, leaving out translational and vibrational terms in our toy model is acceptable.

To perform quantum state reconstruction on the ground state of this toy model of NMAs of endofullerenes, we require a computational basis. At first glance, it is not obvious from Eq. (2.34) what this choice should be, but it should come down to one of two choices corresponding to the two terms in Eq. (2.34): the eigenstates of $\hat{\ell}_i$ or the position operator eigenstates. Consider again the realistic example of HF encapsulated in C_{60}. If the rotor-rotor separation distance r is on the order of the diameter of the C_{60} molecule (19.0 bohr [76]) and given the renormalized rotational constant and screened dipole moment of HF inside C_{60}, we obtain $R \approx 2.64$. In studying many other examples of other dipolar molecules that can be feasibly encapsulated by C_{60}, one generally finds that $R \gg 1$; the rotational energy dominates the Coulombic interaction term. It is therefore natural to select the eigenstates of $\hat{\ell}_i$ as a computational basis,

$$\begin{aligned} \bigotimes_{i=1}^{N} |\ell_i m_i\rangle &= |\ell_1 m_1, \ell_2 m_2, ..., \ell_N m_N\rangle \\ &= |\sigma_1, \sigma_2, ..., \sigma_N\rangle \\ &= |\boldsymbol{\sigma}\rangle \,, \end{aligned} \tag{2.36}$$

where we have ℓ_i and m_i into one integer label, σ_i.

Given that each ℓ_i can take an infinite number of values, the local Hilbert space size, $\mathcal{D}$, of each rotor is infinite. A truncation scheme is therefore in order for numerical purposes. Iouchtchenko *et al.* in Ref. [77] demonstrated that capping the available angular momentum of each rotor to a maximum value $\ell_{\max}$ is a sensible truncation scheme. Not only this, but the DMRG simulations that were performed in Ref. [77] will also be the foundation for generating training datasets in the $|\ell_i m_i\rangle$ basis for our eventual quantum state reconstruction task. Under this truncation scheme,

$$\mathcal{D} = \sum_{\ell=0}^{\ell_{\max}} (2\ell + 1) = (\ell_{\max} + 1)^2 \,, \tag{2.37}$$

and the Hilbert space size of the entire system of N endofullerenes is $\mathcal{D}^N$. In turn, adjustments to the value of $\ell_{\max}$ must be made depending on the value of R. For instance, if

R decreases (Coulombic interaction strength increases), $\ell_{\max}$ must be increased to capture the dipole-dipole interaction and, therefore, the ground state of the system.

With a computational basis in hand, the last step is to ensure that the Hamiltonian is stoquastic in the free-rotor basis (Eq. (2.36)), which requires the evaluation of the dipole-dipole operator matrix elements. The matrix elements of the position operators are shown in Eq. (A.3). Given their form, the dipole-dipole operator in Eq. (2.34) does not satisfy the stoquasticity requirements. However, it can be argued that this does not have to impede our quantum state reconstruction task. The key to this argument is that the Hamiltonian in Eq. (2.34) in the free-rotor basis has real-valued matrix elements, and therefore so must its ground state. It is shown in App. A.2 that the induced error in treating the ground state as having entirely positive coefficients is negligible for the values of R, $\ell_{\max}$, and N studied in the next section. Therefore, we may proceed with our reconstruction task without the burden of having a wavefunction with sign structure.

2.5.2 Reconstructing the ground state

As with Sec. 2.4.2, numerical results presented here are designed to distinguish the differences between training RNNs and RBMs on data in the free-rotor basis from the ground state of Eq. (2.34). The Hamiltonian in Eq. (2.34) is strictly characterized by the parameter R, which indicates the relative strength of the rotational kinetic energy term as compared to the dipole-dipole potential term. Previous Monte Carlo, path integral, and DMRG simulations for 1D systems have indicated that a continuous phase transition exists near $R = 1.0$ [77, 80, 81]. For convenience, we will refer to $R = 1.0$ as the *critical point* for this transition in 1D. The measurement data for was also synthetically acquired via DMRG calculations [6, 7, 77]. Each training dataset was comprised of $|\mathbf{D}| = 10^4$ independent samples produced by the algorithm outlined in Ref. [68]. Preliminary calculations showed that having $|\mathbf{D}| > 10^4$ for system sizes $N \leq 8$ negligibly affected the training outcomes.

Having access to DMRG simulations that produced the training datasets, we may again compare the estimated generative model energy $\langle \hat{H} \rangle_{\psi_{\boldsymbol{\theta}}}$ calculated using Eq. (2.24) with ground state energy from the DMRG simulation $E_{0,\mathrm{DMRG}}$. However, to also simultaneously measure pollution from excited states in the reconstructed state $\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$, the relative difference $\langle \hat{H} \rangle_{\psi_{\boldsymbol{\theta}}} - E_{0,\mathrm{DMRG}}$ can be measured in units of the first energy gap

$$\Delta E_{\mathrm{DMRG}} = E_{0,\mathrm{DMRG}} - E_{1,\mathrm{DMRG}}. \tag{2.38}$$

Specifically, we monitor

$$\delta = \left| \frac{\langle \hat{H} \rangle_{\psi_{\boldsymbol{\theta}}} - E_{0,\mathrm{DMRG}}}{\Delta E_{\mathrm{DMRG}}} \right| \tag{2.39}$$

during the training process. The reason for using such a training metric is as follows. The generative model wavefunction can be written as a linear combination of the eigenstates $|\lambda_n\rangle$ of the Hamiltonian in Eq. (2.34) as

$$|\psi_{\boldsymbol{\theta}}\rangle = \sum_{n=0}^{\infty} c_n \, |\lambda_n\rangle \, ,$$

where the coefficients $c_n \geq 0$ and $\sum_n c_n^2 = 1$. The energy difference between this state and the ground state $|\lambda_0\rangle$ with energy E_0 is

$$\langle \psi_{\boldsymbol{\theta}} | \hat{H} | \psi_{\boldsymbol{\theta}} \rangle - E_0 = \sum_{n=1}^{\infty} c_n^2 (E_n - E_0).$$

Alternatively, isolating the first energy difference $E_1 - E_0$, we find that

$$\begin{aligned} \langle \psi_{\boldsymbol{\theta}} | \hat{H} | \psi_{\boldsymbol{\theta}} \rangle - E_0 &= c_1^2 (E_1 - E_0) + \sum_{n=2}^{\infty} c_n^2 (E_n - E_0) \\ \frac{\langle \psi_{\boldsymbol{\theta}} | \hat{H} | \psi_{\boldsymbol{\theta}} \rangle - E_0}{E_1 - E_0} &= c_1^2 + \sum_{n=2}^{\infty} c_n^2 \frac{E_n - E_0}{E_1 - E_0}. \end{aligned}$$

As such, when $\langle \psi_{\boldsymbol{\theta}} | \hat{H} | \psi_{\boldsymbol{\theta}} \rangle \to E_0$ we expect that the largest contribution to the energy difference will be that of the first excited state. Therefore, the definition of δ in Eq. (2.39) serves as a quantifier for excited state contamination in the reconstructed state $|\psi_{\boldsymbol{\theta}}\rangle$. We will henceforth use $\delta \leq 0.05$ as a sensible definition for *successful* learning. For the RBMs trained, when $\delta \leq 0.05$ was reached, it was decided to terminate the training process due to CPU time constraints.[5]

Hyperparameters used to train RBMs and RNNs are given in Tables 2.3-2.6. As with the hyperparameter choices made in Sec. 2.4.2, the hyperparameter choices in this section were made based on laborious preliminary calculations that yielded the best δ values with minimal fine-tuning. To estimate $\langle \hat{H} \rangle_{\psi_{\boldsymbol{\theta}}}$, 10^4 samples were generated from either generative model to monitor the training. For RBMs, 10^4 samples were generated by allowing for 10^4 independent Markov chains (the block-Gibbs sampling procedure) to evolve for $0.75{\times}10^4$ to 10^4 CD_k steps.

[5]Training RNNs on the same datasets took relatively less CPU time compared to RBMs. However, making rigorous comparisons between algorithm training efficiencies is unfair, since the RNNs and RBMs were coded in two different programming languages.

Table 2.3: Hyperparameters used for training RBMs.

Hyperparameter	Value / Type
Optimizer	SGD
Learning rate	0.001, 0.01
Positive batch size	10
Negative batch size	20
k CD_k steps	10
Hidden units	See Table 2.4

Table 2.4: RBM minimum hidden units to reach $\delta \leq 0.05$

N	$R = 1.0$ (# of parameters)	$R = 1.1$ (# of parameters)
4	5 (389)	4 (324)
6	11 (1163)	9 (872)
8	27 (3611)	19 (2321)

Table 2.5: Hyperparameters used for training RNNs.

Hyperparameter	Value / Type
Loss function	Cross Entropy
Optimizer	SGD
RNN recurrent cell	LSTM
Learning rate	0.001
Batch size	10
Hidden units	See Table 2.6

Table 2.6: RNN LSTM hidden units.

N	$R = 1.0$ (# of parameters)	$R = 1.1$ (# of parameters)
4	4 (424)	4 (424)
6	10 (1276)	7 (814)
8	21 (3586)	16 (2416)

Given the trainability issues that were observed in Sec. 2.4.2 for RBMs, it is sensible to first ask what number of hidden units n_h is required to achieve the learning criterion of $\delta \leq 0.05$. Then, an RNN with a similar number of tunable parameters can be employed for a more fair comparison. It is to be expected that for systems with long correlation lengths or high entanglement, more tunable parameters will be required to obtain the desired $\delta \leq 0.05$ criterion [46]. In the context of the Hamiltonian in Eq. (2.34), ground states near the critical point ($R = 1.0$) should require more tunable parameters than for off-critical ground states ($R = 1.1$). Results for systematically increasing n_h in RBMs for system sizes N =4, 6, and 8 until $\delta \leq 0.05$ is reached are shown in Table 2.4. As expected, for each value of N, training RBMs on data from the off-critical system requires a smaller n_h as compared to that of critical systems.

Now knowing the minimum number of hidden units that an RBM requires to reach the learning criterion, RNNs with LSTM recurrent units that roughly have the same number of tunable parameters can be trained on the same datasets for comparison. The specific number of hidden units in an LSTM recurrent cell required to roughly match the number of parameters in the corresponding RBM for each N and R are given in Table 2.6, and the hyperparameters used to train each RNN are shown in Table 2.5. The value of δ as calculated during the training process for RBMs and RNNs is shown for R =1.0 and 1.1 in Figs. 2.10 and 2.11, respectively. As before with the XY model, training RBMs in this context is still more volatile as compared to the RNNs. This is especially noticeable for $R = 1.0$ and $N = 8$. Not only this, but the RNNs in every case are able to reach the learning criterion several thousand training epochs sooner than the corresponding RBMs, and the value of δ for the RNNs reaches a stable equilibrium well below $\delta = 0.05$ in most cases.[6]

The training stability of RNNs is clearly superior to that of RBMs for these values of N and R. In a real-world reconstruction setting where we lack access to the realized quantum state being simulated in an experiment, this training stability is extremely favourable. For the $N = 8$ systems where the value of δ gets worse, this can easily be corrected with an exponential learning rate decay or by trying adaptive optimization methods. Regardless, it is clear that for rather "trivial" hyperparameters and a relatively strict learning criterion, RNNs outperform RBMs in this setting.

As we saw in Sec. 2.4.2, RNNs with or without a symmetry-enforcement scheme implemented in the autoregressive architecture are still able to effectively learn the underlying symmetry. Given the symmetry constraints in the ground state of the Hamiltonian in

[6]The exceptions here are for $N = 8$, where after approximately 2500 training epochs the value of δ actually increases.

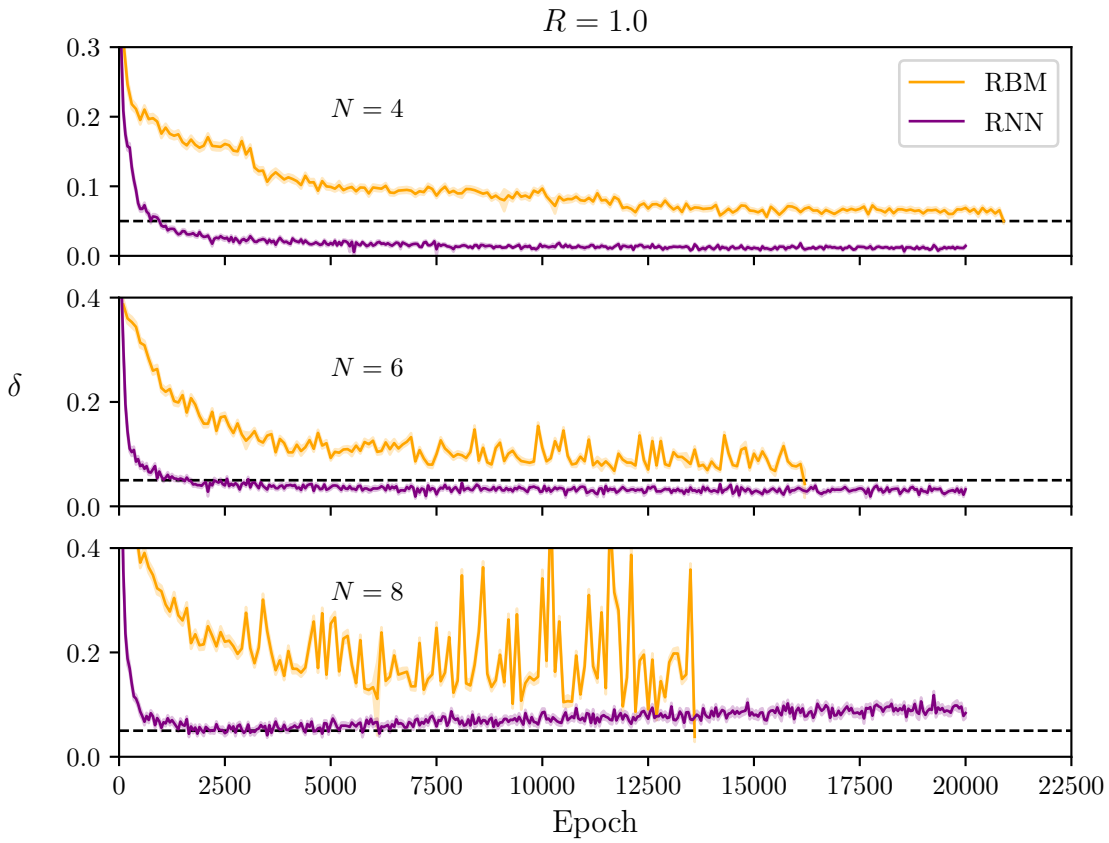

Figure 2.10: Comparing the ground state of NMAs of endofullerene training accuracy of RBMs (orange) and RNNs (purple) for $R = 1.0$ and for system sizes of $N = 4$, 6, and 8. The RBM and RNN hyperparameters are given in Tables 2.3 and 2.5, respectively, and 10^4 samples were generated for each point to estimate $\langle \hat{H} \rangle_{\psi_{\boldsymbol{\theta}}}$ in the definition of δ in Eq. (2.39). The black dashed line corresponds to $\delta = 0.05$.

Eq. (2.34) (total ℓ parity, ℓ_p, and total m), it is a reasonable hypothesis that the RNNs perform better than the RBMs in this setting due to symmetry. By drawing 10^4 independent samples from both generative models, the fraction of samples, f_{NS}, that violate the symmetries $\ell_p = 0$ and $\sum_i m_i = 0$ can be analyzed. More interestingly, however, we may also look at f_{NS} at every block-Gibbs step for the RBM. In doing so, we gain insight into how long the Markov chains take to equilibrate and can therefore make claims pertaining to the RBM's computational efficiency compared to the autoregressive RNN. For a fair comparison, we will use the RNN parameters at the training epoch for which the corre-

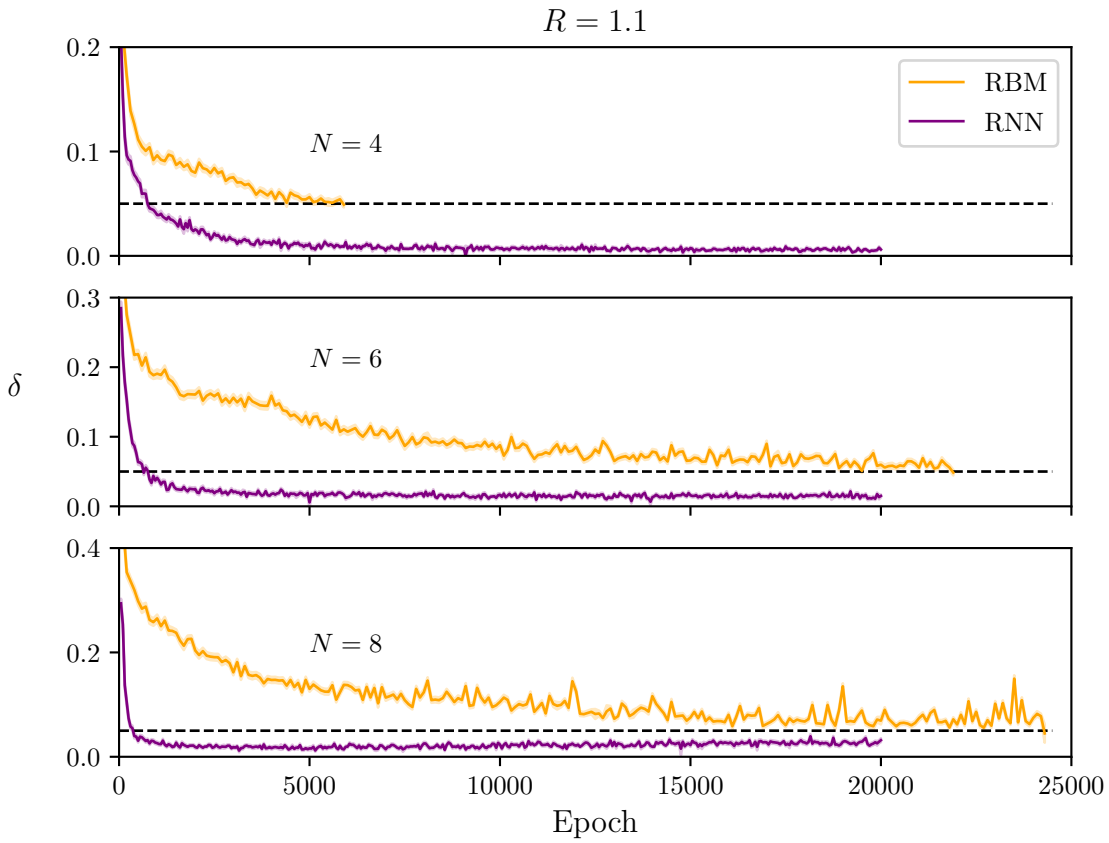

Figure 2.11: Comparing the ground state of NMAs of endofullerene training accuracy of RBMs (orange) and RNNs (purple) for $R = 1.1$ and for system sizes of $N = 4$, 6, and 8. The RBM and RNN hyperparameters are given in Tables 2.3 and 2.5, respectively, and 10^4 samples were generated for each point to estimate $\langle \hat{H} \rangle_{\psi_{\boldsymbol{\theta}}}$ in the definition of δ in Eq. (2.39). The black dashed line corresponds to $\delta = 0.05$.

sponding RBM reached the learning criterion. For the $N = 8$ cases where the RNN's δ value worsened and for the cases where the RBM took more than 20,000 epochs to reach the learning criterion, the RNN parameters that yielded the minimum δ value in the entire training process will be used.

Figs. 2.12 to 2.14 show f_{NS} for both generative models for all values of R and N employed in this section. As was anticipated, the RNN f_{NS} remains at or below that of the RBM for the range of k values for which the Markov chain has certainly reached an

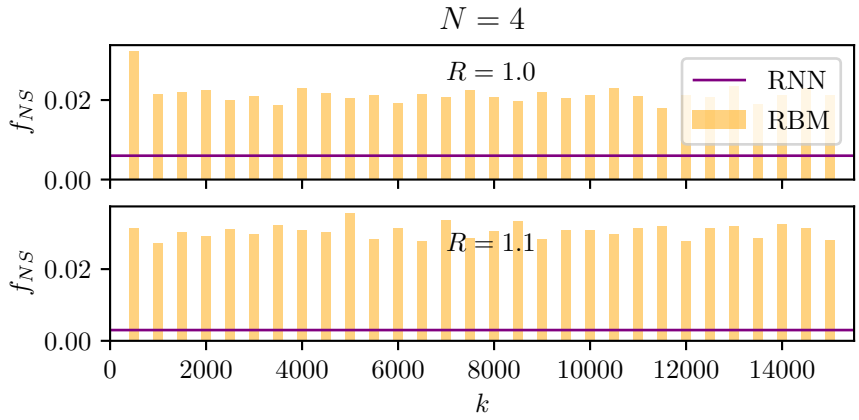

Figure 2.12: The fraction, f_{NS}, of RBM- (orange bars) and RNN-generated (purple line) samples that violate the ground state symmetries of Eq. (2.34) out of 10^4 independent samples for $R = 1.0$ and 1.1 for $N = 4$.

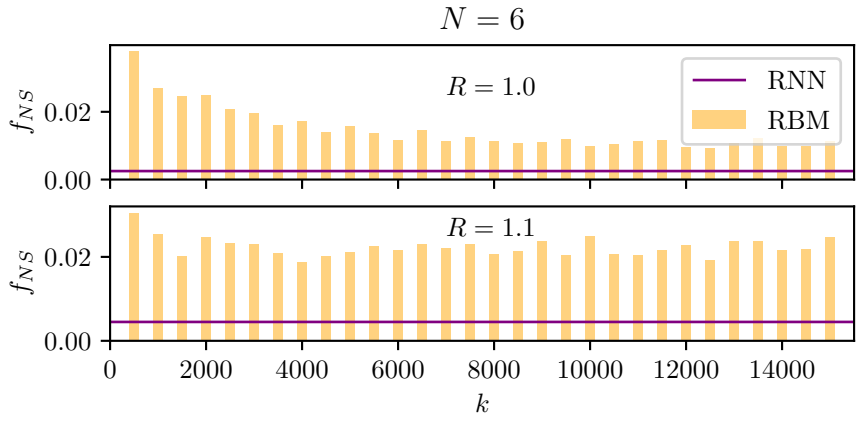

Figure 2.13: The fraction, f_{NS}, of RBM- (orange bars) and RNN-generated (purple line) samples that violate the ground state symmetries of Eq. (2.34) out of 10^4 independent samples for $R = 1.0$ and 1.1 for $N = 6$.

equilibrium. This is in agreement with observations in Sec. 2.4.2, where RNNs were shown to learn the underlying symmetry of the ground state of the XY model quite well. Not only this, but for $N = 6$ and 8 the value of k for which the Markov chain has equilibrated is $\mathcal{O}(10^3)$, in contrast to the perfect-sampling algorithm intrinsic to RNNs.

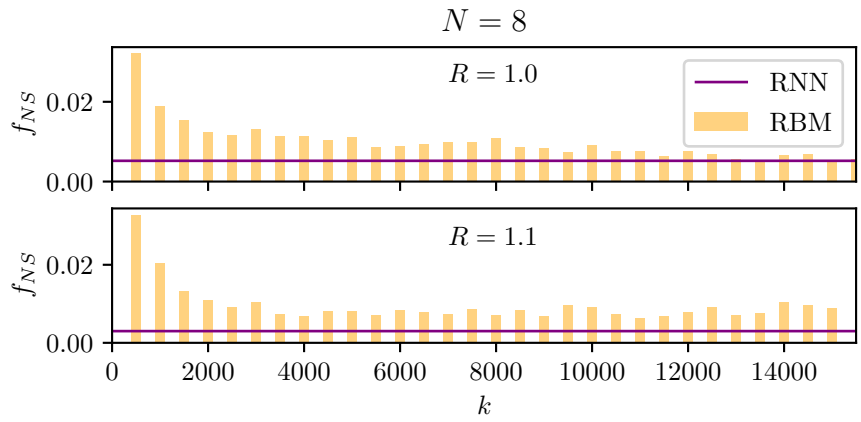

Figure 2.14: The fraction, f_{NS}, of RBM- (orange bars) and RNN-generated (purple line) samples that violate the ground state symmetries of Eq. (2.34) out of 10^4 independent samples for $R = 1.0$ and 1.1 for $N = 8$.

2.6 Preview of Chapter 3

The two learning tasks, reconstructing ground states of the XY model and NMAs of endofullerenes, used for comparing the trainability and expressivity of RBMs and RNNs proved to be quite informative. Both Hamiltonians in Eq. (2.25) and (2.34) have ground states that pose reconstruction tasks unique to quantum many-body physics and have ties to real-world hardware. Still, however, a learning task that is more closely related to today's physical quantum hardware is desired. Synthetic data from numerical simulations might still be required as a proxy for performance on real hardware given that most of today's hardware is not easy to access.

Chapter 3

SSE QMC for Rydberg atoms

3.1 Rydberg atoms

There exists many well-studied experimental platforms in existence today for realizing quantum many-body phenomena, such as trapped ions [19–22], quantum gases [23], photonic simulators [24, 25], and neutral atoms [26, 27] – the platform of focus in this chapter. Neutral atom experiments comprise of atoms (e.g. rubidium), commonly referred to as *Rydberg* atoms, that are trapped and manipulated into specific spatial arrangements via laser light and can be forced to transition between their ground atomic state and an atomic state with a large principal quantum number – a Rydberg state. Rydberg atoms experience a two-body $1/r_{ij}^6$ van der Waals (VDW) interaction, where r_{ij} is the spatial separation between Rydberg atoms i and j, as a product of how they are experimentally prepared [27]. This interaction leads to a phenomenon known as the Rydberg blockade, where the excitation of two Rydberg atoms in close proximity is suppressed [82–84].

The Hamiltonian describing the properties of Rydberg atom arrays is

$$\hat{H} = \frac{\Omega}{2} \sum_{i=1}^{N} \hat{\sigma}_i^x - \delta \sum_{i=1}^{N} \hat{n}_i + \sum_{i<j} V_{ij} \hat{n}_i \hat{n}_j, \tag{3.1}$$

where N is the number of Rydberg atoms. The eigenstates of the operators $\hat{n}_i = |1\rangle\langle 1|_i$, called *occupation* operators, prescribe the natural computational basis as the two atomic energy levels of each Rydberg atom: the ground atomic state and Rydberg state represented by kets $|0\rangle$ and $|1\rangle$, respectively. Specifically, the computational basis is defined by the

eigenequations

$$\hat{n}_i \left|0\right\rangle_j = 0, \tag{3.2a}$$

and

$$\hat{n}_j \left|1\right\rangle_j = \delta_{i,j} \left|1\right\rangle_j . \tag{3.2b}$$

The operator $\hat{\sigma}_i^x = \left|0\right\rangle\!\left\langle 1\right|_i + \left|1\right\rangle\!\left\langle 0\right|_i$ is coupled to the parameter Ω, called the Rabi frequency, which quantifies the atomic energy difference between states $\left|0\right\rangle$ and $\left|1\right\rangle$. The interaction coefficient,

$$V_{ij} = \Omega \left(\frac{R_b}{r_{ij}}\right)^6 , \tag{3.3}$$

depicts the VDW interactions mentioned previously, where R_b is called the blockade radius, $r_{ij} = |\boldsymbol{r}_i - \boldsymbol{r}_j|/a$ is the distance between Rydberg atoms i and j, and a is the lattice spacing. Throughout this thesis, R_b/a is left as a free parameter with $a = 1$. Lastly, the single-body $\hat{n}_i$ term coupling to the parameter δ describes the laser detuning, which also functions as a longitudinal field.

For simulating Eq. (3.1) via QMC methods, potential sign problems must be addressed [16–18]. In its current form, Eq. (3.1) seems sign-problematic in that the off-diagonal $\Omega/2$ term is positive in the Rydberg occupation basis. However, upon unitarily transforming Eq. (3.1) via

$$\hat{U} = \bigotimes_{i=1}^{N} \hat{\sigma}_i^z = \bigotimes_{i=1}^{N} \left[\mathbb{I} - 2\hat{n}_i\right], \tag{3.4}$$

one obtains,

$$\hat{U}^\dagger \hat{H} \hat{U} = -\frac{\Omega}{2} \sum_{i=1}^{N} \hat{\sigma}_i^x - \delta \sum_{i=1}^{N} \hat{n}_i + \sum_{i<j} V_{ij} \hat{n}_i \hat{n}_j, \tag{3.5}$$

where the off-diagonal $\Omega/2$ term is now negative, making Eq. (3.5) sign-problem free.

Experimental demonstrations and characterizations of exotic phases contained within the Rydberg Hamiltonian have been illustrated in Refs. [85,86], and theoretical and numerical explorations are not lagging behind [3,87–92]. Not only this, but numerical techniques are painting an extremely compelling picture outlining novel quantum critical points [91,92] and topologically ordered spin liquid phases [93, 94]. Such numerical studies have so far been limited to employing the density matrix renormalization group (DMRG) [6,7], where approximations to Eq. (3.1) akin to truncations on the Rydberg atom interaction range

are performed to facilitate efficient DMRG calculations. For these reasons, the next logical numerical progression is in developing a corresponding quantum Monte Carlo (QMC) algorithm, where exact sampling and bias inherent to DMRG calculations will (hopefully) be traded favourably for inexact sampling (correlation times and Markov chains) with exact representation. Based on the Stochastic Series Expansion (SSE) framework pioneered by Sandvik [13,95], the presented QMC algorithm in this thesis ushers in the next step in our numerical progression towards a better understanding of Rydberg atoms.

The purpose of Sections 3.2 and 3.3 is to outline the required ingredients for an efficiently functioning Monte Carlo algorithm as outlined in Section 1.3. Namely, formulating weights defined in a sensible configuration space that facilitate sampling, ergodicity and detailed balance, and extracting meaningful observables.

3.2 SSE QMC formalism for ground states

At zero temperature, the expectation value of an operator $\hat{\mathcal{O}}$ given an unnormalized state $|\tilde{\psi}\rangle$ takes the form

$$\langle\hat{\mathcal{O}}\rangle = \frac{\langle\tilde{\psi}|\hat{\mathcal{O}}|\tilde{\psi}\rangle}{\langle\tilde{\psi}|\tilde{\psi}\rangle}. \tag{3.6}$$

If the correspondingly normalized state $|\psi\rangle$ describes a system of N degrees of freedom each with a local Hilbert space size of $\mathcal{D}$, the normalization factor $\langle\tilde{\psi}|\tilde{\psi}\rangle$ requires summing $\mathcal{D}^N$ terms. To avoid this curse of dimensionality, one can devise an importance sampling procedure starting as follows.

Given a Hamiltonian $\hat{H}$ and a computational basis $\{|\alpha\rangle\}$, one can write an arbitrary trial state $|\alpha_r\rangle \in \{|\alpha\rangle\}$ in terms of the eigenstates of $\hat{H}$ $\{|\lambda_m\rangle, m \in 0, 1, \cdots, \mathcal{D}^N - 1\}$, where $\mathcal{D}$ is the local Hilbert space size of each of the system's N degrees of freedom, as

$$|\alpha_r\rangle = \sum_{m=0}^{\mathcal{D}^N-1} c_m |\lambda_m\rangle, \tag{3.7}$$

with $c_m \in \mathbb{C}$. Consider M applications of $(-\hat{H})$ on $|\alpha_r\rangle$,

$$(-\hat{H})^M |\alpha_r\rangle = c_0|E_0|^M \left(|\lambda_0\rangle + \sum_{m=1}^{\mathcal{D}^N-1} \frac{c_m}{c_0} \left(\frac{E_m}{E_0}\right)^M |\lambda_m\rangle \right) \xrightarrow{M\to\infty} c_0|E_0|^M |\lambda_0\rangle, \tag{3.8}$$

where the ground state λ_0 has now been *projected out* of $|\alpha_r\rangle$ in the corresponding $M \to \infty$ limit.[1] In the case that the state of interest is the ground state $|\lambda_0\rangle$, the normalization factor is

$$Z = \langle\lambda_0|\lambda_0\rangle = \langle\alpha_\ell|(-\hat{H})^M(-\hat{H})^M|\alpha_r\rangle\,, \tag{3.9}$$

given a sufficiently large *projector length* $2M$. Here, the trial states $|\alpha_\ell\rangle\,, |\alpha_r\rangle \in \{\,|\alpha\rangle\}$ need not be equal, $|\alpha_\ell\rangle \neq |\alpha_r\rangle$.

To obtain a form for the normalization Eq. (3.9) that is reminiscent of a sum of weights over a configuration space, consider inserting resolutions of the identity in terms of the computational basis states $\{\,|\alpha\rangle\}$ in between each product of $-\hat{H}$ in Eq. (3.9).

$$\begin{aligned}\langle\alpha_\ell|(-\hat{H})^M(-\hat{H})^M|\alpha_r\rangle &= \langle\alpha_\ell|(-\hat{H})\mathbb{I}(-\hat{H})\mathbb{I}\cdots\mathbb{I}(-\hat{H})|\alpha_r\rangle \\ &= \sum_{\{\alpha_1\}}\sum_{\{\alpha_2\}}\cdots\sum_{\{\alpha_{2M-1}\}}\langle\alpha_\ell|(-\hat{H})|\alpha_1\rangle\cdots\langle\alpha_{2M-1}|(-\hat{H})|\alpha_r\rangle \\ &= \sum_{\{\alpha\}}\prod_{p=1}^{2M}\langle\alpha_{p-1}|(-\hat{H})|\alpha_p\rangle \end{aligned} \tag{3.10}$$

Here, $\langle\alpha_0| = \langle\alpha_\ell|$ and $|\alpha_{2M}\rangle = |\alpha_r\rangle$. Furthermore, one can write $\hat{H}$ as a sum,

$$\hat{H} = -\sum_{t,a}\hat{H}_{t,a}, \tag{3.11}$$

of elementary operators $\hat{H}_{t,a}$, where the subscripts t and a refer to the operator *type* (e.g. diagonal or off-diagonal in the computational basis) and the *lattice unit* that comprises $\hat{H}_{t,a}$ (e.g. a two-body or single-body operator), respectively. Employing this,

$$\langle\alpha_\ell|(-\hat{H})^M(-\hat{H})^M|\alpha_r\rangle = \sum_{\{\alpha_p\}}\prod_{p=1}^{2M}\sum_{t,a}\langle\alpha_{p-1}|\hat{H}_{t,a}|\alpha_p\rangle\,, \tag{3.12}$$

or, equivalently,

$$\langle\alpha_\ell|(-\hat{H})^M(-\hat{H})^M|\alpha_r\rangle = \sum_{S_M}\sum_{\{\alpha_p\}}\prod_{p=1}^{2M}\langle\alpha_{p-1}|\hat{H}_{t_p,a_p}|\alpha_p\rangle\,, \tag{3.13}$$

[1] One also assumes that appropriate shifts to the Hamiltonian $\hat{H}$ have been done so as to make E_0 the largest (in magnitude) eigenvalue of $\hat{H}$.

where S_M denotes a sequence of $2M$ elementary operators,

$$S_M = [t_1, a_1], [t_2, a_2], \cdots, [t_{2M}, a_{2M}]. \tag{3.14}$$

At this point, it is helpful to analyze the form of the normalization in Eq. (3.13). For a system in D spatial dimensions, an elevation to $D+1$ dimensions has taken place, where the $+1$ dimension indexed by p can be interpreted as imaginary time. The configuration space where importance sampling takes place is labelled by $\{\alpha\}$ and S_M, and the weights that provide the mechanism for importance sampling are given by

$$W(\{\alpha\}, S_M) = \prod_{p=1}^{2M} \langle \alpha_{p-1} | \hat{H}_{t_p,a_p} | \alpha_p \rangle \,. \tag{3.15}$$

Given this, the matrix elements of the elementary lattice operators $\hat{H}_{t,a}$ in the computational basis *must* be positive in order to interpret Eq. (3.15) as an unnormalized probability distribution. A graphical representation of an example SSE configuration is given in Fig. 3.1. In this diagram, the vertical direction represents the spatial locations of physical sites on a D-dimensional lattice. The state $|\alpha_\ell\rangle = |\alpha_{p=0}\rangle$ is *propagated* from left to right (imaginary time direction) via

$$\hat{H}_{t_p,a_p} |\alpha_{p-1}\rangle \propto |\alpha_p\rangle \,. \tag{3.16}$$

3.2.1 Sampling and calculating observables

In the zero-temperature SSE framework, given an operator sequence S_M, a sample configuration in the computational basis $\{|\alpha\rangle\}$) is obtained by propagating the state $|\alpha_{p=0}\rangle$ all the way to the *middle* imaginary time index M via Eq. (3.16). Conceptually, this can be seen in the projection of the trial state $|\alpha_r\rangle$ in Eq. (3.8), where for a suitably large enough M, $(-\hat{H})^M |\alpha_r\rangle$ recovers the (unnormalized) ground state $|\lambda_0\rangle$ which one wants to sample. Using this, diagonal observables may be calculated similarly to Eq. (2.21).

Expressions for off-diagonal observables are an entirely different story. In general, an observable $\hat{\mathcal{O}}$ can be calculated as

$$\langle \hat{\mathcal{O}} \rangle = \frac{\langle \lambda_0 | \hat{\mathcal{O}} | \lambda_0 \rangle}{\langle \lambda_0 | \lambda_0 \rangle} = \frac{\langle \alpha_\ell | (-\hat{H})^M \hat{\mathcal{O}} (-\hat{H})^M | \alpha_r \rangle}{\langle \alpha_\ell | (-\hat{H})^M (-\hat{H})^M | \alpha_r \rangle}. \tag{3.17}$$

The non-triviality of reducing Eq. (3.17) to something elegant in terms of SSE simulation parameters is clear from this. However, importantly, a derivation of the ground state energy for the Rydberg Hamiltonian using the SSE implementation described in Section 3.3 is offered.

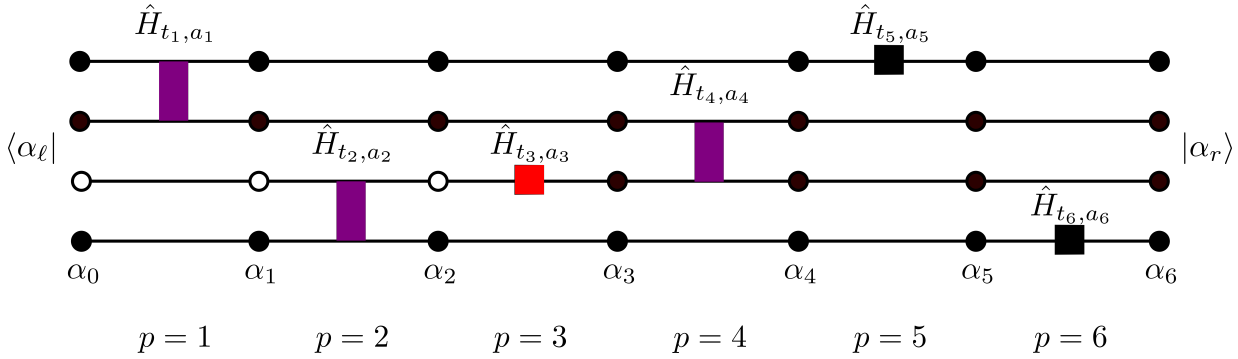

Figure 3.1: A toy example of an SSE configuration. In this example, $M = 3$ and the basis states $\{\alpha_p\}$ are represented by filled/unfilled circles. The elementary operators placed at every imaginary time index are given by purple, red, and black rectangles. Purple and black operators are meant to be diagonal, and therefore do not change the state $|\alpha_p\rangle$ on which they act. The red operator is off-diagonal, since $\alpha_2 \neq \alpha_3$.

3.3 SSE implementation for Rydberg atoms

Having addressed a sign cure to the Rydberg Hamiltonian, Eq. (3.5), an SSE QMC implementation for this Hamiltonian is now feasible. However, in the SSE framework outlined in the previous section, a computational basis $\{|\alpha\rangle\}$ and instructions for defining the specifics of Eq. (3.11) as they pertain to the Rydberg Hamiltonian must be defined. As was mentioned when defining the Rydberg Hamiltonian in Eq. (3.1), the natural computational basis is the Rydberg occupation basis: $\{|\alpha\rangle\} = \{\bigotimes_{i=1}^{N} |n_i\rangle, n_i = 0, 1\}$. How to formulate Eq. (3.11), however, requires extra thought.

Updates to the SSE configuration space labelled by $\{|\alpha\rangle\}$ and S_M have yet to be defined, as there is no "one size fits all" algorithm. In fact, efficient updates to $\{|\alpha\rangle\}$ and S_M must be meticulously conceived for specific SSE implementations, which is the underlying reason why the formulation for Eq. (3.11) is so important. For example, the SSE implementation for the transverse-field Ising model as outlined in Ref. [96] utilizes a non-local and deterministic *cluster* update, while the SSE implementation for the XXZ model utilizes stochastic *directed loops* akin to worm algorithms [97–99]. To summarize, the Hamiltonian and therefore how it is broken into elementary operators $\hat{H}_{t,a}$ drastically affects the update procedure in the $\{|\alpha\rangle\}$, S_M configuration space, and therefore the overall efficiency of the SSE implementation itself.

With an update strategy in mind and given that the Rydberg Hamiltonian Eq. (3.5) takes the form of a quantum Ising model with transverse and longitudinal fields, Refs. [96,

99] motivate the following elementary lattice operators of this SSE implementation as being

$$\hat{H}_{-1,a} = \frac{\Omega}{2}\hat{\sigma}_i^x, \tag{3.18a}$$

$$\hat{H}_{1,a} = \frac{\Omega}{2}\mathbb{I}, \tag{3.18b}$$

and

$$\hat{H}_{1,b} = -V_{ij}\hat{n}_i\hat{n}_j + \delta_b(\hat{n}_i + \hat{n}_j) + C_{ij}, \tag{3.18c}$$

where $t = 1(-1)$ denotes diagonal (off-diagonal) operator types and a and b designate single- and two-body operators, respectively. A reduction to the detuning parameter $\delta_b = \delta/(N-1)$ is required since the sum $\delta\sum_i \hat{n}_i$ has been moved into the sum over pairs $\sum_{i<j}$, and the constant $C_{ij} = |\min(0, \delta_b, 2\delta_b - V_{ij})|$ is required to be added to $\hat{H}_{1,b}$ so that all of its matrix elements remain non-negative.[2] Additionally, a small constant $\varepsilon > 0$ may be included in the definition of C_{ij} to aid numerics [99]. However, in preliminary calculations with this SSE implementation, it was found that the value of ε was negligible to simulation efficiency and therefore $\varepsilon = 0$ was chosen.

Given that the matrix elements of the operators in Eq. (3.18) form the foundation for importance sampling in the $\{|\alpha\rangle\}$ and S_M configuration space, it is helpful to show the exact form of each of their non-zero matrix elements. In the Rydberg occupation basis,

$$\langle 1|\hat{H}_{-1,a}|0\rangle = \langle 0|\hat{H}_{-1,a}|1\rangle = \frac{\Omega}{2}, \tag{3.19a}$$

$$\langle 1|\hat{H}_{1,a}|1\rangle = \langle 0|\hat{H}_{1,a}|0\rangle = \frac{\Omega}{2}, \tag{3.19b}$$

$$W_{ij}^{(1)} \equiv \langle 00|\hat{H}_{1,b}|00\rangle = C_{ij}, \tag{3.19c}$$

$$W_{ij}^{(2)} \equiv \langle 01|\hat{H}_{1,b}|01\rangle = \delta_b + C_{ij}, \tag{3.19d}$$

$$W_{ij}^{(3)} \equiv \langle 10|\hat{H}_{1,b}|10\rangle = \delta_b + C_{ij}, \tag{3.19e}$$

and

$$W_{ij}^{(4)} \equiv \langle 11|\hat{H}_{1,b}|11\rangle = -V_{ij} + 2\delta_b + C_{ij}, \tag{3.19f}$$

where subscripts i, j on matrix elements $W_{ij}^{(1,2,3,4)}$ here contain the spatial location dependence. Fig. 3.2 shows an example of a zero-temperature SSE simulation cell of such an operator breakup in the upper and lower panes.

[2] This must be taken into account when calculating the ground state energy.

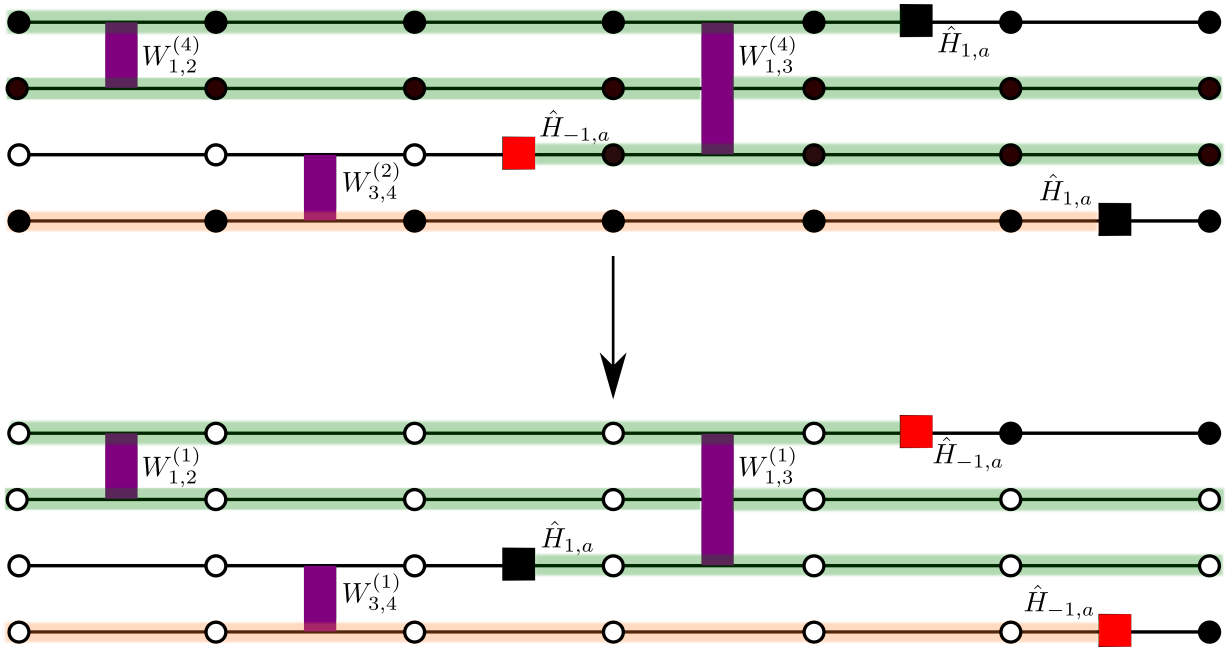

Figure 3.2: Examples of SSE configurations given the operator breakup in Eq. (3.18) with $M = 3$. Here, filled/unfilled circles denote the Rydberg occupation of each of the $N = 4$ sites. The green and orange highlighted regions denote the formation of multibranch and line clusters, respectively. The configuration in the lower pane is obtained by flipping both clusters in the upper pane according to Eq. (3.22).

With this sensible Hamiltonian breakup, one can now outline the specifics of the updates to the $D + 1$-dimensional configurations labelled by $\{|\alpha\rangle\}$ and S_M. In the SSE QMC framework, there are two independent updates to consider: a *diagonal* update and an *off-diagonal/cluster* update, both of which are outlined in Sections 3.3.1 and 3.3.2, respectively. The purpose of the diagonal update in the zero-temperature SSE formalism is to sweep through every imaginary time index p in the SSE simulation cell and replace only elementary diagonal operators with other diagonal operators. In the case of this SSE for Rydberg atoms, this update only involves interchanging operators $\hat{H}_{1,a}$ and $\hat{H}_{1,b}$. In this way, the operator sequence S_M is updated. However, in ignoring off-diagonal operators, $\hat{H}_{-1,a}$ in this case, the computational basis state $\{|\alpha\rangle\}$ is *not* updated, rendering an SSE QMC algorithm void of off-diagonal / cluster updates non-ergodic. This is addressed in Section 3.3.2.

3.3.1 Diagonal update

The procedure for the diagonal update in the zero-temperature SSE formalism is as follows for every imaginary time index $p \in [1, 2M]$.

1. If the operator present at imaginary time index p is one of $\hat{H}_{1,a}$ or $\hat{H}_{1,b}$, remove it and continue to step #2. Otherwise, if the operator is $\hat{H}_{-1,a}$, repeat this step for the next imaginary time index, but propagate the state $|\alpha_p\rangle$ according to Eq. (3.16).

2. Probabilistically choose to *attempt*[3] to either insert $\hat{H}_{1,a}$ at the physical site index i, or $\hat{H}_{1,b}$ at the physical bond (i, j) by sampling the (unnormalized) probability distribution[4]
$$P_{ij} = \begin{cases} \frac{\Omega}{2} & i = j \\ \max\left(W_{ij}^{(1)}, W_{ij}^{(2)}, W_{ij}^{(3)}, W_{ij}^{(4)}\right) & i \neq j \end{cases}. \tag{3.20}$$
In sampling this distribution, one obtains the operator to be inserted corresponding to a given matrix element in Eq. (3.19). Its insertion will be attempted in the following steps at the spatial location (i, j) in the current imaginary time index p ($\hat{H}_{1,a}$ if $i = j$ or $H_{1,b}$ if $i \neq j$).

3. If $\hat{H}_{1,a}$ is chosen, its insertion at site i is accepted.[5] Repeat step #1 at the next imaginary time index and propagate the basis state according to Eq. (3.16). Otherwise, if $\hat{H}_{1,b}$ is chosen, accepting its insertion must still be determined. Proceed to the next step.

4. In the Rydberg occupation basis, the configuration at the current imaginary time index p is given by $|n_{1,p}, n_{2,p}, \cdots, n_{N,p}\rangle$. If $|n_{i,p}, n_{j,p}\rangle$ matches the sampled matrix element of $\hat{H}_{1,b}$ in step #2 (i.e. one of $W_{ij}^{(1)}, W_{ij}^{(2)}$, $W_{ij}^{(3)}$, or $W_{ij}^{(4)}$), the insertion is accepted. Otherwise, the insertion of $\hat{H}_{1,b}$ at location (i, j) is accepted with probability
$$\frac{W_{ij}^{(\text{actual})}}{W_{ij}^{(\text{sampled})}}, \tag{3.21}$$
where $W_{ij}^{(\text{actual})} = \langle n_{i,p}, n_{j,p}|\hat{H}_{1,b}|n_{i,p}, n_{j,p}\rangle$ and $W_{ij}^{(\text{sampled})}$ was the resulting sample from step #2.

[3] This step encompasses the *selection* probability g in Eq. (1.13).
[4] One can sample the distribution in Eq. (3.20) in $\mathcal{O}(1)$ time using the Alias method [100–102].
[5] This step encompasses the *acceptance* probability A in Eq. (1.13).

5. Repeat step #1 at the next imaginary time index and propagate the basis state according to Eq. (3.16).

The key steps in this diagonal update procedure are steps #3 and #4, where Sandvik suggested in Ref. [96] that this is the most efficient way to sample non-uniform diagonal operator matrix elements while satisfying detailed balance.

3.3.2 Cluster updates

Since the diagonal update procedure in Section 3.3.1 ignores the presence of off-diagonal operators $\hat{H}_{-1,a}$, it alone does not yield an ergodic QMC algorithm. Another update is required, and one can consider using the cluster update devised by Sandvik called the *multibranch* cluster update, which is described in Refs. [96, 103, 104]. Originally designed for the SSE implementation of the transverse-field Ising model, the multibranch update is a *deterministic*, highly non-local update – in spatial and imaginary time dimensions – that proceeds as follows.

Employing a graph-based vocabulary, matrix elements in Eq. (3.19) can be pictured as *vertices* in a graph (the SSE simulation cell itself). The vertices from matrix elements of $\hat{H}_{-1/1,a}$ have two *legs* (one leg for the Rydberg state of the ket and same for the bra). Likewise, the vertices from matrix elements of $\hat{H}_{1,b}$ have four legs (two legs for the Rydberg states of the ket and same for the bra). To form a multibranch cluster, perform the following steps.

1. Choose a random leg of any vertex present in the SSE simulation cell as a starting point. If the starting leg belongs to a vertex from $\hat{H}_{1,b}$, add its three remaining legs to the cluster and traverse away (in the imaginary time direction) from it in all four directions. If the starting leg belongs to a vertex from $\hat{H}_{-1/1,a}$, proceed away from the vertex in the imaginary time direction.

2. If a leg from a $\hat{H}_{1,b}$ vertex is encountered, all four of its legs are added to the cluster, and the cluster continues to traverse through the simulation cell by branching out of each of the three exit legs. If a leg from a $\hat{H}_{-1/1,a}$ vertex is encountered, add this leg to the cluster and stop traversing in this direction (this "branch" of the cluster terminates). If the simulation cell edge is encountered (i.e. $|n_1, n_2, \cdots, n_N\rangle_\ell$ or $|n_1, n_2, \cdots, n_N\rangle_r$), add this leg to the cluster and stop traversing in this direction (this "branch" of the cluster terminates).

3. Repeat step #2 until every branch of the cluster has terminated.

Fig. 3.2 shows an example of a ground state projector SSE simulation cell where a multibranch cluster is pictured by the green region in the upper pane. To update a multibranch cluster, all contained legs (Rydberg occupations) are "flipped" and the vertex types within the cluster are correspondingly changed. To satisfy detailed balance when flipping clusters, the flipping probability is given by the Metropolis probability

$$P_{\text{flip}} = \min\left(1, \frac{\mathcal{W}'}{\mathcal{W}}\right), \tag{3.22}$$

where $\mathcal{W}$ is weight of the entire multibranch cluster defined as the product of vertex weights $W(v_i)$ (matrix elements whose values $W(v_i)$ are found in Eq. (3.19)) belonging to the cluster $\mathcal{C}$:

$$\mathcal{W} = \prod_{v_i \in \mathcal{C}} W(v_i). \tag{3.23}$$

Here, $\mathcal{W}'$ is the correspondingly flipped cluster weight. A diagrammatic perspective of this flipping process is shown in Fig. 3.2. The multibranch cluster (green) possesses a weight of

$$\mathcal{W}_{\text{multibranch}} = W^{(4)}_{1,2} \langle 0|\hat{H}_{-1,a}|1\rangle \, W^{(4)}_{1,3} \langle 1|\hat{H}_{1,a}|1\rangle \tag{3.24a}$$

and

$$\mathcal{W}'_{\text{multibranch}} = W^{(1)}_{1,2} \langle 0|\hat{H}_{1,a}|0\rangle \, W^{(1)}_{1,3} \langle 0|\hat{H}_{-1,a}|1\rangle \tag{3.24b}$$

before and after the flip, respectively. Also note that although states $\alpha_{p=0}$ and $\alpha_{p=2M}$ have changed, choosing to initialize these arbitrary states to $\bigotimes_{i=1}^{N} \frac{1}{\sqrt{2}}(\,|0\rangle_i + |1\rangle_i)$ (i.e. randomly initialized) results in no local weight changes at the simulation cell boundaries.

Multibranch cluster updates are the most efficient updates for the transverse-field Ising model SSE implementation for several reasons. Inherent to the model's $\mathbb{Z}_2$ symmetry, multibranch cluster weights are the same before and after flipping, allowing clusters to be flipped according to a Swendsen-Wang procedure [105]. However, that is not the case in the Rydberg Hamiltonian Eq. (3.1) owing to the form of the interactions $\hat{n}_i\hat{n}_j$ and the laser detuning δ terms. Although this reason alone is not a reason for searching for other cluster update algorithms, the multibranch weight changes can be problematic. For many R_b, Ω, and δ combinations, one of the matrix element weights $\{W^{(1-4)}_{i,j}\}$ will be zero, which prohibits a move from its higher-weighted counterpart according to Eq. (3.22) impossible.

For example, in Section 3.4 we show results for $R_b = 1.2$, $\Omega = 1$, and $\delta = 1.1$ for $N = 256$ Rydberg atoms on a square lattice. From this,

$$W^{(1)}_{i,i+1} = W^{(2)}_{i,i+1} = W^{(3)}_{i,i+1} \approx 2.98 \tag{3.25a}$$

and

$$W^{(4)}_{i,i+1} = 0. \tag{3.25b}$$

Given this, $W^{(1)}_{i,i+1}$'s presence in an SSE simulation cell is a likely occurrence. However, a cluster containing $W^{(1)}_{i,i+1}$ cannot be flipped since its counterpart $W^{(4)}_{i,i+1} = 0$, leading to $\mathcal{W}' = 0$. As such, ergodicity issues from the multibranch update may arise since some clusters won't be able to be updated. For this reason, searching for better cluster updates is warranted.

A modification to the multibranch cluster algorithm was formulated called the *line* cluster update in Ref. [3]. Briefly, a line cluster is a local-in-space and non-local-in-imaginary-time cluster inspired by Ref. [106]. Line clusters, like multibranch clusters, terminate on site vertices but do not add all three remaining legs of an $\hat{H}_{1,b}$ vertex to the cluster. Rather, only the adjacent leg (in imaginary time) of an $\hat{H}_{1,b}$ vertex is added to the cluster. The mechanism for flipping such clusters is also given by Eq. (3.22). A diagrammatic perspective of a line cluster's formation and flipping procedure is also given in Fig. 3.2. For instance, the orange line cluster in Fig. 3.2 has a weight

$$\mathcal{W}_{\text{line}} = W^{(2)}_{3,4} \langle 1|\hat{H}_{1,a}|1\rangle$$

and

$$\mathcal{W}'_{\text{line}} = W^{(1)}_{3,4} \langle 0|\hat{H}_{-1,a}|1\rangle$$

before and after it is flipped, respectively.

An advantage of the line update over the multibranch update can be seen in the following example of an impossible move for the multibranch update, where $W^{(1)}_{i,i+1} \to W^{(4)}_{i,i+1}$ is not allowed since $W^{(4)}_{i,i+1} = 0$. With the line update, $W^{(1)}_{i,i+1}$ can instead be flipped to $W^{(2 \text{ or } 3)}_{i,i+1} \approx 2.98$. A diagram of this distinct difference between the two updates is shown in Fig. 3.2. Importantly, many line updates are performed in a single MC step by selecting a random site operator leg as a starting point and continuing in the imaginary time direction until another site operator leg is reached. Therefore, transitions akin to $W^{(1)}_{i,i+1} \to W^{(4)}_{i,i+1}$ can still be proposed, but via two independent line updates (e.g. $W^{(1)}_{i,i+1} \to W^{(2)}_{i,i+1}$ then $W^{(2)}_{i,i+1} \to W^{(4)}_{i,i+1}$).

3.3.3 Directed loop updates

Although employing the line and multibranch cluster updates, as they apply to this SSE implementation for Rydberg atoms, seems very reasonable given where they were inspired from, another prominent off-diagonal update scheme for SSE simulations called *directed loops* should be investigated as a viable candidate for completeness. Directed loops are similar to worm algorithms [97, 107], which are non-deterministic and non-local clusters, and have been successfully demonstrated to be the most efficient cluster update for the SSE implementation of the famous XXZ model [99, 108]. Given the non-triviality of forming such clusters in $D+1$ dimensions, ensuring that detailed balance and ergodicity remain satisfied will be different from one SSE implementation to the other. In this section, it will be demonstrated that directed loops are not an efficient choice of update for the SSE implementation for Rydberg atoms.

To alleviate some of the difficulty in showcasing directed loops for the Rydberg atom SSE in this thesis, we will instead employ a transverse-field Ising model with a longitudinal field (LTFIM),

$$\hat{H} = -h_x \sum_{i=1}^{N} \hat{\sigma}_i^x - h_z \sum_{i=1}^{N} \hat{\sigma}_i^z - J \sum_{\langle i,j \rangle} \hat{\sigma}_i^z \hat{\sigma}_j^z, \tag{3.26}$$

which describes nearest-neighbour interacting spin-1/2 degrees of freedom in the presence of transverse (h_x) and longitudinal (h_z) fields. Note that J, h_x, and $h_z > 0$ and the operators $\hat{\sigma}^x$ and $\hat{\sigma}^z$ are the traditional Pauli operators. The LTFIM Hamiltonian is eerily similar to the Rydberg Hamiltonian in Eq. (3.1). As such, one can decompose the LTFIM Hamiltonian into elementary lattice operators in a very similar fashion to what was done for the Rydberg SSE implementation. Specifically,

$$\hat{H}_{-1,a} = h_x \hat{\sigma}_i^x, \tag{3.27a}$$

$$\hat{H}_{1,a} = h_x, \tag{3.27b}$$

and

$$\hat{H}_{1,b} = J \hat{\sigma}_i^z \hat{\sigma}_j^z + h_{z,b}(\hat{\sigma}_i^z + \hat{\sigma}_j^z) + C, \tag{3.27c}$$

where $h_{z,b} = h_z/(N-1)$ is the modified longitudinal field strength to account for when the sum over sites is put into the sum over pairs, $C = \max(J, 2h_{z,b} - J) + \varepsilon$ is a constant to alleviate sign problems with $H_{1,b}$, and ε is a small non-negative constant to aid with numerics.

The relevant computational basis here is the eigenbasis of the operator $\hat{\sigma}^z$, whose eigenequations are $\hat{\sigma}^z |\uparrow\rangle = |\uparrow\rangle$ and $\hat{\sigma}^z |\downarrow\rangle = -|\downarrow\rangle$. In this basis, the non-zero matrix elements of all elementary lattice operators in Eq. (3.27) are

$$\langle\uparrow|\hat{H}_{-1,a}|\downarrow\rangle = \langle\downarrow|\hat{H}_{-1,a}|\uparrow\rangle = h_x, \tag{3.28a}$$

$$\langle\uparrow|\hat{H}_{1,a}|\uparrow\rangle = \langle\downarrow|\hat{H}_{1,a}|\downarrow\rangle = h_x, \tag{3.28b}$$

$$W_1 \equiv \langle\uparrow\uparrow|\hat{H}_{1,b}|\uparrow\uparrow\rangle = J + 2h_{z,b} + C, \tag{3.28c}$$

$$W_2 \equiv \langle\downarrow\downarrow|\hat{H}_{1,b}|\downarrow\downarrow\rangle = J - 2h_{z,b} + C, \tag{3.28d}$$

and

$$W_3 \equiv \langle\downarrow\uparrow|\hat{H}_{1,b}|\downarrow\uparrow\rangle = \langle\uparrow\downarrow|\hat{H}_{1,b}|\uparrow\downarrow\rangle = -J + C. \tag{3.28e}$$

These matrix elements define the update probabilities involved in the diagonal update portion of the entire SSE configuration space update. The diagonal update for an LTFIM SSE implementation such as this would follow what is outlined in Sec. 3.3.1 with corresponding modifications.

Going back to the previously-employed graphical vocabulary to describe multibranch and line cluster formation, directed loops are formed by starting at a random leg of a vertex in the $D + 1$ SSE simulation cell and traversing in the imaginary time direction to the next closest leg. Whether this next leg is to be added to the directed loop cluster is a question for detailed balance to handle. However, the smoking gun for directed loop inefficiencies here will manifest when vertices from matrix elements of $\hat{H}_{1,b}$ (W_1, W_2, and W_3) are encountered.

When any of the vertices W_{1-3} are encountered in forming a directed loop cluster, the two legs in consideration are the "entrance" leg $e \in [1, 2, 3, 4]$ (four possible legs from the two spin states in the ket and bra each), and the "exit" leg $x \in [1, 2, 3, 4]$. If the four spin states surrounding the vertex are condensed into a label s, Syljuasen and Sandvik showed in Ref. [99] that the detailed balance principle that guides which leg x to exit from is given by

$$\sum_x \omega(s, e, x) = W_s \tag{3.29a}$$

and

$$\omega(s, \ell_1, \ell_2) = \omega(s', \ell_2, \ell_1), \tag{3.29b}$$

where ω are unnormalized weights that are currently unknown, $W_s \in W_{1-3}$, and s' denotes a correspondingly flipped spin configuration from the spins on legs e and x being flipped. Eq. (3.29) defines a set of non-linear equations in ω that, when solved, will provide a prescription for an algorithm that satisfies detailed balance.

What must now be considered is all possible directed loop segments passing through vertices W_{1-3} given an entrance leg e in order to define Eq. (3.29a) concretely. Consider the vertex W_2 in Eq. (3.28d) whose four spin states are all spin-down. Given that the update proposed will be to flip spins on legs e and x, the exit leg x can either be the adjacent leg in imaginary time in the same spatial location ("continue straight") or the entrance leg e ("bounce"). In this way, if the update is accepted and $s \to s'$, one obtains a non-zero vertex weight W_3 ("continue straight") or W_2 ("bounce"). Fig. 3.3 shows all options pictorially, where the imaginary time direction is in the vertical direction, the spatial dimension is the horizontal direction, filled/unfilled circles denote spin-up/down states, and the black bar represents the operator $\hat{H}_{1,b}$. Notice that for the option to "switch and continue" from the spatial exit leg position of the entrance leg, the correspondingly flipped loop segment would yield a vertex with weight $\langle\uparrow\downarrow|\hat{H}_{1,b}|\downarrow\uparrow\rangle = 0$. Therefore, probabilistically this is not allowed. The same can be said for the "switch and reverse" loop segment, which would yield a vertex with weight $\langle\uparrow\uparrow|\hat{H}_{1,b}|\downarrow\downarrow\rangle = 0$.

In doing this exercise for every vertex W_{1-3}, Eq. (3.29a) can now be defined for every $W_s \in W_{1-3}$. All possible directed loop segments for vertices W_{1-3} are shown in Fig. 3.5, but for the example in the previous paragraph one can extract a directed loop equation for W_2 by summing the weights ω that correspond to allowed directed loop segments through W_2 ("continue straight" and "bounce"),[6]

$$W_2 = \omega(2,2,4) + \omega(2,2,2). \tag{3.30}$$

Having done this for every vertex W_{1-3}, the relation in Eq. (3.29b) enforces equivalence between directed loop segments where the entrance and exit legs are interchanged and the vertex spin state s is correspondingly flipped. An example of two such directed loop segments is shown in Fig. 3.4 for the "continue straight" directed loop segments passing through vertices W_2 and W_3. In this case, Eq. (3.29b) is $\omega(2,2,4) = \omega(3,4,2)$. For "bounce" loop segments, they are mapped to themselves under Eq. (3.29b) since the entrance and exit legs are the same.

In using Eq. (3.29b), the directed loop equations akin to Eq. (3.30) can be simplified. At this point, all directed loop equations defined by Eq. (3.29a) must be analyzed to come

[6]The vertex leg enumeration scheme taken here is leg #1 = upper left, leg #2 = upper right, leg #3 = lower left, and leg #4 = lower right.

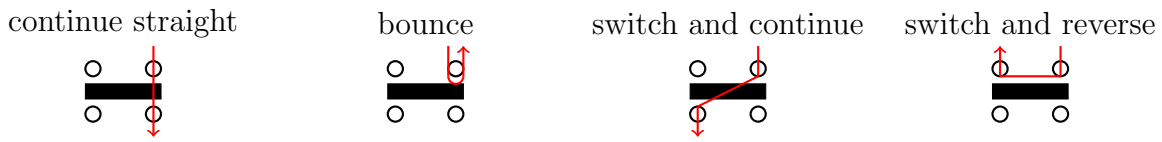

Figure 3.3: Possible directed loop segments through the vertex W_2 defined in Eq. (3.28d). Here, unfilled circles denote spin-down states. The "continue straight" and "bounce" segments, if flipped, will change the vertex weight W_2 to W_3 or back to W_2, respectively. However, the "switch and continue" loop segment would yield a vertex weight of $\langle\uparrow\downarrow|\hat{H}_{1,b}|\downarrow\uparrow\rangle$, which is zero and therefore not allowed. The same can be said for the "switch and reverse" loop segment, which would yield a vertex with weight $\langle\uparrow\uparrow|\hat{H}_{1,b}|\downarrow\downarrow\rangle = 0$.

Figure 3.4: An example of two "continue straight" directed loop segments through vertices W_2 and W_3 that must be related by Eq. (3.29b). Mathematically, this corresponds to $\omega(2,2,4) = \omega(3,4,2)$.

to a solution for the weights ω. Motivated by Ref. [99], Fig. 3.5 shows all possible directed loop segments for all possible vertex configurations divided into four subsets.[7] Each subset contains two different vertices in two rows that have the same entrance leg e, while each configuration in a given row maps, via Eq. (3.29b), to another configuration in the subset. Note again that "bounce" loop segments map to themselves.

Given that each of the four subsets in Fig. 3.4 will yield two directed loop equations from each row, as prescribed by Eq. (3.29a), there will be eight directed loop equations. However, there are two symmetries to take advantage of to reduce the number of equations:[8]

1. permutation of two spins acted on by $\hat{H}_{1,b}$ (i.e. interchanging spins while keeping the orientation of the loop segment) – this relates the upper-left and lower-right subsets – and

2. imaginary time symmetry (swap spins above and below the black bar that represents $\hat{H}_{1,b}$) – in tandem with the previous point, this relates the lower-left and upper-right subsets.

[7] There are technically four more subsets to consider, but they are related to the four subsets in Fig. 3.5 by flipping all of the spins.

[8] We already used spin-flip symmetry to reduce the number of subsets from eight to four!

In utilizing these symmetries, the number of subsets to consider is reduced to only the upper-left and lower-left subsets in Fig. 3.5. Also, in using Eq. (3.29b),

$$\omega(3,4,2) = \omega(2,2,4) = a$$

and

$$\omega(3,2,4) = \omega(1,4,2) = d.$$

In all, the directed loop equations for the upper-left quadrant are

$$W_3 = \omega(3,4,4) + \omega(3,4,2) = b_1 + a = -J + C \tag{3.32a}$$

and

$$W_2 = \omega(2,2,4) + \omega(2,2,2) = a + b_2 = J - 2h_{z,b} + C, \tag{3.32b}$$

and for the lower-left quadrant

$$W_3 = \omega(3,2,2) + \omega(3,2,4) = b_3 + d = -J + C \tag{3.32c}$$

and

$$W_1 = \omega(1,4,2) + \omega(1,4,4) = d + b_4 = J + 2h_{z,b} + C, \tag{3.32d}$$

where the variables b_i are used to denote the corresponding "bounce" weights.

The "best" solution to this system of equations is one where the "bounce" weights b_{1-4} are zero. Since "bounce" loop segments when updated return the same configuration (i.e. they do nothing), ergodicity issues may manifest as a result of "doing nothing" too often. Again, drawing inspiration from Ref. [99], consider a "heat bath" solution to the system of four equations,

$$\frac{b_1}{W_3} = \frac{W_3}{W_3 + W_2}, \tag{3.33a}$$

$$\frac{b_2}{W_2} = \frac{W_2}{W_3 + W_2}, \tag{3.33b}$$

$$\frac{a}{W_3} = \frac{W_2}{W_3 + W_2}, \tag{3.33c}$$

$$\frac{b_3}{W_3} = \frac{W_3}{W_1 + W_3}, \tag{3.33d}$$

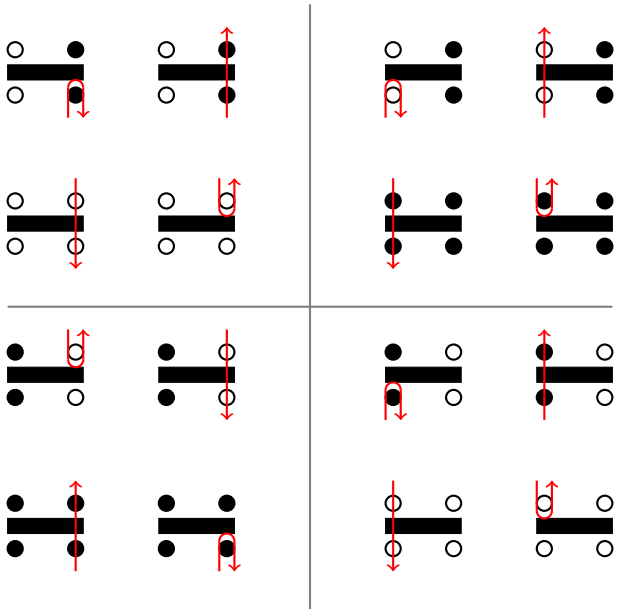

Figure 3.5: Directed loop segments for vertex configurations divided into four subsets. Each subset is organized by containing two different vertices in two rows that have the same entrance leg e, while also containing configurations that map, via Eq. (3.29b), to one and another. For instance, in the upper left subset, the "continue straight" segments are those related loop segments from Fig. 3.4.

$$\frac{b_4}{W_1} = \frac{W_1}{W_1 + W_3}, \tag{3.33e}$$

and

$$\frac{d}{W_3} = \frac{W_1}{W_1 + W_3}. \tag{3.33f}$$

In certain LTFIM Hamiltonian parameter regimes, W_2 or W_3 will be zero.[9] While this does favourably destroy "bounce" weights b_2, b_3, or b_1, it unfavourably destroys the "continue straight" weights a or d. Consider when $W_3 = 0$. This gives

$$\begin{aligned} b_1 &= 0, \\ b_2 &= W_2^2, \\ a &= 0, \\ b_3 &= 0, \\ b_4 &= W_1^2, \end{aligned}$$

and

$$d = 0.$$

Evidently, "bounce" loop segments are only possible. Clearly, the heat-bath solutions to the directed loop equations in Eq. (3.32) are not optimal.

Let us now turn to generally analyzing how b_{1-4} can be minimized while solving Eq. 3.32. Also note that to interpret b_{1-4}, a, and d as unnormalized probabilities, they must all be positive. In solving for b_{1-4} in Eq. (3.32), one obtains

$$b_1 = -a - J + C, \tag{3.35a}$$
$$b_2 = -a + J - 2h_{z,b} + C, \tag{3.35b}$$
$$b_3 = -d - J + C, \tag{3.35c}$$

and

$$b_4 = -d + J + 2h_{z,b} + C. \tag{3.35d}$$

The lines in the $b_{1,2}$ vs a plane are parallel and will never intersect, and similarly for the lines in the $b_{3,4}$ vs d plane. Therefore, $b_{1-4} = 0$ can never occur simultaneously, and inefficient "bounce" loop segments will persist.

[9] This is enforced by the constant C in the definition of $\hat{H}_{1,b}$. We are also taking $\varepsilon = 0$.

Given the inability to determine solutions to the directed loop equations that are optimal, that is at least most of b_{1-4} being zero while a and d remaining non-zero, directed loops for off-diagonal updates in this SSE implementation for the LTFIM Hamiltonian in Eq. (3.26) should not be used over the seemingly efficient and deterministic *line* cluster update outlined in Sec. 3.3.2. This also reasonably applies to the Rydberg atom SSE implementation showcased in Sec. 3.3 given its similarity to the LTFIM SSE implementation presented here.

3.3.4 Ground state energy estimator

In Section 3.2.1, the difficulty in calculating a generic off-diagonal observable in the SSE formalism was alluded to. However, being able to estimate an off-diagonal observable like the energy in an SSE simulation is quite valuable. Recall that the ground state SSE formalism's foundation is in projecting out the (unnormalized) ground state from an arbitrary trial state (Eq. 3.8). It is from this principle that one formulates an importance sampling procedure for the normalization in Eq. 3.9. Upon closer inspection, given a sufficiently large M,

$$\begin{aligned} Z =& \langle \alpha_\ell | (-\hat{H})^M (-\hat{H})^M | \alpha_r \rangle \\ \approx& \langle \lambda_0 | |c_0|^2 E_0^{2M} | \lambda_0 \rangle \\ \propto& E_0^{2M}. \end{aligned}$$

Therefore, the accuracy of the ground state energy estimate is in direct correspondence to the object, Z, that one importance samples, and therefore serves as a vital benchmark for the SSE algorithm itself.

In substituting $\hat{H} = \hat{\mathcal{O}}$ in Eq. 3.17, the goal is to find a compact expression for the ground state energy,

$$\langle \hat{H} \rangle = E_0 = \frac{1}{Z} \langle \alpha_\ell | (-\hat{H})^M \hat{H} (-\hat{H})^M | \alpha_r \rangle \,, \tag{3.36}$$

in terms of parameters in the SSE simulation cell. As the specific form of the Hamiltonian will drastically affect such a derivation, the one presented in this section only applies to an SSE implementation where one of the local operators $\hat{H}_{t,a}$ from Eq. (3.11) is a scalar multiple of unity, $h\mathbb{I}$, that acts on one degree of freedom (a single-body term). Such an operator will be represented as $\hat{H}_{t,a} = \hat{H}_{1,a} = h\mathbb{I}$. Crucially, this applies to the Rydberg SSE implementation presented in this thesis with $h = \Omega/2$ (see Eq. (3.18b)), but also applies to the SSE implementation for the transverse-field Ising model [96].

In the $D+1$ SSE simulation cell, $\hat{H}_{1,a}$ does not alter the basis states $|\alpha_p\rangle$ for which it acts on. Therefore, in the summation over all possible operator strings S_M in Eq. (3.13), one can account for degenerate terms that contain m instances of $\hat{H}_{1,a}$. Given a sequence of operators of length $2M$ (the aforementioned *projector length*), let $\tilde{M} = 2M - m$ represent the "reduced" projector length having removed all $\hat{H}_{1,a}$ operators. There are three simplifications to be aware of.

1. The weight of an SSE configuration with m $\hat{H}_{1,a}$ operators present will receive a contribution of h^m specifically from these operators:

$$\prod_{p=1}^{2M} \langle \alpha_{p-1}|\hat{H}_{t_p,a_p}|\alpha_p\rangle = h^m \prod_{p=1}^{\tilde{M}} \langle \alpha_{p-1}|\hat{H}_{t_p,a_p}|\alpha_p\rangle \,.$$

2. SSE configuration weights are independent of where, *spatially*, each $\hat{H}_{1,a}$ operator is at a given imaginary time index p. This results in a degeneracy factor of N^m, as each of the m operators can be placed in N difference spatial locations.

3. SSE configuration weights are also independent of where, in imaginary time, each $\hat{H}_{1,a}$ operator is for a given spatial location. This results in a combinatorial degeneracy factor of $\binom{2M}{m}$.

Consider the operator sequence $S_{\tilde{M}}$, which is void of $\hat{H}_{1,a}$ occurrences. In replacing S_M with $S_{\tilde{M}}$, SSE configuration space weights are now be labelled by $\{\alpha\}$, $S_{\tilde{M}}$, and m. Using the three simplifications mentioned in the list above, the normalization Z written in this new space is

$$Z = \sum_{\{\alpha\}} \sum_{S_{\tilde{M}}} \sum_{m} N^m \frac{(2M)!}{\tilde{M}!m!} h^m \prod_{p=1}^{\tilde{M}} \langle \alpha_{p-1}|\hat{H}_{t_p,a_p}|\alpha_p\rangle \,.$$

Consider the change of variables $q = m + 1$. Under this change of variables,

$$\tilde{M} \Rightarrow \tilde{M},$$

and the new projector length is

$$2M \Rightarrow 2Q \equiv 2M + 1 = \tilde{M} + q.$$

Applying this to the normalization,

$$Z = \sum_{\{\alpha\}} \sum_{S_{\tilde{M}}} \sum_{q} N^{q-1} \frac{(2Q-1)!}{\tilde{M}!(q-1)!} h^{q-1} \prod_{p=1}^{\tilde{M}} \langle \alpha_{p-1} | \hat{H}_{t_p,a_p} | \alpha_p \rangle \,.$$

To unclutter the summand, let

$$\Phi(\alpha, S_{\tilde{M}}, q) = N^q \frac{(2Q)!}{\tilde{M}!q!} h^{q-1} \prod_{p=1}^{\tilde{M}} \langle \alpha_{p-1} | \hat{H}_{t_p,a_p} | \alpha_p \rangle \,, \tag{3.37}$$

which yields a compact expression for the normalization,

$$Z = \frac{1}{2Q \times N} \sum_{\{\alpha\}} \sum_{S_{\tilde{M}}} \sum_{q} \Phi(\alpha, S_{\tilde{M}}, q) q.$$

Given this compact expression for the normalization, let us turn towards simplifying $\langle \alpha_\ell | (-\hat{H})^M \hat{H} (-\hat{H})^M | \alpha_r \rangle$ in a similar fashion. One can think of this expression simply as an SSE normalization Z with a projector length $2M+1$ instead of $2M$ due to the presence of the additional $\hat{H}$,

$$\begin{aligned} \langle \alpha_\ell | (-\hat{H})^M \hat{H} (-\hat{H})^M | \alpha_r \rangle &= - \langle \alpha_\ell | (-\hat{H})^M (-\hat{H}) (-\hat{H})^M | \alpha_r \rangle \\ &= - \sum_{\{\alpha\}} \sum_{S_M} \prod_{p=1}^{2M+1} \langle \alpha_{p-1} | \hat{H}_{t_p,a_p} | \alpha_p \rangle \,. \end{aligned}$$

Given the similarity of this expression to the normalization in Eq. (3.13), one can apply the same simplifications, degeneracies, and change of variables as was previously done for the normalization. However, to conveniently work with the same SSE configuration space labels $\{\alpha\}$, $S_{\tilde{M}}$, and m (or q), one must note that the total number of $\hat{H}_{1,a}$ occurrences in an operator sequence S_M with $2M+1$ entries is $m+1$. For clarity, these are the three simplifications to make.

1. SSE configuration weights accrue a factor of h for every $\hat{H}_{1,a}$ operator present:

$$\prod_{p=1}^{2M+1} \langle \alpha_{p-1} | \hat{H}_{t_p,a_p} | \alpha_p \rangle = h^{m+1} \prod_{p=1}^{\tilde{M}} \langle \alpha_{p-1} | \hat{H}_{t_p,a_p} | \alpha_p \rangle \,.$$

2. Spatial independence: N^{m+1}.

3. Imaginary-time-index independence: $\binom{2M+1}{m+1}$.

In all,

$$\langle\alpha_\ell|(-\hat{H})^m\hat{H}(-\hat{H})^m|\alpha_r\rangle = -\sum_{\{\alpha\}}\sum_{S_{\tilde{M}}}\sum_m N^{m+1}\binom{2M+1}{m+1}h^{m+1}\prod_{p=1}^{\tilde{M}}\langle\alpha_{p-1}|\hat{H}_{t_p,a_p}|\alpha_p\rangle\,,$$

and using the change of variable $q = m+1$,

$$\begin{aligned}
&= -\sum_{\{\alpha\}}\sum_{S_{\tilde{M}}}\sum_q N^q\binom{2Q}{q}h^q\prod_{p=1}^{\tilde{M}}\langle\alpha_{p-1}|\hat{H}_{t_p,a_p}|\alpha_p\rangle \\
&= -h\sum_{\{\alpha\}}\sum_{S_{\tilde{M}}}\sum_q N^q\frac{(2Q)!}{\tilde{M}!q!}h^{q-1}\prod_{p=1}^{\tilde{M}}\langle\alpha_{p-1}|\hat{H}_{t_p,a_p}|\alpha_p\rangle \\
&= -h\sum_{\{\alpha\}}\sum_{S_{\tilde{M}}}\sum_q \Phi(\alpha, S_{\tilde{M}}, q).
\end{aligned}$$

Plugging in the simplified expressions for Z and $\langle\alpha_\ell|(-\hat{H})^m\hat{H}(-\hat{H})^m|\alpha_r\rangle$ into the expression for $\langle\hat{H}\rangle$,

$$\begin{aligned}
\frac{E_0}{N} &= \frac{\langle\hat{H}\rangle}{N} \\
&= \frac{1}{N}\frac{\langle\alpha_\ell|(-\hat{H})^M\hat{H}(-\hat{H})^M|\alpha_r\rangle}{Z} \\
&= -\frac{1}{N}\frac{\sum_{\{\alpha\}}\sum_{S_{\tilde{M}}}\sum_q \Phi(\alpha, S_{\tilde{M}}, q)}{\frac{1}{2Q\times N}\sum_{\{\alpha\}}\sum_{S_{\tilde{M}}}\sum_q \Phi(\alpha, S_{\tilde{M}}, q)q} \\
&= -2Q\frac{h}{\langle q\rangle}.
\end{aligned} \tag{3.38}$$

To summarize, the ground state energy of an SSE implementation with an elementary operator $\hat{H}_{1,a} = h\mathbb{I}$ can be estimated via the inverse of the number of $\hat{H}_{1,a}$ operators present in the operator string S_M.

3.4 The checkerboard transition on a square lattice

To ensure the accuracy of the energy estimator in Eq. (3.38), exact-diagonalization calculations for a one-dimensional lattice with $N = 10$ sites with open boundary conditions were

performed on the Hamiltonian in Eq. (3.1). Fig. 3.6 shows the absolute value of the energy difference between the exact ground state energy the QMC estimate using Eq. (3.38) with $h = \Omega/2 = 1/2$ for various δ/Ω values. It is clear that the derived energy estimator in Eq. (3.38) is accurate. Interestingly, the result of $\langle \hat{H} \rangle_{\text{QMC}}$ for this one-dimensional system was essentially independent of the type of cluster update – multibranch, line, or some combination of the two.

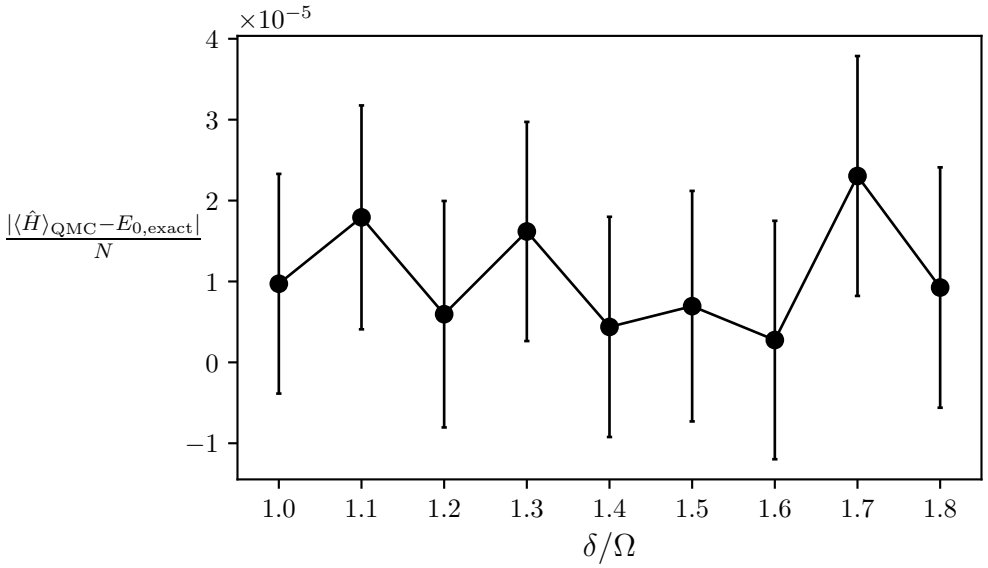

Figure 3.6: The absolute value of the difference between the SSE QMC energy estimator in Eq. (3.38), $\langle \hat{H} \rangle_{\text{QMC}}$, and the exact ground state energy, $E_{0,\text{exact}}$, of $N = 10$ Rydberg atoms in a one dimensional arrangement with open boundaries for various δ/Ω values. Each data point represents an independent SSE QMC simulation where the projector length used was $2M = 4 \times 10^5$ and the cluster update used was the line update. 10^5 equilibration steps were performed, then 10^7 successive measurements were taken. These 10^7 measurements were then processed via a standard jackknife routine to extract the plotted mean value and error bars.

The experimental work from Ebadi *et al.* showcased results from a 16×16 square lattice of Rydberg atoms [86]. Also, for a value of $R_b = 1.2$, Samajdar *et al.* reported the existence of a disordered-to-checkerboard-order quantum phase transition (QPT)in two spatial dimensions in the range of $\delta/\Omega \in [1, 1.2]$ [92]. Given the experimental motivation in this thesis, $L \times L$ lattices with $L = 4$, 10, and 16 with no truncation on the interactions will be used as platforms of choice for assessing the performance of the SSE QMC algorithm

developed in this thesis.

In general, zero-temperature SSE QMC simulations require that the projector length $2M$ be manually converged. It is a reasonable assumption that the converged projector length $2M$ will be larger for systems at criticality than for off-critical systems. Therefore, the converged projector length at the relevant critical point can be safely used for off-critical points as well. However, for the disordered-to-checkerboard-order QPT of interest here, the previously-reported DMRG results from Ref. [92] employed a third-nearest-neighbour truncation on the interactions in the Hamiltonian in Eq. (3.1). Given that the QMC SSE algorithm developed in this thesis does not have the clear dimensionality inefficiencies intrinsic to DMRG simulations, the exact value of the critical point reported in Ref. [92] will not be the same critical point that the the SSE QMC algorithm will produce. Therefore, the value of $\delta/\Omega = 1.12$ will be used as a *near*-critical point to facilitate finding a maximum converged projector length $2M$ for the $L \times L$ lattices of interest.

Table 3.1: Converged projector lengths $2M$ for $L \times L$ lattices.

L	$2M$
4	10^4
10	4×10^4
16	2×10^5

The order parameter across this transition is the absolute value of the staggered magnetization,

$$|M_s| = \left| \sum_{i=1}^{L} \sum_{j=1}^{L} (-1)^{(i+j)} \left(n_{i,j} - \frac{1}{2} \right) \right|, \tag{3.39}$$

where $n_{i,j}$ is the Rydberg occupation at the spatial location (i, j). Therefore, by looking at the value of $|M_s|$ as a function of $2M$, we may extract a suitably converged value of $2M$. Figs. 3.7-3.9 show the estimated $\langle |M_s| \rangle$ and its correlation time [109, 110] as a function of $1/2M$ as measured on a the chosen $L \times L$ lattice sizes with $\delta/\Omega = 1.12$ and $R_b = 1.2$ for three different cluster updates: multibranch ($p = 0$), line ($p = 1$), and a random choice between multibranch and line ($p = 0.5$). It is evident from each system size that the multibranch cluster update ($p = 0$) is less ergodic than the line update ($p = 1$), as the correlation time using the line update only is approximately one order of magnitude less in each case. As such, the remainder of the numerical results in this section will employ the line update only. Finally, in the $p = 0$ subplots, convergence in $\langle |M_s| \rangle / N$ for value of

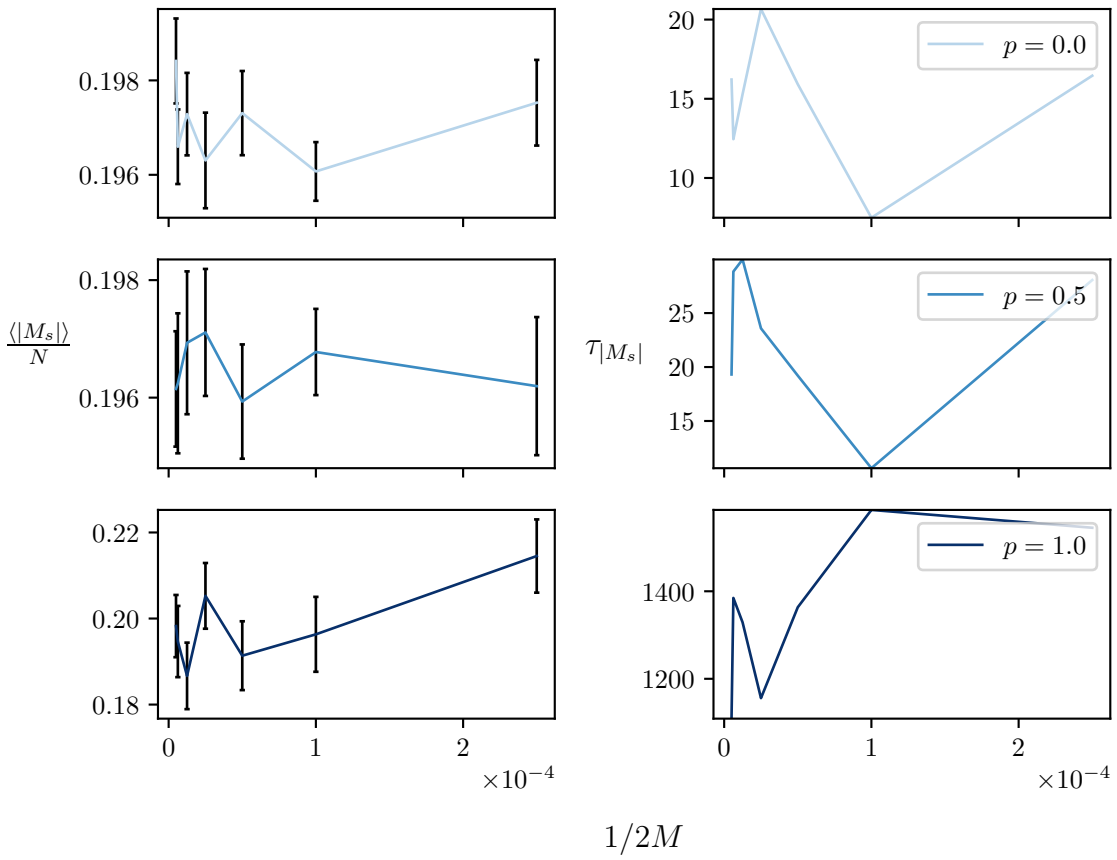

Figure 3.7: The absolute value of staggered magnetization density $\langle|M_s|\rangle/N$ and its correlation time $\tau_{|M_s|}$ versus the inverse projector length $1/2M$ as measured on a $N = 4\times4$ lattice with $\delta/\Omega = 1.12$ and $R_b = 1.2$ for three different cluster updates: multibranch ($p = 0$), line ($p = 1.0$), and a random choice between multibranch and line updates ($p = 0.5$). Each point represents a culmination of three independent – different random seeds – SSE QMC simulations. Per random seed, 2×10^5 consecutive measurements were taken after a 10^5 step equilibration phase, which was proceeded by a standard binning analysis to obtain the plotted mean values and error bars of $\langle|M_s|\rangle/N$.

L is reached for the projector lengths given in Tab. 3.1. Therefore, these projector lengths will be used for subsequent results. Similar plots for the QMC ground state energy density $\langle\hat{H}\rangle/N$ estimated using Eq. (3.38) can be found in App. B.

To qualitatively realize the disordered-to-checkerboard-order QPT in two spatial di-

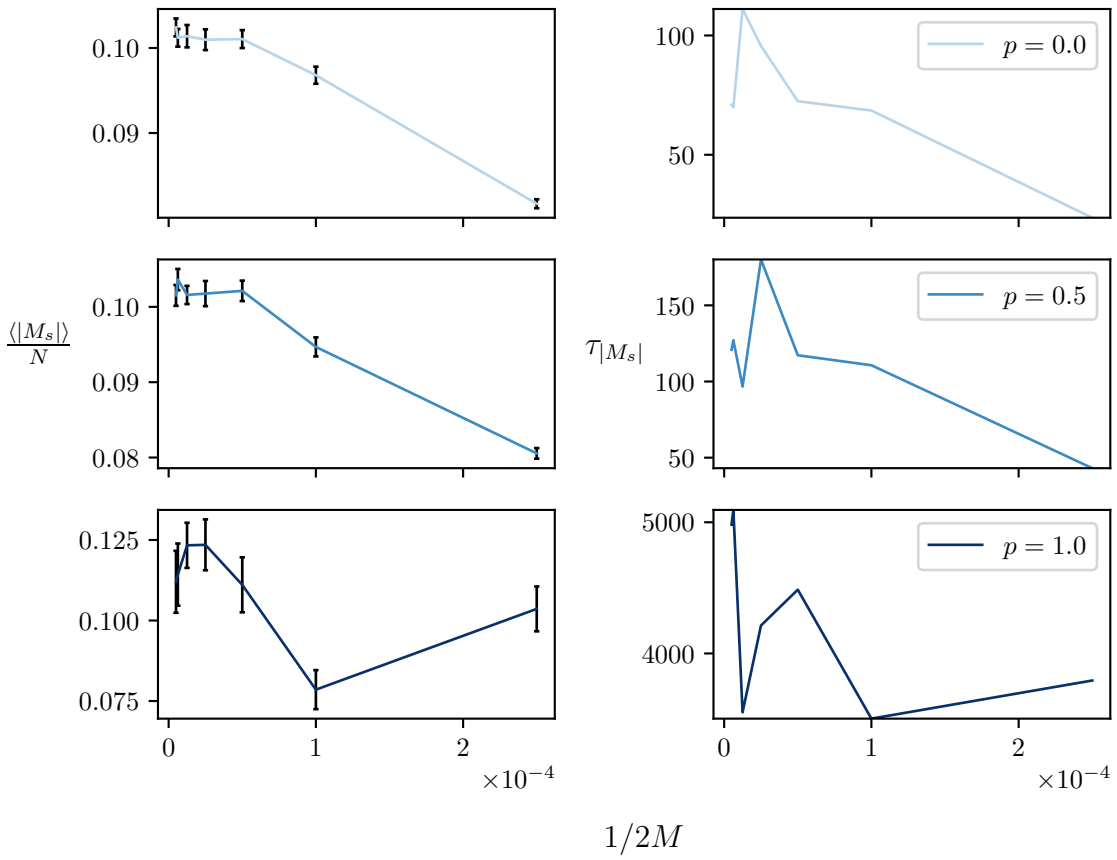

Figure 3.8: The absolute value of staggered magnetization density $\langle |M_s| \rangle / N$ and its correlation time $\tau_{|M_s|}$ versus the inverse projector length $1/2M$ as measured on a $N = 10 \times 10$ lattice with $\delta/\Omega = 1.12$ and $R_b = 1.2$ for three different cluster updates: multibranch ($p = 0$), line ($p = 1.0$), and a random choice between multibranch and line updates ($p = 0.5$). Each point represents a culmination of three independent – different random seeds – SSE QMC simulations. Per random seed, 2×10^5 consecutive measurements were taken after a 10^5 step equilibration phase, which was proceeded by a standard binning analysis to obtain the plotted mean values and error bars of $\langle |M_s| \rangle / N$.

mensions, the absolute value of the staggered magnetization in Eq. 3.39 can be estimated by the SSE QMC for a range of δ/Ω values. In doing so, we expect that $|M_s|$ is zero in the disordered phase and non-zero in the ordered phase and exhibits a relatively large increase in value near the critical δ/Ω value. Fig. 3.10 shows the ground state energy density

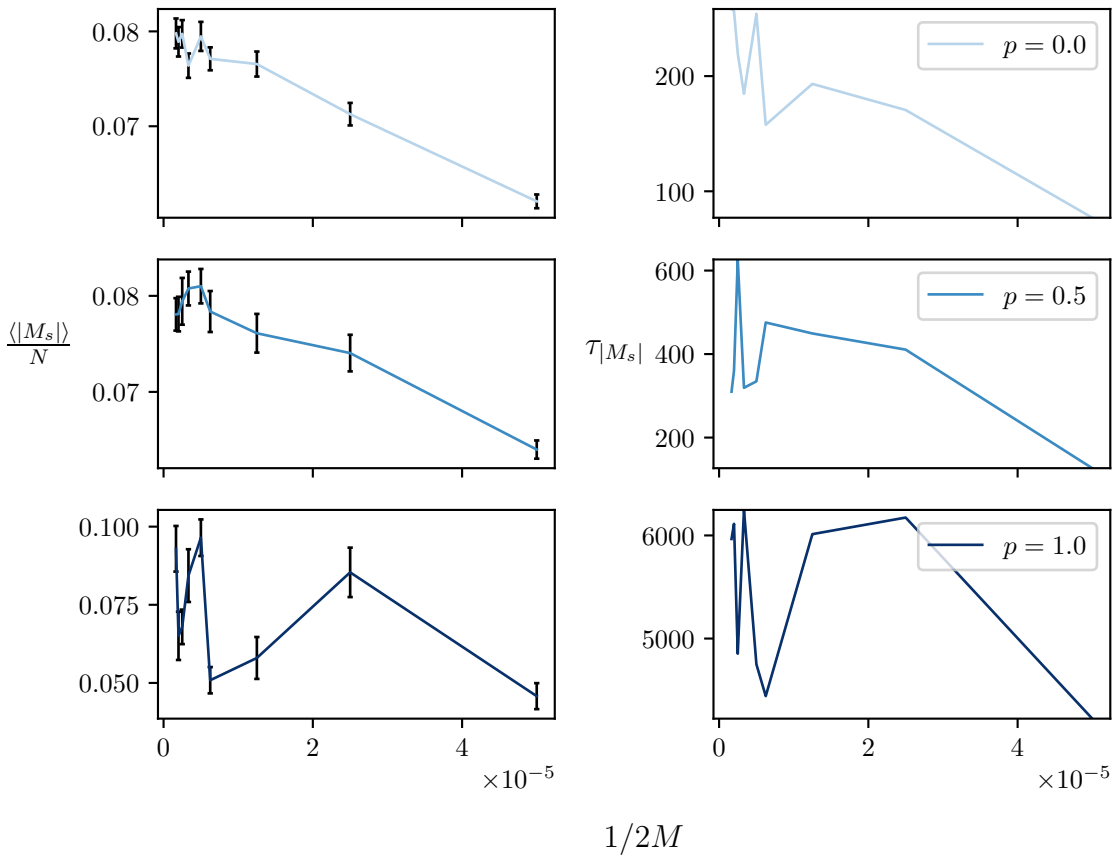

Figure 3.9: The absolute value of staggered magnetization density $\langle|M_s|\rangle/N$ and its correlation time $\tau_{|M_s|}$ versus the inverse projector length $1/2M$ as measured on a $N = 16 \times 16$ lattice with $\delta/\Omega = 1.12$ and $R_b = 1.2$ for three different cluster updates: multibranch ($p = 0$), line ($p = 1.0$), and a random choice between multibranch and line updates ($p = 0.5$). Each point represents a culmination of three independent – different random seeds – SSE QMC simulations. Per random seed, 2×10^5 consecutive measurements were taken after a 10^5 step equilibration phase, which was proceeded by a standard binning analysis to obtain the plotted mean values and error bars of $\langle|M_s|\rangle/N$.

estimate $\langle\hat{H}\rangle/N$ (Eq. (3.38)), the absolute value of the staggered magnetization density $\langle|M_s|\rangle/N$, and the correlation time for the absolute value of the staggered magnetization $\tau_{|M_s|}$ versus δ/Ω. Indeed, the checkerboard QPT is qualitatively realized given the form of $\langle|M_s|\rangle/N$ across the given δ/Ω range. Not only this, but the corresponding correlation

Table 3.2: Values of x used to make $2000 \times x$ consecutive measurements for the data plotted in Fig. 3.10.

L	x
4	50, 150 ($\delta/\Omega \in [1.5, 1.8]$)
10	150, 500 ($\delta/\Omega \in [1.2, 1.3]$)
16	500, 1000 ($\delta/\Omega = 1.2$)

times $\tau_{|M_s|}$ also show a peak that grows in height with N. Although mostly qualitative, the results presented here clearly show that the SSE QMC algorithm developed in Sec. 3.3 can accurately capture physical phenomena and that the ground state energy estimator derived in Sec. 3.3.4 is accurate.

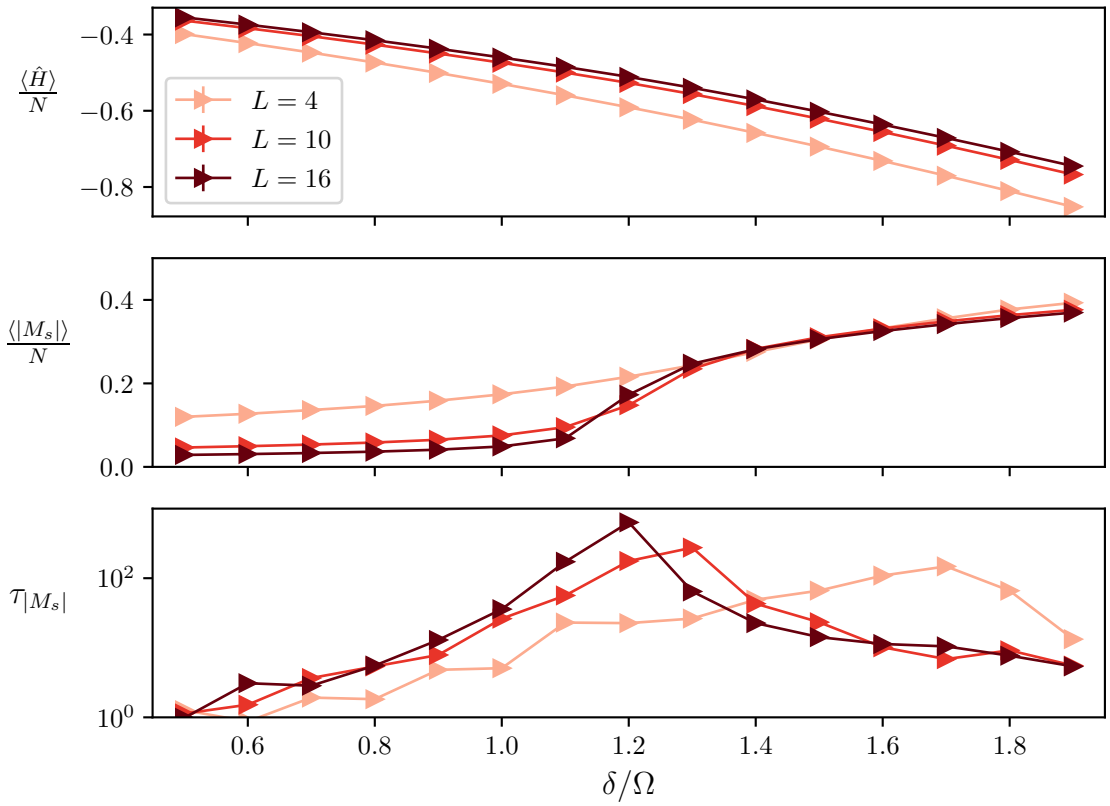

Figure 3.10: The QMC ground state energy density $\langle \hat{H} \rangle / N$, absolute value of the staggered magnetization density $\langle |M_s| \rangle / N$, and the correlation time for the absolute value of the staggered magnetization $\tau_{|M_s|}$ versus δ/Ω for $N = L \times L$ lattices ($L = 4$, 10 and 16). Each data point corresponds to a culmination of 15 independent – different random seeds – SSE QMC simulations. Per random seed, 10^5 equilibration steps were performed, then $2000 \times x$ consecutive measurements were taken, where the value of x is given in Tab. 3.2. Standard binning analysis techniques were employed to estimate the plotted mean value and error bars, which are smaller than the plot markers.

Chapter 4

Estimating the second Renyi entropy for Rydberg atoms

4.1 The Replica trick and SWAP operator

A central part of condensed matter physics is realizing and observing different phases of matter. A very broad class of phase transitions is a disordered-to-ordered transition, like the disordered-to-checkerboard-order QPT shown for $L \times L$ lattices of Rydberg atoms in Sec. 3.4. Second-order transitions like this one have a signature in the divergence of a length scale ξ, the correlation length, near the critical point. Another quantity that pertains to correlations that provides an augmented perspective of what mediates the properties of ground states at the critical point is the *entanglement entropy*. Like the correlation length ξ, the entanglement entropy between two parts of a system will also undergo significant changes around the critical point, but it can also shed light on interesting phenomena like many-body localization [111]. Therefore, its evaluation in studying QPTs is crucial. In this section, the well-known *replica* trick will be outlined for two numerical methods [112].

Given a quantum state represented by a density operator $\hat{\rho}$, one may always represent its matrix elements in a bipartite basis $|\boldsymbol{\sigma}_A, \boldsymbol{\sigma}_{\bar{A}}\rangle$, where the subscripts A and $\bar{A}$ represent the two complementary physical regions for which the bipartition is defined. The quantum state describing only one part of the bipartition, say region A, is called the *reduced* density operator, and is calculated via summing over the complementary degrees of freedom.

Assuming the original state $\hat{\rho}$ is unnormalized,

$$\hat{\rho}_A = \frac{1}{Z} \operatorname{Tr}_{\bar{A}} \hat{\rho} = \frac{1}{Z} \sum_{\{\boldsymbol{\sigma}_{\bar{A}}\}} \langle \boldsymbol{\sigma}_{\bar{A}} | \hat{\rho} | \boldsymbol{\sigma}_{\bar{A}} \rangle \,, \tag{4.1}$$

where $Z = \operatorname{Tr} \hat{\rho}_A = \operatorname{Tr} \hat{\rho}$.

The second Renyi entropy,

$$S_2 = -\ln \operatorname{Tr} \hat{\rho}_A^2, \tag{4.2}$$

is a basis-independent quantity that measures the amount of entanglement for ground states (pure states) between the two regions A and $\bar{A}$. For example, a system is said to be *maximally entangled* if when the complementary degrees of freedom are summed over, the resulting state $\hat{\rho}_A$ is *maximally mixed*: $\hat{\rho}_A = \frac{1}{|\mathcal{D}_A|}\mathbb{I}$, where $|\mathcal{D}_A|$ is the full Hilbert space size of the bipartition A. The second Renyi entropy cannot be calculated traditionally like physical observables and is therefore seemingly out of reach for QMC methods where a prescription for evaluating matrix elements of $\hat{\rho}_A^2$ is absent. Luckily, methods have been devised to facilitate its estimation akin to traditional physical observables with the cost of modifying the QMC algorithm itself [113–115].

Given a bipartition $A/\bar{A}$, the trace over the square of the reduced density matrix can be written as

$$\begin{aligned} \operatorname{Tr} \hat{\rho}_A^2 &= \frac{\sum_{\{\boldsymbol{\sigma}_{1,A}\}} \left\langle \boldsymbol{\sigma}_{1,A} \middle| \hat{\rho}_A^2 \middle| \boldsymbol{\sigma}_{1,A} \right\rangle}{\left(\sum_{\{\boldsymbol{\sigma}_{1,A}\}} \left\langle \boldsymbol{\sigma}_{1,A} \middle| \hat{\rho}_A \middle| \boldsymbol{\sigma}_{1,A} \right\rangle \right)^2} \\ &= \frac{\sum_{\{\boldsymbol{\sigma}_{1,A}\}} \sum_{\{\boldsymbol{\sigma}_{2,A}\}} \left\langle \boldsymbol{\sigma}_{1,A} \middle| \hat{\rho}_A \middle| \boldsymbol{\sigma}_{2,A} \right\rangle \left\langle \boldsymbol{\sigma}_{2,A} \middle| \hat{\rho}_A \middle| \boldsymbol{\sigma}_{1,A} \right\rangle}{\sum_{\{\boldsymbol{\sigma}_{1,A}\}} \sum_{\{\boldsymbol{\sigma}_{2,A}\}} \left\langle \boldsymbol{\sigma}_{1,A} \middle| \hat{\rho}_A \middle| \boldsymbol{\sigma}_{1,A} \right\rangle \left\langle \boldsymbol{\sigma}_{2,A} \middle| \hat{\rho}_A \middle| \boldsymbol{\sigma}_{2,A} \right\rangle} \\ &= \frac{Z[2, A]}{Z[2, \emptyset]}, \end{aligned} \tag{4.3}$$

where $Z[2, A]$ is commonly referred to as a *replicated* partition function. The denominator, $Z[2, \emptyset] = Z^2$, is the square of the usual quantum partition function, which is feasibly importance sampled by QMC algorithms. However, an estimation scheme for $Z[2, A]$ is still not apparent.

Let us now attempt to shape $Z[2, A]$ into something that resembles an expectation value of an operator. Starting from the previous line leading to Eq. (4.3),

$$Z[2, A] = \sum_{\{\boldsymbol{\sigma}_{1,A}\}} \sum_{\{\boldsymbol{\sigma}_{2,A}\}} \left\langle \boldsymbol{\sigma}_{1,A} \middle| \hat{\rho}_A \middle| \boldsymbol{\sigma}_{2,A} \right\rangle \left\langle \boldsymbol{\sigma}_{2,A} \middle| \hat{\rho}_A \middle| \boldsymbol{\sigma}_{1,A} \right\rangle ,$$

using Eq. (4.1) with $Z = 1$,

$$= \sum_{\substack{\{\boldsymbol{\sigma}_{1,A}\}\\ \{\boldsymbol{\sigma}_{2,A}\}}} \sum_{\substack{\{\boldsymbol{\sigma}_{1,\bar{A}}\}\\ \{\boldsymbol{\sigma}_{2,\bar{A}}\}}} \left\langle \boldsymbol{\sigma}_{1,A}, \boldsymbol{\sigma}_{1,\bar{A}} \middle| \hat{\rho} \middle| \boldsymbol{\sigma}_{2,A}, \boldsymbol{\sigma}_{1,\bar{A}} \right\rangle \left\langle \boldsymbol{\sigma}_{2,A}, \boldsymbol{\sigma}_{2,\bar{A}} \middle| \hat{\rho} \middle| \boldsymbol{\sigma}_{1,A}, \boldsymbol{\sigma}_{2,\bar{A}} \right\rangle . \tag{4.4}$$

Let us also make the assumption that $\hat{\rho}$ describes a pure state $|\psi\rangle$[1],

$$\hat{\rho} = |\psi\rangle\langle\psi| .$$

Now,

$$\begin{aligned}
Z[2, A] &= \sum_{\substack{\{\boldsymbol{\sigma}_{1,A}\}\\ \{\boldsymbol{\sigma}_{2,A}\}}} \sum_{\substack{\{\boldsymbol{\sigma}_{1,\bar{A}}\}\\ \{\boldsymbol{\sigma}_{2,\bar{A}}\}}} \left\langle \boldsymbol{\sigma}_{1,A}, \boldsymbol{\sigma}_{1,\bar{A}} \middle| \psi \right\rangle \left\langle \psi \middle| \boldsymbol{\sigma}_{2,A}, \boldsymbol{\sigma}_{1,\bar{A}} \right\rangle \left\langle \boldsymbol{\sigma}_{2,A}, \boldsymbol{\sigma}_{2,\bar{A}} \middle| \psi \right\rangle \left\langle \psi \middle| \boldsymbol{\sigma}_{1,A}, \boldsymbol{\sigma}_{2,\bar{A}} \right\rangle \\
&= \sum_{\substack{\{\boldsymbol{\sigma}_{1,A}\}\\ \{\boldsymbol{\sigma}_{2,A}\}}} \sum_{\substack{\{\boldsymbol{\sigma}_{1,\bar{A}}\}\\ \{\boldsymbol{\sigma}_{2,\bar{A}}\}}} \left\langle \psi \middle| \boldsymbol{\sigma}_{2,A}, \boldsymbol{\sigma}_{1,\bar{A}} \right\rangle \left\langle \psi \middle| \boldsymbol{\sigma}_{1,A}, \boldsymbol{\sigma}_{2,\bar{A}} \right\rangle \left\langle \boldsymbol{\sigma}_{1,A}, \boldsymbol{\sigma}_{1,\bar{A}} \middle| \psi \right\rangle \left\langle \boldsymbol{\sigma}_{2,A}, \boldsymbol{\sigma}_{2,\bar{A}} \middle| \psi \right\rangle \\
&= \langle\psi| \otimes \langle\psi| \left[\sum_{\substack{\{\boldsymbol{\sigma}_{1,A}\}\\ \{\boldsymbol{\sigma}_{2,A}\}}} \left| \boldsymbol{\sigma}_{2,A}, \boldsymbol{\sigma}_{1,\bar{A}}, \boldsymbol{\sigma}_{1,A}, \boldsymbol{\sigma}_{2,\bar{A}} \right\rangle\!\!\left\langle \boldsymbol{\sigma}_{1,A}, \boldsymbol{\sigma}_{1,\bar{A}}, \boldsymbol{\sigma}_{2,A}, \boldsymbol{\sigma}_{2,\bar{A}} \right| \right] |\psi\rangle \otimes |\psi\rangle \\
&= \langle\psi| \otimes \langle\psi| \, \mathrm{SWAP}_A \, |\psi\rangle \otimes |\psi\rangle ,
\end{aligned} \tag{4.5}$$

where the term in squared brackets is aptly termed the *SWAP* operator,

$$\mathrm{SWAP}_A = \left[\sum_{\substack{\{\boldsymbol{\sigma}_{1,A}\}\\ \{\boldsymbol{\sigma}_{2,A}\}}} \left| \boldsymbol{\sigma}_{2,A}, \boldsymbol{\sigma}_{1,\bar{A}}, \boldsymbol{\sigma}_{1,A}, \boldsymbol{\sigma}_{2,\bar{A}} \right\rangle\!\!\left\langle \boldsymbol{\sigma}_{1,A}, \boldsymbol{\sigma}_{1,\bar{A}}, \boldsymbol{\sigma}_{2,A}, \boldsymbol{\sigma}_{2,\bar{A}} \right| \right] , \tag{4.6}$$

since the degrees of freedom in region A, $\boldsymbol{\sigma}_{1,A}$ and $\boldsymbol{\sigma}_{2,A}$, *swap* places in the ket and bra (highlighted in red). Notice that the ratio Eq. (4.3) simplifies to

$$\frac{Z[2, A]}{Z[2, \emptyset]} = \frac{\langle\Psi| \, \mathrm{SWAP}_A \, |\Psi\rangle}{\langle\Psi|\Psi\rangle}, \tag{4.7}$$

[1]This is a valid assumption for the purpose of this thesis.

where $|\Psi\rangle = |\psi\rangle \otimes |\psi\rangle$, which is of the form of a traditional observable. Of course, the caveat to evaluating this ratio as a traditional observable is the requirement of *two* replicas of the state $|\psi\rangle$, while also ensuring that the boundary conditions between replicas for degrees of freedom within the region A are in accordance with Eq. (4.4). Specific examples of estimating Eq. (4.7) are given in the next two sections.

4.1.1 Wavefunctions

Consider a wavefunction $|\psi\rangle$ that possesses no sign structure in a computational basis $\{|\boldsymbol{\sigma}\rangle\}$[2]. Given efficient means to draw a set of samples $\{\boldsymbol{\sigma}\} \in \boldsymbol{\Gamma}$ from the distribution $p(\boldsymbol{\sigma}) = |\langle \boldsymbol{\sigma}|\psi\rangle|^2 = |\psi(\boldsymbol{\sigma})|^2$, one can estimate the ratio in Eq. (4.7) as

$$\begin{aligned}
\frac{Z[2,A]}{Z[2,\emptyset]} &= \frac{\sum_{\{\boldsymbol{\sigma}_{1,A}\}}\sum_{\{\boldsymbol{\sigma}_{2,A}\}}\sum_{\{\boldsymbol{\sigma}_{1,\bar{A}}\}}\sum_{\{\boldsymbol{\sigma}_{2,\bar{A}}\}} \langle\psi|\boldsymbol{\sigma}_{2,A},\boldsymbol{\sigma}_{1,\bar{A}}\rangle\langle\psi|\boldsymbol{\sigma}_{1,A},\boldsymbol{\sigma}_{2,\bar{A}}\rangle\langle\boldsymbol{\sigma}_{1,A},\boldsymbol{\sigma}_{1,\bar{A}}|\psi\rangle\langle\boldsymbol{\sigma}_{2,A},\boldsymbol{\sigma}_{2,\bar{A}}|\psi\rangle}{\sum_{\{\boldsymbol{\sigma}_{1,A}\}}\sum_{\{\boldsymbol{\sigma}_{2,A}\}}\sum_{\{\boldsymbol{\sigma}_{1,\bar{A}}\}}\sum_{\{\boldsymbol{\sigma}_{2,\bar{A}}\}} \langle\psi|\boldsymbol{\sigma}_{1,A},\boldsymbol{\sigma}_{1,\bar{A}}\rangle\langle\psi|\boldsymbol{\sigma}_{1,A},\boldsymbol{\sigma}_{1,\bar{A}}\rangle\langle\boldsymbol{\sigma}_{2,A},\boldsymbol{\sigma}_{2,\bar{A}}|\psi\rangle\langle\boldsymbol{\sigma}_{2,A},\boldsymbol{\sigma}_{2,\bar{A}}|\psi\rangle} \\
&= \frac{\sum_{\{\boldsymbol{\sigma}_{1,A}\}}\sum_{\{\boldsymbol{\sigma}_{2,A}\}}\sum_{\{\boldsymbol{\sigma}_{1,\bar{A}}\}}\sum_{\{\boldsymbol{\sigma}_{2,\bar{A}}\}} \psi(\boldsymbol{\sigma}_{2,A},\boldsymbol{\sigma}_{1,\bar{A}})\psi(\boldsymbol{\sigma}_{1,A},\boldsymbol{\sigma}_{2,\bar{A}})\psi(\boldsymbol{\sigma}_{1,A},\boldsymbol{\sigma}_{1,\bar{A}})\psi(\boldsymbol{\sigma}_{2,A},\boldsymbol{\sigma}_{2,\bar{A}})}{\sum_{\{\boldsymbol{\sigma}_{1,A}\}}\sum_{\{\boldsymbol{\sigma}_{2,A}\}}\sum_{\{\boldsymbol{\sigma}_{1,\bar{A}}\}}\sum_{\{\boldsymbol{\sigma}_{2,\bar{A}}\}} \psi(\boldsymbol{\sigma}_{1,A},\boldsymbol{\sigma}_{1,\bar{A}})\psi(\boldsymbol{\sigma}_{1,A},\boldsymbol{\sigma}_{1,\bar{A}})\psi(\boldsymbol{\sigma}_{2,A},\boldsymbol{\sigma}_{2,\bar{A}})\psi(\boldsymbol{\sigma}_{2,A},\boldsymbol{\sigma}_{2,\bar{A}})} \\
&= \frac{\sum_{\{\boldsymbol{\sigma}_{1,A}\}}\sum_{\{\boldsymbol{\sigma}_{2,A}\}}\sum_{\{\boldsymbol{\sigma}_{1,\bar{A}}\}}\sum_{\{\boldsymbol{\sigma}_{2,\bar{A}}\}} \Psi(\boldsymbol{\sigma}_{2,A},\boldsymbol{\sigma}_{1,\bar{A}},\boldsymbol{\sigma}_{1,A},\boldsymbol{\sigma}_{2,\bar{A}})\Psi(\boldsymbol{\sigma}_{1,A},\boldsymbol{\sigma}_{1,\bar{A}},\boldsymbol{\sigma}_{2,A},\boldsymbol{\sigma}_{2,\bar{A}})}{\sum_{\{\boldsymbol{\sigma}_{1,A}\}}\sum_{\{\boldsymbol{\sigma}_{2,A}\}}\sum_{\{\boldsymbol{\sigma}_{1,\bar{A}}\}}\sum_{\{\boldsymbol{\sigma}_{2,\bar{A}}\}} \Psi(\boldsymbol{\sigma}_{1,A},\boldsymbol{\sigma}_{1,\bar{A}},\boldsymbol{\sigma}_{2,A},\boldsymbol{\sigma}_{2,\bar{A}})\Psi(\boldsymbol{\sigma}_{1,A},\boldsymbol{\sigma}_{1,\bar{A}},\boldsymbol{\sigma}_{2,A},\boldsymbol{\sigma}_{2,\bar{A}})},
\end{aligned}$$

where the compact notation for wavefunction coefficients $\langle\boldsymbol{\sigma}_{1,A},\boldsymbol{\sigma}_{1,\bar{A}}|\psi\rangle = \psi(\boldsymbol{\sigma}_{1,A},\boldsymbol{\sigma}_{1,\bar{A}})$ is used along with $|\Psi\rangle = |\psi\rangle\otimes|\psi\rangle$ and $\Psi(\boldsymbol{\sigma}_{1,A},\boldsymbol{\sigma}_{1,\bar{A}},\boldsymbol{\sigma}_{2,A},\boldsymbol{\sigma}_{2,\bar{A}}) = \psi(\boldsymbol{\sigma}_{1,A},\boldsymbol{\sigma}_{1,\bar{A}})\psi(\boldsymbol{\sigma}_{2,A},\boldsymbol{\sigma}_{2,\bar{A}})$. Since one sums over $\{\boldsymbol{\sigma}_{1/2,A/\bar{A}}\}$, $\Psi(\boldsymbol{\sigma}_{1,A},\boldsymbol{\sigma}_{1,\bar{A}},\boldsymbol{\sigma}_{2,A},\boldsymbol{\sigma}_{2,\bar{A}})$ plays the role of a probability distribution. Therefore,

$$\begin{aligned}
\frac{Z[2,A]}{Z[2,\emptyset]} &= \left\langle \frac{\Psi(\boldsymbol{\sigma}_{2,A},\boldsymbol{\sigma}_{1,\bar{A}},\boldsymbol{\sigma}_{1,A},\boldsymbol{\sigma}_{2,\bar{A}})}{\Psi(\boldsymbol{\sigma}_{1,A},\boldsymbol{\sigma}_{1,\bar{A}},\boldsymbol{\sigma}_{2,A},\boldsymbol{\sigma}_{2,\bar{A}})} \right\rangle_{\Psi(\boldsymbol{\sigma}_{1,A},\boldsymbol{\sigma}_{1,\bar{A}},\boldsymbol{\sigma}_{2,A},\boldsymbol{\sigma}_{2,\bar{A}})} \\
&\approx \left\langle \frac{\Psi(\boldsymbol{\sigma}_{2,A},\boldsymbol{\sigma}_{1,\bar{A}},\boldsymbol{\sigma}_{1,A},\boldsymbol{\sigma}_{2,\bar{A}})}{\Psi(\boldsymbol{\sigma}_{1,A},\boldsymbol{\sigma}_{1,\bar{A}},\boldsymbol{\sigma}_{2,A},\boldsymbol{\sigma}_{2,\bar{A}})} \right\rangle_{\boldsymbol{\Gamma}}
\end{aligned}$$

[2]That is, coefficients $\langle\boldsymbol{\sigma}|\psi\rangle$ are all the same sign, modulo an arbitrary global phase. We may therefore consider $|\psi\rangle$ to be positive-definite in this basis.

$$= \left\langle \frac{\psi(\boldsymbol{\sigma}_{2,A}, \boldsymbol{\sigma}_{1,\bar{A}})\psi(\boldsymbol{\sigma}_{1,A}, \boldsymbol{\sigma}_{2,\bar{A}})}{\psi(\boldsymbol{\sigma}_{1,A}, \boldsymbol{\sigma}_{1,\bar{A}})\psi(\boldsymbol{\sigma}_{2,A}, \boldsymbol{\sigma}_{2,\bar{A}})} \right\rangle_{\boldsymbol{\Gamma}} \tag{4.8}$$

where $\boldsymbol{\sigma}_{1/2} \in \boldsymbol{\Gamma}$. To summarize, in this scenario the ratio Eq. (4.7) is estimated via the ratio of coefficients of the given positive-definite wavefunction $|\psi\rangle$. Crucially, $|\psi\rangle$ need not be normalized to evaluate Eq. (4.8), as the normalization factors from the numerator and denominator cancel each other. This will be vital for evaluating this ratio when $\psi(\boldsymbol{\sigma})$ is parameterized by an RBM, where calculating its normalization factor is still exponentially unfavourable in the number of degrees of freedom (see Section 2.1).

4.1.2 Zero-temperature SSE

The prescription for estimating Eq. (4.7) in the zero-temperature SSE formalism is detailed in Ref. [114]. However, given its complexity and application to the given SSE implementation for Rydberg atoms shown in Ch. 3, an overview of the theory is sensible. Nevertheless, the reader is still encouraged to consult Ref. [114].

As was shown in Sec. 3.2, the (unnormalized) ground state $|\lambda_0\rangle$ of a Hamiltonian $\hat{H}$ is projected out of an arbitrary state via M applications of $-\hat{H}$. After various simplifications, M applications of $-\hat{H}$ translated into importance sampling matrix elements of elementary lattice operators $\hat{H}_{t,a}$ (see Eq. (3.11)) in $D+1$ dimensions in a computational basis $\{|\alpha\rangle\}$ at every index in the $+1$ (imaginary time) dimension. A diagrammatic interpretation of an SSE configuration is given in Fig. 3.1. However, how estimating Eq. (4.3) given SSE configurations is not obvious.

Firstly, it will help to be able to diagrammatically visualize the "interchanging" mechanism in an SSE configuration space. Estimating the SWAP operator directly from a wavefunction in Sec. 4.1.1 involved interchanging spin states in the region A from two replicas of the given wavefunction (see Eq. (4.8)). The "interchanging" mechanism in the SSE configuration space is *precisely* the same as for wavefunctions with the caveat being that *world lines* of degrees of freedom in the region A are interchanged between replicas at the middle of the projection (i.e. at the imaginary time index $p = M$). Consider the second-last line leading up to Eq. (4.5) with conventional SSE variables,

$$Z[2, A] = \sum_{\substack{\{\boldsymbol{\sigma}_{1,A}\} \\ \{\boldsymbol{\sigma}_{2,A}\}}} \sum_{\substack{\{\boldsymbol{\sigma}_{1,\bar{A}}\} \\ \{\boldsymbol{\sigma}_{2,\bar{A}}\}}} \left\langle \lambda_0^{(1)} \middle| \alpha_{2,A}, \alpha_{1,\bar{A}} \right\rangle \left\langle \lambda_0^{(2)} \middle| \alpha_{1,A}, \alpha_{2,\bar{A}} \right\rangle \left\langle \alpha_{1,A}, \alpha_{1,\bar{A}} \middle| \lambda_0^{(1)} \right\rangle \left\langle \alpha_{2,A}, \alpha_{2,\bar{A}} \middle| \lambda_0^{(2)} \right\rangle ,$$

where the superscipts 1 and 2 on λ_0 distinguish the replicas since we require two independently-run SSE simulations to obtain two independent copies of the ground state $|\lambda_0\rangle$. Note that

the ground state $|\lambda_0\rangle$ will be obtained by the SSE propagation rule in Eq. (3.16) starting from ($p = 1$) an arbitrary trial state $|\psi\rangle$: $|\lambda_0\rangle \propto \prod_{p=1}^{M} \hat{H}_{t_p,a_p} |\psi\rangle$. Therefore, when one reaches the middle ($p = M$) of the simulation cell, the first two terms in the summand for the expression for $Z[2, A]$ indicate that the states $\left|\alpha_{1,A}\right\rangle$ and $\left|\alpha_{2,A}\right\rangle$ propagated from the left-hand side of their respective replicated simulation cells and *swapped* between replicas and are then projected back out to the right-hand side of the complementary simulation cell. This is diagrammatically shown in Fig. 4.1 for a projector length $2M = 6$, $N = 4$ degrees of freedom, and a bipartition $A = [1, 2]$ and $\bar{A} = [3, 4]$.

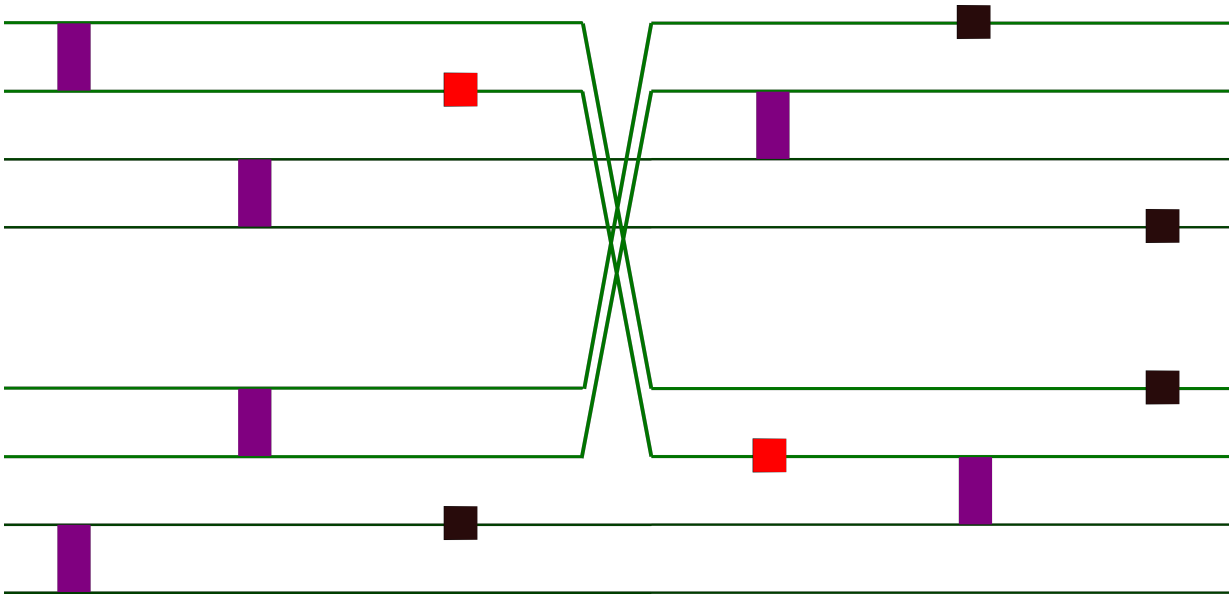

Figure 4.1: An example of the interchanging mechanism for calculating the SWAP operator in the SSE formalism. Given the two replicas with $M = 6$, the green world lines in the bipartition $A = [1, 2]$ continue by switching to the complementary replica at $M = 3$, whereas the world lines in $\bar{A}$ remain in the same replica.

It is at this point that the curious reader is advised to consult Ref. [114] – specifically, Ch. 3 – to understand in detail why the following statements are necessary for evaluating Eq. (4.3) in the SSE language. Instead of working in the conventional SSE configuration space labelled by the operator sequence S_M and basis states $\{|\alpha\rangle\}$, it will be helpful to work in a basis defined by a *linked list of vertices*. Specifically, the labels are:

- l – list of links: this list contains information on the legs of vertices that are linked/connected.

- $v^A(l)$ – set of vertices: given a list of links l, many different vertex types whose legs are linked are compatible with the same list of links. The superscript A denotes a prescription, given two replicas, for how links are made at the middle of the simulation cells (i.e. the *swapping* mechanism outlined in Fig. 4.1).

Given this, one can write

$$Z[2,A] = \sum_{l} \sum_{v^A(l)} \Theta(v^A(l)), \tag{4.9}$$

where $\Theta(v^A(l))$ is a positive weight given to a configuration labelled by the list of links l and set of vertices $v^A(l)$. The list of links is determined by the choice of off-diagonal update. For instance, with a multibranch cluster update (see Sec. 3.3.2), the list of links is a data structure that details how the legs in the corresponding multibranch cluster are linked/connected. Given this list of links, the vertices $v^A(l)$ are simply all vertices that have legs in the multibranch cluster. Finally, the weight $\Theta(v^A(l))$ is similar to Eq. (3.23), but it is a product of all of *all* multibranch cluster weights $\mathcal{W}$.

In all, the entity that requires evaluating is

$$\frac{Z[2,A]}{Z[2,\emptyset]} = \frac{\sum_l \sum_{v^A(l)} \Theta(v^A(l))}{\sum_l \sum_{v^\emptyset(l)} \Theta(v^\emptyset(l))} = \left\langle \frac{\sum_{v^A(l)} \Theta(v^A(l))}{\sum_{v^\emptyset(l)} \Theta(v^\emptyset(l))} \right\rangle_\emptyset . \tag{4.10}$$

Given that its evaluation depends on the off-diagonal update (required to determine the list of links) and the vertex weights, Eq. (4.10) will heavily depend on the SSE implementation. In the interest of capturing bipartite entanglement behaviour for Rydberg atoms using the SSE implementation presented in Ch. 3, we will once again return to the transverse-field Ising model with a longitudinal field (LTFIM) that was shown in Sec. 3.3.3. The LTFIM Hamiltonian in Eq. (3.26) is qualitatively very similar to the Rydberg Hamiltonian in Eq. (3.1), but provides a simpler and more tidy platform that can be used to extrapolate properties of an algorithm that simulates the Rydberg Hamiltonian.

4.2 Important example: TFIM + longitudinal field

4.2.1 $h_z = 0$

First, consider when the longitudinal field h_z in Eq. (3.26) is turned off (i.e. just the TFIM Hamiltonian). In this case,

$$\langle\uparrow|\hat{H}_{-1,a}|\downarrow\rangle = \langle\downarrow|\hat{H}_{-1,a}|\uparrow\rangle = h_x, \tag{4.11a}$$

$$\langle\uparrow|\hat{H}_{1,a}|\uparrow\rangle = \langle\downarrow|\hat{H}_{1,a}|\downarrow\rangle = h_x, \tag{4.11b}$$

$$W_1 \equiv \langle\uparrow\uparrow|\hat{H}_{1,b}|\uparrow\uparrow\rangle = 2J, \tag{4.11c}$$

$$W_2 \equiv \langle\downarrow\downarrow|\hat{H}_{1,b}|\downarrow\downarrow\rangle = 2J, \tag{4.11d}$$

and

$$W_3 \equiv \langle\downarrow\uparrow|\hat{H}_{1,b}|\downarrow\uparrow\rangle = 0. \tag{4.11e}$$

As was previously mentioned in Sec. 3.3.2, the multibranch update is best suited for off-diagonal updates given this SSE implementation [96]. Given a multibranch cluster in this SSE implementation, its weight is simply a function of the number of $\hat{H}_{1,a}$ operators, n_a, and $\hat{H}_{1,b}$ operators, n_b, contained in the cluster. Specifically,

$$\mathcal{W}_{\text{TFIM}}(n_a, n_b) = \prod_{n_a} h_x \prod_{n_b} 2J = 2^{n_b} h_x^{n_a} J^{n_b}. \tag{4.12}$$

Not only this, but given the all-$\downarrow$ / all-$\uparrow$ symmetry present in Eq. (4.11), $\mathcal{W}_{\text{TFIM}}(n_a, n_b) = \mathcal{W}^{\uparrow}_{\text{TFIM}}(n_a, n_b) = \mathcal{W}^{\downarrow}_{\text{TFIM}}(n_a, n_b)$.

Keeping this in mind, let us return to Eq. (4.10). Recall that given a list of links l that defines the shape of the cluster in $D+1$ dimensions, one sums over vertex configurations $v^{\emptyset/A}(l)$ that have the same list of links. The multibranch clusters are independent of each other since they end on site operators $\hat{H}_{-1/1,a}$, which greatly facilitates summing over all possible vertex configurations. Luckily, for a TFIM multibranch cluster, a list of links prescribes *two* possible vertex configurations, an "all-$\uparrow$" one and an "all-$\downarrow$" one like the green multibranch clusters in Fig. 3.2. Therefore, both sums over the vertex sets $v^{\emptyset/A}(l)$ run all possible combinations of each multibranch cluster being "all-$\uparrow$" or "all-$\downarrow$". For example, if there are three multibranch clusters each with an "all-$\downarrow$" weight of $\mathcal{W}^{\downarrow}_{1-3}$, the sum over all vertex sets simplifies to a product of binomials,

$$\sum_{v(l)} \Theta(v(l)) = \mathcal{W}_1^{\downarrow}\mathcal{W}_2^{\downarrow}\mathcal{W}_3^{\downarrow} + \mathcal{W}_1^{\downarrow}\mathcal{W}_2^{\downarrow}\mathcal{W}_3^{\uparrow} + \mathcal{W}_1^{\downarrow}\mathcal{W}_2^{\uparrow}\mathcal{W}_3^{\downarrow} + \cdots + \mathcal{W}_1^{\uparrow}\mathcal{W}_2^{\uparrow}\mathcal{W}_3^{\uparrow} \qquad \text{(8 terms)}$$

$$= \prod_{i=1}^{3} \left(\mathcal{W}_i^{\downarrow} + \mathcal{W}_i^{\uparrow}\right), \tag{4.13}$$

However, in evaluating Eq. (4.10), one must also consider the *topology* of the simulation cell (i.e. swapping world lines or not swapping world lines). The entity $Z[2, \emptyset]$ describes two completely independent SSE simulations cells where world lines of physical degrees of

freedom stay in their respective replicas. $Z[2, A]$, however, describes two SSE simulation cells where world lines *swap* at the middle of the simulation (see Fig. 4.1). As such, multibranch clusters that are formed in the $\emptyset$ topology will be different than those formed in the A topology. Fig. 4.2 demonstrates this. That being said, as one can see in Fig. 4.2, some multibranch clusters are independent of the choice of topology. In general, clusters that do not cross the middle of the simulation cell and do not contain world lines that are in the region A remain unchanged (e.g. the yellow and light blue clusters in Fig. 4.2). Such clusters will be referred to as being in the *bulk* of the simulation cell.

As such, when estimating the ratio Eq. (4.10) given an instance of replicated SSE simulation cells, the bulk cluster weights will cancel. Specifically, in Fig. 4.2 the orange, red, pink, light blue, and yellow clusters will cancel. In general, representing these cancelled cluster weights as $\mathcal{W}_{\text{bulk}}$, Eq. (4.10) evaluates to

$$\begin{aligned}
\frac{Z[2, A]}{Z[2, \emptyset]} &= \left\langle \frac{\sum_{v^A(l)} \Theta(v^A(l))}{\sum_{v^\emptyset(l)} \Theta(v^\emptyset(l))} \right\rangle_\emptyset \\
&= \left\langle \frac{\prod_{\mathcal{C}_A \in \text{non-bulk}} (\mathcal{W}^\downarrow_{\mathcal{C}_A} + \mathcal{W}^\uparrow_{\mathcal{C}_A}) \prod_{\mathcal{C}_A \in \text{bulk}} (\mathcal{W}^\downarrow_{\mathcal{C}_A} + \mathcal{W}^\uparrow_{\mathcal{C}_A})}{\prod_{\mathcal{C}_\emptyset \in \text{non-bulk}} (\mathcal{W}^\downarrow_{\mathcal{C}_\emptyset} + \mathcal{W}^\uparrow_{\mathcal{C}_\emptyset}) \prod_{\mathcal{C}_\emptyset \in \text{bulk}} (\mathcal{W}^\downarrow_{\mathcal{C}_\emptyset} + \mathcal{W}^\uparrow_{\mathcal{C}_\emptyset})} \right\rangle_\emptyset \\
&= \left\langle \frac{\prod_{\mathcal{C}_A \in \text{non-bulk}} (\mathcal{W}^\downarrow_{\mathcal{C}_A} + \mathcal{W}^\uparrow_{\mathcal{C}_A})}{\prod_{\mathcal{C}_\emptyset \in \text{non-bulk}} (\mathcal{W}^\downarrow_{\mathcal{C}_\emptyset} + \mathcal{W}^\uparrow_{\mathcal{C}_\emptyset})} \right\rangle_\emptyset .
\end{aligned} \tag{4.14}$$

Finally, employing Eq. (4.12) to evaluate each cluster weight and noting the previously-mentioned all-$\downarrow$ / all-$\uparrow$ symmetry,

$$\begin{aligned}
\frac{Z[2, A]}{Z[2, \emptyset]} &= \left\langle \frac{\prod_{\mathcal{C}_A \in \text{non-bulk}} 2\mathcal{W}^\downarrow_{\mathcal{C}_A}}{\prod_{\mathcal{C}_\emptyset \in \text{non-bulk}} 2\mathcal{W}^\downarrow_{\mathcal{C}_\emptyset}} \right\rangle_\emptyset \\
&= \left\langle 2^{N_{\mathcal{C}_A} - N_{\mathcal{C}_\emptyset}} \frac{\prod_{\mathcal{C}_A \in \text{non-bulk}} (2J)^{n_{b,\mathcal{C}_A}} h_x^{n_{a,\mathcal{C}_A}}}{\prod_{\mathcal{C}_\emptyset \in \text{non-bulk}} (2J)^{n_{b,\mathcal{C}_\emptyset}} h_x^{n_{a,\mathcal{C}_\emptyset}}} \right\rangle_\emptyset \\
&= \left\langle 2^{N_{\mathcal{C}_A} - N_{\mathcal{C}_\emptyset}} \right\rangle_\emptyset ,
\end{aligned} \tag{4.15}$$

where $N_{\mathcal{C}_{A/\emptyset}}$ is the total number of multibranch clusters that cross the middle of the simulation cell given a particular topology. The cancellation made in the previous line was made since the total number of operators $\hat{H}_{1,b}$ and $\hat{H}_{-1/1,a}$ involved in non-bulk clusters must be conserved regardless of the choice of topology [104, 114]. To summarize, the SWAP operator can be elegantly estimated for the TFIM SSE implementation by counting non-bulk multibranch clusters.

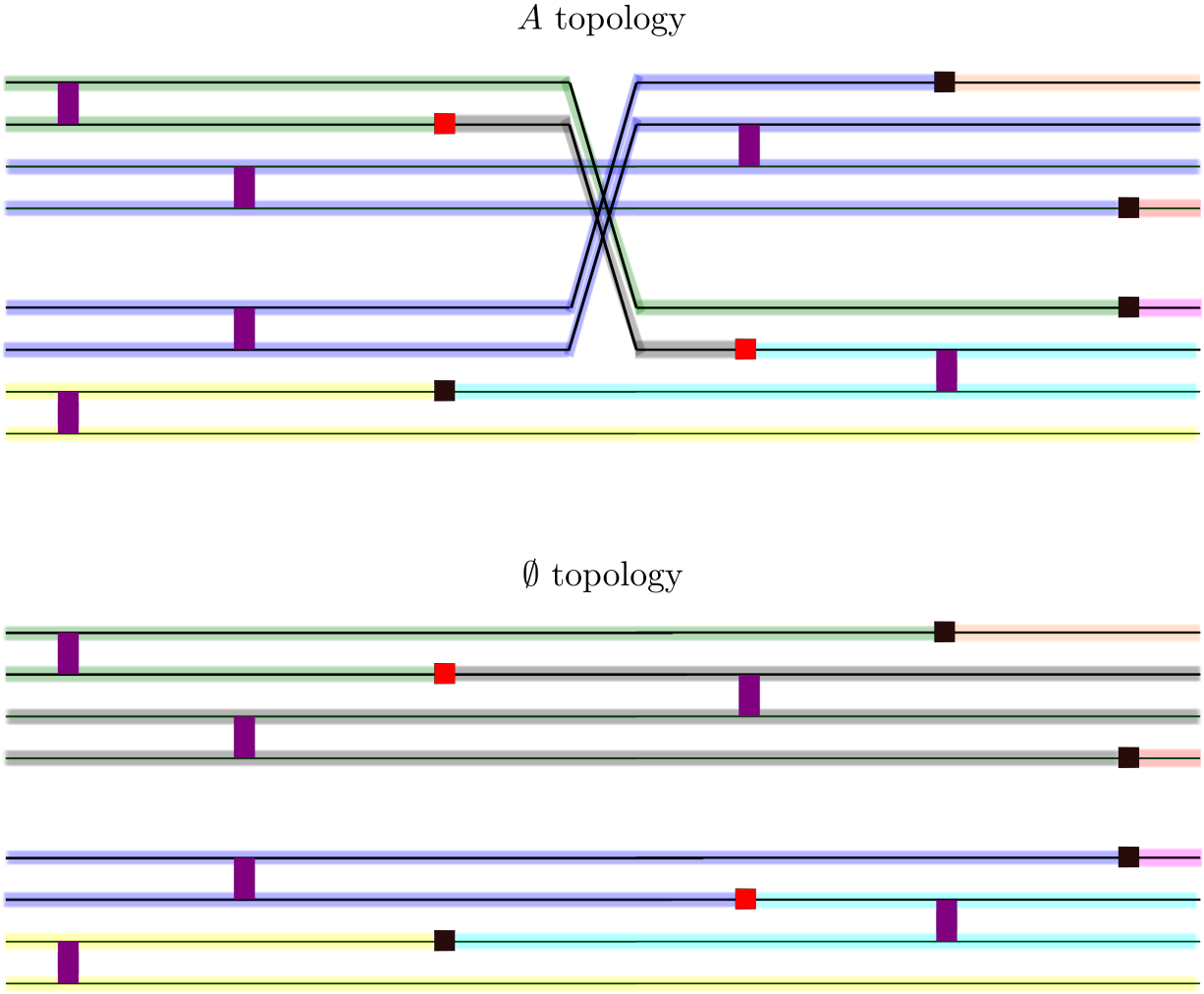

Figure 4.2: An example of two multibranch cluster topologies given a bipartition $A = [1, 2]$ and $\bar{A} = [3, 4]$. The different highlighted regions represent different multibranch clusters. Given an A topology, the corresponding world lines switch between replicas at the middle of the simulation cell, whereas the same world lines in the $\emptyset$ topology remain in their original replica.

4.2.2 $h_z \neq 0$

When the longitudinal field is turned on, $h_z \neq 0$, the negative implications on evaluating Eq. (4.10) arrise from $W_1 \neq W_2$ and $W_3 \neq 0$ in Eq. (3.28). However, the constant C in

each of W_{1-3} can help with evaluating Eq. (4.10) efficiently for some values of J, h_z, and h_x. When $J > h_{z,b}$, $C = J$ and

$$W_1 \equiv \langle\uparrow\uparrow|\hat{H}_{1,b}|\uparrow\uparrow\rangle = 2J + 2h_{z,b}, \tag{4.16a}$$

$$W_2 \equiv \langle\downarrow\downarrow|\hat{H}_{1,b}|\downarrow\downarrow\rangle = 2J - 2h_{z,b}, \tag{4.16b}$$

and

$$W_3 \equiv \langle\downarrow\uparrow|\hat{H}_{1,b}|\downarrow\uparrow\rangle = \langle\uparrow\downarrow|\hat{H}_{1,b}|\uparrow\downarrow\rangle = 0, \tag{4.16c}$$

which is very similar to Eq. (4.11) but with $W_1 \neq W_2$. That being said, of the possible set of vertices that may exist given a list of links defining multibranch clusters, the sum over the set of vertices remains to be two terms: "all-↑" and "all-↓". Not only this, but Eq. (4.13) can still be used since multibranch clusters are independent, and clusters in the bulk will still cancel.

Starting from Eq. (4.14),

$$\begin{aligned}
\frac{Z[2,A]}{Z[2,\emptyset]} &= \left\langle \frac{\prod_{\mathcal{C}_A\in\text{non-bulk}}(\mathcal{W}^{\downarrow}_{\mathcal{C}_A} + \mathcal{W}^{\uparrow}_{\mathcal{C}_A})}{\prod_{\mathcal{C}_\emptyset\in\text{non-bulk}}(\mathcal{W}^{\downarrow}_{\mathcal{C}_\emptyset} + \mathcal{W}^{\uparrow}_{\mathcal{C}_\emptyset})} \right\rangle_\emptyset \\
&= \left\langle \frac{\prod_{\mathcal{C}_A\in\text{non-bulk}}(W_1^{n_{b,\mathcal{C}_A}} h_x^{n_{a,\mathcal{C}_A}} + W_2^{n_{b,\mathcal{C}_A}} h_x^{n_{a,\mathcal{C}_A}})}{\prod_{\mathcal{C}_\emptyset\in\text{non-bulk}}(W_1^{n_{b,\mathcal{C}_\emptyset}} h_x^{n_{a,\mathcal{C}_\emptyset}} + W_2^{n_{b,\mathcal{C}_\emptyset}} h_x^{n_{a,\mathcal{C}_\emptyset}})} \right\rangle_\emptyset \\
&= \left\langle \frac{\prod_{\mathcal{C}_A\in\text{non-bulk}} h_x^{n_{a,\mathcal{C}_A}}(W_1^{n_{b,\mathcal{C}_A}} + W_2^{n_{b,\mathcal{C}_A}})}{\prod_{\mathcal{C}_\emptyset\in\text{non-bulk}} h_x^{n_{a,\mathcal{C}_\emptyset}}(W_1^{n_{b,\mathcal{C}_\emptyset}} + W_2^{n_{b,\mathcal{C}_\emptyset}})} \right\rangle_\emptyset .
\end{aligned}$$

Noting again that the number of operators $\hat{H}_{1,b}$ and $\hat{H}_{-1/1,a}$ involved in non-bulk clusters must be conserved regardless of the choice of topology,

$$\begin{aligned}
\frac{Z[2,A]}{Z[2,\emptyset]} &= \left\langle \frac{\prod_{\mathcal{C}_A\in\text{non-bulk}}(W_1^{n_{b,\mathcal{C}_A}} + W_2^{n_{b,\mathcal{C}_A}})}{\prod_{\mathcal{C}_\emptyset\in\text{non-bulk}}(W_1^{n_{b,\mathcal{C}_\emptyset}} + W_2^{n_{b,\mathcal{C}_\emptyset}})} \right\rangle_\emptyset \\
&= \left\langle \frac{\prod_{\mathcal{C}_A\in\text{non-bulk}} W_1^{n_{b,\mathcal{C}_A}}(1 + (W_2/W_1)^{n_{b,\mathcal{C}_A}})}{\prod_{\mathcal{C}_\emptyset\in\text{non-bulk}} W_1^{n_{b,\mathcal{C}_\emptyset}}(1 + (W_2/W_1)^{n_{b,\mathcal{C}_\emptyset}})} \right\rangle_\emptyset \\
&= \left\langle \frac{\prod_{\mathcal{C}_A\in\text{non-bulk}}(1 + (W_2/W_1)^{n_{b,\mathcal{C}_A}})}{\prod_{\mathcal{C}_\emptyset\in\text{non-bulk}}(1 + (W_2/W_1)^{n_{b,\mathcal{C}_\emptyset}})} \right\rangle_\emptyset .
\end{aligned} \tag{4.17}$$

Although not quite as elegant as Eq. (4.15), Eq. (4.17) is still compact and efficient.

On the contrary, when $h_{z,b} > J$, $C = 2h_{z,b} - J$ and

$$W_1 \equiv \langle\uparrow\uparrow|\hat{H}_{1,b}|\uparrow\uparrow\rangle = 4h_{z,b}, \tag{4.18a}$$

$$W_2 \equiv \langle\downarrow\downarrow|\hat{H}_{1,b}|\downarrow\downarrow\rangle = 0, \tag{4.18b}$$

and

$$W_3 \equiv \langle\downarrow\uparrow|\hat{H}_{1,b}|\downarrow\uparrow\rangle = \langle\uparrow\downarrow|\hat{H}_{1,b}|\uparrow\downarrow\rangle = 2(h_{z,b} - J). \tag{4.18c}$$

In this case, the possible set of vertices that may exist given a list of links defining multibranch clusters is not a sum over two terms ("all-$\uparrow$" and "all-$\downarrow$") as before. The possible sets of vertices will be such that no two adjacent – in space – spins in the cluster can be both in the $\downarrow$ state given that $W_2 = 0$. If $N_{\mathcal{C}}$ world lines are in a multibranch cluster $\mathcal{C}$, there will be $2^{N_{\mathcal{C}}} - x$ number of vertex configurations, where x is the number of configurations where two adjacent spins are down in the $\downarrow$ state. Fig. 4.3 shows $2^{N_{\mathcal{C}}} - x$ versus $N_{\mathcal{C}}$. Clearly, the required sum over $2^{N_{\mathcal{C}}} - x$ terms does not scale favourably with the spatial extent of the multibranch cluster, therefore making estimating Eq. (4.10) a non-scalable task.

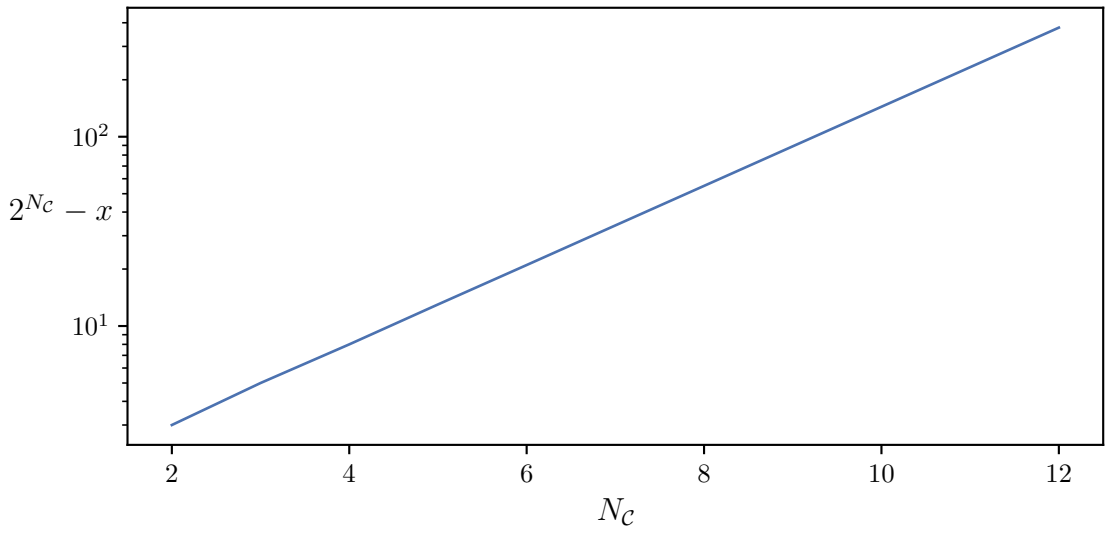

Figure 4.3: The number of possible vertex configurations, $2^{N_{\mathcal{C}}} - x$, versus the spatial extent of the multibranch cluster $N_{\mathcal{C}}$ when $h_{z,b} > J$.

4.2.3 Relation to the Rydberg SSE QMC

Given the similarity between the Rydberg and LTFIM Hamiltonians and their corresponding SSE implementations, the viability of being able to efficiently estimate Eq. (4.10) no matter the value of the Hamiltonian parameters seems to be out of reach. To make matters worse for the Rydberg SSE implementation, there exist cases where all weights $W_{ij}^{(1-4)}$ in Eq. (3.19) are non-zero. This results in having to sum over an exponentially-scaling 2^{N_c} terms.

Throughout this entire discussion, however, we've left out the newly-developed *line* clusters in favour of multibranch clusters. However, line clusters are *not independent* from other line clusters. If one line cluster has a particular vertex configuration given a list of links, its weight will depend on other line clusters near it. To summarize, implementing an algorithm that calculates Eq. (4.10) for the Rydberg SSE outlined in Ch. 3 is an algorithmically and computationally challenging task. Experimentally, the entanglement entropy can be measured at the cost of highly non-trivial and hardware-specific protocols [22, 32, 116, 117]. To get around these non-trivialities, consider the SWAP operator given a compact wavefunction representation (Eq. (4.8)). Although a QMC simulation or an experiment does not provide us with a wavefunction, they can efficiently provide us with *measurements* in a computational basis. In generating a dataset comprised of such measurements, one could overcome the challenges of estimating the SWAP operator in the SSE formalism by framing the problem in a wavefunction representation: train a neural network, like an RBM (Sec. 2.1) or an RNN (Sec. 2.2), estimate the SWAP operator using Eq. (4.8), then estimate the second Renyi entropy as

$$S_2 = -\ln\langle \text{SWAP} \rangle, \tag{4.19}$$

similarily to what was done in Ref. [32].

Instead of going through the hassle of training a neural network on QMC-generated data to obtain a parameterized wavefunction representation like in Eq. (1.8b), one may instead propose the use of a *frequency distribution* (FD),

$$p_{\text{FD}}(\boldsymbol{\sigma}) = \frac{1}{|\mathbf{D}|} \sum_{\boldsymbol{\sigma}' \in \mathbf{D}} \delta(\boldsymbol{\sigma} - \boldsymbol{\sigma}'). \tag{4.20}$$

However, Torlai *et al.* in Ref. [32] showed that the fidelity between the FD wavefunction $\psi_{\text{FD}}(\boldsymbol{\sigma}) = \sqrt{p_{\text{FD}}(\boldsymbol{\sigma})}$ and the target state decays exponentially with system size, therefore rendering the use of FD wavefunctions a moot point.

4.3 Second Renyi entropy reconstruction from Rydberg SSE QMC data

The $L \times L$ systems of interest in Sec. 3.4 will again be the focus here. In generating training datasets comprising of Rydberg occupation measurements of the $N = L \times L$ sites from the SSE QMC algorithm in Sec. 3.3, we must ensure that each sample is independent and uncorrelated. Given that the absolute value of the staggered magnetization $|M_s|$ across the checkerboard transition at $R_b = 1.2$ characterizes the physics at hand, it is reasonable to assume that its correlation time $\tau_{|M_s|}$ dictates the Markov time separation required for Rydberg occupation samples to be uncorrelated. The last requirement to perform quantum state reconstruction like in Secs. 2.4.2 and 2.5.2 is the stoquasticity of the Hamiltonian in Eq. (3.1). The trivial sign cure to Eq. (3.1) is given in Eq. (3.5), which satisfies the stoquasticity requirement. Therefore, Rydberg occupation measurements $(n_1, n_2, \cdots, n_N)$ suffice.

The data generation scheme that was employed here was as follows. Using the correlation times in Fig. 3.10 as a guide, samples from the SSE QMC algorithm were drawn at specific Markov time intervals. With 15 different random seeds, each random seed generated 2000 Rydberg occupation samples, culminating in $|\mathbf{D}| = 3 \times 10^5$ measurements for each δ/Ω value. The hyperparameter choices used to train the RBMs and RNNs on the QMC-generated Rydberg occupation measurement datasets $\mathbf{D}$ were guided by the results in Secs. 2.4.2 and 2.5.2. However, given the two-dimensional architecture, we instead use the 2D-RNN recurrent cell given in Eq. (2.18). The hyperparameter choices for both RBMs and RNNs are summarized in Tabs. 4.2 and 4.3. Again, for a fair comparison between both generative models, both the RBM and RNN hidden unit values n_h were chosen such that the total number of tunable parameters was approximately the same. All n_h values used are given in Tab. 4.1.

Each RBM and RNN was trained for 2×10^5 epochs and then 10^5 Rydberg occupation measurements were drawn to calculate the energy E, the absolute value of the staggered magnetization $|M_s|$, and the second Renyi entropy S_2 using Eq. (4.19). The bipartition scheme A and $\bar{A}$ that was used to calculate S_2 was to split the square $L \times L$ lattices directly in half. Although the specific values of the reconstructed S_2 from either generative model is not of interest across the transition, the goal of these results is to offer some perspective on 1) using this method for measuring Renyi entropies as opposed to experimentally or variationally (i.e. using variational methods like variational Monte Carlo or DMRG), and 2) what can be extrapolated from comparisons on using RBMs or RNNs in this setting.

Figs. 4.4 and 4.5 show the reconstructed energy density E/N, absolute value of the stag-

Table 4.1: RBM and RNN hidden unit values n_h used for training on $L \times L$ Rydberg occupation data.

L	RBM n_h (# of parameters)	RNN n_h (# of parameters)
4	119 (2039)	30 (2042)
	293 (4997)	48 (4994)
	588 (10,012)	69 (10,076)
10	19 (2019)	30 (2042)
	49 (5049)	48 (4994)
	99 (10,099)	69 (10,076)

Table 4.2: Hyperparameters used for training RBMs on $L \times L$ Rydberg data ($L = 4$ and 10).

Hyperparameter	Value / Type
Optimizer	SGD
Learning rate	0.001×0.9999^t
Positive batch size	100
Negative batch size	200
k CD_k steps	10
Hidden units	See Tab. 4.1

Table 4.3: Hyperparameters used for training RNNs on $L \times L$ Rydberg data ($L = 4$ and 10).

Hyperparameter	Value / Type
Loss function	Cross Entropy
Optimizer	SGD
RNN recurrent cell	2D (Eq. 2.18)
Learning rate	0.001
Batch size	100
Hidden units	See Tab. 4.1

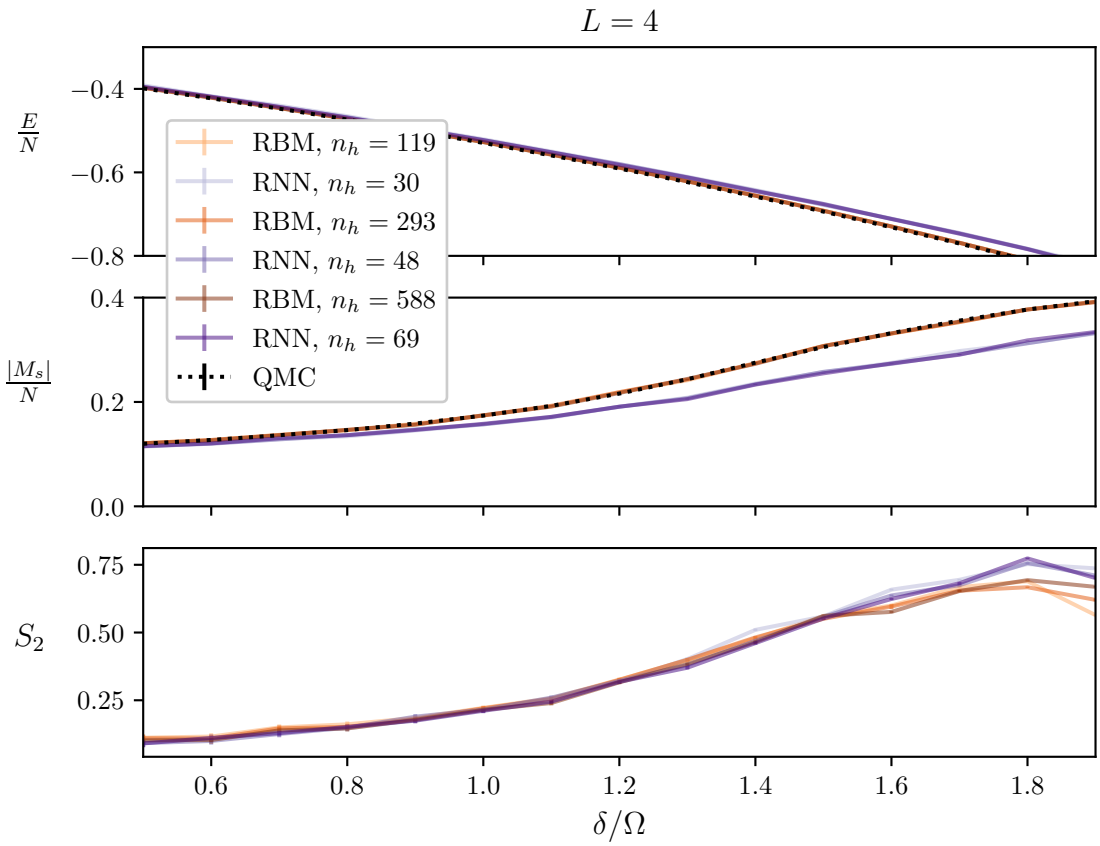

Figure 4.4: The RBM (orange) and RNN (purple) reconstructed energy density E/N, absolute value of the staggered magnetization density $|M_s|/N$, and second Renyi entropy S_2 observables across the checkerboard transition at $R_b = 1.2$ for an $N = 4 \times 4$ lattice. The black dashed line in the E/N and $|M_s|/N$ subplots is the corresponding observable as calculated from the QMC simulation that generated the training datasets $\mathbf{D}$. Three different hidden unit values n_h for the RBM and RNN were used to reconstruct the aforementioned observables. Error bars are smaller than the line widths.

gered magnetization density $|M_s|/N$, and second Renyi entropy S_2 across the checkerboard transition at $R_b = 1.2$ for $L = 4$ and 10. For both system sizes and all values of n_h, the RNN's reconstructed $|M_s|$ is well below that of the QMC's in the ordered phase, whereas the RBM's reconstructed $|M_s|$ is nearly identical to the QMC's across the entire range of δ/Ω values. However, these observations do not perfectly carry over to the reconstructed energy. The RNN's reconstructed energy density steers away from the QMC value in the

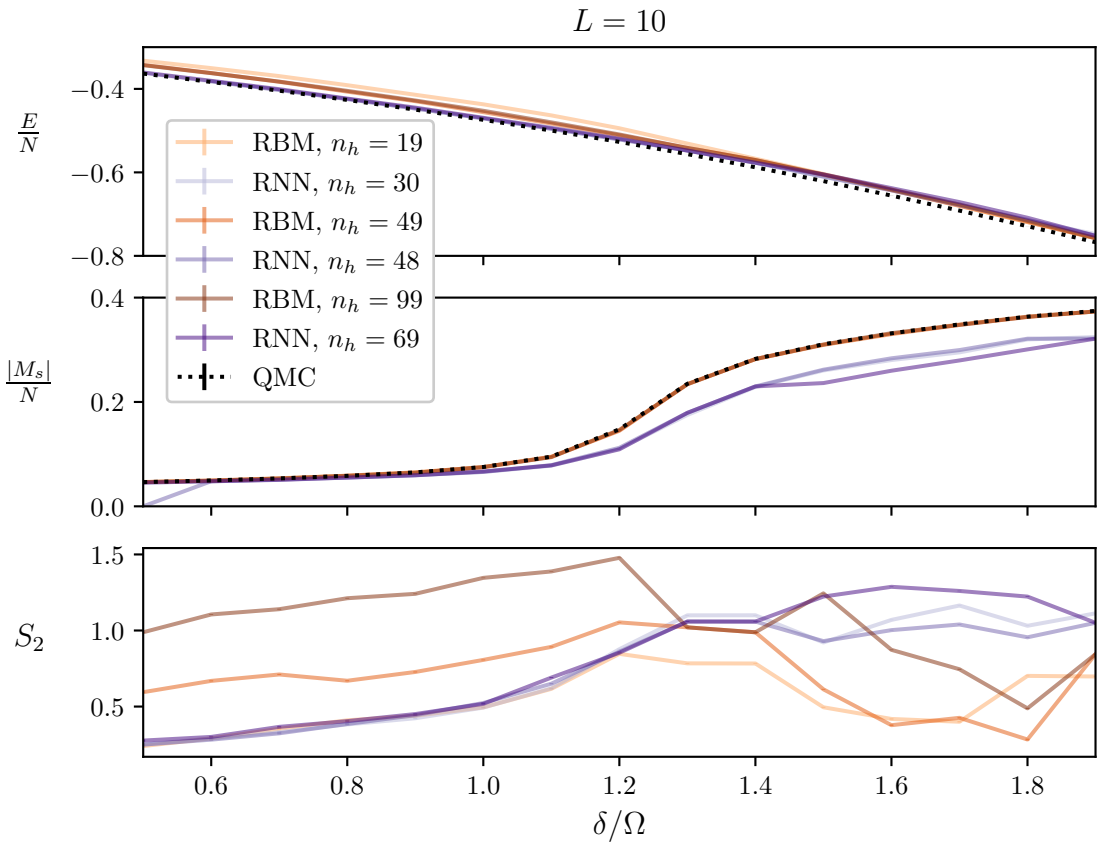

Figure 4.5: The RBM (orange) and RNN (purple) reconstructed energy density E/N, absolute value of the staggered magnetization density $|M_s|/N$, and second Renyi entropy S_2 observables across the checkerboard transition at $R_b = 1.2$ for an $N = 10 \times 10$ lattice. The black dashed line in the E/N and $|M_s|/N$ subplots is the corresponding observable as calculated from the QMC simulation that generated the training datasets $\mathbf{D}$. Three different hidden unit values n_h for the RBM and RNN were used to reconstruct the aforementioned observables. Error bars are smaller than the line widths.

ordered phase like with the reconstructed $|M_s|$, but the RBM's reconstructed energy for the $L = 10$ system is above the QMC energy throughout the entire δ/Ω range. In general, for this transition, accurately reconstructed order parameters $|M_s|$ do not translate into accurately reconstructed energies. Moreover, for the hyperparameters in Tab. 4.2 and 4.3, increasing the number of hidden units n_h in hopes of increasing the reconstruction accuracy has little to no effect on the reconstructed $|M_s|$ and energy; a *saturation* point is reached

at the smallest n_h value employed here.

More striking differences between the RBM and RNN can be seen in their respective reconstructed second Renyi entropies. For $L = 4$, we again see that both generative models have reached a saturation point with the smallest value of n_h, as there are no glaring qualitative differences between the reconstructed S_2 across the δ/Ω range for the given values of n_h. However, for $L = 10$, increasing the value of n_h in the RBM leads to unpredictable volatility across the phase transition.[3] Comparatively, increasing the RNN's number of hidden units seems to smoothen the reconstructed S_2 in the ordered phase. In all, the discrepancy between accurate reconstruction with some observables and inaccurate reconstruction with others is apparent for both generative models across this transition, which all but complicates the general conclusions that can be made with respect to the trainability and expressivity of the RBM and RNN.

Regarding the actual accuracies of the second Renyi entropy reconstructions, there are no exact values to compare to at the time of writing. However, Samajdar *et al.* in Ref. [92] reported DMRG results on the bipartite second Renyi entropy for square lattices on a cylinder. Although on a different lattice and finite-size effects must be taken into consideration, the disordered-checkerboard phase transition reported in Ref. [92] demonstrated a peak in S_2 at the critical point that dropped sharply in the ordered phase. Although the range of δ/Ω values used here is a small window, it does appear that the reconstructed values of S_2 reported here start to decrease well into the ordered phase.

[3] In fact, one cannot say anything regarding the presence of a phase transition from simply looking at the RBM's reconstructed second Renyi entropy for $L = 10$.

Chapter 5

Conclusions

5.1 Summary

In Ch. 2 of this thesis, the underlying theory behind two prominent generative models, the RBM and RNN, was outlined. Comparisons were then made between both of these generative models using physically-motivated quantum state reconstruction tasks for ground states of stoquastic – or approximately-stoquastic Hamiltonians – models: the XY model and nanomolecular assemblies of endofullerenes. In both cases, determining RBM hyperparameters that yielded stability during the training process and, ultimately, accurate reconstruction of relevant observables was difficult compared to determining RNN hyperparameters that provided the aforementioned characteristics. Not only this, but the RNN's ability to learn discrete symmetries underlying each model's ground state as compared to the RBM was highlighted. Lastly, the correlation times intrinsic to sampling the RBM as compared to autoregressive sampling in RNNs all but confirms the RNNs advantage over RBMs for symmetry-laden quantum state reconstruction tasks that are experimentally relevant.

Impelled by the inclination towards a reconstruction task that is relevant to one of *today*'s quantum hardware juggernauts – Rydberg atoms – while noting that synthetic measurement data might still be required as an intermediary outlet, a ground state projector SSE QMC algorithm for simulating Rydberg atom arrays was developed in Ch. 3. Notably, a derivation was given for a ground state energy estimator applicable to SSE QMC algorithms of the same flavour as the one developed in this thesis. To assess the algorithm's performance, the known disordered-checkerboard transition in two spatial dimensions on $L \times L$ lattices was qualitatively confirmed for system sizes up to $N = 16 \times 16$.

The energy estimator derivation in Sec. 3.3.4 showcased how non-trivial certain off-diagonal estimators can be to formulate in the language of a given QMC flavour. However, feasibly extracting estimates for some off-diagonal observables is vital to the usefulness of any algorithm. Although not a traditional observable, the entanglement entropy is a quantity that is highly sought after due to the signatures it provides for certain physical phenomena. By using the well-known replica trick [112], the second Renyi entropy can be estimated in the ground state SSE QMC formalism. However, for the SSE QMC algorithm developed in this thesis for Rydberg atoms, traditional second Renyi entropy estimations possess an intrinsic curse of dimensionality. However, given access to a compact wavefunction representation, estimating the second Renyi entropy is more feasible.

Guided by the results in Ch. 2 and Ch. 3, RNNs and RBMs were trained on QMC-generated datasets in Ch. 4 to reconstruct the Rydberg energy, absolute value of the staggered magnetization, and second Renyi entropy for $L \times L$ lattices up to $L = 10$ across the disordered-checkerboard transition. Interestingly, the resulting reconstruction accuracy from the RBMs and RNNs varied depending on the observable. For instance, the RNN's reconstructed staggered magnetization was well off of the QMC-calculated value in the ordered phase compared to the RBM, but the reconstructed second Renyi entropy was without question more stable and indicative of the physics involved in the disordered-checkerboard transition. Therefore, the RBMs and RNNs trained in Ch. 4 seemed to be biased towards learning different aspects of the physics at hand, be they related to entanglement or the order parameter.

5.2 Future work

A theme pervading the results presented in Ch. 2 was that the RBMs trained were unstable even after performing a modest manual hyperparameter search compared to RNNs. Cumbersome manual labour in the preprocessing stages of research alone is enough of a deterrent to most, which makes RNNs all the more appealing when put next to RBMs. That being said, making definitive statements regarding comparisons between RNN and RBM trainability and expressivity for quantum state reconstruction tasks still requires more empirical and theoretical work.

The SSE QMC algorithm developed for Rydberg atoms displayed promising results for the disordered-checkerboard transition in two spatial dimensions at $R_b = 1.2$. However, the amount of realizable phenomena contained in ground states of Rydberg atom arrays is staggering [85–90, 118, 119]. In this spirit, the QMC algorithm developed in this thesis

should be utilized for rigorous finite-size scaling studies to add to the growing amount of DMRG and variational Monte Carlo work in the literature [91, 92, 118, 120, 121].

Although variational methods are much different in their construction compared to quantum state reconstruction, the results for training generative models on QMC-generated Rydberg occupation data presented in this thesis demonstrate the inherent bias present in assuming a representational form of the target ground state wavefunction. In this case, RBMs learned the order parameter of the relevant transition much better, whereas RNNs seemed to learn the entanglement. The SSE QMC algorithm presented in this thesis is an exact representation of the problem at hand with the catch being the presence of autocorrelation times. Hopefully, this algorithm will offer a much-needed perspective.

The hybrid approach of using neural networks trained on QMC-generated data as mediums for estimating non-trivial quantities in the given QMC flavour must be compared to other methods wherein those quantities are feasibly measured. Arguably, variational methods require less heavy lifting than the aforementioned hybrid approach in that a single optimization routine takes place in favour of a full QMC simulation and then an optimization routine. The hybrid approach is founded in an exact representation in that the biasedly represented generative model is trained on an exact representational form. That being said, both the hybrid approach and variational methods possess elements of bias as evident in Sec. 4.3 and therefore should be fervently compared.

References

[1] S. Morawetz, I. J. S. De Vlugt, J. Carrasquilla, and R. G. Melko. U(1)-symmetric recurrent neural networks for quantum state reconstruction. *Phys. Rev. A*, 104:012401, 2021.

[2] I. J. S. De Vlugt, D. Iouchtchenko, E. Merali, P.-N. Roy, and R. G. Melko. Reconstructing quantum molecular rotor ground states. *Phys. Rev. B*, 102(3):035108, 2020.

[3] E. Merali, I. J. S. De Vlugt, and R. G. Melko. Stochastic series expansion quantum monte carlo for rydberg arrays. *arXiv:2107.00766*, 2021.

[4] G. Carleo and M. Troyer. Solving the Quantum Many-Body Problem with Artificial Neural Networks. *Science*, 355(6325):602, 2017.

[5] J. Carrasquilla. Machine learning for quantum matter. *Advances in Physics: X*, 5(1):1797528, 2020.

[6] S. R. White. Density Matrix Formulation for Quantum Renormalization Groups. *Phys. Rev. Lett.*, 69(19):2863, 1992.

[7] S. R. White. Density-Matrix Algorithms for Quantum Renormalization Groups. *Phys. Rev. B*, 48(14):10345, 1993.

[8] B. M. Austin, D. Yu. Zubarev, and W. A. Lester. Quantum Monte Carlo and Related Approaches. *Chem. Rev.*, 112(1):263–288, 2012.

[9] M. Cerezo, A. Arrasmith, R. Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles. Variational Quantum Algorithms. *Nat Rev Phys*, 3(9):625–644, 2021. arXiv: 2012.09265.

[10] U. Schollwoeck. The Density-Matrix Renormalization Group in the Age of Matrix Product States. *Annals of Physics*, 326(1):96, 2011.

[11] A. Peruzzo, J. McClean, Peter 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. *Nat Commun*, 5(1):4213, 2014.

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

[13] A. W. Sandvik. A generalization of Handscomb's quantum Monte Carlo scheme-application to the 1D Hubbard model. *J. Phys. A: Math. Gen.*, 25(13):3667–3682, 1992.

[14] D. P. Landau and K. Binder. *A Guide to Monte Carlo Simulations in Statistical Physics*. Cambridge University Press, 4 edition, 2014.

[15] J. A. Barker. A quantum-statistical Monte Carlo method; path integrals with boundary conditions. *J. Chem. Phys.*, 70(6):2914–2918, 1979.

[16] P. Henelius and A. W. Sandvik. Sign problem in Monte Carlo simulations of frustrated quantum spin systems. *Phys. Rev. B*, 62(2):1102–1113, 2000.

[17] R. K. Kaul, R. G. Melko, and A. W. Sandvik. Bridging Lattice-Scale Physics and Continuum Field Theory with Quantum Monte Carlo Simulations. *Annual Review of Condensed Matter Physics*, 4(1):179–215, 2013.

[18] L. Gupta and I. Hen. Elucidating the Interplay between Non-Stoquasticity and the Sign Problem. *Advanced Quantum Technologies*, 3(1):1900108, 2020.

[19] R. Blatt and C. F. Roos. Quantum simulations with trapped ions. *Nature Phys*, 8(4):277–284, 2012.

[20] K. Kim, M.-S. Chang, S. Korenblit, R. Islam, E. E. Edwards, J. K. Freericks, G.-D. Lin, L.-M. Duan, and C. Monroe. Quantum simulation of frustrated Ising spins with trapped ions. *Nature*, 465(7298):590–593, 2010.

[21] G.-D. Lin, S.-L. Zhu, R. Islam, K. Kim, M.-S. Chang, S. Korenblit, C. Monroe, and L.-M. Duan. Large-scale quantum computation in an anharmonic linear ion trap. *EPL (Europhysics Letters)*, 86(6):60004, 2009.

[22] K. Kim, M.-S. Chang, R. Islam, S. Korenblit, L.-M. Duan, and C. Monroe. Entanglement and tunable spin-spin couplings between trapped ions using multiple transverse modes. *Physical Review Letters*, 103(12), 2009.

[23] I. Bloch, J. Dalibard, and S. Nascimbène. Quantum simulations with ultracold quantum gases. *Nature Physics*, 8(4):267–276, 2012.

[24] A. Aspuru-Guzik and P. Walther. Photonic quantum simulators. *Nature Physics*, 8(4):285–291, 2012.

[25] D. E. Chang, V. Vuletić, and M. D. Lukin. Quantum nonlinear optics — photon by photon. *Nature Photonics*, 8(9):685–694, 2014.

[26] L. Henriet, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Reymond, and C. Jurczak. Quantum computing with neutral atoms. *Quantum*, 4:327, 2020.

[27] A. Browaeys and T. Lahaye. Many-body physics with individually controlled rydberg atoms. *Nature Physics*, 16(2):132–142, 2020.

[28] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, 2018.

[29] 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(4):045002, 2019.

[30] M. Beach, I. De Vlugt, A. Golubeva, P. Huembeli, B. Kulchytskyy, X. Luo, R. Melko, E. Merali, and G. Torlai. QuCumber: Wavefunction Reconstruction with Neural Networks. *SciPost Phys.*, 7(1):009, 2019.

[31] G. Torlai. *Augmenting Quantum Mechanics with Artificial Intelligence*. PhD Thesis, UWSpace, 2018.

[32] G. Torlai, B. Timar, E. P. L. van Nieuwenburg, H. Levine, A. Omran, A. Keesling, H. Bernien, M. Greiner, V. Vuletićć, M. D. Lukin, R. G. Melko, and M. Endres. Integrating Neural Networks with a Quantum Simulator for State Reconstruction. *Phys. Rev. Lett.*, 123(23):230504, 2019.

[33] G. Torlai and R. G. Melko. Latent Space Purification via Neural Density Operators. *Phys. Rev. Lett.*, 120(24):240503, 2018.

[34] G. Torlai and R. G. Melko. Learning Thermodynamics with Boltzmann Machines. *Phys. Rev. B*, 94(16):165134, 2016.

[35] G. Torlai and R. G. Melko. Machine-Learning Quantum States in the NISQ Era. *Annu. Rev. Condens. Matter Phys.*, 11(1):325, 2020.

[36] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. G. Melko, and G. Carleo. Neural-Network Quantum State Tomography. *Nat. Phys.*, 14(5):447–450, 2018.

[37] J. Carrasquilla, G. Torlai, R. G. Melko, and L. Aolita. Reconstructing Quantum States with Generative Models. 2018. arXiv: 1810.10584.

[38] A. I. Lvovsky. Iterative maximum-likelihood reconstruction in quantum homodyne tomography. *Journal of Optics B: Quantum and Semiclassical Optics*, 6(6):S556–S559, 2004.

[39] D. F. V. James, P. G. Kwiat, W. J. Munro, and A. G. White. Measurement of qubits. *Phys. Rev. A*, 64(5):052312, 2001.

[40] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert. Quantum state tomography via compressed sensing. *Phys. Rev. Lett.*, 105(15):150401, 2010.

[41] M. Cramer, M. B. Plenio, S. T. Flammia, D. Gross, S. D. Bartlett, R. Somma, O. Landon-Cardinal, Y.-K. Liu, and D. Poulin. Efficient Quantum State Tomography. *Nature Communications*, 1(9):149, 2010.

[42] G. Tóth, W. Wieczorek, D. Gross, R. Krischek, C. Schwemmer, and H. Weinfurter. Permutationally Invariant Quantum Tomography. *Phys. Rev. Lett.*, 105(25):250403, 2010.

[43] B. P. Lanyon, C. Maier, M. Holzäpfel, T. Baumgratz, C. Hempel, P. Jurcevic, I. Dhand, A. S. Buyskikh, A. J. Daley, M. Cramer, M. B. Plenio, R. Blatt, and C. F. Roos. Efficient tomography of a quantum many-body system. *Nature Phys*, 13(12):1158–1162, 2017.

[44] P. Tarazaga, M. Raydan, and A. Hurman. Perron–Frobenius theorem for matrices with some negative entries. *Linear Algebra and its Applications*, 328(1-3):57, 2001.

[45] S. Bravyi, D. P. DiVincenzo, Roberto I. Oliveira, and B. M. Terhal. The complexity of stoquastic local hamiltonian problems. 2006. arXiv:quant-ph/0606140v4.

[46] D. Sehayek, A. Golubeva, M. Albergo, B. Kulchytskyy, G. Torlai, and R. Melko. The Learnability Scaling of Quantum States: Restricted Boltzmann Machines. *Phys. Rev. B*, 100(19):195125, 2019.

[47] M. E. J. Newman and G.T. Barkema. *Monte Carlo Methods in Statistical Physics*. Clarendon Press, 1998.

[48] N. Metropolis and S. Ulam. The Monte Carlo Method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

[49] G. E. Hinton. A Practical Guide to Training Restricted Boltzmann Machines. In G. Montavon, G. B. Orr, and K.-R. Müller, editors, *Neural Networks: Tricks of the Trade*, volume 7700, page 599. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[50] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted Boltzmann Machines for Collaborative Filtering. In *Proceedings of the 24th International Conference on Machine Learning - ICML '07*, page 791. ACM Press, 2007.

[51] G. E. Hinton. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, 14(8):1711–1800, 2002.

[52] K.P. Murphy. Machine Learning: A Probabilistic Perspective. In *Machine Learning: A Probabilistic Perspective*. The MIT Press, Cambridge, Massachusetts, 2012.

[53] M. Á. Carreira-Perpiñan and G. Hinton. On contrastive divergence learning. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 33–40. Society for Artificial Intelligence and Statistics, 2005.

[54] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. 2019.

[55] B. Widrow and M.A. Lehr. 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proceedings of the IEEE*, 78(9):1415–1442, 1990.

[56] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: a survey. 2018. arXiv:1502.05767.

[57] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[58] M. Innes, E. Saba, K. Fischer, D. Gandhi, M. Concetto Rudilosso, N. Mariya Joy, T. Karmali, A. Pal, and V. Shah. Fashionable modelling with flux. *CoRR*, abs/1811.01457, 2018.

[59] M. Innes. Flux: Elegant machine learning with julia. *Journal of Open Source Software*, 2018.

[60] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[61] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 1997.

[62] 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. 2014. arXiv:1406.1078v3.

[63] P. Richerme, Z.-X. Gong, A. Lee, C. Senko, J. Smith, M. Foss-Feig, S. Michalakis, A. V. Gorshkov, and C. Monroe. Non-local propagation of correlations in quantum systems with long-range interactions. *Nature*, 511(7508):198–201, 2014.

[64] P. Jurcevic, B. P. Lanyon, P. Hauke, C. Hempel, P. Zoller, R. Blatt, and C. F. Roos. Quasiparticle engineering and entanglement propagation in a quantum many-body system. *Nature*, 511(7508):202–205, 2014.

[65] C. Monroe, W. C. Campbell, L.-M. Duan, Z.-X. Gong, A. V. Gorshkov, P. W. Hess, R. Islam, K. Kim, N. M. Linke, G. Pagano, P. Richerme, C. Senko, and N. Y. Yao. Programmable quantum simulations of spin systems with trapped ions. *Rev. Mod. Phys.*, 93:025001, 2021.

[66] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. volume 28 of *Proceedings of Machine Learning Research*, pages 1310–1318. PMLR, 2013.

[67] R. J. Williams and J. Peng. An efficient gradient-based algorithm for on-line training of recurrent network trajectories. *Neural Computation*, 2(4):490–501, 1990.

[68] A. J Ferris and G. Vidal. Perfect Sampling with Unitary Tensor Networks. *Phys. Rev. B*, 85(16):165146, 2012.

[69] *ITensor Library (version 2.0.11) http://itensor.org.*

[70] D. Gottesman, A. Kitaev, and J. Preskill. Encoding a qubit in an oscillator. *Phys. Rev. A*, 64(1):012310, 2001.

[71] V. V. Albert, J. P. Covey, and J. Preskill. Robust Encoding of a Qubit in a Molecule. 2019. arXiv: 1911.00099.

[72] S. Berber, Y.-K. Kwon, and D. Tománek. Microscopic formation mechanism of nanotube peapods. *Phys. Rev. Lett.*, 88(18):185502, 2002.

[73] K. Komatsu, M. Murata, and Y. Murata. Encapsulation of molecular hydrogen in fullerene C60 by organic synthesis. *Science*, 307(5707):238, 2005.

[74] A. Krachmalnicoff, R. Bounds, S. Mamone, S. Alom, M. Concistrè, B. Meier, Karel Kouřil, M. E. Light, M. R. Johnson, S. Rols, A. J. Horsewill, A. Shugai, U. Nagel, T. Rõõm, M. Carravetta, M. H. Levitt, and R. J. Whitby. The Dipolar Endofullerene HF@C60. *Nat. Chem.*, 8:953, 2016.

[75] M. Xu, S. Ye, and Z. Bačić. General Selection Rule in the Inelastic Neutron Scattering Spectroscopy of a Diatomic Molecule Confined Inside a Near-Spherical Nanocavity. *J. Phys. Chem. Lett.*, 6(18):3721, 2015.

[76] P. M. Felker and Z. Bačić. Electric-Dipole-Coupled H20@C60 Dimer: Translation-Rotation Eigenstates from Twelve-Dimensional Quantum Calculations. *J. Chem. Phys.*, 146(8):084303, 2017.

[77] D. Iouchtchenko and P. N. Roy. Ground States of Linear Rotor Chains via the Density Matrix Renormalization Group. *J. Chem. Phys.*, 148(13):194110, 2018.

[78] T. Halverson, D. Iouchtchenko, and P. N. Roy. Quantifying Entanglement of Rotor Chains Using Basis Truncation: Application to Dipolar Endofullerene Peapods. *J. Chem. Phys.*, 148(7):074112, 2018.

[79] Yulia N. Kalugina and Pierre-Nicholas Roy. Potential energy and dipole moment surfaces for HF@C60: Prediction of spectral and electric response properties. *J. Chem. Phys.*, 147(24):244303, 2017.

[80] B. P. Abolins, R. E. Zillich, and K. B. Whaley. A Ground State Monte Carlo Approach for Studies of Dipolar Systems with Rotational Degrees of Freedom. *J. Low Temp. Phys.*, 165(5):249, 2011.

[81] T. Sahoo, D. Iouchtchenko, C. M. Herdman, and P.-N. Roy. A Path Integral Ground State Replica Trick Approach for the Computation of Entanglement Entropy of Dipolar Linear Rotors. *J. Chem. Phys.*, 152(18):184113, 2020.

[82] D. Jaksch, J. I. Cirac, P. Zoller, S. L. Rolston, R. Côté, and M. D. Lukin. Fast quantum gates for neutral atoms. *Phys. Rev. Lett.*, 85:2208–2211, 2000.

[83] M. D. Lukin, M. Fleischhauer, R. Cote, L. M. Duan, D. Jaksch, J. I. Cirac, and P. Zoller. Dipole blockade and quantum information processing in mesoscopic atomic ensembles. *Phys. Rev. Lett.*, 87:037901, 2001.

[84] P. Fendley, K. Sengupta, and S. Sachdev. Competing density-wave orders in a one-dimensional hard-boson model. *Phys. Rev. B*, 69:075106, 2004.

[85] H. Bernien, S. Schwartz, A. Keesling, H. Levine, A. Omran, Hannes Pichler, Soonwon Choi, Alexander S. Zibrov, Manuel Endres, Markus Greiner, Vladan Vuletić, and Mikhail D. Lukin. Probing Many-Body Dynamics on a 51-Atom Quantum Simulator. *Nature*, 551(7682):579–584, 2017.

[86] S. Ebadi, T. T. Wang, H. Levine, A. Keesling, G. Semeghini, A. Omran, D. Bluvstein, R. Samajdar, H. Pichler, W. W. Ho, S. Choi, S. Sachdev, M. Greiner, V. Vuletic, and M. D. Lukin. Quantum Phases of Matter on a 256-Atom Programmable Quantum Simulator. 2020. arXiv:2012.12281.

[87] F. Liu, S. Whitsitt, P. Bienias, R. Lundgren, and A. V. Gorshkov. Realizing and Probing Baryonic Excitations in Rydberg Atom Arrays. 2020. arXiv:2007.07258.

[88] S. Whitsitt, R. Samajdar, and S. Sachdev. Quantum field theory for the chiral clock transition in one spatial dimension. *Phys. Rev. B*, 98(20):205118, 2018.

[89] M. C. Bañuls, R. Blatt, J. Catani, A. Celi, J. I. Cirac, M. Dalmonte, L. Fallani, K. Jansen, M. Lewenstein, S. Montangero, C. A. Muschik, B. Reznik, E. Rico, L. Tagliacozzo, K. Van Acoleyen, F. Verstraete, U.-J. Wiese, M. Wingate, J. Zakrzewski, and P. Zoller. Simulating Lattice Gauge Theories within Quantum Technologies. *Eur. Phys. J. D*, 74(8):165, 2020.

[90] P. Nikolić and T. Senthil. Theory of the kagome lattice Ising antiferromagnet in weak transverse fields. *Phys. Rev. B*, 71(2):024401, 2005.

[91] R. Samajdar, S. Choi, H. Pichler, M. D. Lukin, and S. Sachdev. Numerical study of the chiral z3 quantum phase transition in one spatial dimension. *Phys. Rev. A*, 98(2):023614, 2018.

[92] R. Samajdar, W. W. Ho, H. Pichler, M. D. Lukin, and S. Sachdev. Complex Density Wave Orders and Quantum Phase Transitions in a Model of Square-Lattice Rydberg Atom Arrays. *Phys. Rev. Lett.*, 124(10):103601, 2020.

[93] 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(4), 2021.

[94] 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. Vuletic, and M. D. Lukin. Probing topological spin liquids on a programmable quantum simulator. 2021. arXiv:2104.04119.

[95] A. W. Sandvik and J. Kurkijärvi. Quantum Monte Carlo simulation method for spin systems. *Phys. Rev. B*, 43(7):5950–5961, 1991.

[96] A. W. Sandvik. Stochastic series expansion method for quantum Ising models with arbitrary interactions. *Phys. Rev. E*, 68(5):056701, 2003.

[97] N. Prokof'ev and B. Svistunov. Worm Algorithms for Classical Statistical Models. *Phys. Rev. Lett.*, 87(16):160601, 2001.

[98] H. G. Evertz. The Loop Algorithm. *Advances in Physics*, 52(1):1–66, 2003.

[99] O. F. Syljuasen and A. W. Sandvik. Quantum Monte Carlo with Directed Loops. *Phys. Rev. E*, 66(4):046701, 2002.

[100] A. J. Walker. New fast method for generating discrete random numbers with arbitrary frequency distributions. *Electronics Letters*, 10:127–128, 1974.

[101] A. J. Walker. An efficient method for generating discrete random variables with general distributions. *ACM Trans. Math. Softw.*, 3(3):253–256, 1977.

[102] M. D. Vose. A linear algorithm for generating random numbers with a given distribution. *IEEE Trans. Softw. Eng.*, 17(9):972–975, 1991.

[103] R. G. Melko. Stochastic Series Expansion Quantum Monte Carlo. In Adolfo Avella and Ferdinando Mancini, editors, *Strongly Correlated Systems*, volume 176, pages 185–206. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[104] S. Inglis. Implementations and applications of Renyi entanglement in Monte Carlo simulations of spin models. 2013.

[105] R. H. Swendsen and J.-S. Wang. Nonuniversal critical dynamics in monte carlo simulations. *Phys. Rev. Lett.*, 58:86–88, Jan 1987.

[106] T. Nakamura. Efficient Monte Carlo algorithm in quasi-one-dimensional Ising spin systems. *Phys. Rev. Lett.*, 101(21):210602, 2008.

[107] N. Kawashima and K. Harada. Recent Developments of World-Line Monte Carlo Methods. *J. Phys. Soc. Jpn.*, 73(6):1379–1414, 2004.

[108] R. G. Melko. Simulations of Quantum XXZ Models on Two-Dimensional Frustrated Lattices. *J. Phys.: Condens. Matter*, 19(14):145203, 2007.

[109] M. Wallerberger. Efficient estimation of autocorrelation spectra. 2019. arXiv:1810.05079.

[110] V. Ambegaokar and M. Troyer. Estimating errors reliably in monte carlo simulations of the ehrenfest model. *American Journal of Physics*, 78(2):150–157, 2010.

[111] D. A. Abanin, E. Altman, I. Bloch, and M. Serbyn. Many-body localization, thermalization, and entanglement. *Rev. Mod. Phys.*, 91(2):021001, May 2019.

[112] P. Calabrese and J. Cardy. Entanglement Entropy and Quantum Field Theory. *J. Stat. Mech.: Theor. Exp.*, 2004(06):P06002, June 2004.

[113] M. B. Hastings, I. González, A. B. Kallin, and R. G. Melko. Measuring Renyi Entanglement Entropy in Quantum Monte Carlo Simulations. *Phys. Rev. Lett.*, 104(15):157201, April 2010. Publisher: American Physical Society.

[114] B. Kulchytskyy. Probing universality with entanglement entropy via quantum Monte Carlo. 2019.

[115] F. Alet, S. Capponi, N. Laflorencie, and M. Mambrini. Valence bond entanglement entropy. *Physical Review Letters*, 99(11), Sep 2007.

[116] Rajibul Islam, Ruichao Ma, Philipp M. Preiss, M. Eric Tai, Alexander Lukin, Matthew Rispoli, and Markus Greiner. Measuring entanglement entropy in a quantum many-body system. *Nature*, 528(7580):77–83, December 2015.

[117] Tiff Brydges, Andreas Elben, Petar Jurcevic, Benoît Vermersch, Christine Maier, Ben P. Lanyon, Peter Zoller, Rainer Blatt, and Christian F. Roos. Probing Rényi entanglement entropy via randomized measurements. *Science*, 364(6437):260–263, April 2019. Publisher: American Association for the Advancement of Science.

[118] R. Verresen, M. D. Lukin, and A. Vishwanath. Prediction of Toric Code Topological Order from Rydberg Blockade. 2020. arXiv:2011.12310.

[119] S. de Léséleuc, V. Lienhard, P. Scholl, D. Barredo, S. Weber, N. Lang, H. P. Büchler, T. Lahaye, and A. Browaeys. Observation of a symmetry-protected topological phase of interacting bosons with Rydberg atoms. *Science*, 365(6455):775–780, 2019.

[120] J. Carrasquilla and G. Torlai. Neural networks in quantum many-body physics: a hands-on tutorial. January 2021. arXiv2101.11099.

[121] 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(4), 2021.

[122] 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*, pages 103–111, Doha, Qatar, 2014. Association for Computational Linguistics.

[123] Z. C. Lipton, J. Berkowitz, and C. Elkan. *A Critical Review of Recurrent Neural Networks for Sequence Learning*. 2015. arXiv:1506.00019.

[124] S. Berber, Y.-K. Kwon, and D. Tománek. Microscopic Formation Mechanism of Nanotube Peapods. *Phys. Rev. Lett.*, 88(18):185502, 2002.

[125] J. Carrasquilla and R. G. Melko. Machine Learning Phases of Matter. *Nat. Phys.*, 13(5):431, 2017.

[126] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural Language Processing (Almost) from Scratch. *Natural Language Processing*, page 45.

[127] D. Deng, X. Li, and S. Sarma. Quantum Entanglement in Neural Network States. *Phys. Rev. X*, 7(2):021021, 2017. arXiv: 1701.04844.

[128] D. Gottesman, A. Kitaev, and J. Preskill. Encoding a Qubit in an Oscillator. *Phys. Rev. A*, 64(1):012310, 2001.

[129] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Process. Mag.*, 29(6):82, 2012.

[130] G. E. Hinton. Learning Multiple Layers of Representation. *Trends in Cognitive Sciences*, 11(10):428–434, 2007.

[131] Y. N. Kalugina and P.-N. Roy. Potential Energy and Dipole Moment Surfaces for HF@C60: Prediction of Spectral and Electric Response Properties. *J. Chem. Phys.*, 147(24):244303, 2017.

[132] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. Chow, and J. Gambetta. Hardware-Efficient Variational Quantum Eigensolver for Small Molecules and Quantum Magnets. *Nature*, 549(7671):242, 2017.

[133] H. J. Kappen. Learning Quantum Models from Quantum or Classical Data. 2018. arXiv:1803.11278.

[134] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*, 60(6):84, 2017.

[135] Y. LeCun, Y. Bengio, and G. E. Hinton. Deep Learning. *Nature*, 521(7553):436, 2015.

[136] C. H. Lee, K. T. Kang, K. S. Park, M. S. Kim, H. S. Kim, H. G. Kim, J. E. Fischer, and A. T. Johnson. The Nano-Memory Devices of a Single Wall and Peapod Structural Carbon Nanotube Field Effect Transistor. *Jpn. J. Appl. Phys.*, 42(8R):5392, 2003.

[137] M. Kieferová and N. Wiebe. Tomography and Generative Training with Quantum Boltzmann Machines. *Phys. Rev. A*, 96(6):062327, 2017.

[138] K. Mills, M. Spanner, and I. Tamblyn. Deep Learning and the Schrodinger Equation. *Phys. Rev. A*, 96(4):042113, 2017.

[139] N. Moll, P. Barkoutsos, L. Bishop, J. Chow, A. Cross, D. J. Egger, S. Filipp, A. Fuhrer, J. M. Gambetta, M. Ganzhorn, A. Kandala, A. Mezzacapo, P. Müller, W. Riess, G. Salis, J. Smolin, I. Tavernelli, and K. Temme. Quantum Optimization Using Variational Algorithms on Near-Term Quantum Devices. *Quantum Sci. Technol.*, 3(3):030503, 2018.

[140] K. P. Murphy. Machine Learning: A Probabilistic Perspective. In *Machine Learning: A Probabilistic Perspective*. The MIT Press, Cambridge, Massachusetts, 2012.

[141] A. Melkani, C. Gneiting, and F. Nori. 2019. arXiv: 1911.07506.

[142] W. Rothschild. Pure Rotational Absorption Spectrum of Hydrogen Fluoride Vapor between 22 and 250 Microns. *J. Opt. Soc. Am., JOSA*, 54(1):20, 1964.

[143] K. Ryczko, D. Strubbe, and I. Tamblyn. Deep Learning and Density-Functional Theory. *Phys. Rev. A*, 100(2):022512, 2019.

[144] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[145] P. Tarazaga, M. Raydan, and A. Hurman. Perron–Frobenius Theorem for Matrices with Some Negative Entries. *Linear Algebra and its Applications*, 328(1-3):57, 2001.

[146] E. van Nieuwenburg, Y. Liu, and S. Huber. Learning Phase Transitions by Confusion. *Nat. Phys.*, 13(5):435, 2017.

[147] L. Wang. Discovering Phase Transitions with Unsupervised Learning. *Phys. Rev. B*, 94(19):195105, 2016.

[148] B. Xu and X. Chen. Electrical-Driven Transport of Endohedral Fullerene Encapsulating a Single Water Molecule. *Phys. Rev. Lett.*, 110(15):252001, 2013.

[149] J. Zhang, G. Pagano, P. W. Hess, A. Kyprianidis, P. Becker, H. Kaplan, A. V. Gorshkov, Z.-X. Gong, and C. Monroe. Observation of a Many-Body Dynamical Phase Transition with a 53-Qubit Quantum Simulator. *Nature*, 551(7682):601, 2017.

[150] A. W. Sandvik and K. S. D. Beach. Monte Carlo Simulations of Quantum Spin Systems in the Valence Bond Basis. *arXiv:0704.1469 [cond-mat]*, 2007.

[151] A. W. Sandvik. Ground State Projection of Quantum Spin Systems in the Valence-Bond Basis. *Phys. Rev. Lett.*, 95(20):207203, 2005.

[152] A. W. Sandvik and H. G. Evertz. Loop updates for variational and projector quantum Monte Carlo simulations in the valence-bond basis. *Phys. Rev. B*, 82(2):024407, 2010.

[153] A. W. Sandvik. Computational Studies of Quantum Spin Systems. pages 135–338, 2010. arXiv:1101.3281.

[154] A. W. Sandvik. Stochastic Series Expansion Methods. 2019. arXiv:1909.10591.

[155] R. G. Melko. Stochastic Series Expansion Quantum Monte Carlo. In Adolfo Avella and Ferdinando Mancini, editors, *Strongly Correlated Systems*, volume 176, pages 185–206. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[156] D. C. Handscomb. A Monte Carlo method applied to the Heisenberg ferromagnet. *Mathematical Proceedings of the Cambridge Philosophical Society*, 60(1):115–122, 1964.

[157] A. J. Walker. New fast method for generating discrete random numbers with arbitrary frequency distributions. *Electronics Letters*, 10(8):127–128, 1974.

[158] A. J. Walker. An Efficient Method for Generating Discrete Random Variables with General Distributions. *ACM Trans. Math. Softw.*, 3(3):253–256, 1977.

[159] K. Fukui and S. Todo. Order-N cluster Monte Carlo method for spin systems with long-range interactions. *J. Comput. Phys.*, 228(7):2629–2642, 2009.

[160] D. Jaksch, J. I. Cirac, P. Zoller, S. L. Rolston, R. Côté, and M. D. Lukin. Fast Quantum Gates for Neutral Atoms. *Phys. Rev. Lett.*, 85(10):2208–2211, 2000.

[161] M. D. Lukin, M. Fleischhauer, R. Cote, L. M. Duan, D. Jaksch, J. I. Cirac, and P. Zoller. Dipole Blockade and Quantum Information Processing in Mesoscopic Atomic Ensembles. *Phys. Rev. Lett.*, 87(3):037901, 2001.

[162] P. Fendley, K. Sengupta, and S. Sachdev. Competing density-wave orders in a one-dimensional hard-boson model. *Phys. Rev. B*, 69(7):075106, 2004.

[163] D. Barredo, V. Lienhard, S. de Léséleuc, T. Lahaye, and A. Browaeys. Synthetic three-dimensional atomic structures assembled atom by atom. *Nature*, 561(7721):79–82, 2018.

[164] 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. Vuletic, and M. D. Lukin. Probing Topological Spin Liquids on a Programmable Quantum Simulator. 2021.

[165] N. Kawashima and K. Harada. Recent Developments of World-Line Monte Carlo Methods. *Journal of the Physical Society of Japan*, 73(6):1379–1414, 2004.

[166] A. W. Sandvik. Finite-size scaling of the ground-state parameters of the two-dimensional Heisenberg model. *Phys. Rev. B*, 56(18):11678–11690, 1997.

[167] D. Barredo, V. Lienhard, S. de Léséleuc, T. Lahaye, and A. Browaeys. Synthetic three-dimensional atomic structures assembled atom by atom. *Nature*, 561(7721):79–82, 2018.

[168] M. Endres, H. Bernien, A. Keesling, H. Levine, E. R. Anschuetz, A. Krajenbrink, C. Senko, V. Vuletic, M. Greiner, and M. D. Lukin. Atom-by-atom assembly of defect-free one-dimensional cold atom arrays. *Science*, 354(6315):1024–1027, 2016.

[169] Q. Wang, X. Gao, K. Wan, F. Li, and Z. Hu. A Novel Restricted Boltzmann Machine Training Algorithm with Fast Gibbs Sampling Policy. *Mathematical Problems in Engineering*, 2020:4206457, 2020.

APPENDICES

Appendix A

NMAs of Endofullerenes: Ground state wavefunction sign structure

A.1 Position operator matrix elements

The matrix elements of the dipole-dipole operator in the free-rotor basis must be calculated by evaluating the matrix elements of the position operators,

$$\left\langle l'm'\middle|\hat{x}\middle|\ell m\right\rangle = \int_{-1}^{1} d(\cos\theta) \int_{0}^{2\pi} d\phi Y_{l'}^{m'}(\theta,\phi) \sin\theta\cos\theta Y_{l}^{m}(\theta,\phi), \tag{A.1a}$$

$$\left\langle l'm'\middle|\hat{y}\middle|\ell m\right\rangle = \int_{-1}^{1} d(\cos\theta) \int_{0}^{2\pi} d\phi Y_{l'}^{m'}(\theta,\phi) \sin\theta\sin\theta Y_{l}^{m}(\theta,\phi), \tag{A.1b}$$

and

$$\left\langle l'm'\middle|\hat{z}\middle|\ell m\right\rangle = \int_{-1}^{1} d(\cos\theta) \int_{0}^{2\pi} d\phi Y_{l'}^{m'}(\theta,\phi) \cos\theta Y_{l}^{m}(\theta,\phi), \tag{A.1c}$$

where the functions $Y_l^m(\theta,\phi)$ are the conventional spherical harmonic functions that define an orthonormal basis via

$$\int_{-1}^{1} d(\cos\theta) \int_{0}^{2\pi} d\theta d\phi \sin\theta Y_{l}^{m}(\theta,\phi) Y_{l'}^{m'}(\theta,\phi) = \delta_{m,m'}\delta_{\ell,\ell'}. \tag{A.2}$$

Evaluating these integrals, one finds that the matrix elements of the position operators in the free-rotor basis are,

$$\begin{aligned}
\langle l'm'|\hat{x}|\ell m\rangle =& -\frac{1}{2}\delta_{m',m+1}\delta_{\ell',\ell+1}\sqrt{\frac{(\ell+m+1)(\ell+m+2)}{(2\ell+1)(2\ell+3)}} \\
&+\frac{1}{2}\delta_{m',m+1}\delta_{\ell',\ell-1}\sqrt{\frac{(\ell-m-1)(\ell-m)}{(2\ell-1)(2\ell+1)}} \\
&+\frac{1}{2}\delta_{m',m-1}\delta_{\ell',\ell+1}\sqrt{\frac{(\ell-m+1)(\ell-m+2)}{(2\ell+1)(2\ell+3)}} \\
&-\frac{1}{2}\delta_{m',m-1}\delta_{\ell',\ell-1}\sqrt{\frac{(\ell+m-1)(\ell+m)}{(2\ell-1)(2\ell+1)}},
\end{aligned} \tag{A.3a}$$

$$\begin{aligned}
\langle l'm'|\hat{y}|\ell m\rangle =& -\frac{i}{2}\delta_{m',m+1}\delta_{\ell',\ell+1}\sqrt{\frac{(\ell+m+1)(\ell+m+2)}{(2\ell+1)(2\ell+3)}} \\
&-\frac{i}{2}\delta_{m',m+1}\delta_{\ell',\ell-1}\sqrt{\frac{(\ell-m-1)(\ell-m)}{(2\ell-1)(2\ell+1)}} \\
&+\frac{i}{2}\delta_{m',m-1}\delta_{\ell',\ell+1}\sqrt{\frac{(\ell-m+1)(\ell-m+2)}{(2\ell+1)(2\ell+3)}} \\
&-\frac{i}{2}\delta_{m',m-1}\delta_{\ell',\ell-1}\sqrt{\frac{(\ell+m-1)(\ell+m)}{(2\ell-1)(2\ell+1)}},
\end{aligned} \tag{A.3b}$$

and

$$\begin{aligned}
\langle l'm'|\hat{z}|\ell m\rangle =& \delta_{m',m}\delta_{\ell',\ell+1}\sqrt{\frac{(\ell-m+1)(\ell+m+1)}{(2\ell+1)(2\ell+3)}} \\
&+\delta_{m',m}\delta_{\ell',\ell-1}\sqrt{\frac{(\ell-m)(\ell+m)}{(2\ell-1)(2\ell+1)}}.
\end{aligned} \tag{A.3c}$$

A.2 Wavefunction sign structure

Consider a normalized state $|\psi\rangle$ whose coefficients in an orthonormal basis $\mathcal{B} = \{|\boldsymbol{\sigma}\rangle\}$ are real: $\langle\boldsymbol{\sigma}|\psi\rangle \in \mathbb{R}$. In choosing a sign convention as "positive" when coefficients $\langle\boldsymbol{\sigma}|\psi\rangle$ have

the same sign as a "fixed" coefficient $\langle \boldsymbol{\sigma}^* | \psi \rangle$, one can divide the basis $\mathcal{B}$ into two classes,

$$\mathcal{B}^+ = \{ |\boldsymbol{\sigma}\rangle \in \mathcal{B} | \, \langle \boldsymbol{\sigma} | \psi \rangle \, \langle \boldsymbol{\sigma}^* | \psi \rangle \geq 0 \} \tag{A.4a}$$

and

$$\mathcal{B}^- = \{ |\boldsymbol{\sigma}\rangle \in \mathcal{B} | \, \langle \boldsymbol{\sigma} | \psi \rangle \, \langle \boldsymbol{\sigma}^* | \psi \rangle < 0 \} \tag{A.4b}$$

such that

$$\mathcal{B} = \{ |\boldsymbol{\sigma}\rangle \} = \mathcal{B}^+ \cup \mathcal{B}^-, \tag{A.4c}$$

where the latter class of coefficients who are oppositely-signed as $\langle \boldsymbol{\sigma}^* | \psi \rangle$ are aptly called "negative." We may also define projection operators

$$\hat{\mathcal{P}}^{\pm} = \sum_{|\boldsymbol{\sigma}\rangle \in \mathcal{B}^{\pm}} |\boldsymbol{\sigma}\rangle\langle\boldsymbol{\sigma}| \, , \tag{A.5}$$

which allows us to quantify the amount of "positivity" and "negativity" in our wavefunction coefficients as

$$\tau^{\pm} = \langle \psi | \hat{\mathcal{P}}^{\pm} | \psi \rangle = \sum_{|\boldsymbol{\sigma}\rangle \in \mathcal{B}^{\pm}} | \langle \boldsymbol{\sigma} | \psi \rangle |^2. \tag{A.6}$$

We may define a *sign-corrected* or *rectified* version of our state $|\psi\rangle$ as

$$\big| \psi_{||} \big\rangle = \hat{\mathcal{P}}^+ |\psi\rangle - \hat{\mathcal{P}}^- |\psi\rangle \, , \tag{A.7a}$$

whose coefficients, by design, are "positive,"

$$\langle \mathbf{n} | \psi_{||} \rangle = \mathrm{sgn}\big(\langle \boldsymbol{\sigma}^* | \psi \rangle \big) \big| \langle \boldsymbol{\sigma} | \psi \rangle \big|. \tag{A.7b}$$

Clearly, approximating $|\psi\rangle$ as $\big|\psi_{||}\big\rangle$ will become better and better as $\tau^- \to 0$. The overlap of the rectified state $\big|\psi_{||}\big\rangle$ with our original state $|\psi\rangle$ conveniently quantifies the error of such an approximation,

$$\big\langle \psi_{||} \big| \psi \big\rangle = \langle \psi | \left(\hat{\mathcal{P}}^+ |\psi\rangle - \hat{\mathcal{P}}^- |\psi\rangle \right) = 1 - 2\tau^-. \tag{A.8}$$

What can be said about calculating observables? For instance, given an observable $\hat{\mathcal{O}}$, we wish to determine the error induced by making a rectified approximation

$$\varepsilon = \big\langle \psi_{||} \big| \hat{\mathcal{O}} \big| \psi_{||} \big\rangle - \langle \psi | \hat{\mathcal{O}} | \psi \rangle \, . \tag{A.9}$$

Using Eq. (A.7a) and noting that $\hat{\mathcal{P}}^+ + \hat{\mathcal{P}}^- = \mathbb{I}$ by construction,

$$\begin{aligned}
\varepsilon &= \left(\langle\psi|\,\hat{\mathcal{P}}^+ - \langle\psi|\,\hat{\mathcal{P}}^-\right)\hat{\mathcal{O}}\left(\hat{\mathcal{P}}^+\,|\psi\rangle - \hat{\mathcal{P}}^-\,|\psi\rangle\right) - \langle\psi|\hat{\mathcal{O}}|\psi\rangle \\
&= \langle\psi|\hat{\mathcal{P}}^+\hat{\mathcal{O}}\hat{\mathcal{P}}^+|\psi\rangle - \langle\psi|\hat{\mathcal{P}}^-\hat{\mathcal{O}}\hat{\mathcal{P}}^+|\psi\rangle - \langle\psi|\hat{\mathcal{P}}^+\hat{\mathcal{O}}\hat{\mathcal{P}}^-|\psi\rangle + \langle\psi|\hat{\mathcal{P}}^-\hat{\mathcal{O}}\hat{\mathcal{P}}^-|\psi\rangle - \langle\psi|\hat{\mathcal{O}}|\psi\rangle \\
&= \langle\psi|\Big[\hat{\mathcal{P}}^+\hat{\mathcal{O}}\left(\mathbb{I}-\hat{\mathcal{P}}^-\right) - \hat{\mathcal{P}}^-\hat{\mathcal{O}}\hat{\mathcal{P}}^+ - \hat{\mathcal{P}}^+\hat{\mathcal{O}}\hat{\mathcal{P}}^- + \hat{\mathcal{P}}^-\hat{\mathcal{O}}\left(\mathbb{I}-\hat{\mathcal{P}}^+\right) - \hat{\mathcal{O}}\Big]|\psi\rangle \\
&= \langle\psi|\Big[\hat{\mathcal{P}}^+\hat{\mathcal{O}} - \hat{\mathcal{P}}^+\hat{\mathcal{O}}\hat{\mathcal{P}}^- - \hat{\mathcal{P}}^-\hat{\mathcal{O}}\hat{\mathcal{P}}^+ - \hat{\mathcal{P}}^+\hat{\mathcal{O}}\hat{\mathcal{P}}^- + \hat{\mathcal{P}}^-\hat{\mathcal{O}} - \hat{\mathcal{P}}^-\hat{\mathcal{O}}\hat{\mathcal{P}}^+ - \hat{\mathcal{O}}\Big]|\psi\rangle \\
&= \langle\psi|\Big[\left(\hat{\mathcal{P}}^+ + \hat{\mathcal{P}}^-\right)\hat{\mathcal{O}} - \hat{\mathcal{O}} + \hat{\mathcal{P}}^+\hat{\mathcal{O}}\hat{\mathcal{P}}^- - \hat{\mathcal{P}}^-\hat{\mathcal{O}}\hat{\mathcal{P}}^+ - \hat{\mathcal{P}}^+\hat{\mathcal{O}}\hat{\mathcal{P}}^- - \hat{\mathcal{P}}^-\hat{\mathcal{O}}\hat{\mathcal{P}}^+\Big]|\psi\rangle \\
&= -\,2\,\langle\psi|\Big(\hat{\mathcal{P}}^+\hat{\mathcal{O}}\hat{\mathcal{P}}^- + \hat{\mathcal{P}}^-\hat{\mathcal{O}}\hat{\mathcal{P}}^+\Big)|\psi\rangle\,,
\end{aligned}$$

and using the property of hermitianity,

$$\begin{aligned}
&= -\,2\left[\,\langle\psi|\hat{\mathcal{P}}^+\hat{\mathcal{O}}\hat{\mathcal{P}}^-|\psi\rangle + \langle\psi|\hat{\mathcal{P}}^+\hat{\mathcal{O}}\hat{\mathcal{P}}^-|\psi\rangle^*\right] \\
&= -\,4\mathrm{Re}\,\langle\psi|\hat{\mathcal{P}}^+\hat{\mathcal{O}}\hat{\mathcal{P}}^-|\psi\rangle\,. \qquad \text{(A.10)}
\end{aligned}$$

Now, consider when $\hat{\mathcal{O}} = \hat{H}$ and $|\psi\rangle = |\lambda_0\rangle$, the ground state of the Hamiltonian $\hat{H}$. Eq. (A.10) becomes

$$\begin{aligned}
\varepsilon &= -\,4\mathrm{Re}\,\langle\lambda_0|\hat{\mathcal{P}}^+\hat{H}\hat{\mathcal{P}}^-|\lambda_0\rangle \\
&= -\,4\mathrm{Re}\,\langle\lambda_0|\hat{\mathcal{P}}^-\hat{H}\left(\mathbb{I}-\hat{\mathcal{P}}^-\right)|\lambda_0\rangle \\
&= -\,4\mathrm{Re}\left[\,\langle\lambda_0|\hat{\mathcal{P}}^-\hat{H}|\lambda_0\rangle - \langle\lambda_0|\hat{\mathcal{P}}^-\hat{H}\hat{\mathcal{P}}^-|\lambda_0\rangle\right] \\
&= -\,4\mathrm{Re}\left[E_0\,\langle\lambda_0|\hat{\mathcal{P}}^-|\lambda_0\rangle - \langle\lambda_0|\hat{\mathcal{P}}^-\hat{H}\hat{\mathcal{P}}^-|\lambda_0\rangle\right] \\
&= -\,4E_0\tau^- + 4\,\langle\lambda_0|\hat{\mathcal{P}}^-\hat{H}\hat{\mathcal{P}}^-|\lambda_0\rangle\,. \qquad \text{(A.11)}
\end{aligned}$$

For the model Hamiltonian describing NMAs of endofullerenes in Eq. (2.34), we can analyze how ε and τ^- depend on the Hamiltonian parameter R and the truncation $\ell_{\max}$ for exactly-diagonalizable system sizes. Fig. A.1 and Fig. A.2 show how $\varepsilon/\Delta E_1$ and τ^- depend on the truncation $\ell_{\max}$ and R, respectively. Here, ΔE_1 is the energy difference between the ground state and the first excited state. Fig. A.1 demonstrates that the values of $\varepsilon/\Delta E_1$ and τ^- are sufficiently small and converged at $\ell_{\max} = 3$. Moreover, the peaks in Fig. A.2 clearly drop in height as N is increased, and the values of $\varepsilon/\Delta E_1$ and τ^- diminish quickly when $R \geq 1.0$. Thus, for the chosen values of R and N in Sec. 2.5, the rectified state approximation is robust.

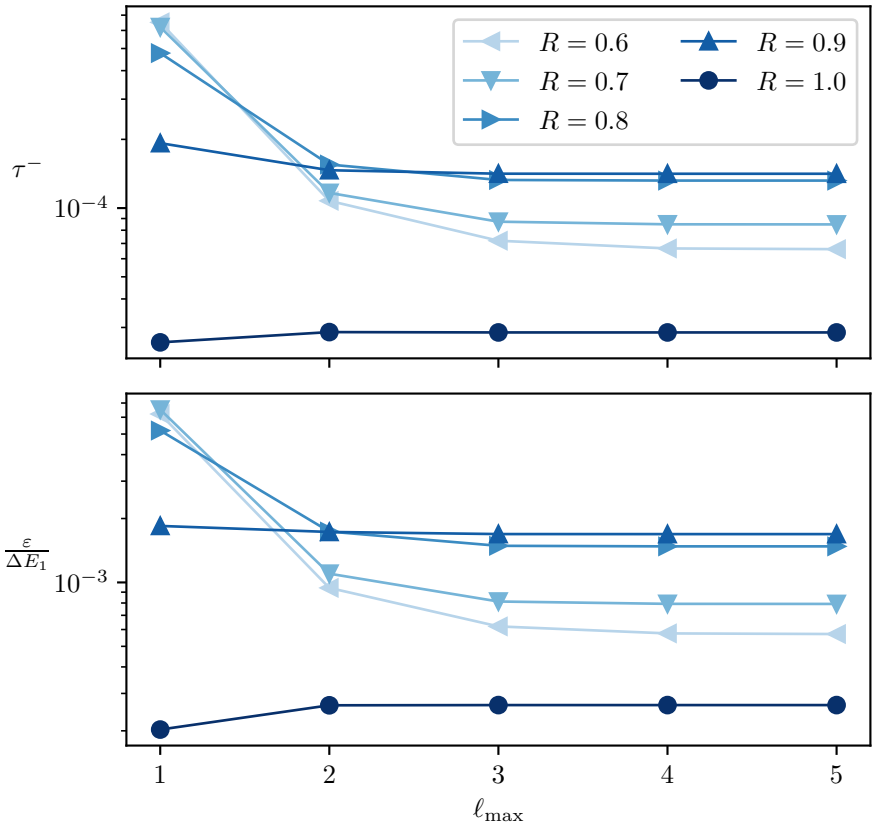

Figure A.1: The convergence of the rectified state error measures τ^- (top pane) and $\varepsilon/\Delta E_1$ (bottom pane) with the truncation $\ell_{\max}$ for a linear system of $N = 6$ endofullerenes for several different R values. All calculations here were performed with exact diagonalization.

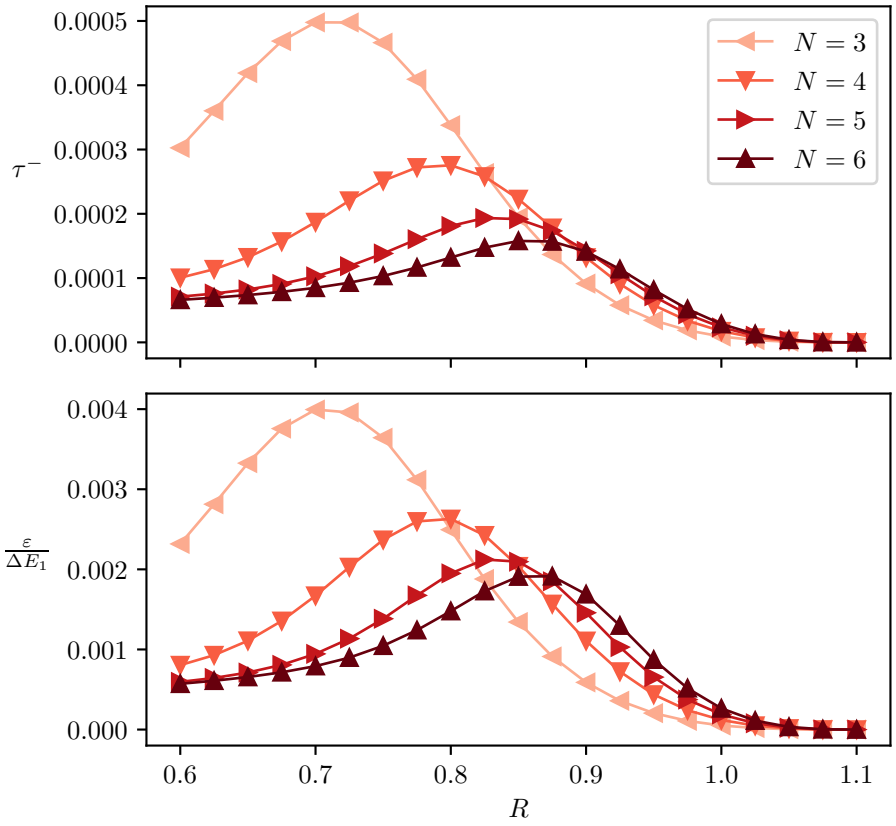

Figure A.2: The convergence of the rectified state error measures τ^- (top pane) and $\varepsilon/\Delta E_1$ (bottom pane) with R for linear systems of $N = 3$ to 6 endofullerenes using $\ell_{\text{max}} = 5$. All calculations here were performed with exact diagonalization.

Appendix B

Extra plots from Sec. 3.4

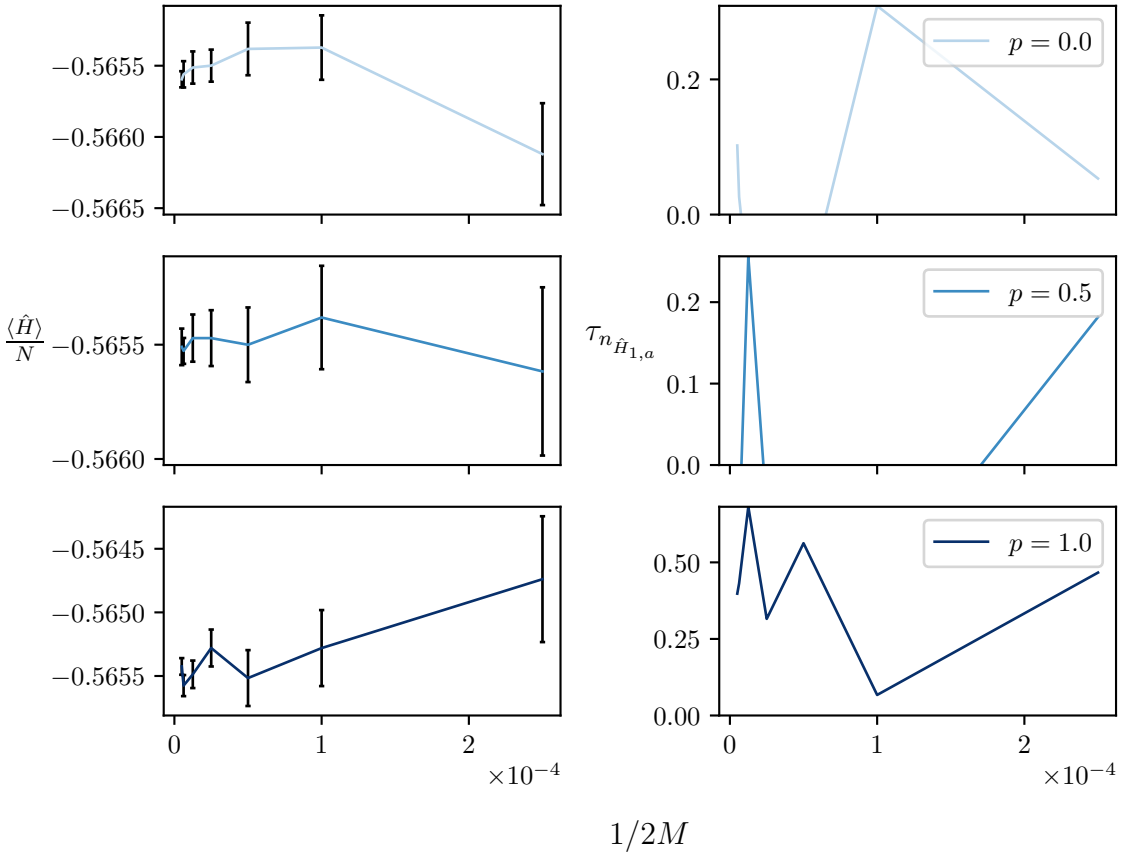

Figure B.1: The QMC ground state energy density $\langle \hat{H} \rangle / N$ and the correlation time for the number of $\hat{H}_{1,a}$ operators in the operator sequence S_M versus the inverse projector length $1/2M$ as measured on a $N = 4 \times 4$ lattice with $\delta/\Omega = 1.12$ and $R_b = 1.2$ for three different cluster updates: multibranch ($p = 0$), line ($p = 1.0$), and a random choice between multibranch and line updates ($p = 0.5$). Each point represents a culmination of three independent – different random seeds – SSE QMC simulations. Per random seed, 2×10^5 consecutive measurements were taken after a 10^5 step equilibration phase, which was proceeded by a jackknife binning analysis to obtain the plotted mean values and error bars of $\langle \hat{H} \rangle / N$.

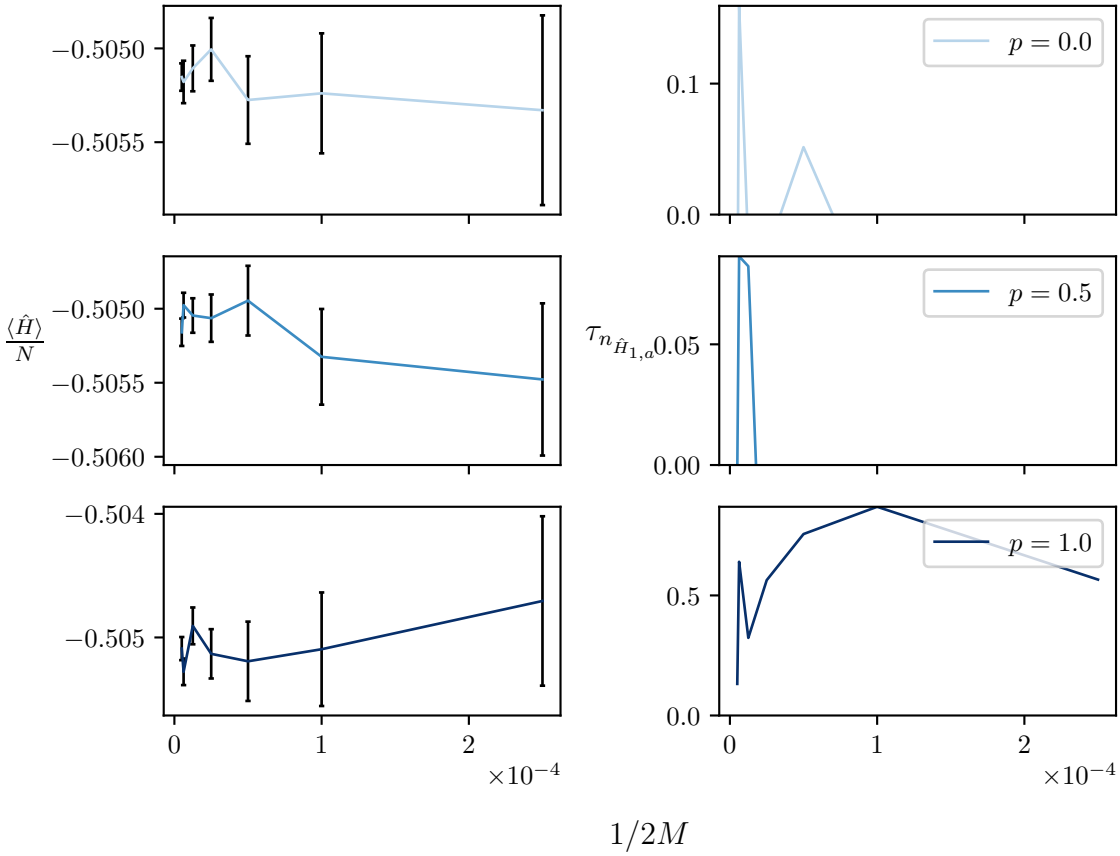

Figure B.2: The QMC ground state energy density $\langle \hat{H} \rangle / N$ and the correlation time for the number of $\hat{H}_{1,a}$ operators in the operator sequence S_M versus the inverse projector length $1/2M$ as measured on a $N = 10 \times 10$ lattice with $\delta/\Omega = 1.12$ and $R_b = 1.2$ for three different cluster updates: multibranch ($p = 0$), line ($p = 1.0$), and a random choice between multibranch and line updates ($p = 0.5$). Each point represents a culmination of three independent – different random seeds – SSE QMC simulations. Per random seed, 2×10^5 consecutive measurements were taken after a 10^5 step equilibration phase, which was proceeded by a jackknife binning analysis to obtain the plotted mean values and error bars of $\langle \hat{H} \rangle / N$.

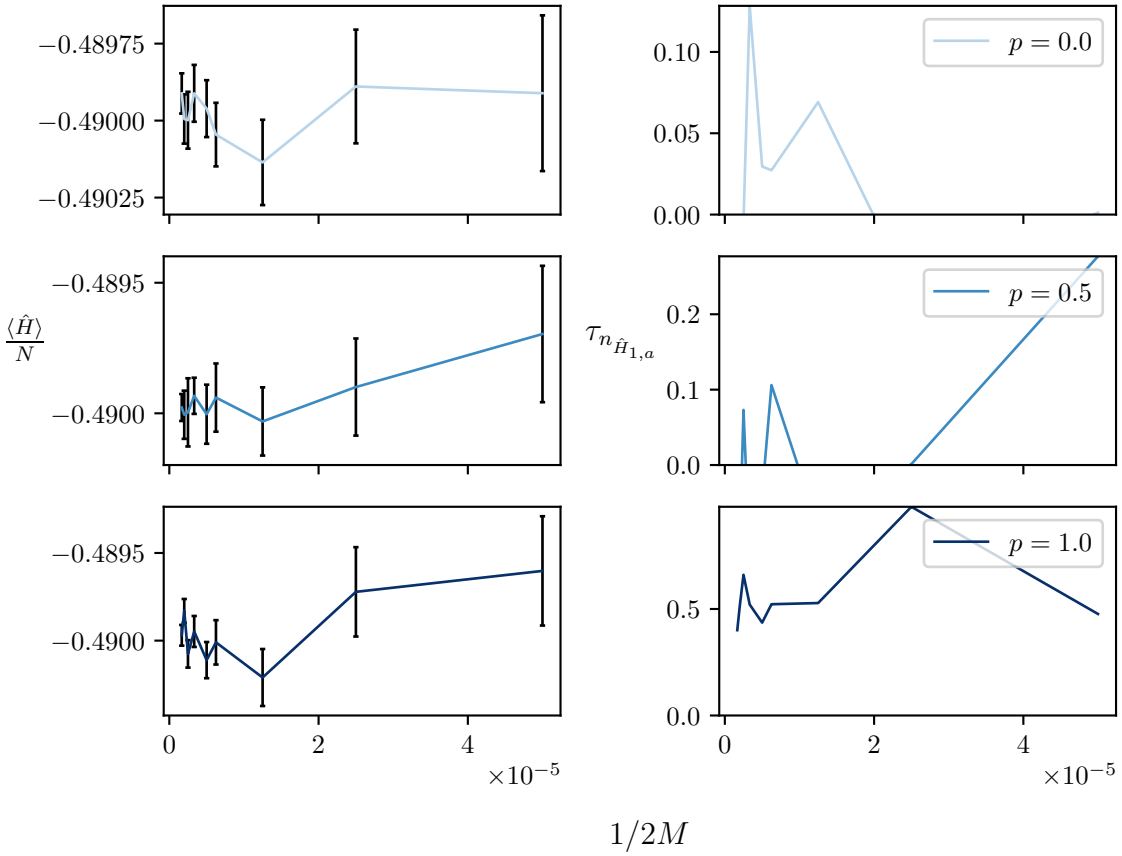

Figure B.3: The QMC ground state energy density $\langle \hat{H} \rangle / N$ and the correlation time for the number of $\hat{H}_{1,a}$ operators in the operator sequence S_M versus the inverse projector length $1/2M$ as measured on a $N = 16 \times 16$ lattice with $\delta/\Omega = 1.12$ and $R_b = 1.2$ for three different cluster updates: multibranch ($p = 0$), line ($p = 1.0$), and a random choice between multibranch and line updates ($p = 0.5$). Each point represents a culmination of three independent – different random seeds – SSE QMC simulations. Per random seed, 2×10^5 consecutive measurements were taken after a 10^5 step equilibration phase, which was proceeded by a jackknife binning analysis to obtain the plotted mean values and error bars of $\langle \hat{H} \rangle / N$.