Mapping the Ising model onto two-dimensional Rubidium atom arrays

Mapping the Ising model onto two-dimensional Rubidium atom arrays

Abhishek Anand

Advisor: Prof. Mikhail Lukin

An undergraduate thesis submitted in partial fulfillment of the requirements for the joint degree of Bachelor of Arts in Computer Science and Physics with Honors

Harvard University
Cambridge, Massachusetts
April 17, 2020

Abstract

We describe an architecture to map the Ising model problem onto the dynamics of two-dimensional neutral trapped atom arrays. Specifically, we show that the NP-complete Ising model decision problem on n variables can be reduced to deciding whether, for $O(n^2)$ Rubidium atoms in two dimensions, interacting via Rydberg excitations, there exists a state with energy below a certain threshold. This provides an alternative proof of the NP-hardness of two dimensional Rydberg dynamics with a single species in the regime where long-range interactions are negligible. Therefore, we can encode NP-complete problems onto the ground state of Rydberg atom arrays and use quantum optimization techniques such as quantum adiabatic algorithms and variational algorithms to estimate the ground state. Furthermore, as the Ising model describes many-body phenomena in statistical physics, we can probe a plethora of emergent phenomena via quantum simulators.

Our result is based on the Lechner-Hauke-Zoller (LHZ) scheme that enables the encoding of an all-to-all connected Ising model onto systems with local-only interactions. We improve the scheme in the following ways to make the it feasible for near-term Rydberg simulators: (1) Implementations of the scheme in literature that use qubits suffer from a double-counting error causing inconsistencies in the strength of the interactions. Identical interactions must have different strengths based on their spatial location. We describe and resolve this error without impacting the $O(n^2)$ scaling of the atom array by introducing additional ancillary qubits. (2) We introduce new formalism and simpler proofs via novel approaches regarding the validity of the LHZ scheme and its variants. These proofs lend more easily to complexity-theoretic arguments regarding the hardness of Rydberg dynamics. (3) Finally, we circumvent the LHZ scheme constraint of requiring four-body interactions, qutrits, or multiple species of atoms by providing an implementation that uses only Rubidium atoms to encode both physical and ancillary qubits. To engineer the required interactions, the atoms are coupled to different Rydberg energy levels.

We also provide a brief review of classical and quantum computation, quantum computing platforms, near-term quantum experiments, and Rydberg physics.

Acknowledgements

I am grateful to my thesis adviser, Prof. Mikhail Lukin, for introducing me to this field, enabling me to work on an exciting research topic and helping me learn the foundations of quantum computation and Rydberg physics.

I am indebted to Sheng-Tao Wang and Leo Zhou, who were key contributors to the project. They proposed the problem, assisted with brainstorming and discussing various approaches, provided numerous insights and background information, and patiently helped me whenever I got stuck.

I am thankful to Prof. Boaz Barak for useful conversations, teaching me complexity theory, and for introducing me to the wonderful world of theoretical computer science. Furthermore, I thank Prof. Madhu Sudan, Stefan Krastanov, Alexander Keesling, Jambay Kinley, and Shraddha Anand for useful discussions. Stefan's, Jambay's, and Shraddha's feedback on drafts of the thesis have been extremely useful in enhancing its clarity. I am also grateful to Prof. Boaz Barak, Prof. Prineha Narang, and Prof. Mikhail Lukin for agreeing to the readers for my thesis.

Lastly, I would like to thank my parents, my sister, and friends, for their encouragement and support throughout my academic career.

Contents

1 | Introduction

In the last four decades, quantum mechanics has emerged as the latest challenger to the extended Church-Turing thesis. There are computational problems for which we have known quantum algorithms that provide an exponential speed-up compared to the best known classical algorithms: for example, Shor's factoring algorithm and quantum algorithms for Hamiltonian simulation. However, the experimental implementation of a large number of coherent qubits has been challenging. We are now approaching an exceptional period with the advent of the first noisy intermediate-scale quantum (NISQ) devices with > 50 qubits that may outperform their classical counterparts on certain problems.

One potentially scalable approach to building quantum simulators and computers is using neutral atoms. Recently, significant progress has been made in demonstrating high-fidelity entanglement between neutral atoms that interact via Rydberg excitations. It has been shown that these atoms can be arranged in programmable configurations of arrays using optical tweezers [End+16; Bar+16], and high two-qubit and three-qubit gate fidelities have been reported [Lev+19]. *To design an experiment that demonstrates quantum computational advantage and solve useful problems using the Rydberg platform, it is critical to understand and characterize the computational power of Rydberg atom arrays and design near-term algorithms taking into account current experimental capabilities.*

Our focus is on quantum optimization via neutral atoms, which is a paradigm of adiabatic quantum computation. The key idea is to steer the dynamics of the system in such a way that the final state encodes the solution to the computational problem we set out to solve. The kinds of problems we can solve via this paradigm depend on the types of dynamics our quantum simulator can implement. In essence, we can only solve the computational problems that can be mapped onto the device. Recently, it has been shown that there is a natural mapping between the NP-complete maximum independent set problem on unit disk graphs [CCJ90] and the quantum dynamics governing Rubidium atom in two dimensions [Pic+18a]. It has also been shown that it is NP-complete to decide whether the ground state of Rydberg atoms in two dimensions is below a certain fixed constant [Pic+18b]. Therefore, we can map all NP problems onto the ground state of Rydberg atom arrays with polynomial overhead. However, given the limited number

of qubits in this NISQ era, it is critical to find efficient mappings from other *useful* and hard problems.

Ising model problem: We consider the Ising model problem [Len20] which is NP-complete [Bar82]. This problem was inspired by models in statistical mechanics, and there exists literature on the efficient mapping of other canonical NP problems onto the Ising model problem [Luc14]. The problem is defined as:

Definition 1.1 (**Ising model decision problem**)**.** We define a function $Ising : \{0,1\}^* \rightarrow \{0,1\}$ that takes in an Ising model optimization problem parameterized by $n \in \mathbb{N}$, $h_i \in \mathbb{R}\ \forall i \in [n], J_{i,j} \in \mathbb{R}\ \forall i,j \in [n]$ such that $0 \leq i < j < n$, and $k \in \mathbb{R}$

$$Ising(n,h,J,k) = \begin{cases} 1 \text{ if } \exists S \in \{-1,+1\}^n \text{ such that } \left(\sum_{i=0}^{n-1} h_i S_i + \sum_{i<j} J_{i,j} S_i S_j\right) \leq k \\ 0 \text{ otherwise} \end{cases} \tag{1.1}$$

The dynamics of Rubidium atom arrays: We consider the model of Rydberg atom arrays in two dimensions using a single species of atoms and certain long-range interactions to be zero. In this setup, we encode qubits by considering the internal ground state of the atom as $|g\rangle$ and exciting it coherently via lasers to the long-lived Rydberg state $|r\rangle$. The dynamics of Rydberg atom arrays are controlled by the following Hamiltonian:

$$H_{Ryd} = \sum_{\nu} (\Omega_\nu \sigma_\nu^x - \Delta_\nu n_\nu) + \sum_{\nu<w} V(|\vec{x}_\nu - \vec{x}_w|, \nu, w) n_\nu n_w \tag{1.2}$$

where ν, w are qubits realized by atoms and the n operator has eigenvalue 1 for state $|r\rangle$ and 0 for state $|g\rangle$. Tunable parameters Ω_ν and Δ_ν are the Rabi frequency and laser detuning for qubit ν, respectively, and $\vec{x}_\nu$ refers to the position of qubit ν. Furthermore, V is the Rydberg potential, and is a deterministic function of the distance between two qubits and the properties of the Ryberg state that the qubits are coupled to. Hence, V is not tunable. Given this, it is not clear what problems can be efficiently reduced to Rydberg dynamics a priori.

We show that we can map the problem with n variables onto the dynamics of two-dimensional Rubidium atom arrays using $O(n^2)$ atoms. Note that the challenges in reducing this problem lies in the fact that J_{ij} is an arbitrary real number (the graph is "all-to-all" connected) while Rydberg dynamics are local. Our work is based on the Lechner-Zoller-Hauke scheme [LHZ15] which offers a way to map the Ising model onto the dynamics of local systems. Our original work builds off the scheme and makes it implementable on 2D Rubidium atom arrays by improving it in the following three ways:

- Implementations of this scheme that use qubits [Gla+17] suffer from a double-counting error (described formally in Section 5.2): for the mapping to be successful, we want interactions between two atoms at a distance s to vary depending on where the atoms are in the configuration. We resolve this error by introducing additional (linear in n) ancillary qubits and hence, resolve the issue without impacting the $O(n^2)$ scaling.

- We introduce new formalism and simpler proofs via novel approaches regarding the validity of the LHZ scheme and its variants. These proofs lend more easily to complexity-theoretic arguments regarding the hardness of Rydberg dynamics. Specifically, the proofs of Theorem refproof1, refproof2, refproof3, and refproof4 are via new approaches.

- We circumvent the LHZ scheme requirement of four-body interactions, qutrits, or multiple species of atoms by using ^{87}Rb to encode both physical and ancillary qubits but coupling them to different Rydberg energy levels.

The existence of this mapping has two-fold consequences:

- From a complexity standpoint, we provide a proof for the following:

  Theorem 1.1. (Informal) We can map the Ising model problem of n variables onto the dynamics of 2D Rydberg atom arrays of a single species with $O(n^2)$ atoms. For a given set of positions of atoms, parameters $\{\Omega, \Delta\}$ and the Rydberg state each atom is coupled to, for some number k if there is a state of the form $\{|r\rangle, |g\rangle\}^n$ on n atoms with ψ such that $\langle\psi|H_{Ryd}|\psi\rangle \leq k$, then function $Rydberg(\{\Omega, \Delta, x, n, l\}, k) : \{0,1\}^* \to \{0,1\}$ outputs 1 and 0 otherwise. This implies that finding the ground state of this Hamiltonian is NP-hard.

  This means that we can encode NP-complete problems onto 2D Rydberg atom arrays; however, the heuristic nature of the adiabatic algorithm (described in Section 2.2.2) makes it hard to predict speed-up for problems theoretically. In general, we do not expect to solve an NP-complete problem using quantum computation efficiently. However, there may exist non-exponential speed-ups that we may be able to probe experimentally. Furthermore, we could explore sampling versions of the problem and different regimes of the J matrix (which may be considered classically hard but not NP-hard) for exponential speed-ups. Therefore, this formulation may provide a guide towards experiments that could be conducted to explore and compare quantum advantage across different structures of the J matrix.

- Secondly, from a physics perspective, many systems in statistical physics and many-body physics are modeled as Ising spin models, and therefore, using this reduction,

our simulator can be used to probe exotic many-body phenomena with $O(n^2)$ overhead.

The thesis is structured as follows. In Chapter 2, we discuss the preliminaries: we introduce the classical and quantum computation paradigms as well as Rydberg dynamics. In Chapter 3, we formally introduce the Ising model problem and discuss challenges to its mapping. In Chapter 4, we discuss the original LHZ scheme and provide new proofs regarding its validity. In Chapter 5, we discuss a feasible implementation of the LHZ scheme using only qubits, resolving the double-counting error. In Chapter 6, we discuss how we may implement our scheme using only Rubidium atoms, present the proof of our main result, and end by detailing future steps. Chapters 4, 5, and 6 encompass most of the original work carried out for this thesis.

2 | Preliminaries

In this chapter, we will introduce the foundations of classical and quantum computation, complexity theory, quantum computing platforms, and Rydberg physics. We will cover both the circuit and adiabatic model of quantum computation, discuss quantum simulator platforms, and recent results on the classical hardness of near-term experiments on quantum devices. Then, we introduce Rydberg physics and certain schemes for computation via Rydberg atom arrays.

2.1 Classical computation

We first overview classical computation and complexity theory. This section is primarily based on [AB09; Kit+02; Bar].

2.1.1 Models of computation and computability

We are concerned with functions $f : \{0,1\}^* \to \{0,1\}$ that we want to compute (note that we can extend all definitions and discussions below to functions with multi-bit outputs). Computational complexity theory studies the resources required to compute a function. These resources may include time/steps taken, space/memory used, or queries made to some black-box oracle. Naively, the amount of resources required should depend on our computing apparatus. As a simple example, a machine that can add only two bits together in 1 unit of time will take more time to add n bits compared to machine that can add n bits together in 1 unit of time. Therefore, a priori, it seems that the choice of our model of computation is crucial. Soon, we will see why this choice may not matter in the classical case. First, let us introduce the standard model of classical computation, a Turing machine, invented by Alan Turing in 1936 [Tur36]:

Definition 2.1 (**Turing machine** (adapted from [Bar]))**.** A Turing machine M consists of the following parts:

- k states
- alphabet $\Sigma \supseteq \{0, 1, \triangleright, \varnothing\}$

- a tape T where each cell $T[i]$ can house a member of Σ
- a transition function $\delta_M : [k] \times \Sigma \to [k] \times \Sigma \times \{\mathsf{L}, \mathsf{R}, \mathsf{S}, \mathsf{H}\}$

For every $x \in \{0,1\}^*$, the output of M is denoted by $M(x)$, and is the result of the following process:

- We initialize the tape T such that $T[0] = \triangleright$ (the starting symbol), $T[i+1] = x_i$ for $i \in [n]$, and $T[i] = \varnothing$ for $i > n$ and initialize variables $i = s = 0$ where i denotes the position of the head and s denotes the state of TM M.
- We then repeat the following process:
  - Let $(s', \sigma', D) = \delta_M(s, T[i])$.
  - We set $s \to s'$, $T[i] \to \sigma'$.
  - If $D = \mathsf{R}$ then we set $i \to i+1$, if $D = \mathsf{L}$ then we set $i \to \max(i-1, 0)$, if $D = \mathsf{S}$ then we keep i the same, if $D = \mathsf{H}$ then we halt the process.
- If $D = \mathsf{H}$ and the process above halts, then we set $M(x)$ to be the string $y \in \{0,1\}^*$ obtained by concatenating all the symbols in $\{0,1\}$ in positions $T[0], \ldots, T[i]$ where i is the final head position.
- If M does not halt then we denote $M(x) = \perp$.

Note that the above definition is for a one-tape Turing machine and can be generalized for multi-tape Turing machines. To summarize, the Turing machine consists of a tape which is initialized to be the input to the computation, x. The transition function δ_M tells us how the tape changes with each time step. Either the Turing machine never reaches state H, and we see that this machine does not halt set $M(x) = \perp$. However, if the machine does reach H state, the symbols on the tape encode the result of the computation by the Turing machine, and we call their concatenation $M(x)$.

Computability: If we claim that a Turing machine M_f computes a function $f : \{0,1\}^* \to \{0,1\}$, this means that for all $x \in \{0,1\}^*$, $M_f(x) = f(x)$. Using this, we can define our first complexity class R:

Definition 2.2 (class R). R is the set of functions such that for every function $f \in R$, there exists a Turing machine M_f that computes f.

We also call the set of functions in R, *computable* functions. That is a strong claim: we are saying that functions for which corresponding Turing machines do not exist are not *computable*. What makes the model of Turing machines so expressive that it can

encapsulate all computation? Shouldn't the model of computation change what is *computable*, and what is not? Many models of computation have been proposed since the invention of Turing machines since 1936. But for every model of computation, we have been able to show that Turing machines can simulate the model. Therefore, based on these experiences, we believe that what is in R and what is not, does not change. This belief can be expressed as the following conjecture by Church and Turing, which they came up with independently:

Law of computation 2.1 (**Church-Turing thesis**)**.** Every physically realizable model of computation can be simulated by a Turing machine.

The Church-Turing thesis is not a mathematical theorem and, perhaps, cannot be proved. Though, it can be shown to be false via a counterexample. However, computation done via all models of computation that we have thought of implementing: pen and paper, Post machine, Godel's theory of recursive functions, DNA computing, lambda calculus, cellular automata, and even quantum computation has been shown to be simulatable by a Turing machine given enough time. This idea can be formalized by the idea of Turing completeness and equivalence. A model of computation A is Turing-complete if for any Turing machine M_f that computes the function f there is an instance of A (let's call it A_f) such that A_f that also computes function f. A model of computation A is Turing-equivalent if it is Turing-complete, and Turing machines are A-complete, i.e., for every instance of A_f that computes the function f, there exists a Turing machine M_f that computes the function f. The models of computation considered above and popular programming languages C, Python, and JavaScript are Turing-complete. In fact, spreadsheet programs are Turing-complete as well!

Universal Turing machine: Note that given a Turing machine M and an input $x \in \{0,1\}^*$, we can go through the steps listed in Definition 2.1 and compute $M(x)$ using a pen and paper. Church-Turing thesis now implies that this procedure must also be emulatable by a Turing machine. More formally, there exists a universal Turing machine M such that it takes as input a representation of Turing machine M and x and:

$$U(\text{representation of } M, x) = M(x) \tag{2.1}$$

U is also called the meta-circular evaluator. In fact, we do not need to rely on the Church-Turing thesis. We can explicitly construct U and prove its existence by simply formalizing and converting the process in Definition 2.1 (link this) to a programming language and then into sub-routines that can be encoded into the transition function of U. For more details, refer to [AB09; Kit+02].

Uncomputability: Are there functions outside R? By a counting argument, it can be shown that the set of functions that cannot be computed by a Turing machine is non-empty. This is because the set of Turing machines is countable while the set of functions from $\{0,1\}^* \to \{0,1\}$ is uncountable. Hence, he same technique of diagonalization that is used to prove that size of the the set of real numbers is greater than the size of the set of the natural numbers can be used. A canonical example of an uncomputable function is the Halting problem:

Definition 2.3 (**Halting problem**)**.** We define $H : \{0,1\}^* \to \{0,1\}$ to be the function that takes in two arguments: representation of a Turing machine M and $x \in \{0,1\}^*$ and:

$$H(\text{representation of } M, x) = \begin{cases} 0 \text{ if } H(x) = \perp \\ 1 \text{ otherwise} \end{cases} \tag{2.2}$$

That is H returns 1 if M halts on x and 0 if M does not halt on x. One way to show that H is uncomputable by assuming that it is computable and constructing adversial arguments for the Turing machine that cause a contradiction. For more details refer to [Str65]. A more physically-motivated example of an uncomputable problem is the spectral gap problem: given a many-body Hamiltonian, deciding whether it is gapped (there is a finite gap between the ground state and the first excited state) or gapless (the gap between the ground state and the first excited state is not finite: the energy levels are continuous) [CPW15].

2.1.2 Efficient and randomized computation

P and EXP: We have talked about what can be and cannot be computed by Turing machines, but we have not yet talked about the resources required for the computation. We will be focusing on the time complexity. Let us begin by discussing measures of time for the Turing machine. Following the notation in [AB09], let us define the class DTIME standing for deterministic time:

Definition 2.4 (**class DTIME**)**.** Let $T : \mathbb{N} \to \mathbb{R}$ be some function. A function f belongs to the set $DTIME(T(n))$ if and only if there exists a $n_0 \in \mathbb{N}$ such that for all $n > n_0$, for all $x \in \{0,1\}^n$, the Turing machine computing f, M_f halts in at most $T(n)$ evaluations of the transition function on input x.

Here, we are using the number of times the transition function is called as a proxy for the amount of time it takes for the Turing machine to compute the function in question. We have the clause regarding large enough n as we care about asymptotic time complexity of functions. More simply, we can state that a function f belongs to DTIME($T(n)$) if the Turing machine that computes f runs in at most $T(n)$ time when n is the length of the input to Turing machine. We can use DTIME to define the class P (for polynomial):

Definition 2.5 (**class P**). A function f belongs to the class P if it belongs to DTIME($T(n)$) where $T(n) = n^a$ for any $a \in \mathbb{N}$. More succinctly, we have that:

$$P = \cup_{a \geq 1} DTIME(n^a) \tag{2.3}$$

These are functions for which it takes polynomial time in the input length for the Turing machines to compute them. We will consider a function f to have an *efficient* algorithm if and only if f is in P. Examples of problem within P are 2SAT, testing primality and shortest path in graphs. Similarly, we can define the class EXP (for exponential):

Definition 2.6 (**class EXP**). Similarly to P, we have that:

$$EXP = \cup_{a \geq 1} DTIME(\exp(n^a)) \tag{2.4}$$

These are functions for which it takes exponential time in the input length for the Turing machines to compute them. As we can always find an exponential that grows faster than any polynomial, P is contained in EXP. Examples of problems in EXP but not known to be in P are 3SAT, factoring, and longest path in graphs. Note that because of the Definition **??**, the time-bound should hold for all $x \in \{0,1\}^*$. Therefore, it is the worst-possible $x \in \{0,1\}^*$ that determines the time complexity class of a function f. That is why we sometimes call these classes, worst-case time complexity classes. Lastly, due to the time hierarchy theorem [HS65], it is known that there are functions in EXP that are not in P, i.e., if you give a Turing machine more time to compute, it can compute more functions.

Randomized computation: Till now, we only considered deterministic models of computation: given an input $x \in \{0,1\}^*$ and a Turing machine M, no matter how many times we try to determine $M(x)$, we follow exactly the same steps determined by the transition function δ_M and either the machine does not halt or if it does, we get the same value for $M(x)$. Let us consider a *probabilistic* Turing machine that has two transition functions δ_0 and δ_1 and for every step, it chooses either δ_0 or δ_1 by flipping a fair coin. Then, this probabilistic Turing machine would not output the same answer from set $(0, 1, \perp)$ for every single run as the answer would depend on the choice of the transition function at every step. What would be a good way to measure whether such a probabilistic Turing machine computes a function f?

Definition 2.7 (**Probabilistic computation**). Let M be a probabilistic Turing machine. We say that M computes function $f : \{0,1\}^* \to \{0,1\}$ if and only if for all $x \in \{0,1\}^*$, M halts and:

$$Pr[M(x) = f(x)] \geq \epsilon \tag{2.5}$$

where $\frac{1}{2} < \epsilon \leq 1$ and where the probability is taken over all paths (combinations) that the Turing machine could've taken over selecting δ_0 and δ_1.

To see why this is a reasonable definition, consider the worst case when:

$$Pr[M(x) = f(x)] = \frac{1}{2} + \delta \tag{2.6}$$

where $0 < \delta < \frac{1}{2}$. We run the probabilistic Turing machine M, t times and take a majority vote. The probability that we compute $f : \{0,1\}^* \to \{0,1\}$ correctly on some arbitrary input $x \in \{0,1\}^*$ is given by (based on argument in [Kit+02]):

$$p_{success} = 1 - p_{failure} \tag{2.7}$$

$$p_{failure} = Pr[\sum_{i \in [t]} (M_i(x) = f(x)) < \frac{t}{2}] \tag{2.8}$$

where we are indexing run $i \in [t]$ of the Turing machine as M_i. Now the event $\sum_{i \in [t]} (M_i(x) = F(x)) \leq \frac{t}{2}$ can divided into the following events;

$$\sum_{i \in [t]} (M_i(x) = F(x)) \leq \frac{t}{2} = \cup_{s \in [\frac{t}{2}+1]} \left(\sum_{i \in [t]} (M_i(x) = f(x)) = s \right) \tag{2.9}$$

We can compute the probability of event $\left(\sum_{i \in [t]} (M_i(x) = f(x)) = s \right)$ for arbitrary s as follows:

$$Pr\left[\sum_{i \in [t]} (M_i(x) = f(x)) = s \right] = \binom{t}{s} \left(\frac{1}{2} + \delta \right)^s \left(\frac{1}{2} - \delta \right)^{t-s} \tag{2.10}$$

Now we can combine the above equations using the union bound and we get that:

$$p_{failure} \leq \sum_{s \in [\frac{t}{2}+1]} Pr[\left(\sum_{i \in [t]} (M_i(x) = f(x)) = s \right)] \tag{2.11}$$

$$= \sum_{s \in [\frac{t}{2}]+1} \binom{t}{s} \left(\frac{1}{2} + \delta \right)^s \left(\frac{1}{2} - \delta \right)^{t-s} \tag{2.12}$$

$$= \left(\frac{1}{2} + \delta \right)^{\frac{t}{2}} \left(\frac{1}{2} - \delta \right)^{\frac{t}{2}} \sum_{s \in [\frac{t}{2}+1]} \binom{t}{s} \left(\frac{\frac{1}{2} - \delta}{\frac{1}{2} + \delta} \right)^{\frac{t}{2}-s} \tag{2.13}$$

As $\delta > 0$, $0 < \left(\frac{\frac{1}{2}-\delta}{\frac{1}{2}+\delta} \right) < 1$ and therefore,

$$\sum_{s \in [\frac{t}{2}+1]} \binom{t}{s} \left(\frac{\frac{1}{2} - \delta}{\frac{1}{2} + \delta} \right)^{\frac{t}{2}-s} < \sum_{s \in [\frac{t}{2}]} \binom{t}{s} < \sum_{s \in [t]} \binom{t}{s} = 2^t \tag{2.14}$$

Therefore, we get:

$$p_{success} > 1 - \left(2\sqrt{\frac{1}{4} - \delta^2} \right)^t \tag{2.15}$$

Given the bounds on δ, $\left(2\sqrt{\frac{1}{4}-\delta^2}\right) < 1$ and hence, the larger the t, the higher the probability of success. Moreover, if we want $p_{success} > 1-\delta'$ for some δ', we can set:

$$t = \frac{\log(\delta')}{\log\left(2\sqrt{\frac{1}{4}-\delta^2}\right)} \tag{2.16}$$

which is a constant that does not depend on n. Therefore, we can amplify the probability of success by taking the majority vote after running the Turing machine multiple times. With the definition of randomized computation, we can define the class RTIME and BPP (for bounded-error probabilistic polynomial time) as follows:

Definition 2.8 (**class RTIME**). Let $T : \mathbb{N} \to \mathbb{R}$ be some function. A function f belongs to the set $DTIME(T(n))$ if and only if there exists a $n_0 \in \mathbb{N}$ such that for all $n > n_0$, for all $x \in \{0,1\}^n$, there exists a probabilistic Turing machine M such that it halts in at most $T(n)$ evaluations of the transition function on input x and:

$$Pr[M(x) = f(x)] \geq \epsilon \tag{2.17}$$

where $\frac{1}{2} < \epsilon \leq 1$ and where the probability is taken over all paths (combinations) that the Turing machine could've taken over selecting δ_0 and δ_1.

Definition 2.9 (**class BPP**). A function $f : \{0,1\}^* \to \{0,1\}$ belongs to the class BPP if it belongs to RTIME($T(n)$) where $T(n) = n^a$ for any $a \in \mathbb{N}$. More succinctly, we have that:

$$BPP = \cup_{a \geq 1}\ RTIME(n^a) \tag{2.18}$$

In fact, we can show something stronger than our result in 2.16. Using the Chernoff bound, we can show that if we have a polynomial time probabilistic Turing machine M that correctly computes function f with probability:

$$Pr[M(x) = f(x)] \geq \frac{1}{2} + \frac{1}{p(|x|)} \tag{2.19}$$

where p is any polynomial, we can construct another probabilistic polynomial time Turing machine that for any polynomial q can compute f with probability for all x [Bar]:

$$Pr[M(x) = f(x)] \geq 1 - 2^{-q(|x|)} \tag{2.20}$$

Therefore, even if the success rate is slightly more than $\frac{1}{2}$, we can make it exponentially close to 1 with respect to input size in polynomial time. We can define BPP in terms of non-probabilistic Turing machines as well: consider that a deterministic Turing machine takes input both $x \in \{0,1\}^*$ (the input to the function) as well as $r \in \{0,1\}^{poly(|x|)}$. Now we are providing the randomness from outside the Turing machine: the transition function is deterministic but the input r vector tells the machine the results to all the coin tosses it requires. Therefore,

Definition 2.10 (**class BPP via deterministic Turing machines**)**.** A function $f : \{0,1\}^* \to \{0,1\}$ belongs to the class BPP if there exists a deterministic Turing machine which halts in polynomial time and there exists a polynomial $p : \mathbb{N} \to \mathbb{R}$ such that for all $x \in \{0,1\}^*$:

$$Pr_{r \in \{0,1\}^{p(n)}}[M(x,r) = f(x)] \geq \epsilon \tag{2.21}$$

where $\frac{1}{2} < \epsilon \leq 1$ and where the probability is taken over all possible r vectors.

We know that by setting $\epsilon = 1$, BPP reduces to P and, therefore, $P \subseteq BPP$ but we do not know if they are equal. A famous problem for which we knew a probabilistic polynomial-time algorithm but not a deterministic polynomial-time algorithm was primality checking, but it was shown to be in P as well. Polynomial identity testing (checking whether two multivariate polynomials are identical) is known to be in BPP, but we do not know of an efficient deterministic algorithm. From above, BPP is bounded by EXP as if $f : \{0,1\}^* \to \{0,1\}$ is in BPP, for input $x \in \{0,1\}^*$ we could enumerate over all possible (exponential number of) paths or r vectors and run the corresponding Turing machine for all of them. Then, we take a majority vote, and we are guaranteed that the majority vote is $f(x)$ due to the definition of BPP. Therefore, P $\subseteq$ BPP $\subseteq$ EXP, but we know that at most one of the two subsets is strict due to the time hierarchy theorem. However, most computer scientists believe that P = BPP. This is because it is a corollary of a conjecture believed by many that exponentially strong pseudorandom generators exist (we can take a small number of random bits and convert them into an exponentially large number of pseudorandom bits such that a polynomial-time machine cannot tell that the pseudorandom bits are not *actually* random).

Extended Church-Turing thesis: Note that we do not label the names P, BPP, or EXP with the fact that these definitions are for Turing machines, as our model for computation. We saw in the last section that the definition of R should not change if we assume the Church-Turing thesis. Similarly, there is a stronger conjecture, the Extended Church-Turing thesis:

Law of computation 2.2 (**Extended Church-Turing thesis**)**.** Every computation on a physically realizable and scalable model of computation can be simulated on a probabilistic Turing machine with at most a polynomial overhead. More precisely, if to compute function f, another model of computation takes time $T(n)$ for all $x \in \{0,1\}^n$ for all $n \in \mathbb{N}$, a probabilistic Turing machine takes at most time $aT(n)^b$ for any $a, b \in \mathbb{R}$ for all $x \in \{0,1\}^n$ for all $n \in \mathbb{N}$ to compute $f(x)$.

Just like the non-extended version, this is not a mathematical law or theorem, merely a belief. A consequence of believing in this thesis is that if there is a function that

takes polynomial time in another model of computation, it takes polynomial (perhaps, a larger polynomial but a polynomial nevertheless) time via a probabilistic Turing machine. Therefore, by changing our model of computation, we cannot take something that is in EXP but not in BPP (or P) with respect to Turing machines and implement it in polynomial time on another model of computation as that would lead to a contradiction. Therefore, assuming the extended Church-Turing thesis, in some sense, maintains a fixed wall between functions in BPP and outside BPP as this wall does not change no matter what Turing-complete model of computation we used. Note that within BPP, the model of computation *can* make a difference in the time complexity of a function. It has been shown that parallel computing, lambda calculus, popular programming languages Python, C++ and JavaScript, cellular automata, and DNA-based computers all abide by the Extended Church-Turing thesis.

However, the extended version is more controversial compared to the standard version precisely due to quantum computers - we know functions for which the best known classical algorithm takes exponential time (they might not be in BPP), but a quantum algorithm takes polynomial time. For example, Shor's algorithm for factoring. Therefore, the feasibility of speed-ups through quantum computation threatens a very foundational pillar of computation and complexity theory.

Alternate characterization of P via circuits: Until now we have been dealing *uniform* computation. Our objects of study were functions of the form: $f : \{0,1\}^* \to \{0,1\}$, that is, the input length is arbitrary. In *non-uniform* computation, we restrict our attention to computing functions that only take in a string of length $n \in \mathbb{N}$, $f : \{0,1\}^n \to \{0,1\}$. In this regime, we have a very powerful model of computation (in terms of computability), circuits. Formally, a circuit that computes function $f : \{0,1\}^n \to \{0,1\}$ can be represented as a directed acyclic graph such that there are n vertices with each one of them representing one of the input bits, one output vertex that does not have any fan-out edges and multiple gate edges that have a certain number of fan-in edges and one fan-out edge. Each vertex is associated with a value: either 0 or 1. The gate vertices combine the value of the vertices that point to it and put the output into the vertex that the gate vertex points to, according to some fixed function. The output for some function C on input $x \in \{0,1\}^*$ is defined as the value of the output vertex. For boolean functions, the disjunctive normative form, uses only AND, OR and NOT gates defined in the usual way:

$$AND(i,j) = 1 \text{ iff } i = j = 1 \tag{2.22}$$

$$OR(i,j) = 0 \text{ iff } i = j = 0 \tag{2.23}$$

$$NOT(i) = (i+1) \ \% \ 2 \tag{2.24}$$

Therefore, AND, OR, NOT gates form a universal set of gates: all boolean functions can be computed using circuits with $AND/OR/NOT$ gates available. Informally, we can think of circuit computing $f : \{0,1\}^n \to \{0,1\}$ as applying $AND/OR/NOT$ gates on the input and the output of the gates such that we can, if we follow the same procedure for all $x \in \{0,1\}^n$, the final output (that is not fed into another gate), is $f(x)$.

It might be tempting to define P as the class of functions that require a polynomial number of gates in the input length. However, P is defined over arbitrary input functions, and this definition allows for different circuits for different input lengths. This class is called $P_{/poly}$, and in fact, it can be shown that it contains uncomputable functions. To define P properly, we need to take into account that even though simulating a circuit with a polynomial number of gates will take a polynomial number of steps for a Turing machine, finding the circuit may take super-polynomial time. Therefore, we can alternatively characterize P as:

Definition 2.11 (**class P**)**.** A function $f : \{0,1\}^* \to \{0,1\}$ belongs to P if and only if for every $n \in \mathbb{N}$, there is a polynomial-time Turing machine M_f such that on input 0^n, it outputs the description of a circuit C_n such that $\forall x \in \{0,1\}^n$, $C_n(x) = f(x)$

For more details on boolean circuits, please refer to Chapter 6 of [AB09]. We introduce this here to draw connections to our definition of BQP in later sections.

2.1.3 NP and NP-completeness

In theoretical computer science, it has been extremely challenging to prove lower bounds for functions: statements of the form 'function $f : \{0,1\}^* \to \{0,1\}$ must at least take $g(n)$ time' where g is some function and n is in the input size. Therefore, for most functions, we don't know what is the most efficiently they might be computed. However, to deal with this, computer scientists have been able to divide functions into useful categories such that if we know an efficient algorithm for one function in that category, we know an efficient algorithm for all functions in that category. A very well-studied and extremely useful category is of search problems. To illustrate this, we introduce the computational class of NP:

Definition 2.12 (**class NP**)**.** A function $f : \{0,1\}^* \to \{0,1\}$ is in NP if and only if there exists a polynomial time Turing machine V_f such that:

$$\forall x \in \{0,1\}^*, f(x) = 1 \text{ iff } \exists_{w \in \{0,1\}^{p(|x|)}} V_f(x, w) = 1 \tag{2.25}$$

where p is some polynomial.

We sometimes call w, the 'proof' of $f(x) = 1$. We can think of $f(x)$ as a search problem where we need to find some string $w \in \{0,1\}^*$ (which is at most polynomial in input size) that satisfies some conditions which can be checked in polynomial time. If such a w (proof) exists then $f(x) = 1$ and if not, then $f(x) = 0$. Note that for functions in P, we can simply replace V_f with the Turing machine M_f that computes f in polynomial-time and satisfy the definition. Therefore, $P \subseteq NP$. Whether $NP \subseteq P$ is unknown and is one of the fundamental questions in theoretical computer science but it is believed to be false. If, $P = NP$, then the consequences would be spectacular: we would be able to solve all optimization problems, break certain cryptosystems, and find proofs for many mathematical theorems and much more in polynomial time. Moreover, we know that $NP \subseteq EXP$ as for some $f(x)$ such that $f \in NP$, we could simply try all possible proofs in exponential time using the verifier V_f and if for any of them verifier outputs 1, we know that $f(x) = 1$ and $f(x) = 0$ otherwise.

Definition 2.13 (**class NP-hard**)**.** A function $f : \{0,1\}^*$ is $NP - hard$ if and only if for all $g : \{0,1\}^* \to \{0,1\} \in NP$, g reduces to f with polynomial overhead.

That is, if we knew how to solve $f \in NP - hard$, then, for every input x of $g \in NP$, we can massage the input using a polynomial-time Turing machine R such that $g(x) = f(R(x))$. Therefore, using this, if we can solve any NP-hard problem in polynomial time, we could solve all problems in NP in polynomial time. The last classical class we discuss is $NP - complete$:

Definition 2.14 (**class NP-complete**)**.** A function $f : \{0,1\}^*$ is $NP - complete$ if it is $NP - hard$ and $f \in NP$.

Therefore, in some sense, these are the hardest functions to compute inside NP such that if we can compute these efficiently, we can compute all functions in NP efficiently. Examples of functions that are $NP - complete$ include 3SAT, maximum independent set decision problem, integer programming, and as we will show in this thesis, the Ising model decision problem and the Rydberg dynamics (in the negligible long-range interactions model).

2.2 Quantum computation

2.2.1 Quantum circuits

One of the most significant development of the 20^{th} century has been that we have realized that reality is not *classical* and, in fact, is based on quantum mechanical particles. When trying to simulate large quantum mechanical systems, physicists realized that the task was intractable on classical computers. Even for 20 two-level particles, the Hilbert

space governing the system had size 2^{20} and hence, exponential in the number of particles, making simulations inefficient. Noting this, Feynman suggested that a computer based on quantum mechanical effects such as interference and entanglement may do better [Fey99]. Deutsch formalized these ideas and showed that quantum computers could compute the XOR of two bits faster than classical computers could [Deu85]. Here, we introduce quantum computation through the standard circuit model. Quantum Turing machines were also defined by Deutsch and are equivalent to the circuit model.

Quantum states: The basic unit of computation for a quantum computer is the qubit (this can be generalized to more than 2 dimensions). Unlike classical computers, a qubit is not defined to be either in a 1 or 0 state, and not even a probability distribution over 1 and 0 states but instead, we define the state of the qubit as:

$$|\psi\rangle = c_0 |0\rangle + c_1 |1\rangle \tag{2.26}$$

where $|0\rangle, |1\rangle$ are two orthonormal basis states of a two-level quantum and $c_0, c_1 \in \mathbb{C}$ such that $|c_0|^2 + |c_1|^2 = 1$. When we measure the qubit, the amplitude squared $|c_i|^2$ is the probability of measuring the qubit in state $|i\rangle$ (this is called the Born rule). For a state defined over n qubits, we can generalize this to:

$$|\psi\rangle = \sum_{i}^{2^n-1} c_i |i\rangle \tag{2.27}$$

where i runs over all possible 2^n orthonormal basis states of the system, which we can describe as all possible $i \in \{0,1\}^n$ strings. Furthermore, it must be that $\sum_i^{2^n-1} |c_i|^2 = 1$. A quantum circuit takes in a state ψ_i, evolves it using quantum gates and outputs the final state ψ_f.

Quantum gates: We saw that the disjunctive normal form told us that $AND/OR/NOT$ were a universal gate set for classical computation. Analogously, we define gates for quantum computation. Quantum gates are operators defined from $\mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$ where n is the number of qubits they act on. For quantum mechanics to be consistent, all quantum operators must uphold two properties:

- It must preserve norm, that is, if we have initial state $|\psi_i\rangle = \sum_i^{2^n-1} c_i |i\rangle$ and after applying a quantum gate $G : \mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$, let's say we get $G(|\psi_i\rangle) = |\psi_f\rangle = \sum_i^{2^n-1} d_i |i\rangle$, then:

$$\sum_{i}^{2^n-1} |c_i|^2 = 1 \implies \sum_{i}^{2^n-1} |d_i|^2 = 1 \tag{2.28}$$

- It must be a linear operator, that is, its action can be completely defined by how it act on the basis vectors of Hilbert space: we apply it to the different basis vectors

and then add the results up weighted by the c_i coefficients. Therefore, for a quantum gate $G : \mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$:

$$G(|\psi_i\rangle) = \sum_{i}^{2^n-1} c_i G(|i\rangle) \tag{2.29}$$

Combining these two properties tells us that the operators are *unitary*. This means that they can be expressed as matrices of size $\mathbb{C}^{2^n} \times \mathbb{C}^{2^n}$ and are invertible with $GG* = I$. This is exactly the reason why Hamiltonians governing quantum systems are Hermitian as Schrödinger's equation tell us that for a time-independent Hamiltonian H, the quantum operation it is equivalent to is given by:

$$G = e^{-iHt/\hbar} \tag{2.30}$$

It can be proven that if H is unitary, then G is Hermitian. Let us describe some noteworthy quantum gates. In the following, we have assumed that:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{2.31}$$

- **Single-qubit gates:** Pauli gates, defined as follows:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{2.32}$$

  where X is akin to a bit-flip. The Hadamard gate:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.33}$$

  It maps $|0\rangle \to \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $|1\rangle \to \frac{|0\rangle-|1\rangle}{\sqrt{2}}$ and therefore, there is equal probability of each state after measurement when the Hadamard gate is applied to the basis states.

- **Two qubit gates:** The swapping gate simply swaps the states of two qubits i and j. If we consider the basis states in the order $|00\rangle$,$|01\rangle, |10\rangle, |11\rangle$, we can represent the swapping gate as the following matrix:

$$SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.34}$$

The CNOT gate flips the state of the second qubit if the first qubit is in state $|1\rangle$. Using the same basis order as above, we can define it as:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.35}$$

- **Three qubit gates:** The Toffoli gate acts on three qubits and flips the third qubit if and only if the first two qubits were both in state $|1\rangle$. That is, it's action is defined by $|a, b, c\rangle \rightarrow |a, b, c \oplus a.b\rangle$. Note that we can implement NAND using a Toffoli gate, as Toffoli $|ab\ 1\rangle = |ab\ NAND(a, b)\rangle$. As we can implement $AND/OR/NOT$ using only $NAND$ gates, the Tofolli gate is universal for classical computation. We can define the Toffoli gate as a matrix:

$$\text{Toffoli} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.36}$$

Given that the elements of a quantum gate can be arbitrary complex numbers, we do not have a universal gate set that can implement all quantum gates in the same sense as the classical case. However, multiple gate sets that can approximate (with arbitrarily precision) any quantum gate exist. [Deu89; Kit97] proved that for every $\epsilon > 0$, every gate G acting on n qubits can be decomposed into $O(\left(n \log \frac{1}{\epsilon}\right)^3$ gates from the set: $\{\text{Hadamard}, \text{Toffoli}, \sqrt{Z}\}$ such every element of the G differs from the corresponding element of the constructed $\tilde{G}$ by at most ϵ:

$$|G_{i,j} - \tilde{G}_{i,j}| < \epsilon \tag{2.37}$$

for all $i, j < n$. Similarly, it has been shown that the set $\{H, T, CNOT\}$ is also universal in the same sense where:

$$T = \begin{bmatrix} 1 & 0 \\ 0 & \exp\left(\frac{i\pi}{4}\right) \end{bmatrix} \tag{2.38}$$

Furthermore, the Solovay-Kitaev algorithm tells us that we can efficiently find the constructed gate $\tilde{G}$ in time $O(poly(\log(1/\epsilon)))$ [DN05].

Defining quantum computation: Now, what does it mean for the quantum circuit to compute a function $f : \{0,1\}^* \to \{0,1\}$? For every $x \in \{0,1\}^*$, with $|x| = n$, we do the following procedure:

- Let m be some constant such that $m > n$. We initialize an m qubit state $|\psi_i\rangle = |x0^{m-|x|}\rangle$.
- Then, we apply a series of quantum gates to this state from our favourite universal gate set:

$$|\psi_f\rangle = G_{k-1}G_{k-2}...G_1G_0\,|\psi_i\rangle \tag{2.39}$$

  where k is the number of gates we apply.
- Then, we measure the last qubit of $|\psi_f\rangle$ in some computational basis and call it $C(x)$.

Now, if our circuit computed f, it must be that:

$$\Pr[C(x) = f(x)] \geq \frac{1}{2} + \epsilon \tag{2.40}$$

for some $\epsilon > 0$. The choice of ϵ is arbitrary as we can amplify this probability by running the circuit again a polynomial number of time. Another way to understand the condition is that we can think of the final state as $|\psi_f\rangle = \sum_i^{2^m-1} c_i\,|i\rangle$ where $i \in \{0,1\}^m$. We sum up $|c_i|^2$ for all i such that $i_{-1} = f(x)$ and this sum must be strictly greater than $\frac{1}{2}$.

Now that we have defined quantum circuit, we can define the quantum analogue of BPP just as we defined P using classical circuits. It is called BQP (for bounded-error quantum polynomial time):

Definition 2.15 (class BQP). A function $f : \{0,1\}^* \to \{0,1\}$ belongs to BQP if and only if for every $n \in \mathbb{N}$, there is a polynomial-time Turing machine M_f such that on input 0^n, it outputs the description of a *quantum* circuit C_n that computes the restriction of f to input length n, $f_n : \{0,1\}^n \to \{0,1\}$.

We saw earlier that the Toffoli gate is universal for classical computation. Therefore, we see that classical computation is a subcase of quantum computation. That is, we know that $P \subseteq BQP$. Whether the converse is true is unknown but believed to be false. Shor's algorithm [Sho94] shows that factoring is in BQP, but if $P = BQP$, we would have polynomial-time classical algorithms for factoring, a very well-studied problem that is considered classically hard. Unlike running time, there are provable speed-ups shown for quantum algorithms for other resources: for example, queries to an oracle in Grover's algorithm and the Bernstein–Vazirani algorithm. Also, given exponential time, we can

compute all matrix products of Equation 2.39 and compute $|\psi_f\rangle$ explicitly. Then, we could output 1 or 0 based on which one of the two has the higher probability of being measured via the last qubit of $|\psi_f\rangle$. Therefore, every function that is in BQP is also in EXP. The Quantum Algorithms Zoo (https://quantumalgorithmzoo.org/) is a useful resource for an overview of all known quantum algorithms.

Similarly to the classical cases, we can define quantum analogs for $NP, NP-hard$, and $NP-complete$ classes as $QMA, QMA-hard$, and $QMA-complete$ where QMA stands for Quantum Merlin Arthur. The only difference is that the proof is a quantum state instead of a classical string, and the verifier is a polynomial-time quantum algorithm. A canonical example of a QMA-complete problem is 2-local Hamiltonian [KKR06], where for a given Hamiltonian that is a sum of Hamiltonians acting only on 2 qubits, we must find whether the ground state energy of the Hamiltonian is below a constant a or above a constant b. We can modify the class QMA by permitting only classical proofs; this class is called QCMA, and whether $QMA = QCMA$ is unknown.

2.2.2 Adiabatic quantum computation

Adibatic theorem: There is another equivalent paradigm of modelling quantum computation: adiabatic quantum computation. This section is primarily based on [AL18; Chi]. Let's say there is a system with n qubits initialized in the state $|\psi(0)\rangle$. Schrödinger's equation tells us that the time-dependent Hamiltonian $H(t)$ associated with the system governs how the system evolves with time, through the following differential equation:

$$i\hbar\frac{d}{dt}|\Psi(t)\rangle = H(t)|\Psi(t)\rangle \tag{2.41}$$

The adiabatic theorem, which underlies adiabatic quantum computation, was introduced by Max Born and Vladimir Fock in 1928 [BF28]. If $|\psi(0)\rangle$ is the n^{th} eigenstate of $H(0)$, then according to the theorem, if we vary $H(t)$ infinitesimally slowly and smoothly from $H(t=0) = H_{initial}$ to $H(t=T) = H_{final}$, then at the final time T, $|\psi(T)\rangle$ would be the n^{th} eigenstate of $H(T)$ as long as there is a gap between the n^{th} eigenvalue of system and the rest of the eigenspectra at all times t. That is, the state evolves to remain in the eigenstate as long as the evolution is very slow. Let us say we start in the ground state of $H(0)$ and evolve it to $H(1)$ where $H(s), s \in [0,1]$ and we vary s smoothly and reach $s=1$ when $t=T$. Then, the precise statement is that:

$$T \to \infty \implies |\langle\psi(T)|\phi_0(1)\rangle|^2 = 1, \tag{2.42}$$

where $\phi_n(s)$ represents the n^{th} eigenstate of $H(s)$. But if T is not infinity, how slow is slow enough for this to be approximately true?

An approximate analysis in [Ami09] shows that the adiabatic theorem would be approximately valid if the following inequality is maintained for all $i \neq 0$:

$$\frac{1}{T} \max_s \frac{|\langle \phi_0(s)|\partial_s H|\phi_i(s)\rangle|}{|E_0(s) - E_i(0)|} << 1 \tag{2.43}$$

where $E_n(s)$ is the n^{th} eigenvalue of $H(s)$. Note that this takes into account the condition that there must be gap between the ground state and the rest of the eigenvalues. If there isn't a gap for some value of s, the denominator will go to 0 and we will not satisfy the condition. Due to this inequality, the minimum spectral gap is an important quantity defined as:

$$\Delta = \min_s [E_1(s) - E_0(s)] \tag{2.44}$$

In fact, it can be shown that T must be proportional to the inverse square of Δ for there to be a high probability that we are in the ground state of $H(1)$ [AL18]. It was shown in [EH12], that for T:

$$T \sim \frac{\max_s ||\dot{H(s)}||^2}{\Delta^2} \tag{2.45}$$

is the nearly optimal upper bound on the running time in the adiabatic theorem.

Adiabatic quantum computation: In 2000, [Far+00] introduced a model for computation that exploits the adiabatic theorem. The model works as follows, let H_0, and H_1 be two $k-$local Hamiltonians. The ground state of H_0 is known and can be prepared. The ground state of H_1 encodes the solution to the problem we want to solve: that is, after finding the ground state of H_1, we can perform an efficient computation to compute the function $f : \{0,1\}^* \to \{0,1\}$. We define the schedule to be a function $s : [0,T] \to [0,1]$ and run adiabatic evolution on the Hamiltonian given by $H(s) = (1-s)H_0 + sH_1$. T is the running time of the algorithm and is defined to be the minimum time required, such that $|\langle \psi(T)|\phi_0(1)\rangle| < \epsilon$ for some defined $\epsilon > 0$. We have already seen that T is proportional to $\frac{1}{\Delta^2}$ where Δ depends on H_0, H_1 and the schedule function s. This makes analyzing the run time of computing a function f challenging since solving for Δ analytically is not always possible. Furthermore, the run time strongly depends on the choice of H_0, H_1 and s, as there may be multiple choices that will compute f.

As an example, consider the NP-complete 3SAT problem as per [Far+00; Chi]: let there be n variables and m clauses, each consists of three literals which may be the variable or its negation. The clauses are of the form: $C = (x_i \vee x_j \vee \bar{x_k})$ and all clauses are connected with conjunctions: $C_m \wedge ..... \wedge C_n$. The function outputs 1 if and only if there is an assignment x^* such that the formula $C_m \wedge ..... \wedge C_n$ outputs 1, that is, all clauses are satisfied. To encode this problem via a Hamiltonian, we let each qubit represent one

variable of 3SAT, then for $|i\rangle$ such that $i \in \{0,1\}^n$, $H_c(|i\rangle) = 0$ if string i satisfies clauses c and 1 otherwise. We define:

$$H_1(|i\rangle) = \sum_c^m H_c(|i\rangle) \tag{2.46}$$

Note that $H_1(|i\rangle) = 0$ if and only if all clauses are satisfied by assignment i. Therefore, by finding the ground state of H_1, we can solve 3SAT.

When this model was introduced, it was shown that it could be simulated using the circuit model efficiently. A few years later, in [Aha+08], it was in fact, shown that an arbitrary quantum circuit can be efficiently simulated via adiabatic quantum computation with a linear schedule: that is, it was shown that for a collection of gates applied to some initial state, we can determine a Hamiltonian such that the ground state encodes the resulting state of the circuit. Therefore, this model is equivalent to the circuit model. In fact, adiabatic versions of the Grover's, Deutsch-Jozsa, and Bernstein-Vazirani algorithm have been shown [AL18].

Similar to 3SAT, multiple optimization problems have been shown to be encodable via Hamiltonians on particles. This has lead to the rise of a new paradigm of adiabatic quantum computation: adiabatic quantum optimization. The key idea is to steer the dynamics of a quantum mechanical system such that the ground state encodes the solution to the optimization problem we wish to solve. Note that for quantum simulators attempting to implement this technique, the *classes* of Hamiltonians they can implement is critical in determining what optimization problems they can potentially solve. It is with this motivation that we hope to show that we can map the Ising model problem onto the dynamics of Rydberg atom arrays.

Lastly, there is a connection between $QMA-complete$ problems and universal quantum computation (computations that can be done in the circuit model) via adiabatic evolution. Often when proving whether a class of Hamiltonians can implement universal quantum computation adiabatically, it turns out that if they can, then determining whether the ground state energy of that class of Hamiltonians is below a certain constant is $QMA-complete$ [Aha+09]. Therefore, one way to show that a quantum simulator can perform arbitrary quantum computation is proving that it can implement one of these $QMA-complete$ problems in [BL08]

Quantum Approximate Optimization Algorithm: Here, we describe a hybrid quantum algorithm - the quantum approximate optimization algorithm [FGG14] inspired partly by the adiabatic quantum computation. Let H_1 be the cost Hamiltonian encoding

a combinatorial optimization problem defined on n qubits whose maximum value we want to find. We define the operator:

$$U(H_1, \gamma) = \exp(-iH_1\gamma) \tag{2.47}$$

We define $B = \sum_{i=0}^{n} \sigma_i^x$ where i runs over all qubits. We define:

$$U(B, \beta) = \exp(-iB\gamma) \tag{2.48}$$

Now, we begin with some initial state $|\psi_i\rangle$: the a uniform superposition over all possible states. Let p be the depth of our circuit, we run the following computation:

$$|\psi_f\rangle_{\gamma,\beta} = U(B, \beta_p)U(H_1, \gamma_p)...U(B, \beta_0)U(H_1, \gamma_0)\,|\psi_i\rangle \tag{2.49}$$

for some values of $\{\beta\}$ and $\{\gamma\}$. If we define $\{\beta\}^*$ and $\{\gamma\}^*$ such that $\langle\psi_{f_{\gamma^*,\beta^*}}|H_1|\psi_{f_{\gamma^*,\beta^*}}\rangle$ is maximized then, it can be shown that:

$$p \to \infty \implies \langle\psi_{f_{\gamma^*,\beta^*}}|H_1|\psi_{f_{\gamma^*,\beta^*}}\rangle = \max(H_1) \tag{2.50}$$

where $\max(H_1)$ is the maximum value of H_1. Given this, the algorithm works as follows: let depth be set to some fixed p, we initialize the $\{\beta\}, \{\gamma\}$ vectors to some value and implement circuit defined by Equation 2.49. Then, we measure the value of H_1 for the final state. We do this multiple times and use a classical sub-routine to select the next $\{\beta\}, \{\gamma\}$ vectors. One interesting feature of this algorithm is that as p increases, in theory, we are guaranteed that the maximum value of H_1 we have observed either increases or stays the same as we can set $\gamma_{p-1} = \beta_{p-1} = 0$ and we would get the same value of H_1 as in with $p-1$ depth. The algorithm's performance has been emperically analyzed for constraint satisfaction problems [FGG14] and MaxCut [Zho+18] and it is an attractive candidate for near-term implementation. Here again, like the adiabatic algorithm, we note that that the classes of Hamiltonians that can be implemented on a quantum system determines for which problems we can run QAOA on the system.

2.2.3 Quantum computing platforms and near-term experiments

Quantum computing platforms: Even though we have focused on quantum computation, programmable quantum mechanical systems have another important and closely related application: simulating complex many-body phenomena. There have been proposals to probe aspects of many-body physics, quantum chemistry, quantum gravity proposals and topological physics using the upcoming platforms [Ale+19].

Multiple platforms are being pursued for potential candidate qubits for scalable quantum simulators and computers. These include trapped ions, ultracold atoms, Rydberg

atom arrays, superconducting qubits, and diamond vacancy centers. [Alt+19] offers a useful review of these platforms and their idiosyncrasies. In general, we note that each platform differs in qubit connectivity, noise properties, coherence times, gate speeds, and the kind of gates that can be natively expressed [Ale+19]. Therefore, the type of computational problems that can be attempted on each platform in the near-term differ. The thesis will be focusing on Rydberg atom arrays consisting of ^{87} Rubidium atoms in two-dimensional arrays trapped via optical tweezers.

Near-term experiments to demonstrate quantum computational advantage: The end goal of building quantum computers is to solve useful problems that are intractable on classical hardware. However, to attain that goal, in the near-term, we must first demonstrate experimentally that there are problems that we may potentially solve using near-term hardware that we believe are hard to solve classically. Therefore, we must focus on problems whose classical complexity can be analyzed and proved to be hard under plausible complexity-theoretic assumptions [HM17]. There have been several proposals:

- It has been shown that an efficient classical algorithm that samples approximately from a QAOA circuit even with depth $p = 1$ would lead to the collapse of the polynomial hierarchy (PH) [FH16] which is strongly believed to be unlikely. Non-collapse of PH is similar to the belief that $P = NP$.

- Similarly, it has been shown that approximately and efficiently sampling from the output distribution of instantaneous quantum circuits (IQP) would lead to the collapse of the PH [BJS11]. These are circuits for which all native gates commute with each other. Furthermore, it has been shown that efficient sampling from the distribution of identical bosons that have been scatted due to a linear interferometer would also lead to the collapse of the PH [AA11].

- The recent experimental demonstration of quantum computational advantage or quantum computational supremacy by the Martinis group and Google Quantum AI [Aru+19] was based on random circuit sampling for qubits in a two-dimensional grid. The classical hardness of this task relies on the QUATH conjecture [AC16], which says that there is no classical polynomial-time algorithm that can predict certain proprieties relating how the application of a random quantum circuit to a specific input relates to the application of the random quantum circuit on other inputs.

Therefore, these problems are good candidates for near-term experiments.

2.3 Rydberg atom arrays for quantum computation

There has been tremendous progress made in the last decade towards the scalability and gate fidelities of Rydberg atom arrays as a quantum computing platform. [] and [] showed the ability to arbitrary trap atoms in one and two dimensions using optical tweezers. [] demonstrated the use of 51 Rubidium atoms to probe many-body physical phenomena, while [] has demonstrated high gate fidelities for a universal gate set. In this section, we describe some fundamentals of Rydberg atoms and their properties and end with a review of what problems can be encoded onto the dynamics of Rydberg atom arrays.

Rydberg atoms and interactions: This section is primarily based on [SWM10; BL16]. An atom is in a Rydberg state if its valence electron is excited to a very large principal quantum number ($n >> 1$), and it is far from the nucleus core. A Rydberg atom is labelled by the quantum number of this outermost electron. The energy is given by [SWM10]:

$$E_{nlj} = -\frac{Ry}{(n - \delta_{lj}(n))^2} \tag{2.51}$$

where $Ry = 13.6$ eV is the Rydberg constant of energy where δ_{lj} represents the defect based on the l, j numbers of the electron. In some sense, $n^* = n - \delta_{lj}(n)$ represents the effective principal quantum number. The wave function of the electron of a Rydberg excited atom is given by:

$$|nljm_J\rangle = P_{nl}(r)\frac{|ljm_j\rangle}{r} \tag{2.52}$$

where P_{nl} is the associated Legendre polynomials. Rydberg states can be long-lived and their lifetime t_{nl} is given by [SWM10]:

$$\frac{1}{t_{nl}} = \frac{1}{t^0_{nl}} + \frac{1}{t^{bb}_{nl}} \tag{2.53}$$

where T^0_{nl} is the life time at temperature $T = 0$ and is of the form $O(poly(n^*))$ where the degree is approximately 3 for all alkali atoms. Moreover, the t^{bb}_{nl} term decreases the lifetime due to blackbody radiation and is given by:

$$\frac{1}{t^{bb}_{nl}} = \frac{4\alpha^3 k_B T}{3\hbar n^2} \tag{2.54}$$

where k_B is the Boltzmann constant and α is the fine structure constant and T is the temperature under consideration. We note that for both terms n occurs in the denominator with a positive power. Therefore, the larger the n, the longer the lifetime and it scales approximately like $O(n^3)$.

As the valence electron is at a large distance from the nucleus or the atom core, the atom in the Rydberg state can develop a large dipole moment, a measure of how charge is distributed in the system. Given this, Rydberg atoms are very sensitive to external fields.

Now, let us consider the case where there are two Rydberg atoms. To compute the Rydberg potential, i.e., the Van der Waals shift of being in sate $|n_a, l_a, j_a, (m_j)_a\rangle \otimes |n_b, l_b, j_b, (m_j)_b\rangle$, Therefore, two atoms a and b in their respective Rydberg state which are at a distance s (let $\hat{s}$ be the unit vector separating the two atoms) interact mainly through an electric dipole-dipole interaction [BL16] given by:

$$V_{dd} \approx \frac{\hat{d}_a \hat{d}_b - 3(\hat{d}_a.\hat{s})(\hat{d}_b.\hat{s})}{4\pi\epsilon_0 s^3} \tag{2.55}$$

where $\hat{d}_a$ and $\hat{d}_b$ are the electrical dipole moment of each of the two atoms and ϵ_0 is the permittivity of a vacuum. The task of computing the Van der Waal energy shifts therefore, is reduced to computing the matrix elements of the form:

$$|\,|n'', l'', j'', m''_j\rangle \otimes \langle n', l', j', m'_j|V_{dd}|n_a, l_a, j_a, (m_j)_a\rangle \otimes |n_b, l_b, j_b, (m_j)_b\rangle\,|^2 \tag{2.56}$$

where we must sum over all possible n'', l'', j'', m''_j and n', l', j', m'_j states of the atom pair and normalize to obtain the energy shift. One way to understand this it that this encodes how likely or unlikely it is to remain in state $|r_a\rangle$ and $|r_g\rangle$. To find any particular element, we must compute the matrix element of $\langle \frac{1}{s^3} \rangle$ and $\hat{d}$ which is possible given that we know the wave-functions of an electron associated with a particular state. In theory, to compute the energy shift and hence, this sum, we must iterate over all possible other states. In the case when $n_a = n_b$ and $l_a = l_b = 0$, both Rydberg atoms will be excited to the $|nS\rangle$ state. In this case, when R is large enough, the energy shift to the Van der Waal's force takes the form:

$$V(s) = \frac{C_6}{s^6} \tag{2.57}$$

where C_6 scales as $n^1 1$. Therefore, these interactions scale fast with increase in shell number.

The dynamics of Rydberg atom arrays: As in the Introduction, the dynamics of Rydberg atom arrays are governed by the following Hamiltonian:

$$H_{Ryd} = \sum_{\nu} (\Omega_\nu \sigma^x_\nu - \Delta_\nu n_\nu) + \sum_{\nu<w} V(|\vec{x}_\nu - \vec{x}_w|, \nu, w) n_\nu n_w \tag{2.58}$$

where ν, w are qubits realized by atoms and the n operator has eigenvalue 1 for state $|r\rangle$ and 0 for state $|g\rangle$. Each atom is individually controlled by a laser, and Ω_ν is the Rabi frequency determining the intensity of the laser that is associated with qubit ν and

determines the amplitude of flipping between state $|g\rangle$ and $|r\rangle$ and therefore, is the coefficient of the σ_ν^x operator while Δ_ν is the laser detuning for qubit ν controlling the n_μ operator. We assume that these two constants are tunable parameters. $\vec{x}_\nu$ refers to the position of qubit ν, and V is the Rydberg potential, a function of the distance between two qubits and also, which particular excited Rydberg state qubit ν and w are coupled to. That is, it depends on $n_a, l_a, j_a, (m_j)_a$ and $n_b, l_b, j_b, (m_j)_b$, the Rydberg states that ν and w are coupled to, respectively.

Quantum Optimization of Maximum Independent Set via Rydberg atom arrays: Implementations of universal gate sets mediated via the Rydberg interactions have been both theoretical described and experimentally implemented with high fidelity [Jak+00; Lev+19; Mal+15; Gra+19].

Here, we will be discussing schemes to implement adiabatic optimization and QAOA on Rydberg atom arrays to solve the Maximum Independent Set (MIS) problem on unit disk graphs, as proposed in [Pic+18a]. Maximum Independent Set is a canonical NP-complete problem stated as follows: given a graph $G = (V, E)$ and a number k, we output 1 if and only if there is an independent set in G which size greater than k. A set of vertices in a graph is independent if no two vertices of the set are connected by an edge in the graph. Unit disk graphs are a special category of graphs that are geometric in nature: we place vertices in Euclidian space and connect two vertices if and only if their distance is less than some critical radius r. Note that MIS on unit disk graphs is NP-complete as well. We can also rephrase MIS as a search problem of finding the size of the largest independent set.

The key to encoding this problem is in the concept of Rydberg blockade. Consider an array of atoms such that all atoms are coupled to the excited Rydberg state $|nS\rangle$, and therefore, V only depends on the distance between two atoms and is given by $V(s) \propto \frac{1}{s^6}$. If we set $\delta_\nu = \delta$ and $\omega_\nu = \omega$ for all qubits ν, there is a critical distance, the blockade distance $r_B = \left(\frac{C_6}{\sqrt{(2\omega)^2+\delta^2}}\right)^{\frac{1}{6}}$ such that if the two atoms are closer than r_B, effectively, the atoms cannot both be in the excited Rydberg state in the ground state of the system no matter what Δ and Ω we chose.

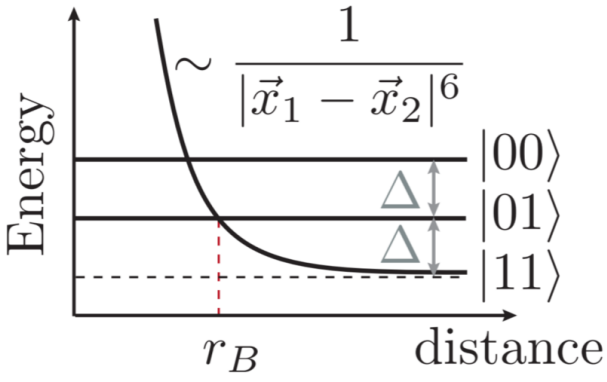

Figure 2.1: Rydberg blockade interactions between two qubits from [Pic+18a]

Therefore, if we relate every qubit in a two-dimensional plane to a vertex of a unit disk graph with $r_B = r$, and set $\Delta > 0$ such that every atom is biased to be in the Rydberg state, the ground state of the system will the largest number of atoms in the Rydberg state provided that no two atoms that are within r_B distance of each other are both in the Rydberg state, effectively encoding the solution to the search version of the MIS problem. In this case, if we ignore long-range interactions, we have shown that by solving the dynamics of Rydberg atom arrays, we can solve an NP-complete problem and hence, Rydberg dynamics (ignoring long-range interactions) are NP-hard.

The thesis will be showing the same result but by reducing the Ising model problem to Rydberg dynamics. In fact, [Pic+18b] showed that the result holds even if we consider long-range interactions by reducing directly from MIS on 3-planar graphs. However, we must find native implementations for useful and hard problems that minimize overhead.

3 | The Ising model and challenges to mapping

We first introduce the Ising model formulation and its associated decision problem, and discuss why it may be useful to map it onto quantum simulators, and the challenges associated with the mapping.

3.1 The Ising model

The Ising model was introduced in [Len20] as a model for studying ferro-magnetism in materials. The original model consists of discrete binary $\{\pm 1\}$ variables on a lattice which can interact with each other. The states of these variables and their interactions with each other determine the energy of the system. Physically, these discrete variables represent the magnetic dipole moments of atoms in a material and their energy dynamics were used to study magnetism in materials. These are one of the simplest of analytically modelling many-body physical phenomena such as phase transitions. We will discuss their role in many-body physics later in this section.

For our purposes, we treat the Ising model as formalism to represent a quadratic binary optimization problem with linear terms. Formally, we define:

Definition 3.1 (**Ising model**)**.** An instance of an Ising spin optimization problem can be completely characterized by $n \in \mathbb{N}$, $\forall i \in [n], h_i \in \mathbb{R}$ and $\forall i, j \in [n]$ such that $0 \leq i < j < n, J_{i,j} \in \mathbb{R}$. We define a configuration $S \in \{-1, +1\}^n$. Each configuration S is associated with an energy given by:

$$E(S) = \sum_{i=0}^{n-1} h_i S_i + \sum_{i<j} J_{i,j} S_i S_j \tag{3.1}$$

The task is to find the configuration S that minimizes $E(S)$, i.e.:

$$\min_{S \in \{-1,+1\}^n} \left(\sum_{i=0}^{n-1} h_i S_i + \sum_{i<j} J_{i,j} S_i S_j \right) \tag{3.2}$$

We can reformulate this search problem as a decision-problem as follows:

Definition 3.2 (**Ising model decision problem**)**.** We define a function $Ising : \{0,1\}^* \to \{0,1\}$ that takes in an Ising spin glass optimization problem parameterized by $n \in \mathbb{N}$, $\forall i \in [n], h_i \in \mathbb{R}$, $\forall i, j \in [n]$ such that $0 \leq i < j < n, J_{i,j} \in \mathbb{R}$, and $k \in \mathbb{R}$ and:

$$Ising(n, h, J, k) = \begin{cases} 1 \text{ if } \exists S \in \{-1, +1\}^n \text{ such that } \left(\sum_{i=0}^{n-1} h_i S_i + \sum_{i<j} J_{i,j} S_i S_j\right) \leq k \\ 0 \text{ otherwise} \end{cases} \tag{3.3}$$

Note that if here each variable can take value from $\{-1, +1\}$. An alternate model would be variables taking value from $\{0, 1\}$ which we can define as follows:

Definition 3.3 (**Quadratic unconstrained binary optimization (QUBO)**)**.** An instance of an QUBO problem can be completely characterized by $n \in \mathbb{N}$, $\forall i \in [n], h_i \in \mathbb{R}$ and $\forall i, j \in [n]$ such that $0 \leq i < j < n, J_{i,j} \in \mathbb{R}$. We define a configuration $N \in \{0,1\}^n$. Each configuration N is associated with an energy given by:

$$E(N) = \sum_{i=0}^{n-1} h_i N_i + \sum_{i<j} J_{i,j} N_i N_j \tag{3.4}$$

The task is to find the configuration N that minimizes $E(N)$, i.e.:

$$\min_{N \in \{0,1\}^n} \left(\sum_{i=0}^{n-1} h_i N_i + \sum_{i<j} J_{i,j} N_i N_j\right) \tag{3.5}$$

We can define the decision problem associated with QUBO in exactly the same way as the Ising model decision problem. Note that this problem is equivalent to the Ising model problem as we can simply construct the one-to-one and onto function $\phi : \{0,1\} \to \{-1,1\}$ for $N_i \in \{0,1\}$ and $S_i \in \{-1,+1\}$:

$$S_i = \phi(N_i) = 2N_i - 1 \tag{3.6}$$

we can verify that $\phi(0) = -1$ and $\phi(1) = 1$. Note that as the function is both onto and one-to-one, we can invert it and get for $S_i \in \{-1,+1\}$ and $N_i \in \{0,1\}$:

$$N_i = \phi^{-1}(S_i) = \frac{S_i + 1}{2} \tag{3.7}$$

Therefore, for every Ising model, we have an equivalent QUBO problem instance:

$$\begin{aligned} \left(\sum_{i=0}^{n-1} h_i N_i + \sum_{i<j} J_{i,j} N_i N_j\right) &= \left(\sum_{i=0}^{n-1} h_i \frac{S_i + 1}{2} + \sum_{i<j} J_{i,j} \frac{S_i + 1}{2} \frac{S_j + 1}{2}\right) \\ &= C + \left(\sum_{i=0}^{n-1} h'_i S_i + \sum_{i<j} J'_{i,j} S_i S_j\right) \end{aligned} \tag{3.8}$$

where $C = \frac{n}{2} + \frac{n(n-1)}{8}$ is a constant, $J'_{ij} = \frac{J_{ij}}{4}$, $h'_i = \frac{h_i}{2} \sum_{j=0}^{n-1} \frac{J_{ij}}{4}$. Therefore, when solving the search problem we can simply solve for $Ising(n, h', J')$ and then add C in the end to get the minimized energy for the QUBO. And for the decision problem, we can simply compute $Ising(n, h', J', k + C)$ if we want to compute $QUBO(n, h, J, k)$. Therefore, we can convert it to an Ising model instance to a QUBO instance and vice-versa with no extra overhead in terms of number of variables and the two problems are equivalent.

NP-completeness: It is classically hard to compute the function $Ising$; the Ising model decision problem is NP-complete. This was proved first in the 1980s in [Bar82]. Here, we provide a simple proof by reducing Maximum Independent Set onto a QUBO problem which is equivalent the Ising model decision problem as seen above:

Theorem 3.1. The QUBO and Ising model decision problem is NP-complete.

Proof. We prove that QUBO is in NP and is NP-hard.

- $QUBO$ is in NP. Here, the proof is simply the configuration of N such that $\left(\sum_{i=0}^{n-1} h_i N_i + \sum_{i<j} J_{i,j} N_i N_j\right) \leq k$. The verifier simply computes $E(N)$ which takes polynomial time as at maximum, we loop over all variables twice (to compute the interaction terms)

- $QUBO$ is NP-hard. We reduce Maximum Independent Set onto the Ising model decision problem via the following reduction:

  - Given: A graph $G = (V, E)$ and k_{MIS}.
  - Associate every vertex i of the graph with a QUBO variable N_i with $h_i = -1$
  - For every edge (i, j), in the graph, set $J_{ij} = n^2 + 1$. For all other pairs of vertices set $J_{ij} = 0$.
  - set $k_{QUBO} = -k_{MIS}$

  Note that we loop over all vertices and all edges and $|E| \leq |V|^2$. Moreover, the input to the problem must be at least $|V|$ large as it specifies number of edges. Therefore, this reduction runs in polynomial time. We show both completeness and soundness:

  - If $MIS(G, k_{MIS}) = 1$, then there exists a set of k_{MIS} vertices in V such that they are not connected by edges among themselves by definition of MIS. Let this set be called M. For every vertex $i \in M$ in this set, we set $N_i = 1$ and for all other vertices $j \in (V - M)$, we set $N_j = 0$. As per the definition of MIS, $\forall i, j \in [|V|]$, $J_{ij} N_i N_j = 0$ unless $i = j$. This is because if $J_{ij} N_i N_j = 1$, then it means that both N_i and N_j are 1 but then this means both i and j

are in set M and by construction, they do not share an edge and $J_{ij} = 0$. Therefore, the interaction terms are all 0. At least k_{MIS} variables are set to 1 as $MIS(G, k_{MIS}) = 1$ and therefore, we get that $E(N) \leq -k_{MIS} = k_{QUBO}$ due to the linear terms.

– $QUBO(n, h, J, k_{QUBO}) = 1 \implies MIS(G, k) = 1$. If k_{QUBO} is non-negative, then k_{MIS} is non-positive and hence, $MIS(G, k_{MIS}) = 1$ as we can simply chose the empty set to be our maximum independent set of size greater or equal to 0. If k_{QUBO} is is negative, then it must be that there exists a configuration of variables $N*$ such that $E(N*) < k_{QUBO} \leq 0$. For all $i \in [n]$, if $N*_i = 1$, then let vertex i be part of the set M which we will show is independent set. Note that this means all the interaction terms must be 0 as if any of the interaction terms were on, then the energy would be non-negative as $n^2 \geq n$. Therefore, we know that the vertices chosen to be inside M do not share any edges. Therefore, M is a independent set. Moreover, as the energy is less than k_{QUBO} and only the linear terms contribute to it, we know that at least $-k_{QUBO} = k_{MIS}$ variables are set to 1 as for all $i \in [n], h_i = -1$ and therefore, the size of M is at least k_{MIS}. Therefore, we have that $MIS(G, k_{MIS}) = 1$.

Therefore, we have that the QUBO decision problem and equivalently the Ising model decision problem are NP-complete and hence, believed to be *classically hard*. □

Why is it useful to map this problem onto our quantum simulator?:

- **Computation and speed-ups:** If we know that the ground state of a Hamiltonian encodes the solution to a classically *hard* problem, then if we can prepare this special state in any way, we would solve the *hard* problem. This is the fundamental principle behind quantum optimization: we want to steer the dynamics of the system such that the resulting state encodes the solution to a classically *hard* problem. To prepare this special state, we can use the adiabatic algorithm or variational algorithms such as variational quantum eigensolvers (VQE) and quantum approximate optimization algorithms (QAOA). The adiabatic algorithm backed by the adiabatic theorem tells us that if we can prepare a Hamiltonian with a known ground state and can smoothly transition to an *interesting* Hamiltonian with an unknown ground state, given enough time, we are guaranteed to find the ground state of the *interesting* Hamiltonian. In standard implementations of VQE, and QAOA, we discretely change between the known and *interesting* Hamiltonians. Therefore, it is critical to be able to implement the *interesting* Hamiltonian. We must study what families of Hamiltonians can be implemented on physical platforms and therefore, what family of problems may be solved via these physical platforms.

As solving the Ising model is NP-hard, it is classically hard and a good candidate for attempting to map onto physical simulators. Note that as it has been shown that we can map unit disk Maximum Independent Set onto 2D Rubidium atom arrays, in theory, there exists a reduction from Ising model onto 2D Rubidium atom arrays but in the regime where number of qubits is limited to a few hundreds, finding a native and efficient reduction is crucial. Whether we can find an efficient and experimentally feasible reduction or not determines whether the problem can be implemented and probed in the near-term. Therefore, the task at hand is to find a mapping that requires a small number of qubits in overhead, and that is experimentally feasible. Moreover, there exists vast literature on how to map many useful and hard problems onto Ising models efficiently [Luc14].

However, a general disadvantage of this approach is that the adiabatic algorithm as well as the variational algorithms are heuristic in nature. For example, the time taken for the adiabatic algorithm to converge to the ground state of the *interesting* Hamiltonian depends on the inverse squared of the minimum energy gap encountered while transitioning from the initial to the final Hamiltonian. A priori, we do not know how to analytically determine this gap. Moreover, even determining whether a Hamiltonian is gapless or gapped is undecidable [CPW15]. Experimental testing on different types of problems may shed light onto the mechanism behind speed-ups via these adiabatic techniques. In general, we do not expect exponential speed-up for NP-complete problems. However, experimentally we may observe sub-exponential speed-ups, and exploring different aspects of the J matrix can help us illuminate the theory of quantum speed-ups in the non-circuit setting. Compared to UD-MIS, this may provide a more straightforward way to test different regimes of problems.

- **Simulation of many-body physics**: Apart from the computational perspective, Ising models are extremely useful both in physics and beyond due to the rich dynamics of the system. Initially proposed as a mathematical model for ferromagnetism in statistical mechanics, it is now widely used to probe phase transitions in different materials. The Ising model and phase transitions have a rich history of connections with the theory of computation. [MM11] is an excellence resource to explore the same. Additionally, it can be used to explain the dynamics of the so-called 'lattice gas' where the motion of the atoms in the gas is modelled as the filled or unfilled positions on a space-time lattice. Many other networks made of two-level systems are studied using the Ising Model. In neuroscience, it can be used as a model for neural function in the brain, where each neuron can either be active or inactive. In fact, the famous Hopfield network is identical to the Ising model description for spin glasses where one is no longer restricted to nearest neighbor interactions [Roj13].

Outside of science, this versatile framework can be use to model socio-economic phenomena in economic opinions, urban segregation, and language change [Sta08].

3.2 Challenges to the mapping

The first step towards the mapping from the classical Ising model problem to a quantum Hamiltonian that represents it is to identify the quantum operators and interactions that play the same role as the classical variables and interaction parameters.

Quantum operators as classical variables: First, let us begin translating the $\{S\}$ and $\{N\}$ classical variables to quantum operators. We hope to use the Rydberg state of the our qubits to encode classical variables. We define the ground state and excited Rydberg state of the qubit as follows:

$$|g\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{3.9}$$

$$|r\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{3.10}$$

We note that the σ^z and n operators in the Hamiltonian will play the role of the S and N classical variables, respectively as:

$$\sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{3.11}$$

$$\implies \sigma^z |g\rangle = -|g\rangle\,,\ \sigma^z |r\rangle = |r\rangle \tag{3.12}$$

$$n = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \tag{3.13}$$

$$\implies n|g\rangle = 0,\ n|r\rangle = |r\rangle \tag{3.14}$$

The $|g\rangle$ quantum state represents the classical $\{-1_{Ising}, 0_{QUBO}\}$ state while the $|r\rangle$ quantum state represent the classical $\{1_{Ising}, 1_{QUBO}\}$ state.

Interaction terms: Consider the Hamiltonian for arbitary values of the h array and J matrix:

$$H_{QUBO} = \sum_{i=0}^{n-1} h_i n_i + \sum_{i<j} J_{ij} \left(n_i \otimes n_j\right) \tag{3.15}$$

$$H_{Ising} = \sum_{i=0}^{n-1} h_i \sigma_i^z + \sum_{i<j} J_{ij} \left(\sigma_i^z \otimes \sigma_j^z\right) \tag{3.16}$$

where $n_i \otimes n_j$ and $\sigma_i^z \otimes \sigma_j^z$ represents that n_i and σ_i^z act on the subspace spanned by i while n_j and σ_j^z act on the subspace spanned by qubit j. For a basis vector of H, $|\psi\rangle$, its energy is given by:

$$H_{QUBO} |\psi\rangle = E |\psi\rangle \tag{3.17}$$

Based on the rules given in Equations 3.12 and 3.14, one energy basis of H_{QUBO} and H_{Ising} is $\{|r\rangle, |g\rangle\}^{\otimes n}$ no matter whether the i^{th} qubit is in state $|r\rangle$ or $|g\rangle$, as each term of H_{QUBO} or $H_{Ising}$is expressed with the n or σ^z and $|r\rangle$ and $|g\rangle$ are eigenstates of n or σ^z operators. We get each state back with a coefficient determined by h array and J matrix. More precisely, to compute the energy associated with a basis vector $|\psi\rangle$, we use the expressions in 5.12 and 5.14 and get that:

$$E_{QUBO}(|\psi\rangle) = \sum_{i=0}^{n-1} h_i N_i + \sum_{i<j} J_{ij} (N_i N_j) \tag{3.18}$$

$$E_{Ising}(|\psi\rangle) = \sum_{i=0}^{n-1} h_i S_i + \sum_{i<j} J_{ij} (S_i S_j) \tag{3.19}$$

where $N_i = 0$ and $S_i = -1$ if qubit i is in state $|g\rangle$ and $N_i = S_i = 1$ if qubit i is in state $|r\rangle$. In fact, the choice of variable N and S was deliberate, Equations 3.18 and 3.19 are exactly the QUBO and Ising description with the classical variable $0/-1$ identified with $|g\rangle$ and variable 1 identified with $|r\rangle$.

Now, if we are able to implement these Hamiltonians with arbitrary h and J values, we would have found a way to map an arbitrary Ising model or QUBO instance onto a quantum simulator. Now, let us see how we may implement the Hamiltonian with Rydberg atom arrays. As we saw in Chapter 2, we can use the laser detuning for individual atoms to control the coefficient of the n operator. However, the coefficient of the interaction terms is a deterministic function of the distance between two atoms i and j, and their quantum state. Let us grant that we can position each atom in three-dimensional space with arbitrary accuracy and can prepare the excited Rydberg state coupled with any n, l quantum numbers. Therefore, for each atom we have 5 real numbers that we can control. We can think of solving equations of the form:

$$V(\vec{r_i}, n_i, l_i, \vec{r_j}, n_j, l_j) = J_{ij} \tag{3.20}$$

where V tells interaction strength between two atoms at location $\vec{r_i}$ and $\vec{r_j}$ coupled to the Rydberg state n_i, l_i and n_j, l_j and J_{ij} is the interaction strength we want to implement and is a constant provided to use by the QUBO instance. To implement an arbitrary QUBO problem through Hamiltonian in 5.15, we would have to solve $\frac{n(n-1)}{2}$

non-linear equations (representing each interaction term) with $5n$ (specifying the properties of each atom) variables. Given this, solutions will not exist for all QUBO instances.

Moreover, Rydberg interactions are local, i.e., they depend on the distance between atoms. For example, for fixed $n_i = n_j, l_i = l_j = 0$, the interaction term is given by:

$$V(|\vec{r_i} - \vec{r_j}|) = \frac{C}{|\vec{r_i} - \vec{r_j}|^6} \tag{3.21}$$

We can construct simply adversarial counterexamples, which make implementing the Hamiltonian a physical impossibility. For example, the J_{ij} may require atom pairs (1, 2) and (2,3) to be extremely close to each other, but require pair (1,3) to be extremely far from each other, causing a contradiction. This is precisely the key challenge in implementing arbitrary infinite-range QUBO or Ising problems and more generally is a fundamental challenge in building completely programmable quantum simulators (at least in the spin glass picture): how can we encode arbitrary interactions using only access to local interactions?

In literature, two major schemes have been proposed to resolve this: the Minor Embedding scheme [Cho08] and the Lechner-Zoller-Hauke scheme:

- Minor embedding scheme: One way to retain all information is by representing each variable with multiple atoms. This is possible if the atoms representing the same variable have the same spin at all times. Here, atoms representing the same variable are connected in chains via short-range interactions. As the size of the chain increases (as each variable is represented many times), they may break: all atoms representing the same variable may not have the same spin. Then, classical heuristics are required to decide the assignment of that variable. This scheme works well when there exists a fixed architecture and we can determine in polynomial time whether we can map an arbitrary Ising model problem onto it. But for fixed n: some Ising models may be mappable but some might not be. However, when both the architecture and the Ising model to be mapped are not fixed, it has been shown that finding a mapping is NP-hard [CMR14]. Only heuristic polynomial time algorithms are known to find mappings of arbitrary Ising models onto feasible architectures. The scheme requires at most n^2 atoms to encode n variables.

- Lechner-Hauke-Zoller scheme: The key idea behind the scheme is that physical atoms represent interactions between two variables in the classical setting and not the variables themselves. For an arbitrary Ising model with n variables, there is an efficient algorithm that tells us a Hamiltonian with only local interactions on $O(n^2)$

atoms. Another advantage of this scheme is that for fixed n, the same architecture can encode all possible Ising models with $\leq n$ variables by simply varying the local fields. Furthermore, it has been observed that this scheme has an an intrinsic fault tolerance property. Given these advantages for our setting, this will be scheme we will be building off and will be explained further in the next chapter.

4 | The Lechner-Hauke-Zoller Scheme: all-to-all connectivity from local interactions

Then, we review the Lechner-Hauke-Zoller (LHZ) scheme [LHZ15]. Our original work builds off the scheme and improves the scheme in making it implementable on 2D Rubidium atom arrays.

Encoding logical interactions as physical qubits: We will be considering the Ising model formulation for describing the Lechner-Hauke-Zoller scheme. The Lechner-Hauke-Zoller scheme works in a setting when arbitrary local fields h can be achieved while the interaction terms J are only local using one's quantum simulator. This is exactly the regime we are in. To explain the scheme, we begin with a Ising model with only interaction terms and we call this the logical Hamiltonian we want to implement:

$$H_{logical}(S) = \sum_{i<j} J_{ij} \left(S_i S_j\right) \tag{4.1}$$

As we explained before, the key idea is to think of a physical qubit representing an interaction and not a logical qubit. Notice that as per Equation 3.19, for every $0 \leq i < j < n$, J_{ij} is added to the energy term if $S_i S_j = 1$ that is $S_i = S_j = \pm 1$. Furthermore, $-J_{ij}$ is added to the energy if $S_i \neq S_j$. In the LHZ scheme, each physical qubit is labeled ij for $0 \leq i < j < n$ where n is the number of variables in the Ising model problem we are considering. Each physical qubit represents the product of the two classical variables $S_i S_j$. We identify $|ij\rangle = |g\rangle$ with $S_i S_j = -1$ or $S_i \neq S_j$ and $|ij\rangle = |r\rangle$ with $S_i S_j = 1$ or $S_i = S_j$. To get the desired effect, we simply map Equation 4.1 to the Hamiltonian:

$$H_{physical} = \sum_{i<j} J_{ij} \sigma^z_{ij} + \text{ constraints} \tag{4.2}$$

where σ^z_{ij} is the Pauli operator applied to qubit labelled ij. Note that here J_{ij} is physically playing the role of local field and not the interaction terms. As our Hamiltonian only consists of σ^z terms, its eigenstates are the eigenstates of the σ^z terms. Therefore, we can compute the energy of a configuration of atoms in this eigenbasis, by computing

the energy contribution of each term:

$$S_iS_j = 1 \implies |ij\rangle = |r\rangle \implies J_{ij}\sigma_{ij}^z |ij\rangle = J_{ij} |ij\rangle \tag{4.3}$$

$$S_iS_j = -1 \implies |ij\rangle = |g\rangle \implies J_{ij}\sigma_{ij}^z |ij\rangle = -J_{ij} |ij\rangle \tag{4.4}$$

which matches whether J_{ij} should be added or subtracted based on the value of S_iS_j as in the original Ising model energy formulation as in Equation 4.1. Therefore, if we ignore the constraint terms for now, we would be able to implement arbitrary Ising model Hamiltonians through simply changing the local field which we have precise control over. That is, for every classical configuration of Ising model variables, there exists a state of qubits in the LHZ setting which have the same energy (ignoring the constraint terms). Note that we require a physical qubit for every interaction, we will require $\frac{n(n-1)}{2}$ physical qubits to encode an Ising model problem with n variables.

Single plaquette constraints: Note that simply $\sum_{i<j} J_{ij}\sigma_{ij}^z$ is not the correctly mapped Hamiltonian. If the Hamiltonian is correctly mapped from the Ising model problem, we want that the ground state of the Hamiltonian to encode the solution to the Ising model problem. However, here, the ground state is trivial to determine:

- we set $|ij\rangle = |g\rangle$ if $J_{ij} > 0$
- we set $|ij\rangle = |r\rangle$ if $J_{ij} \leq 0$

In this way, for all $0 \leq i < j < n$, we simply add up $-|J_{ij}|$ and therefore, our in the ground state of $\sum_{i<j} J_{ij}\sigma_{ij}^z$. However, crucially, this may not be the ground state of the Ising model problem. To see this explicitly, consider the following example with 3 variables: $J_{01} = -1$, $J_{02} = -1$, $J_{12} = 1$. We know that the ground state of $\sum_{i<j} J_{ij}\sigma_{ij}^z$ will be -3 by setting $|01\rangle = |r\rangle$, $|02\rangle = |r\rangle$ and $|12\rangle = |g\rangle$. However, if we try to compute the resulting configuration of S_0, S_1 and S_2, we get the following conditions as per what states $|g\rangle$ and $|r\rangle$ mean in this setting:

$$S_0S_1 = 1 \tag{4.5}$$

$$S_0S_2 = 1 \tag{4.6}$$

$$S_1S_2 = -1 \tag{4.7}$$

Note that if we multiply the above equations, we get:

$$S_0^2S_1S_2 = 1 \tag{4.8}$$

but $S_0^2 = 1$ as $S_0 = \pm 1$ and we get:

$$\implies S_1S_2 = 1 \tag{4.9}$$

which is a contradiction. Therefore, we have seen that not all ground states of $\sum_{i<j} J_{ij}\sigma_{ij}^z$ will satisfy the internal logical structure of the Ising model we are trying to model. In some sense, the different physical qubits were not compatible which each other. Conversely, the actual solution to the Ising model will give the same energy as this Hamiltonian provided we translate from S to $|ij\rangle$ correctly, by construction. Therefore, we want to add constraint terms to the Hamiltonian such that we are searching only in the regime where the physical qubits are compatible.

To do the same, Lechner, Zoller and Hauke proposed placing the physical qubits in the following two-dimensional set up:

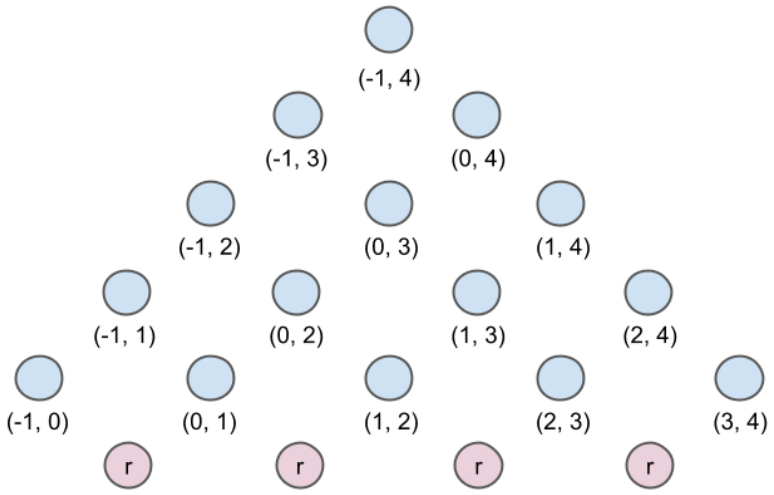

Figure 4.1: The LHZ scheme. Blue qubits represent product of classical spins; the state of the red qubits are fixed to $|r\rangle$.

The best way to analyze why such a setup would ensure the correct constraints, we must focus on an individual plaquette: four physical qubits arranged on the vertices of a square. First, let us see how the qubits are arranged. In the bottom row, we place qubits associated with $|01\rangle, |12\rangle, ..., |n-2|n-1\rangle$ in a straight line. Then, for each consecutive pair of qubits of the form: $|i|i+1\rangle$ and $|i+1|i+2\rangle$, we place a qubit representing $|i|i+2\rangle$ at the center of the pair but above their level by half the distance between them. We proceed with the same steps until we are left with a single qubit at the top of this triangle-type shape. Given this setup, we can see that the arrangement is based on these plaquettes. Every qubit (except the ones on the boundary) are associated with four plaquettes: one on top, one on bottom, one on the left and one on the right.

In the plaquette, we have four qubits representing the states $|ij\rangle, |ik\rangle, |lk\rangle$ and $|lj\rangle$. The key idea behind constraints in one plaquette is the following:

Theorem 4.1. A plaquette has even number of states in the $|g\rangle$ state if and only if the state of the four qubits is consistent with an possible assignment of variables in the Ising problem we are mapping.

Proof. We prove both directions:

- We want to prove that an even number of $|g\rangle$ states have consistent interpretation in the classical Ising model setup. There are three cases:
  - None of the qubits are in state $|g\rangle$: All of the states are in state $|r\rangle$ which means that:
    $$S_iS_j = S_iS_k = S_lS_k = S_lS_j = 1 \tag{4.10}$$
    which is true if $S_i = S_j = S_k = S_l = \pm 1$ and hence, consistent with the Ising model variables.
  - All of the qubits are in state $|g\rangle$:
    $$S_iS_j = S_iS_k = S_lS_k = S_lS_j = -1 \tag{4.11}$$
    which is true if $S_i = S_l = \pm 1$, $S_j = S_k = \mp 1$ and hence, consistent with the Ising model variables.
  - Two of qubits are in state $|g\rangle$ and two are in state $|r\rangle$. Here, we have two cases:
    * The two qubits that are in state $|g\rangle$ share a classical Ising model variable. Without loss of generality, let $|ij\rangle = |ik\rangle = |g\rangle$ and $|lj\rangle = |lk\rangle = |r\rangle$. We have that:
      $$S_iS_j = S_iS_k = -1 \tag{4.12}$$
      $$S_lS_j = S_lS_k = 1 \tag{4.13}$$
      which is true if $S_i = \pm 1$, $S_j = S_k = S_l = \mp 1$ and hence, consistent with the Ising model variables.
    * The two qubits that are in state $|g\rangle$ do not share a classical Ising model variable. Without loss of generality, let $|ij\rangle = |lk\rangle = |g\rangle$ and $|lj\rangle = |ik\rangle = |r\rangle$. We have that:
      $$S_iS_j = S_lS_k = -1 \tag{4.14}$$
      $$S_lS_j = S_iS_k = 1 \tag{4.15}$$
      which is true if $S_i = S_k = \pm 1$, $S_j = S_l = \mp 1$ and hence, consistent with the Ising model variables.

- All possible configurations of the classical spin variables in the Ising model lead to an even number of $|g\rangle$ states in a plaquette. Notice that in the first part of the proof we exhausted all possibilities of what the four classical Ising model variables can be. As all four variables i, j, k, l are symmetric (except when in pairs), without loss of generality:
  - $S_i = S_j = S_k = S_l = \pm 1 \implies |ij\rangle = |ik\rangle = |lj\rangle = |lk\rangle = |r\rangle$ which means none of the qubits are in the ground state (even).
  - $S_i = \pm 1, S_j = S_k = S_l \mp 1 \implies |ij\rangle = |ik\rangle = |g\rangle, |lj\rangle = |lk\rangle = |r\rangle$ which means two of the qubits are in the ground state (even).

  For the case when two variables are 1 and the other two are -1, we have two cases based on whether the two variables that are the same also share a state in the plauqette:

  - The two variables that are the same, share a state. Then, without loss of generality, let $S_i = S_j = \pm 1, S_k = S_l = \mp 1 \implies |ij\rangle = |lk\rangle = |r\rangle, |ik\rangle = |lj\rangle = |g\rangle$ which means that two of the qubits are in the ground state (even).
  - The two variables thare are the same, do not share a state. Then, without loss of generality, let $S_i = S_l = \pm 1, S_j = S_k = \mp 1 \implies |ik\rangle = |lj\rangle = |ik\rangle = |lk\rangle = |g\rangle$ which means all the four qubits are in the ground state (even).

Combing the two directions, we see that each plaquette must have even number of states to be in the $|g\rangle$ state for the four states to be consistent with the internal logical structure of the Ising problem we are mapping. □

Handling the last row and linear terms: However, yet we have not shown that this result holds globally. In order to do the same, we need to discuss some details:

- **Last row:** Notice that as per our construction, the qubits in the last row do not form a plaquette. Therefore, our result in Theorem 4.1 does not hold. We notice that they do form triangles of the form: $|ij\rangle, |jk\rangle, |ik\rangle$. In order to convert this into a plaquette, we notice that that the state of two of the qubits completely determines the state of the third: for example, if $|ij\rangle = |jk\rangle = |g\rangle$, this means that $S_iS_j = S_jS_k = -1$, we can multiple both equations and get that $S_is_k = 1$ as $S_j^2 = 1$ and therefore, $|ik\rangle = |r\rangle$. Taking into account all combinations as we did in Theorem 4.1, we note that either none of the qubits are in state $|g\rangle$ or two of them are. Therefore, to use the same plaquette structure for all other rows, we simply append a fourth qubit opposite the upper vertice of the triangle and fix its state to $|r\rangle$. Now, if we consider the plauqette, the "good" configurations are the ones with none of the qubits in state $|g\rangle$, two of the qubits in state $|g\rangle$ or (trivially as this can

never be the case), all of the qubits in state $|g\rangle$. Therefore, for all plaquettes now, we have to implement the same odd/even separation.

- **Linear terms:** Till now, we were only thinking of logical Ising model problems with energy of the form: $\sum_{i<j} J_{ij} S_i S_j$ till now. For completeness, if we also want to include the linear terms, we can create a dummy classical variable (let's call it S_{-1}) which is fixed to 1 (procedure to do this will be discussed next) and interacts with all other variables with interaction strength given by $J_{-1j} = H_j \forall j \in [n]$. As we fixed S_{-1} to be of value 1, now, $\forall j \in [n], \pm J_{-1j} = \pm h_j$ will be added the final energy term if the dynamics set $S_j = \pm 1$ and therefore, this interaction term with the dummy variable will act as the linear terms.

Global consistency proof: By Theorem 4.1, we showed that each plaquette must have even ground states if and only if the plaquette corresponds to a valid assignment of the Ising model variables. However, we were thinking of valid assignments among the four Ising model variables that were involved in the plaquette. Is it possible that all plaquettes have even number of ground states but the Ising model assignment for all n variables that the qubits correspond to is not consistent? We will show that this not the case.

Theorem 4.2. Every plaquette has even number of ground states if and only if they qubit states correspond to a valid assignment of all n variables in the Ising model.

Proof. Again, we must show both the sides:

- Every configuration of qubits such that each plaquette has an even number of ground state can be mapped onto a valid assignment of Ising model variables: To prove this, we provide an algorithm in order to find a suitable assignment for all variables given any set of states for the qubits provided that all plaquettes have an even number of ground states. We first set $S_{-1} = 1$ as per our discussion about linear terms above. Then, we begin with the bottom row and set $S_0 = 1$ if $|-10\rangle = |r\rangle$ and $S_0 = -1$ if $|-10\rangle = |g\rangle$ and so on:

$$S_i = \begin{cases} \pm 1 \text{ if } |i-1|i\rangle = |r\rangle \text{ and } S_{i-1} = \pm 1 \\ \mp 1 \text{ if } |i-1|i\rangle = |g\rangle \text{ and } S_{i-1} = \pm 1 \end{cases} \tag{4.16}$$

  This simply follows the meaning of the $|g\rangle$ and $|r\rangle$ state via Equations 4.3 and 4.4. Once we are done with the bottom row we would have assigned values to all n variables, $i \in [n], S_n$. This assignment is consistent with all qubits of the bottom row but we claim it is also consistent with all other qubits above the bottom row given that each plaquette has even number of ground states. We can show this by induction. Our claim is that the assignment we set above is consistent with qubits with all rows. Let us number the rows starting at 0 at the bottom.

– **Base case:** At row 0, as per our construction of the assignment, we know that the assignment is consistent with the qubit states in row 0.

– **Induction assumption:** Let the assignments be consistent with qubit states upto row i

– **Induction step:** For row $i+1$, we must show that the assignment is consistent with all qubits in this row. Notice that each qubit is at the top of a plaquette with states $|ij\rangle, |ik\rangle, |lk\rangle, |lj\rangle$ with $|ij\rangle$ be the state of the qubit at the top. Given the induction assumption, the assignment is consistent with all other qubits of that particular plaquette as they belong to row numbers $< i+1$. Now, we can simply show that in all cases, the assignments are consistent with each qubit of row $i+1$:

  * $|ij\rangle = |lk\rangle = |lj\rangle = |r\rangle$. We know that an even number of states are in the ground state in each plaquette so it must be that $|ik\rangle = |r\rangle$. As we know that the assignment is consistent with the three other states, we know that:

$$S_i S_j = S_l S_k = S_l S_j = 1 \tag{4.17}$$

$$\implies S_i S_l = 1 \implies S_i S_k = 1 \tag{4.18}$$

    where we used that $S_l^2 = S_j^2 = 1$. Note that $S_1 S_k = 1$ is consistent with $|ik\rangle = |r\rangle$ which we get due to the even parity constraint. We show similarly for all other possible configurations, in brief.

  * $|ij\rangle = |lk\rangle = |lj\rangle = |g\rangle$. Even parity of ground states $\rightarrow |ik\rangle = |g\rangle$. Assignment consistency with lower qubits implies:

$$S_i S_j = S_l S_k = S_l S_j = -1 \tag{4.19}$$

$$\implies S_i S_l = 1 \implies S_i S_k = -1 \tag{4.20}$$

    where we used that $S_l^2 = S_j^2 = 1$. $S_1 S_k = -1$ is consistent with $|ik\rangle = |g\rangle$.

  * $|ij\rangle = |lk\rangle = |g/r\rangle$ while $|lj\rangle = |r/g\rangle$. Even parity of ground states $\rightarrow |ik\rangle = |r/g\rangle$. Assignment consistency with lower qubits implies:

$$S_i S_j = S_l S_k = -S_l S_j = \pm 1 \tag{4.21}$$

$$\implies S_i S_l = -1 \implies S_i S_k = \mp 1 \tag{4.22}$$

    where we used that $S_l^2 = S_j^2 = 1$. $S_i S_k = \mp 1$ is consistent with $|ik\rangle = |r/g\rangle$.

  * As qubit ij and lk are symmetric with respect to qubit in consideration ik (they both share only one classical variable), we can without loss of

generality consider only one remaining case: $|ij\rangle = |lj\rangle = |g/r\rangle$ while $|lk\rangle = |r/g\rangle$. Even parity of ground states $\rightarrow |ik\rangle = |r/g\rangle$. Assignment consistency with lower qubits implies:

$$S_i S_j = S_l S_j = -S_l S_k = \pm 1 \tag{4.23}$$

$$\implies S_i S_l = 1 \implies S_i S_k = \mp 1 \tag{4.24}$$

where we used that $S_l^2 = S_j^2 = 1$. $S_i S_k = \mp 1$ is consistent with $|ik\rangle = |r/g\rangle$. Exactly the same analysis follows for $|lk\rangle = |lj\rangle = |g/r\rangle$ while $|ij\rangle = |r/g\rangle$.

Therefore, the assignment we constructed is consistent with row $i+1$. By induction, we see that every configuration of qubit states with even parity constraint on all plaquettes correspond to a valid assignment of Ising model variables.

- Every assignment of Ising model variables leads to an even number of ground states in every plaquette: This is the easier direction. For every plaquette associated with four variables S_i, S_j, S_k, S_l, we have shown in the second direction of Theorem 4.1, that there can be only an even number of ground states in the plaquette for every possible assignment of the four variables. Therefore, for every assignment of Ising model variables, each plaquette has even number of ground states.

□

[RBL16] provides an alternate proof to the same statement via the formalism of stabilizer codes.

Finding the Ising model variable assignments (read-out procedure): Now let us assume we are in the setting where we have evolved our many-body system such that now we are in the ground state of the physical Hamiltonian under the constraint that each plaquette has an even number of their qubits in $|g\rangle$ state. Given Theorem 4.2 and the form of our physical Hamiltonian, in this condition, for all possible states of qubits, there exists a configuration of Ising model variables such as (ignoring the constraint terms), the energy of the configuration and the states of qubits is the same. Now, to convert the final states of all qubits to the corresponding assignment of the Ising model, we follow the same procedure as described in the proof of Theorem 4.2:

- set $S_{-1} = 1$ as per our discussion the linear terms section
- Then, we begin with the bottom row and set $S_0 = 1$ if $|-10\rangle = |r\rangle$ and $S_0 = -1$ if $|-10\rangle = |g\rangle$ and so on $\forall i \in [n]$:

$$S_i = \begin{cases} \pm 1 \text{ if } |i-1|i\rangle = |r\rangle \text{ and } S_{i-1} = \pm 1 \\ \mp 1 \text{ if } |i-1|i\rangle = |g\rangle \text{ and } S_{i-1} = \pm 1 \end{cases} \tag{4.25}$$

as per the meaning we have set for a qubit to be in state $|g\rangle$ and $|r\rangle$.

Note that this procedure is both one-to-one and onto (given the even parity constraint) and therefore, we can also efficiently determine the states for all qubits in the physical Hamiltonian (using Equations 4.3 and 4.4) from a configuration of Ising model variables. What we mean by mapping an Ising model problem to a physically implementable Hamiltonian is exactly this: there is a polynomial time algorithm to go from the any state of the physical Hamiltonian to the the corresponding configuration of the Ising model instance and vice-versa. However, note that we only read n of the available $\frac{n(n+1)}{2}$ qubit (also considering the extra terms due to S_{-1}). To read out the classical assignments, we could have also simply traversed through the qubits labelled $\forall i \in [n], -1i$ and set $S_i = 1$ if $|-1i\rangle = |r\rangle$ and $S_i = -1$ otherwise. Given this redundancy in encoding the classical assignment of the lowest energy state of the Ising model, the LHZ has a intrinsic fault-tolerance and is therefore, resistant to bit flips in individual qubits. To take advantage of this property, we can decode the assignment in $O(n)$ different ways and take a majority vote. However, these might not be the primary errors in adiabatic quantum computation. [AVL16] compares the LHZ scheme and the minor embedding scheme with respect to fault tolerance in quantum adiabatic evolution.

Summary of the theoretical LHZ scheme: If we can implement arbitrary local fields on qubits in two dimensions, and ensure the even parity constraint in all plaquettes, we can map an arbitrary Ising model problem with n variables onto a two-dimensional quantum system with $\boxed{\frac{n(n+1)}{2} + n - 1 \in O(n^2)}$ qubits. The remaining challenge is to implement the even parity constraint in an experimentally feasible manner.

Implementation for even $|g\rangle$ states constraint: First, we overview the proposed techniques in [LHZ15] for ensuring the even parity constraint in all plaquettes. These techniques are infeasible for our current experimental setup.

Now that we are discussing physical implementations, what we mean by mapping an Ising model to a physical Hamiltonian changes:

- The energies between the valid states of our system and corresponding variable assignments may not exactly match but be shifted by some constant amount.
- There may be states that do not correspond to any consistent variable assignment (invalid states) but the energy of such states must be greater that the energy of all states with consistent variable assignments.

These two conditions are motivated by the adiabatic and variational algorithms which let us find ground states of physical systems given enough time. The two conditions would

ensure lowest energy state of the physical Hamiltonian corresponds to the lowest energy state of the Ising model instance. Furthermore, it would give a fixed energy regime, probing which would let us probe all possible Ising model variable assignments. These two conditions would also help us prove the computational complexity of Rydberg dynamics.

The key idea is to add a term to Hamiltonian for each plaquette such that for configurations with odd number of $|g\rangle$ states, a large and positive penalty is added to the Hamiltonian. The energy penalty must be large enough so that for every plaquette, there is a separation between the the energy manifold of the states with even and the states with odd number of $|g\rangle$ states. In effect, this would create an energy separation between the desirable configurations of the qubits in which all plaquettes satisfy the even parity constraint and all other configurations. Therefore, the total Hamiltonian looks like:

$$H_{physical} = \sum_{i<j} J_{ij}\sigma_{ij}^{z} + \sum_{\text{all plaquettes}} (\text{penalty for odd parity configuration}) \tag{4.26}$$

In the initial paper [LHZ15], two different methods of implementing the parity constraint were proposed. Let the four qubits of a single plaquette a belong to a set $Q_a = \{ij, ik, lk, lj\}$:

- **Four-body term:** The penalty term for each plaquette a looks like:

$$P_a = -C \prod_{m \in Q_a} \sigma_m^z = -C\sigma_{ij}^z\sigma_{ik}^z\sigma_{lk}^z\sigma_{lj}^z \tag{4.27}$$

$$= \begin{cases} -C \text{ if number of } |g\rangle \text{ states is even} \\ C \text{ if number of } |g\rangle \text{ states is odd} \end{cases} \tag{4.28}$$

  where C is a positive constant and we used that for a qubit in state $|g\rangle$, σ^z returns -1 while it returns 1 for a qubit in state $|r\rangle$. Therefore, if number of $|g\rangle$ states is even then, the resulting -1 couple up, and we end up with $-C \times 1$, and otherwise, they don't and we end up with $-C \times -1 = C$. Each plaquette either contributes C or $-C$ to the total energy. Therefore, the energy of all valid configurations of qubit states is given by:

$$\begin{aligned} H_{physical,valid} &= \sum_{i<j} J_{ij}\sigma_{ij}^z - C \times (\text{number of plaquettes}) \\ &= \sum_{i<j} J_{ij}\sigma_{ij}^z - C\left(\frac{n(n+1)}{2} - n\right) \end{aligned} \tag{4.29}$$

  Number of plaquette can be computed by noting at each qubit except the qubits on the last row (n qubits) and therefore, number of plaquettes is $\frac{n(n+1)}{2} - n$ where n is the number of variables in the Ising model instance. Therefore, we could adapt the readout procedure and add $C\left(\frac{n(n+1)}{2} - n\right)$ to the energy of the physical

Hamiltonian to find the corresponding Ising model energy. Note that the maximum this value can be is:

$$\max(H_{physical,valid}) = \sum_{i<j} |J_{ij}| - C\left(\frac{n(n+1)}{2} - n\right) \tag{4.30}$$

The energy for the invalid states would be given by:

$$\begin{aligned} H_{physical,invalid} = \sum_{i<j} J_{ij}\sigma_{ij}^z - C \times (\text{number of even plaquettes}) \\ + C \times (\text{ number of odd plaquettes}) \end{aligned} \tag{4.31}$$

As for invalid states, there must be at least one plaquette with an odd number of $|g\rangle$ states, therefore, the minimum this value can be is given by:

$$\min(H_{physical,invalid}) = \sum_{i<j} -|J_{ij}| - C\left(\frac{n(n+1)}{2} - n - 1\right) + C \tag{4.32}$$

$$\implies \min(H_{physical,invalid}) - \max(H_{physical,valid}) = \sum_{i<j} -2|J_{ij}| + 2C \tag{4.33}$$

If we let $C > \sum_{i<j} |J_{ij}|$, then we ensure that $\min(H_{physical,invalid}) - \max(H_{physical,valid}) > 0$ and therefore, satisfy the second condition of our mapping.

The key challenge in implementing this is that current atomic quantum simulators do not have native three- or four-body term implementations given how weak these interactions are.

- **Ancillary qutrits:** The penalty term for each plaquette a looks like:

$$P_a = C\left(\sum_{m \in Q_a} \sigma_m^z - S_a\right)^2 \tag{4.34}$$

where C is a positive constant and every plaquette a is associated with an ancillary qutrit which has an operator S_a that takes on value from the set $\{-4, 0, 4\}$. Note that if the number of $|g\rangle$ states is even, then the sum of eigenvalues of the σ^z operator on the four qubits can only be $-4, 0$ or 4. Therefore, if the state of qubits in a plaquette is valid, then, there is a choice of the ancillary qutrit, such that the $P_a = 0$. Therefore, the energy of the physical system with a valid Ising model variable assignment and correct orientation of all qutrits, would be given by:

$$H_{physical,valid} = \sum_{i<j} J_{ij}\sigma_{ij}^z \tag{4.35}$$

$$\implies \max(H_{physical,valid}) = \sum_{i<j} |J_{ij}| \tag{4.36}$$

In this setting, the energy of a valid state is the same as the energy of the corresponding variable assignment. Note that if the number of $|g\rangle$ states is odd, then the sum of eigenvalues of the σ^z operator on the four qubits can only be -2 or 2. Therefore, the penalty term would at least add $4C$ in this case. In the case where a plaquette is valid but the ancillary qutrit's state does not match the sum, we get an addition of at least $16C$. We group both these kinds of states as invalid states and note that the minimum this energy can be is given by:

$$\min(H_{physical,invalid}) = \sum_{i<j} -|J_{ij}| + 4C \tag{4.37}$$

which corresponds to the case where a single plaquette has an odd number of $|g\rangle$ states.

$$\min(H_{physical,invalid}) - \max(H_{physical,valid}) = \sum_{i<j} -2|J_{ij}| + 4C \tag{4.38}$$

Therefore, if we let $C > \sum_{i<j} \frac{|J_{ij}|}{2}$, then we ensure that the minimum energy of an invalid state is more that the maximum energy of a valid state and therefore, satisfy the second condition of the mapping.

The challenge in implementing this way to constrain the parity of each plaquette is that it adds experimental complexity by requiring the coherent control of a three-level system in addition to two-level systems.

5 | Feasible implementation with qubits: resolving the double-counting issue

In the previous chapter, we introduced the theoretical LHZ scheme that lets us map arbitrary Ising model on n variables instances to the dynamics of a Hamiltonian over $O(n^2)$ atoms and provided a new proof regarding its validity. Then, we defined the conditions of mapping the Ising model onto a physical system and overviewed two proposals to physically implement the LHZ scheme but saw that both of them added experimentally complexity: neither can be implemented using just Rubidium atom arrays in two dimensions. We now describe our new scheme that enables exactly the mapping onto 2D Rubidium atom arrays with the $O(n^2)$ scaling.

5.1 Ancillary qubits

In the second proposal of implementing the LHZ scheme we discussed in the above chapter, we required ancillary qutrits to be associated with every plaquette: these qutrits would add penalty terms whenever the number of $|g\rangle$ states was not even. The reason we required qutrits and not qubits was precisely because the sum of σ^z operators of all qubits in a plaquette could be three different values: $-4, 0$ or 4 when number of $|g\rangle$ states are even. Now, if we were able to change the condition to force the number of $|g\rangle$ states to be odd, then, the sum of σ^z operators can only be two different values: -2 or 2 and hence, we shall only require an ancillary qubit S_a that takes value from set $\{-2, 2\}$ for some operator $(2\sigma^z)$ and implement the penalty term for every plaquette:

$$P_a = C\left(\sum_{m\in Q_a} \sigma_m^z - 2\sigma_{S_a}^z\right)^2 \tag{5.1}$$

where C is a positive constant. Analogously to our analysis in the qutrit setting, note that if the number of $|g\rangle$ states is odd, there is a choice of the ancillary qutrit, such that the $P_a = 0$. Therefore, the energy of the physical system with a valid Ising model variable

assignment and correct orientation of all ancillary qubits, would be given by:

$$H_{physical,valid} = \sum_{i<j} J_{ij}\sigma^z_{ij} \tag{5.2}$$

$$\implies \max(H_{physical,valid}) = \sum_{i<j} |J_{ij}| \tag{5.3}$$

Note that if the number of $|g\rangle$ states is even, then the sum of eigenvalues of the σ^z operator on the four qubits can only be -4, 0, or 4. Therefore, the penalty term would at least add $4C$ in this case. In the case where a plaquette is valid but the ancillary qutrit's state does not match the sum, we get an addition of at least $16C$. We group both these kinds of states as invalid states and note that the minimum this energy can be is given by:

$$\min(H_{physical,invalid}) = \sum_{i<j} -|J_{ij}| + 4C \tag{5.4}$$

which corresponds to the case where a single plaquette has an even number of $|g\rangle$ states.

$$\min(H_{physical,invalid}) - \max(H_{physical,valid}) = \sum_{i<j} -2|J_{ij}| + 4C \tag{5.5}$$

Therefore, if we let $C > \sum_{i<j} \frac{|J_{ij}|}{2}$, then we ensure that the minimum energy of an invalid state is more that the maximum energy of a valid state and therefore, satisfy the second condition of the mapping. We can view the ancillary qubit "correcting" the energy of the system due to some constraints, therefore, the LHZ scheme lends itself to a stabilizer formalism. [RBL16] analyzed the scheme via such techniques and their analysis found that an equivalent formulation with the odd parity constraint exists as well. Following their general approach but without invoking stabilizer theory, we provide a proof of the validity of the odd-parity scheme. Our construction differs from [RBL16] but the key idea remains the same: we want to flip the meaning of qubit ij from semantically being connected to the product (S_iS_j) to the negative of the product $(-S_iS_j)$ and we want to do this for exactly one qubit for every plaquette. First, we shall show that this change would change the number of $|g\rangle$ required from even to odd. Then, we show that choosing k qubits such that for every plaquette exactly 1 qubit is chosen is possible (this is where our construction varies from [RBL16]). Finally, in the next chapter, we will describe how to enact this experimentally.

Even to odd parity constraint:

Definition 5.1 (Odd-parity LHZ scheme)**.** We translate the Ising spin model exactly as described as above for every plaquette, exactly one qubit ij now identifies with the variable product $-S_iS_j$ and not S_iS_j as before.

Theorem 5.1. In the odd-parity LHZ scheme, every plaquette has odd number of ground states if and only if they qubit states correspond to a valid assignment of all n variables in the Ising model.

Proof. We prove both directions are corollaries of Theorem 4.2.

- Let all plaquettes of the odd-parity LHZ scheme have an odd number of $|g\rangle$ states. For a plaquette, let the qubits be labelled ij_{odd}, ik_{odd}, lk_{odd}, and lj_{odd}. Without loss of generality, let the qubit that was flipped (to come up with the odd-parity LHZ scheme) in this plaquette be qubit ij_{odd}. If $|ij_{odd}\rangle = |g\rangle \implies S_iS_j = 1$, this implies that in the usual (even-parity) LHZ scheme, we have that $|ij_{even}\rangle = |r\rangle$. And if $|ij_{odd}\rangle = |r\rangle \implies S_iS_j = -1$, this implies that in the usual (even-parity) LHZ scheme, we have that $|ij_{even}\rangle = |g\rangle$. As only qubit is flipped, for all other qubits x, we have that $|x_{even}\rangle = |x_{odd}\rangle$. As we either reduced or increased the number of $|g\rangle$ states by 1, the same plaquette in the even-LHZ scheme has an even number of $|g\rangle$ states now. We can do the same for all plaquettes. Therefore, odd-parity LHZ scheme with odd-parity constraint on all plaquettes is equivalent to even-parity LHZ scheme with even-parity constraint on all plaquettes. By Theorem 5.3, we know there is a valid variable assignment for the latter and hence, there is a valid variable assignment for the former.

- Let there be a valid variable assignment of an Ising model. From Theorem 4.2, we can convert this to the usual LHZ scheme such that all plaquettes have even number of $|g\rangle$ states. We can convert to a odd-parity LHZ scheme via definition by simply flipping the meaning of one qubit in each plaquette. For each plaquette, let us flip one qubit and call it $|ij\rangle$ without loss of generality. If $|ij_{even}\rangle = |g\rangle \implies S_iS_j = -1$, this implies that we have that $|ij_{odd}\rangle = |g\rangle$. And if $|ij_{even}\rangle = |r\rangle \implies S_iS_j = 1$, this implies that $|ij_{odd}\rangle = |r\rangle$. As only qubit is flipped, for all other qubits x, we have that $|x_{even}\rangle = |x_{odd}\rangle$. As we either reduced or increased the number of $|g\rangle$ states by 1, the same plaquette in the odd-LHZ scheme has an odd number of $|g\rangle$ states now.

□

Existence of flipping procedure:

Theorem 5.2. There is set of qubits that we can select such that for every plaquette, the number of qubits selected is exactly one.

Proof. To prove this, we provide a simple procedure to do the same. We label rows starting with 1 at the top and each qubit in every row with 1 at the left-most qubit of a particular row. Now, we select all qubits that are in the odd rows and at an odd number in their row. For example, in row 1, we select the only qubit, in row 2, we select no

qubits and in row 3, we select qubit labelled 1 and 3. Now, we have to prove that every plaquette, exactly one qubit is selected via this scheme. Consider a plaquette with top, left, right and bottom qubits. There are two cases:

- The top qubit is in a row $2n, n \in \mathbb{N}$. If this is the case, then we know that both the top and bottom qubits are not selected as they fall in even rows. Now the left and right qubits are in an odd row $2n+1, n \in \mathbb{N}$ and within the row, they are consecutive. Therefore, exactly one of them is odd and is therefore, selected in our scheme while the other qubit is not.

- The top qubit is in a row $2n+1, n \in \mathbb{N}$. The left and right qubit are in an even row and hence, not selected. The top and bottom qubit have the same vertical position. We note that the number of qubits in each row is one more than the number in the previous row. Therefore, number of qubits in the row with bottom qubit is two more than the number of qubits in the row with the top qubit. As per our LHZ construction, these extra qubits fall on either side of the bottom qubit. Therefore, the bottom qubit has one extra qubit on its left compared to the top qubit. Therefore, if the label of the top qubit in its row is $X \in \mathbb{N}$, then the label of the bottom qubit is $X+1$. Either X or $X+1$ is odd and both cannot be odd. Therefore, we select the one that is odd.

In both these cases we see that only one qubit is selected in each plaquette as per our selection scheme. □

In the following image, we color the selected set green for $n = 5$:

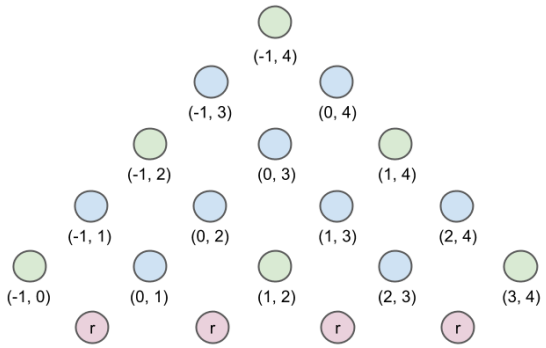

Figure 5.1: The odd-parity LHZ scheme for $n = 5$. Green qubits are selected for the flipping set. Note that for every plaquette, exactly 1 qubit is selected.

Experimental implementation of qubits selected to be negated: If we are able to negate the meaning of qubit ij selected in the fashion described in the proof above

to represent the product $-S_iS_j$ instead of S_iS_j, we would have to implement the ancillary qubit-based penalty term as now we want odd number of qubits in $|g\rangle$ state in all plaquettes. For these selected qubits we require that:

$$|ij\rangle = |g\rangle \implies -S_iS_j = -1 \implies J_{ij} \text{ added to the energy} \tag{5.6}$$

$$|ij\rangle = |r\rangle \implies -S_iS_j = 1 \implies -J_{ij} \text{ added to the energy} \tag{5.7}$$

We can attain this behaviour by simply negating the local field associated with qubit ij:

$$S_iS_j = 1 \implies |ij\rangle = |g\rangle \implies -J_{ij}\sigma^z_{ij}|ij\rangle = J_{ij}|ij\rangle \tag{5.8}$$

$$S_iS_j = -1 \implies |ij\rangle = |r\rangle \implies -J_{ij}\sigma^z_{ij}|ij\rangle = -J_{ij}|ij\rangle \tag{5.9}$$

Therefore, we can define the matrix $J^{physical}$ from the original Ising model J matrix such that for qubit ij if it is selected in our set to be flipped, we get $J^{physical}_{ij} = -J_{ij}$ and if not, $J^{physical}_{ij} = J_{ij}$. And the final Hamiltonian becomes:

$$H_{physical} = \sum_{i<j} J^{physical}_{ij}\sigma^z_{ij} + \sum_a C\left(\sum_{m\in Q_a}\sigma^z_m - 2\sigma^z_{S_a}\right)^2 \tag{5.10}$$

where a runs over all plaquettes, Q_a is the set of physical qubits in plaquette a and S_a is the ancillary qubit of plaquette a.

Implementation of ancillary qubit in a single plaquette: We have assumed that we can implement arbitary local fields on each qubit. The challenge remains to see whether $\sum_a C\left(\sum_{m\in Q_a}\sigma^z_m - 2\sigma^z_{S_a}\right)^2$ can be implemented. [Gla+17] showed that the exact form $C\left(\sum_{m\in Q_a}\sigma^z_m - 2\sigma^z_{S_a}\right)^2$ is not crucial. Here, we reproduce their analysis which is crucial to understanding the double-counting error. They considered a more general formulation: let the four physical qubits of a plaquette belong to one species of atoms and let the associated ancillary qubit belong to another species of atoms. As all four qubits are symmetric, we must place the ancillary at a symmetric location. However, only one such location exists: the center of the plaquette. Let the side of the plaquette square be s. If we consider only binary interactions, three types exist (we are using the same notation as [Gla+17]):

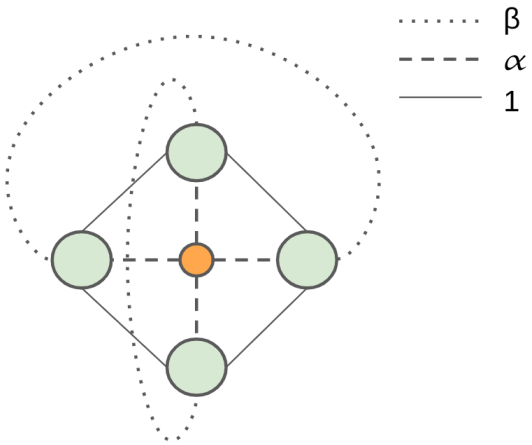

Figure 5.2: Plaquette interactions

- Four physical-physical interactions over s distances. These are between the physical qubits that share sides of the plaquette. We normalize all interactions such that this interaction is 1.
- Two physical-physical interactions over $\sqrt{2}s$ distance. These are between qubits that are across the diagonal. We define this to be β times the first interaction.
- Four physical-ancillary interactions over $\frac{1}{\sqrt{2}}s$ distance. These are between each physical qubit and the ancillary qubit. We define this to be α times the first interaction.

Therefore, the interaction Hamiltonian of the plaquette (ignoring the local fields) will be given by:

$$P_a = \left(\sum_{m,n \in Q_{a,edges}} \sigma_m^z \sigma_n^z \right) + \left(\beta \sum_{m,n \in Q_{a,diag}} \sigma_m^z \sigma_n^z \right) + \left(\alpha \sum_{m \in Q_a} \sigma_m^z \sigma_{S_a}^z \right) \tag{5.11}$$

Note that if we had expanded equation 5.10, we would have found the same form with $\beta = 1$ and $\alpha = 2$ and some constant:

$$\left(\sum_{m \in Q_a} \sigma_m^z - 2\sigma_{S_a}^z \right)^2 \propto \left(\sum_{m,n \in Q_{a,edges}} \sigma_m^z \sigma_n^z \right) + \left(\sum_{m,n \in Q_{a,diag}} \sigma_m^z \sigma_n^z \right) + \left(2 \sum_{m \in Q_a} \sigma_m^z \sigma_{S_a}^z \right) + 4 \tag{5.12}$$

Therefore, in fact, we are considering a much more general setting of possibilities for our penalty term. Using the second condition for our mapping, we must have that all invalid states of the plaquette have more energy than the valid plaquettes. Moreover, as the shift of energy due to the penalty terms must be a constant, all possible valid states must be

degenerate, i.e., have the same energy. If the odd-parity constraint is met then either 1 qubit is in state $|g\rangle$ and 3 are in state $|r\rangle$ or 1 qubit is in state $|r\rangle$ and 3 are in state $|g\rangle$. In this general setting, we will define valid states to be those where the odd-parity constraint is met and the ancillary qubit has the same state that only 1 of the physical qubits has. The reasons for the same will become apparent after the following analysis. Let us consider all possible states:

- **Valid states**: Note that in these states, we have one qubit in state $|g/r\rangle$ and then three qubits in the other spot with state $|r/g\rangle$. Therefore, two interactions on the side for qubits m and n have eigenvalue -1 for $\sigma_m^z\sigma_n^z$ operator while the other two have eigenvalue 1 and therefore, these cancel. Similarly, both diagonal operators have opposite sign as for one diagonal both states are the same while for the other, they are different and these cancel. We are only left with the interaction with the ancilla. As the ancilla is also in state $|g/r\rangle$ (condition of validity), we get that the total sum of the penalty energy is $\alpha - 3\alpha = -2\alpha$. Therefore, we have satisfied the condition that all valid states are degenerate.
- **Invalid states**: There are three possible cases:
  - Odd-parity constraint but wrong ancilla orientation: in this cases, the side and diagonal terms cancel as above but the ancillary interactions is the opposite and we get total energy $-\alpha + 3\alpha = 2\alpha$. For valid states to have lower energy that these states, we get the constraint that $\boxed{\alpha > 0}$.
  - Two are in state $|g/r\rangle$ while two are in state $|r/g\rangle$. In this case, the ancillary interactions cancel as no matter what state of the ancillary qubit it interacts with two $|g\rangle$ state and two $|r\rangle$ states. If the two states that are the same are on the diagonal, all side terms result in a -1 and both diagonal terms result in a $+\beta$. Hence, the total energy is $-4 + 2\beta$. The condition we want to satisfy is that $\boxed{-4 + 2\beta > -2\alpha}$. Now, conversely, if the two states that are the same share a side, then there are two positive side terms and two negative side terms and hence, they cancel. Both diagonal operators have -1 eigenvalue and hence, the total energy is given by: -2β and hence, the condition becomes $-2\beta > -2\alpha$ which is true if $\boxed{\alpha > \beta}$.
  - The last kind of invalid states are those in which all four physical qubits are in the same state. No matter what state, they are in the side terms and diagonal term operator would always have eigenvalue 1 and hence, we get $4 + 2\beta$. Now, either the ancillary qubit can have the same state as the physical qubits or not; therefore, either -4α or 4α is added. Therefore, the condition is that $\boxed{4 + 2\beta \pm 4\alpha > -2\alpha}$.

Therefore, if we find implementations of physical qubits and ancillary qubits such that interaction strengths of β and α can be manifested such that:

$$\alpha > 0 \tag{5.13}$$

$$\alpha > \beta \tag{5.14}$$

$$-4 + 2\beta > -2\alpha \tag{5.15}$$

$$4 + 2\beta \pm 4\alpha > -2\alpha \tag{5.16}$$

are satisfied then we would satisfy the conditions of the mapping for a single plaquette. Note that in this setting there is no C parameter we can tune to ensure that the separation exists with arbitrary Ising model instance. Conversely, we scale down the J matrix in order to ensure the separation. Note that that we perform a constant scaling on the J matrix and h array, we do not change the problem: to get the energy of the original problem for a particular configuration, we simply multiple the inverse of scaling to the energy found for the modified problem. We note that if all plaquettes are valid, then, each plaquette contributes -2α to the total energy. Hence, the maximum the total energy can be is:

$$\max(H_{physical,valid}) = c\sum_{i<j} |J_{i,j}^{physical}| - 2\alpha\left(\frac{n(n+1)}{2} - n\right) \tag{5.17}$$

as $\left(\frac{n(n+1)}{2} - n\right)$ is the number of plaquettes and where c is the scaling factor. Therefore, for a valid state, to get the original energy of the corresponding variable assignment of the Ising model, we simply add $2\alpha\left(\frac{n(n+1)}{2} - n\right)$ and multiply by $\frac{1}{c}$. For the invalid states, we get a penalty of 2α, -2β, $-4+2\beta$ or $4+2\beta\pm4\alpha$. If even a single plaquette is wrong, then the energy is increased and the minimum the energy can be is given by:

$$\begin{aligned}\min(H_{physical,invalid}) = &- c\sum_{i<j} |J_{i,j}^{physical}| - 2\alpha\left(\frac{n(n+1)}{2} - n - 1\right) \\ &+ \min(2\alpha, -2\beta, -4+2\beta, 4+2\beta\pm4\alpha)\end{aligned} \tag{5.18}$$

which we can compute once we know what β and α is set to. We get that the difference:

$$\begin{aligned}\min(H_{physical,invalid}) - \max(H_{physical,valid}) = &- 2c\sum_{i<j} |J_{i,j}^{physical}| + 2\alpha \\ &+ \min(2\alpha, -2\beta, -4+2\beta, 4+2\beta\pm4\alpha)\end{aligned} \tag{5.19}$$

and we can set c such that this number is positive. Using a particular value of $0 < \beta < 1$ (physical reason for this is that two atoms of the same species that are further away would interact weakly compared to the same atoms which are closer), [Gla+17] plotted the following instructive diagram: Through this diagram, they concluded that $1.1 < \alpha < 2.5$

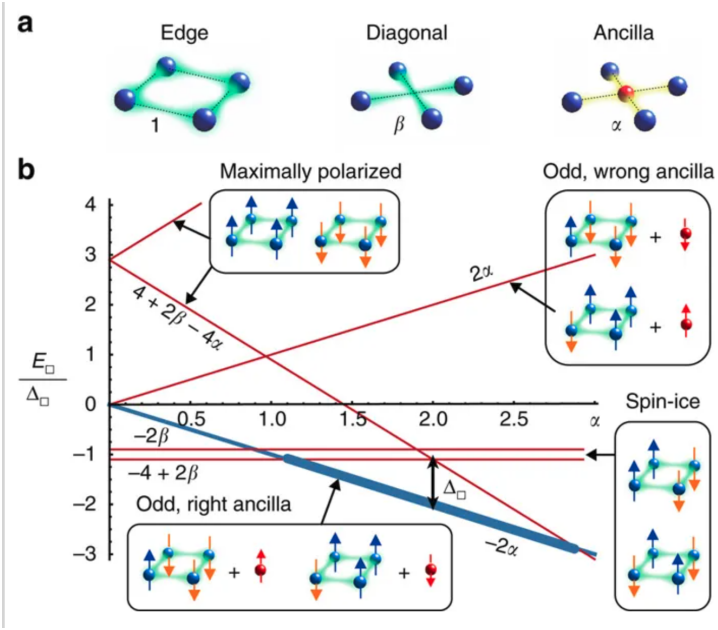

Figure 5.3: Interaction parameter search from [Gla+17]

would suffice for the mapping and presented the hyperfine states of ^{87}Rb and ^{133}Cs as candidate qubits for the physical and ancilla qubits respectively giving them $\alpha = 2$ and $\beta = 1$ at certain specific distance scales. However, we will see in the next section that this analysis is incomplete and these conditions and the current scheme alone does not suffice a successful mapping.

5.2 The double-counting error and its resolution

Going beyond a single plaquette: The key error in the above analysis is that it is limited to one plaquette. For a single plaquette, the conditions (5.13-5.16) along with the plaquette structure with an ancillary qubit at the center of the plaquette do lead to a separation in the valid (odd-parity constraint, ancilla is in state that only one physical qubit has) and invalid (all other) states. However, once we begin combining plaquettes together to form the odd-parity LHZ scheme, the separation between states in which all plaquettes are valid and at least one plaquette is invalid is not maintained. This is because some interactions (sides) of plaquettes are shared between multiple plaquettes. The penalty term for a single qubit looks like:

$$P_a = \left(\sum_{m,n \in Q_{a,edges}} \sigma_m^z \sigma_n^z \right) + \left(\beta \sum_{m,n \in Q_{a,diag}} \sigma_m^z \sigma_n^z \right) + \left(\alpha \sum_{m \in Q_a} \sigma_m^z \sigma_{S_a}^z \right) \tag{5.20}$$

with conditions as in (5.13-5.16). However, for our scheme to work our proof relies on the fact that this penalty term is applied to every plaquette and the total Hamiltonian

is:

$$H_{physical} = \sum_{i<j} J_{ij}^{physical} \sigma_{ij}^{z} + \sum_{a} \left(\left(\sum_{m,n \in Q_{a,edges}} \sigma_m^z \sigma_n^z \right) + \left(\beta \sum_{m,n \in Q_{a,diag}} \sigma_m^z \sigma_n^z \right) + \left(\alpha \sum_{m \in Q_a} \sigma_m^z \sigma_{S_a}^z \right) \right) \tag{5.21}$$

Therefore, if an interaction term is in two plaquettes (the plaquettes are side by side and share an edge), then, for our mapping to work $H_{physical}$ counts this interaction twice. However, in the implementation in [Gla+17] and discussed in the previous section, the side interactions will be determined simply by the interaction of two ^{87}Rb atoms at distance s and won't depend on whether the interaction is being shared by two different plaquettes. We call this the *double-counting* error: theoretically, shared interactions must be double-counted but in implementation, they are counted only once.

To see the error in practice, consider two plaquettes that share only one edge and set α, β which satisfies conditions (5.13-5.16):

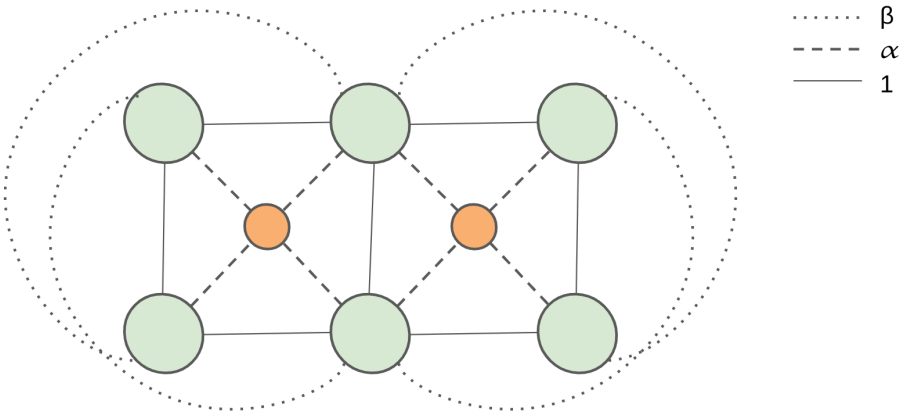

Figure 5.4: The double-counting error due to shared edges

Now, the penalty energy of the system is given by (correctly):

$$(P_a + P_b)_{logical} = \left(\left(\sum_{m,n \in Q_{a,edges}} \sigma_m^z \sigma_n^z \right) + \left(\beta \sum_{m,n \in Q_{a,diag}} \sigma_m^z \sigma_n^z \right) + \left(\alpha \sum_{m \in Q_a} \sigma_m^z \sigma_{S_a}^z \right) \right) + \left(\left(\sum_{m,n \in Q_{b,edges}} \sigma_m^z \sigma_n^z \right) + \left(\beta \sum_{m,n \in Q_{b,diag}} \sigma_m^z \sigma_n^z \right) + \left(\alpha \sum_{m \in Q_b} \sigma_m^z \sigma_{S_b}^z \right) \right) \tag{5.22}$$

If we are in the state where both plaquettes have an odd number of $|g\rangle$ states then based on the last section, each plaquette gets a penalty of -2α and therefore,

$$(P_a + P_b)_{logical} = -4\alpha \tag{5.23}$$

no matter what exact valid state each plaquette is in. Let the middle edge consist of qubits i and j. In the implemented version, this edge is counted only one but it should have been counted twice as in the logical version. Therefore, the penalty in the implemented version is given by:

$$(P_a + P_b)_{implemented} = (P_a + P_b)_{logic} - \sigma_i^z \sigma_j^z \tag{5.24}$$

In the case when both plaquettes are still in valid states, we get that:

$$(P_a + P_b)_{implemented} = -4\alpha - \sigma_i^z \sigma_j^z \tag{5.25}$$

However, note that now the energy depends on the eigenvalue of the operator $\sigma_i^z \sigma_j^z$. There are two cases:

- There exist valid states in which $|i\rangle = |j\rangle = |g/r\rangle$ where one of the other qubits in either plaquette is in state $|r/g\rangle$. In this case, the eigenvalue of $\sigma_i^z \sigma_j^z$ is 1.
- But there also exist valid states in which $|i\rangle = |g/r\rangle$ but $|j\rangle = |r/g\rangle$ with the remaining states in both plaquettes being the same. In this case, the eigenvalue of $\sigma_i^z \sigma_j^z$ is -1.

Therefore, in the implemented version, $(P_a + P_b)_{implemented}$ can be different for two valid states which ensures that mapping is not possible as the energy shift from the original energy scale depends on the configuration. This condition is critical for algorithmic reasons, as the adiabatic algorithm may find incorrect ground states due to this discrepancy. In other words, some valid states will be preferred more than other valid states (without looking at the J matrix). Therefore, it is critical to resolve this error. Note that for this section, we will be neglecting long-range interactions (interactions beyond one plaquette) and assuming them to be very small compared to 1, α and therefore, can be safely ignored. Further justification will be provided in Chapter 6.

Resolving the double-counting error: The key observation in resolving this issue is that all interactions except the boundary interactions must be double-counted. That is precisely the definition of a non-boundary interaction: an interaction that has a plaquette on both sides while boundary interactions have plaquettes only on one side. Therefore, we want that the $ZZ-$interaction between two physical qubits that form a non-boundary interaction be 2 (in the units discussed in the previous section), while the $ZZ-$interaction between physical qubits on the boundary term to be 1 (in the same units). Note that

this double-counting only happens for side terms and never for diagonal terms as they are exclusive to a plaquette.

The procedure works as follows: let β, α be parameters such that inequalities are satisfied such that we have the valid-invalid separation in a single qubit. Now, implement qubits in such a way such that the physical-physical $ZZ-$interaction over distance s has magnitude 2 (instead of 1 in the previous section), the physical-physical $ZZ-$interaction at distance $\sqrt{2}s$ be β while the $ZZ-$interaction between a physical and ancillary qubit at $\frac{1}{\sqrt{2}}s$ be α. As per our discussion above, if the physical-physical $ZZ-$interaction at the side of our plaquette is a non-boundary interaction, then the magnitude 2 interaction can "split up" between the penalty terms of the two plaquettes the interaction is a part of. The problem now exists in boundary $ZZ-$interactions, these have twice the magnitude that they should have if we want our mapping scheme to be successful.

We device a "gadget" of sorts to neglect half of the interaction for every boundary term. The construction works as follows: for every boundary side, associate another ancillary qubit (let's call it gadget qubit) such that the $ZZ-$interaction between this qubit and the two qubits of the boundary term is 1 (in the units as above). This can be practically done by placing a qubit outside the LHZ boundary, on the line that is equidistant from both physical qubits of the boundary term:

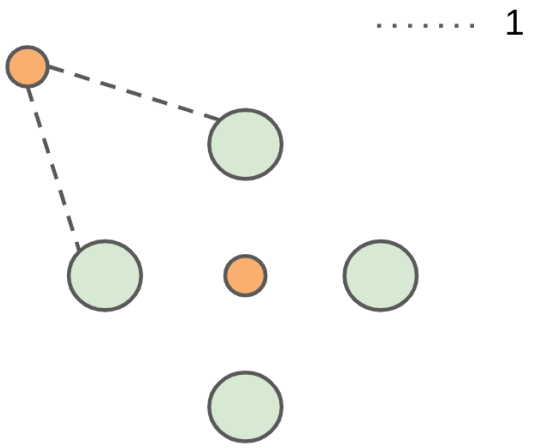

Figure 5.5: Gadget qubit to resolve double-counting error for top and left qubit

The species of the gadget qubit, interaction strength with the physical qubit dictates how far this qubit must be placed (note that we are under the assumption that the gadget qubit does not interact strongly with other gadget qubits and we will see how this may be implemented with Rubidium atoms in Chapter 6). Now we let the validity of states

is defined by all three:

- Number of $|g\rangle$ states in each plaquette must be odd
- The ancillary qubit's state for each plaquette must be the same as the state that appears only once in the physical qubits of the plaquette
- The state of the gadget qubit for each boundary term must be opposite to the state of at least one of the two qubits in the boundary term. That is, if i, j represent the qubits forming the boundary term, and x is the associated gadget qubit then:

$$|i\rangle = |j\rangle = |r/g\rangle \implies |x\rangle = |g/r\rangle \tag{5.26}$$

$$|i\rangle \neq |j\rangle = |r/g\rangle \implies |x\rangle = |g/r\rangle \text{ or } |r/g\rangle \tag{5.27}$$

To see why this condition works, we must analyze it for a single boundary term first. A boundary term has magnitude 2. Half of this magnitude is bracketed with the plaquette penalty term. However, half of it remains unengaged yet. Let o, p represent the qubits forming the boundary term, and x is the associated gadget qubit. Therefore, without the gadget, the penalty Hamiltonian of a plaquette with one boundary term looks like (let us assume that for non-boundary terms, we have divided the magnitude 2 into its two plaquettes):

$$(P_a)_{implement,no\ gadget} = \left(\left(\sum_{m,n\in Q_{a,edges}} \sigma_m^z \sigma_n^z\right) + \left(\beta \sum_{m,n\in Q_{a,diag}} \sigma_m^z \sigma_n^z\right) + \left(\alpha \sum_{m\in Q_a} \sigma_m^z \sigma_{S_a}^z\right)\right) + \sigma_o^z \sigma_p^z \tag{5.28}$$

P_a depending on $\sigma_o^z \sigma_p^z$ is not a good sign as now the penalty terms for valid states might differ: this was exactly the source of our double-counting problem. If we include the gadget qubit interactions, we get that:

$$(P_a)_{implement,gadget} = \left(\left(\sum_{m,n\in Q_{a,edges}} \sigma_m^z \sigma_n^z\right) + \left(\beta \sum_{m,n\in Q_{a,diag}} \sigma_m^z \sigma_n^z\right) + \left(\alpha \sum_{m\in Q_a} \sigma_m^z \sigma_{S_a}^z\right)\right) + \sigma_o^z \sigma_p^z + \sigma_x^z \sigma_o^z + \sigma_x^z \sigma_p^z \tag{5.29}$$

Now, if we are in a newly defined valid state, we know that the first part of the penalty would output -2α as per the last section and the first two conditions of validity. The third condition of validity means that for at least one of o and p, they are in a different state compared to x. Without loss of generality let it be o, then we know that the eigenvalue of $\sigma_x^z \sigma_o^z$ is -1. Now there are two cases:

- $|o\rangle = |p\rangle$: if this is the case then the eigenvalue of $\sigma_o^z \sigma_p^z$ is 1, while the eigenvalue of $\sigma_x^z \sigma_o^z$ is same as $\sigma_x^z \sigma_p^z$ and is -1 and these two values cancel.

- $|o\rangle \neq |p\rangle$: if this is the case then the eigenvalue of $\sigma_o^z\sigma_p^z$ is -1, while the eigenvalue of $\sigma_x^z\sigma_p^z$ is opposite to $\sigma_x^z\sigma_o^z$ and is 1 and these two values cancel.

Therefore, we get that no matter what, the gadget qubit interactions along with 1 magnitude of the $ZZ-$interaction between the boundary term qubits subtracts -1 from the energy penalty. Therefore, we satisfy the condition that all valid states of the plaquette has the same penalty energy. Note that we do not fall into the same trap as the double-counting error as the boundary terms and the gadget only interact in this particular plaquette. Therefore, we can safely add up all plaquette penalty energy terms, and we will find that both boundary and non-boundary interactions between physical qubits required magnitude 2 which is what we implemented (from the two plaquettes for the non-boundary terms, and from the plaquette and the gadget for the boundary term). As the boundary terms do not interact with any other plaquette, we can generalize this for plaquettes with more than 1 or no boundary terms. Therefore, if we are in state such that all plaquettes are valid, the penalty term simply adds -2α for each plaquette and -1 for each boundary term. Therefore, the physical Hamiltonian is:

$$H_{physical,valid} = c\sum_{i<j} J_{i,j}^{physical}\sigma_{ij}^z + (-2\alpha)\left(\frac{n(n+1)}{2} - n\right) + (-1)\times(\text{number of boundary terms}) \tag{5.30}$$

where c is the scaling constant and we have that the maximum this can be is:

$$\max(H_{physical,valid}) = c\sum_{i<j} |J_{i,j}^{physical}| + (-2\alpha)\left(\frac{n(n+1)}{2} - n\right) + (-1)\times(\text{number of boundary terms}) \tag{5.31}$$

Conversely, now we need to show that failing the gadget qubit validity condition increases the penalty energy of the plaquette. The only scenario in which we do not pass the condition is that when all three of the qubits are in the same state. In such a situation, the eigenvalue of each of the ZZ term is 1 and therefore, eigenvalue of $\sigma_o^z\sigma_p^z + \sigma_x^z\sigma_p^z + \sigma_x^z\sigma_o^z$ is 3. Therefore, 3 is added to the penalty energy if the plaquette state is invalid due to the gadget qubit condition. The minimum the energy of an invalid state can be if only one plaquette is invalid (as each invalid plaquette adds up penalty terms) in one of two ways (as both the gadget invaldity and odd-parity invalidity serve to increase the penalty due to the plaquette). Therefore, the minimum energy an invalid state can have is given by:

$$\begin{aligned}\min(H_{physical,invalid}) = c\sum_{i<j} -|J_{i,j}^{physical}| + \min\Bigg(&(-2\alpha))\left(\frac{n(n+1)}{2} - n\right)\\ &+(-1)\times(\text{number of boundary terms - 1}) + 3,\ (-2\alpha)\left(\frac{n(n+1)}{2} - n - 1\right)\\ &+(-1)\times(\text{number of boundary terms}) + \min(2\alpha, -2\beta, -4+2\beta, 4+2\beta\pm 4\alpha)\Bigg)\end{aligned} \tag{5.32}$$

As before, once we know α and β, we can set c such that $\min(H_{physical,invalid}) - \max(H_{physical,valid}) > 0$. In this setting to go from the physical Hamiltonian energy of a valid to the energy of the corresponding configuration of Ising model variables, we first add $1 \times$ (number of boundary terms) and then scale by $\frac{1}{c}$. Note that the number of boundary terms can be computed a priori, noting that there are $n-1$ boundary terms on the left, $n-1$ on the right, and $2(n-1)$ in the bottom row, giving us $4n-4$ terms. As we are associating a qubit with every boundary term this adds at most $\boxed{4n-4}$ qubits to our scheme where n is the number of variables in the Ising model instance. Therefore, by adding these gadget qubits for each boundary term with the defined interaction strengths and fixing the magnitude of physical-physical interactions on plaquette sides to 2, we have resolved the double-counting issue.

5.3 Fine-grained scaling of our implementation

Earlier, we saw that the number of qubits required for the odd-parity LHZ scheme is given by $\frac{n(n+1)}{2} + n - 1$ where $n-1$ are the terms of the bottom row and we have $\frac{n(n+1)}{2}$ instead of $\frac{n(n-1)}{2}$ as we added a dummy variable labelled -1 to simulate the effect of having local fields (coefficients of linear terms) in the Ising model instance. A key challenge of resolving the double-counting issue was doing it without adding many additional qubits. In fact, we added only $4n-4$ and therefore, our solution to resolving the double-counting error does not change the asymptotic scaling of scheme which is still: to map a Ising model problem onto the dynamics of qubits in two dimensions, we require $\boxed{O(n^2)}$ qubits. However, in near-term settings, the exact scaling of the overhead is critical to determining how large problems we can implement on our quantum simulators. Our mapping requires $\boxed{\frac{n(n+1)}{2} + 5n - 5 = \frac{n^2}{2} + \frac{11n}{2} - 5}$. To be concrete, let us say we want to solve an arbitrary Ising model problem with 50 variables, which is in the regime where current classical computers cannot solve the problem in feasible amount of time. For 50 variables, using our scheme, we would require 1520 qubits which we may have access to (although noisy) in the coming years given the rapid development in the field. Of course, we do not expect an exponential speed-up given that the Ising model is NP-complete. We may attain sub-exponential speedups, and more importantly, be able to probe approximate sampling versions of the problem and explore different regimes of the J matrix which might not be NP-hard thereby solving practically interesting problems.

6 | Feasible implementation with Rubidium atom arrays

Now that we have resolved the double-counting error in the qubit-only implementations of the odd-parity LHZ scheme, the scheme is feasible if we can programmably implement the following Hamiltonian:

$$H_{physical} = \sum_{i<j} J_{ij}^{physical} \sigma_{ij}^{z} + \sum_{a} \left(\left(\sum_{m,n \in Q_{a,edges}} \sigma_m^z \sigma_n^z \right) + \left(\beta \sum_{m,n \in Q_{a,diag}} \sigma_m^z \sigma_n^z \right) + \left(\alpha \sum_{m \in Q_a} \sigma_m^z \sigma_{S_a}^z \right) \right) + \sum_{m,n \in b} (\sigma_m^z \sigma_n^z + \sigma_{xmn}^z \sigma_n^z + \sigma_{xmn}^z \sigma_m^z) \tag{6.1}$$

where a runs over all plaquettes, Q_a is the set of all physical qubits in a plaquette a, S_a is the ancillary qubit of plaquette a, b is the set of all boundary interactions defined between two physical qubits and x_{mn} is the gadget qubit associated with physical qubit m and n. Therefore, as per our discussion in the last section, in essence, we want that:

- For every plaquette, the $ZZ-$interaction strength between two physical qubits s distance apart is normalized to be 2, the strength between two physical qubits $\sqrt{2}s$ distance apart is β and the strength between a physical and ancillary qubit at distance $\frac{1}{\sqrt{2}}s$ is α where α and β satisfy our inequalities (5.13 - 5.16)
- For every boundary interaction, we can associate a gadget qubit that is s_g distance away from both physical qubits of the boundary interaction such that the interaction strength between the gadget and physical qubit is of magnitude 1 (in the same units as above).
- Interactions beyond a single plaquette are negligible compared to the magnitude of $1, \alpha$ and β.

As we saw in chapter 2, the dynamics of Rydberg atom arrays are governed by the following Hamiltonian:

$$H_{Ryd} = \sum_{\nu} (\Omega_\nu \sigma_\nu^x - \Delta_\nu n_\nu) + \sum_{\nu<w} V(|\vec{x}_\nu - \vec{x}_w|, \nu, w) n_\nu n_w \tag{6.2}$$

where ν, w are qubits realized by atoms and the n operator has eigenvalue 1 for state $|r\rangle$ and 0 for state $|g\rangle$. Each atom is individually controlled by a laser and Ω_ν is the Rabi frequency determining the intensity of the laser that is associated with qubit ν and determines the amplitude of flipping between state $|g\rangle$ and $|r\rangle$ and therefore, is the coefficient of the σ_ν^x operator while Δ_ν is the laser detuning for qubit ν controlling the n_μ operator. We assume that these two constants are tunable parameters. $\vec{x}_\nu$ refers to the position of qubit ν and V is the Rydberg potential, a function of the distance between two qubits and also, which particular excited Rydberg state qubit ν and w are coupled to. If both are coupled to the state $|r\rangle = |nS\rangle$ (where n is the principal quantum number and S means $l = 0$) for some large $n \approx 50$, then the energy of the $|rr\rangle$ state (where two interacting qubits are in their excited Rydberg state) is shifted by a van der Waals interaction with magnitude:

$$V(|\vec{x}_\nu - \vec{x}_w|, \nu, w) = \frac{C_6}{|\vec{x}_\nu - \vec{x}_w|^6} \tag{6.3}$$

where the C_6 coefficient scales approximately as n^{11}. Let us go over all three parts of equation 6.1 and show how it may be implemented using equation 6.2.

Encoding $J_{ij}^{physical}$: The key challenge in encoding $\sum_{i<j} J_{ij}^{physical}\sigma_{ij}^z$ is that it is in terms of the σ^z operator while our Rydberg hamiltonian is in terms of the n operator. However, noting the definitions (equations 3.11 and 3.13), we can simply convert from one to the other using:

$$2\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix} - \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix} = \sigma^z \tag{6.4}$$

$$\implies \sigma^z = 2n - 1 \tag{6.5}$$

Subsisting equation 6.5 into the first part of $H_{physical}$, we get that:

$$\sum_{i<j} J_{ij}^{physical}\sigma_{ij}^z = 2\sum_{i<j} J_{ij}^{physical} n_{ij} - \sum_{i<j} J_{ij}^{physical} \tag{6.6}$$

$\sum_{i<j} J_{ij}^{physical}$ is simply a constant shift in energy that we can precompute and add when converting from the energy of our implementation to the energy of the $H_{physical}$ for a particular valid state on our way to determining the energy of the Ising model instance for the corresponding variable assignment. Moreover, to implement the first part of equation 6.6, we match with equation 6.2 and set the laser detuning for qubit labelled ij $\Delta_{ij} = -2J_{ij}^{physical}$.

Implementing plaquette terms: Now, let us consider a single plaquette a. We want to implement the following:

$$\left(\left(\sum_{m,n\in Q_{a,edges}} 2\sigma_m^z\sigma_n^z\right) + \left(\beta \sum_{m,n\in Q_{a,diag}} \sigma_m^z\sigma_n^z\right) + \left(\alpha \sum_{m\in Q_a} \sigma_m^z\sigma_{S_a}^z\right)\right) \tag{6.7}$$

Again, we can use equation 6.5 to represent the $\sigma_m^z\sigma_n^z$ operators with $n_m n_n$ operators. We get that:

$$\sigma_m^z\sigma_n^z = (2n_m - 1)(2n_n - 1) = 4n_m n_n - 2n_m - 2n_n + 1 \tag{6.8}$$

Here for every interaction we get a constant shift of energy by 1 which again we can precompute as we know the number of plaquettes and number of interactions within a plaquette and add back later. We can address the additional single n operator terms by incrementing the Δ by the required quantity, given than Δ is completely programmable. For example, for the β interaction, we get a $-2\beta n_m$ term in the expansion and therefore, for qubit labelled m, we can decrement δ_m by 2β from what was set while encoding $J_{ij}^{physical}$ terms. Now we have can implement the nn terms with the required interaction strengths, we have effectively implemented all plaquette interactions. Note that the coeffecient of the ZZ interaction gets multiplied by a constant (4) to get the new coefficient of the corresponding nn and therefore, the ratio between the required interactions remains the same. That is, the nn interaction strengths we want are 8, 4α and 4β (in the normalized units) for physical-physical side, physical-ancillary and physical-physical diagonal interactions, respectively such that α and β satisfy our inequalities. Therefore, if we find qubit implementations such that interactions are $2, \alpha', \beta'$, respectively, and these values satisfy our inequalities, then we can simply set $\alpha = \frac{\alpha'}{4}$ and $\beta = \frac{\beta'}{4}$ as our inequalities are scale-invariant. Therefore, the following diagram tells us the interactions we want to implement between different qubits in the plaquette in some normalized units.

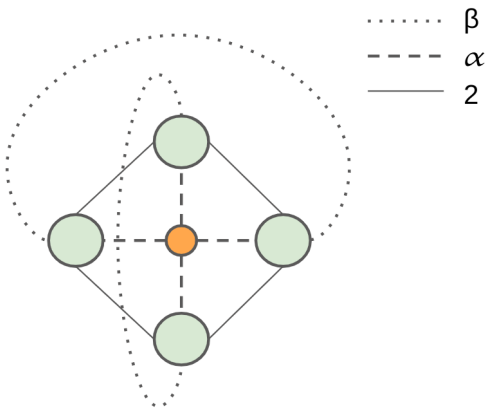

Figure 6.1: Corrected plaquette interactions note the 2 for the physical-physical interactions

To engineer these interactions, [Gla+17] used to different species of atoms. Our aim is to use a single species of atoms (^{87}Rb) to reduce experimental complexity. The main tool at our disposal is that we can couple different atoms to different excited Rydberg states by varying the frequency that controls the particular atom. Therefore, now our Rydberg potentials will no longer be limited to equation 6.3. For every atom, we can control the principal and angular momentum quantum number (n and l). Here are the attempts with increasing complexity:

- **Attempt 1**: Both physical and ancillary qubits are coupled to Rydberg state $|r\rangle = |nS\rangle$ ($l = 0$). In this case, our interactions are indeed, governed by equation 6.3, and we can show that this would not satisfy the inequalities. Let the interaction between two atoms at distance s be 2 as desired. Now, we know that:

$$V(s) = \frac{C_6}{s^6} = 2 \implies C_6 = 2s^6 \tag{6.9}$$

  Using this, we can compute the interactions at distance $\sqrt{2}s$ (diagonal interaction, β) and at distance $\frac{1}{\sqrt{2}}s$ (physical-ancillary interaction, β) and we get that:

$$\beta = \frac{C_6}{\sqrt{2}^6 s^6} = \frac{2s^6}{8s^6} = \frac{1}{4} \tag{6.10}$$

$$\alpha \frac{C_6}{\left(\frac{1}{\sqrt{2}}\right)^6 s^6} = \frac{2s^6}{\frac{1}{8}s^6} = 16 \tag{6.11}$$

  However, note that $4 + 2\beta - 4\alpha = -59.5 < -32 = -2\alpha$ and hence, inequality in equation (5.16) is not satisfied.

- **Attempt 2:** Here, we let the physical qubits be coupled to Rydberg state $|r_p\rangle = |nS\rangle$ while the ancillary qubit is coupled to state $|r_a\rangle = |n'S/P\rangle$. To compute the Rydberg potential, as discussed in Chapter 2, we must compute the matrix elements of the form:

$$|\,|n'', l'', j'', m_j''\rangle \otimes \langle n', l', j', m_j'|V_{dd}|n_a, l_a, j_a, (m_j)_a\rangle \otimes |n_b, l_b, j_b, (m_j)_b\rangle\,|^2 \tag{6.12}$$

  where we must sum over all possible n'', l'', j'', m_j'' and n', l', j', m_j' states of the atom pair and normalize to obtain the energy shift. In theory, to compute the energy shift and hence, this sum, we must iterate over all possible other states. However, numerical methods have been developed [Rei+07], with heuristic measures on which of these "virtual" states are important to consider and contribute the most to the sum. We, in particular, will be looking at states such that $n_a - 7 < n'', n < n_a + 7$ and $n_b - 7 < n'', n < n_b + 7$. We search for $10 < n, n' < 130$ with $s \in (0, 5)\mu m$. However, $S - S$ interactions are very strong: this means that the interactions between the ancillary qubits of two different plaquettes was not negligible. The

minimum ratio we could find of the strength of the ancillary-ancillary interactions to physical-physical interactions was of the order 10^{-1}. We can decrease this ratio by noting that $P-P$ interactions are weaker then $S-S$ interactions given the larger overlap between $S-S$ orbitals. Indeed, while searching with this constraint, we find the following α, β, n_p, n_a values:

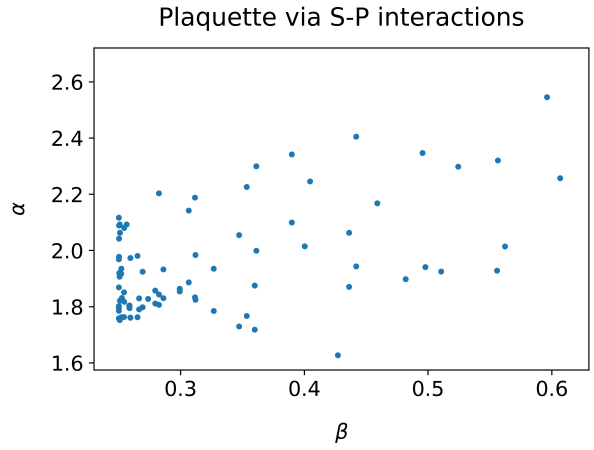

Figure 6.2: Implementable α, β pairs

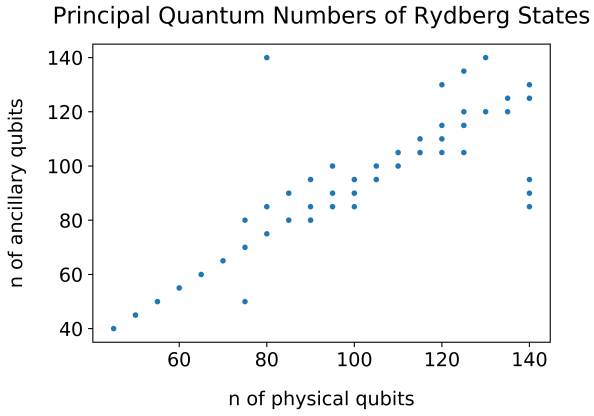

Figure 6.3: Principle quantum numbers required to implement LHZ scheme

For the S-P case, we are able to find states such that the cross-talk ratio is as low as 10^{-3}. We note that n, n' increase with each other in order to maintain the inequalities (5.13-5.16)

Implementing gadget qubits: To implement the gadget qubit interaction, we need $ZZ-$interactions of strength 1 (in the units we have been discussing). Again, we can use that:

$$\sigma_m^z \sigma_n^z = (2n_m - 1)(2n_n - 1) = 4n_m n_n - 2n_m - 2n_n + 1 \tag{6.13}$$

We know the number of boundary terms and therefore, the energy shift of +1 can be precomputed and added later on when computing the corresponding energy of the Ising model. Note that the factor of 4 existed in the physical-physical side term expansion as well considered in the plaqutte interactions and therefore, whatever strength we normalize and set to 1 in the last section is what we would have to implement for the gadget-physical interaction. We can decrement the Δ, for qubits on the boundary by 2 and therefore, effectively implementing the $ZZ-$interaction up to a constant energy shift. To implement the interaction with strength 1, we will use the same Rydberg state for the gadget qubit as we chose for the ancillary qubit. Let us use the state $|50S\rangle$ and the state $|45P\rangle$ for the physical and ancillary/gadget qubits respectively, as an example. Now, we know that at distance $\frac{1}{\sqrt{2}}$ the interaction between these two is of strength $\alpha = 1.9$. Therefore, we must place the gadget qubit further away than this distance to attain a strength of 1. If we plot the interaction strength as a function of distance, we find numerically that the curve seems continuous: in theory, if we can place an atom with arbitrary precision, we can get as close to 1 as we want:

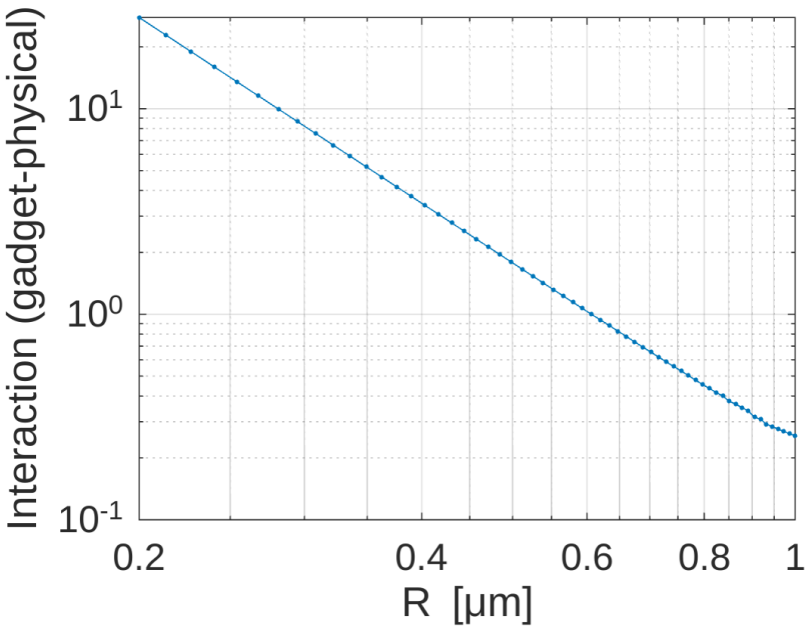

Figure 6.4: Gadget-physical interactions. We can set the distance between the gadget and physical qubit such that the interaction is arbitary close to 1

Long-range interactions: Using the ancillary qubit as P, we reduced long-range ancillary interactions to be about 10^{-3} compared to the relevant physical-physical interaction. However, our proof works in the case when long-range interactions are effectively zero. Therefore, the result in the next section is restricted to the case when we set all long-range interactions to zero and whether it holds when we consider long-range interactions is open, but we believe that the proof can be adapted using techniques in [Pic+18b] that

proved that Rydberg dynamics are NP-hard by reducing from maximum independent set problem on planar graphs with maximum degree three, by engineering certain interactions that overwhelm long-range interactions. In our setting, that may be equivalent to ensuring that all $|J_{ij}|$ are above a certain threshold C. Note that J_{ij} are constrained from both sides: guaranteed separation between odd and even-parity states require them to be smaller in magnitude while non-zero long-range interactions require them to large to overwhelm the long-range interactions.

6.1 Computational complexity of Rubidium array dynamics

The last section concludes our reduction for mapping an arbitrary Ising model instance of the dynamics of Rubidium atoms in two dimensions interacting with Rydberg states. Therefore, what we have proved is the following:

Theorem 6.1. Consider the following Hamiltonian that describes the dynamics of Rubidium atom arrays on n atoms:

$$H_{Ryd} = \sum_{\nu} (\Omega_{\nu}\sigma_{\nu}^{x} - \Delta_{\nu}n_{\nu}) + \sum_{\nu<w} V(|\vec{x}_{\nu} - \vec{x}_{w}|, \nu, w)n_{\nu}n_{w} \tag{6.14}$$

For a given set of positions of atoms, parameters $\{\Omega, \Delta\}$ and the Rydberg state each atom is coupled to, for some number k if there is a state of the form $\{|r\rangle, |g\rangle\}^n$ on n atoms ψ such that $\langle\psi|H_{Ryd}|\psi\rangle < k$ function $Rydberg(\{\Omega, \Delta, x, n, l\}, k) : \{0,1\}^* \to \{0,1\}$ outputs 1 and 0 otherwise. This implies, that finding the ground state of this Hamiltonian is NP-hard.

Proof. We will prove both that the problem is in NP and is $NP - hard$:

- If $Rydberg(\{\Omega, \Delta, x, n, l\}, k) = 1$, then there must be a product state of the form $|\psi\rangle \in \{|r\rangle, |g\rangle\}^n$ for which H_{Ryd} has energy $< k$. This state can be described using a string of siz4 n. Furthermore, we can verify that $Rydberg(\{\Omega, \Delta, x, n, l\}, k) = 1$ by computing the eigenvalue of H_{Ryd} for this state. As at maximum, we need to compute two-body terms, we can perform this computation in $O(n^2)$ time which is polynomial in the input size as the input contains details for n qubits and therefore, must have size at least in $O(n)$.

- As we have shown in Chapters 4, 5 and 6, we can reduce the Ising model decision problem on $\{J, h, k\}$, which is NP-complete onto the Rydberg energy decision problem. Let us say we are given an arbitrary Ising model instance. We use the Rubidium-atom based implementation odd-parity LHZ scheme with gadget

qubits to determine the $\{\Omega, \Delta, x, n, l\}$ input to *Rydberg*. This will take $O(n^2)$ time as we need to define these parameters for $O(n^2)$ atoms (for example, by finding $J_{ij}^{physical}$ based on the flipping set selection rules, finding $\{x\}$ based on the geometry of the LHZ scheme which can be done in constant time analytically). Let these values be called $\{\Omega, \Delta, x, n, l\}_{LHZ}$. To compute k' that goes into *Rydberg*, we first convert from the *nn* implementation to the *ZZ* implementation and then from that to the Ising model energy values and set $k' = ck + (-2\alpha)(\frac{n(n+1)}{2} - n) + (-1)(4n-4) - \text{number of ZZ interactions} + \sum_{i<j} J_{ij}^{physical}$ where the number of ZZ interactions is a function fixed of n.

– *Completeness:* If there is an Ising model variable assignment such that its energy is less than or equal to k, we know that there is a valid state in the odd-parity LHZ scheme that corresponds to the variable assignment being considered as per Chapter 5. As per equation 5.31, the ZZ physical Hamiltonian has energy and our construction:

$$\sum_{i<j} J_{i,j}^{physical} \sigma_i^z \sigma_j^z = \sum_{i<j} J_{i,j} S_i S_j + \sum_i h_i S_i \leq k \tag{6.15}$$

$$\implies H_{physical,valid} \leq ck + (-2\alpha)\left(\frac{n(n+1)}{2} - n\right) + (-1)(4n-4) \tag{6.16}$$

Now, to go to the the *nn* implementation of the Hamiltonian, using equations 6.6 and 6.8, we have that:

$$\sum_{i<j} J_{ij}^{physical} + \sum_{i<j} J_{ij}^{physical} \sigma_{ij}^z = 2 \sum_{i<j} J_{ij}^{physical} n_{ij} \tag{6.17}$$

$$-1 + \sigma_m^z \sigma_n^z = (2n_m - 1)(2n_n - 1) = 4n_m n_n - 2n_m - 2n_n \tag{6.18}$$

where the second equation applies to all ZZ interactions. Therefore, we have that:

$$H_{physical,valid} + \sum_{i<j} J_{ij}^{physical} - 1(\text{number of ZZ interactions}) = H_{Ryd} \tag{6.19}$$

Combing 6.16 and 6.19, we get that:

$$H_{Ryd} \leq ck + (-2\alpha)\left(\frac{n(n+1)}{2} - n\right) + (-1)(4n-4) + \sum_{i<j} J_{ij}^{physical} - 1(\text{number of ZZ interactions}) = k' \tag{6.20}$$

and hence, $Rydberg(\{\Omega, \Delta, x, n, l\}_{LHZ}, k') = 1$.

– *Soundness:* Let $Rydberg(\{\Omega, \Delta, x, n, l\}_{LHZ}, k') = 1$. To show that $Ising(\{J, h\}, k) = 1$, we simply reverse the steps we followed in the completeness part.
If $Rydberg(\{\Omega, \Delta, x, n, l\}_{LHZ}, k') = 1$ and there is a state $\{|r\rangle, |g\rangle\}^n$ such that

the energy of the state is less than or equal to $-k$. Then that means that using equation 6.17) and equation 6.18), there is $\{|r\rangle, |g\rangle\}^n$ such that:

$$H_{physical} \leq k' + \text{number of ZZ interactions} - \sum_{i<j} J_{ij}^{physical}$$

$$= ck + (-2\alpha)(\frac{n(n+1)}{2} - n) + (-1)(4n-4) \tag{6.21}$$

There are two cases:

* $\{|r\rangle, |g\rangle\}^n$ is a valid state. If this is true, then using (5.31), we know that:

$$H_{physical,valid} = c\sum_{i<j} J_{i,j}^{physical}\sigma_{ij}^z + (-2\alpha)\left(\frac{n(n+1)}{2} - n\right) + (-1)(4n-4) \tag{6.22}$$

Combining this with 6.21), we get that:

$$c\sum_{i<j} J_{i,j}^{physical}\sigma_i^z\sigma_j^z + (-2\alpha)\left(\frac{n(n+1)}{2} - n\right) + (-1)(4n-4)$$

$$\leq ck + (-2\alpha)(\frac{n(n+1)}{2} - n) + (-1)(4n-4) \tag{6.23}$$

$$\implies \sum_{i<j} J_{i,j}^{physical}\sigma_{ij}^z \leq k \tag{6.24}$$

Given that we are in a valid state, we know that there exists $\{S\}$ such that $\sum_{i<j} J_{ij}S_iS_j + \sum_i h_iS_i = \sum_{i<j} J_{i,j}^{physical}\sigma_{ij}^z \implies \sum_{i<j} J_{ij}S_iS_j + \sum_i h_iS_i \leq k$ and therefore $Ising(\{J,h\},k) = 1$.

* $\{|r\rangle, |g\rangle\}^n$ is not a valid state. If this is true, then we know that the energy of this state must be more than the maximum energy of the valid state. Let the actual energy be E. Using Equation 5.32:

$$ck + (-2\alpha)(\frac{n(n+1)}{2} - n) + (-1)(4n-4) \geq E \geq \max(H_{physical,valid}) \tag{6.25}$$

$$ck + (-2\alpha)(\frac{n(n+1)}{2} - n) + (-1)(4n-4)$$
$$\geq c\sum_{i<j} |J_{i,j}^{physical}| + (-2\alpha)\left(\frac{n(n+1)}{2} - n\right) + (-1) \times (\text{number of boundary terms}) \tag{6.26}$$

$$\implies k \geq \sum_{i<j} |J_{i,j}^{physical}| \tag{6.27}$$

$\sum_{i<j} |J_{i,j}^{physical}|$ is the maximum that energy any variable assignment of the Ising model can have and therefore, there exists a variable assignment that has energy less than or equal to k and we have that $Ising(\{J,h\},k) = 1$.

□

6.2 Future directions

Mapping other problems: We know that we can obtain a universal gate set by using Rydberg interactions [Lev+19; Jak+00]. Mapping an NP-complete problem onto the dynamics of Rydberg atom arrays does not suffice to show that we can attain universal adiabatic quantum computation. Therefore, a future project could be attempting to map a problem that is powerful enough to prove universality. QMA-complete problems are good candidates for the same [BL08].

Furthermore, it would be useful to develop schemes to implement the sampling problems discussed in Section 2.1 (random circuits, IQP, and BosonSampling) as we have strong evidence that these problems are hard to simulate classically but are not known to be NP-complete and hence, we may expect super-polynomial speed-ups.

Simulating and analyzing performance of the adiabatic algorithm and QAOA In this thesis, we only mapped the Ising model problem onto Rydberg dynamics but did not analyze the performance of approximating the problem or sampling low energy Ising model variable assignments using the adiabatic algorithm or QAOA. The same analysis that was done for MIS [Pic+18a] could be done for this problem. Furthermore, for QAOA, we could study whether there exist any patterns in the optimized angles β, γ as done for the MaxCut problem here [Zho+18].

References

[End+16] Manuel Endres et al. “Atom-by-atom assembly of defect-free one-dimensional cold atom arrays”. In: *Science* 354.6315 (2016), pp. 1024–1027.

[Bar+16] Daniel Barredo et al. “An atom-by-atom assembler of defect-free arbitrary two-dimensional atomic arrays”. In: *Science* 354.6315 (2016), pp. 1021–1023.

[Lev+19] Harry Levine et al. “Parallel implementation of high-fidelity multiqubit gates with neutral atoms”. In: *Physical review letters* 123.17 (2019), p. 170503.

[CCJ90] Brent N Clark, Charles J Colbourn, and David S Johnson. “Unit disk graphs”. In: *Discrete mathematics* 86.1-3 (1990), pp. 165–177.

[Pic+18a] Hannes Pichler et al. “Quantum optimization for maximum independent set using Rydberg atom arrays”. In: *arXiv preprint arXiv:1808.10816* (2018).

[Pic+18b] Hannes Pichler et al. “Computational complexity of the Rydberg blockade in two dimensions”. In: *arXiv preprint arXiv:1809.04954* (2018).

[Len20] W Lenz. “Contributions to the understanding of the magnetic properties in solid bodies , v s”. In: *physical Z* 21 (1920), pp. 613–615.

[Bar82] Francisco Barahona. “On the computational complexity of Ising spin glass models”. In: *Journal of Physics A: Mathematical and General* 15.10 (1982), p. 3241.

[Luc14] Andrew Lucas. “Ising formulations of many NP problems”. In: *Frontiers in Physics* 2 (2014), p. 5.

[LHZ15] Wolfgang Lechner, Philipp Hauke, and Peter Zoller. “A quantum annealing architecture with all-to-all connectivity from local interactions”. In: *Science advances* 1.9 (2015), e1500838.

[Gla+17] Alexander W Glaetzle et al. “A coherent quantum annealer with Rydberg atoms”. In: *Nature communications* 8 (2017), p. 15813.

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[Kit+02] Alexei Yu Kitaev et al. *Classical and quantum computation*. 47. American Mathematical Soc., 2002.

[Bar] Boaz Barak. *Introduction to Theoretical Computer Science*. Under preperation.

[Tur36] Alan Mathison Turing. "On computable numbers, with an application to the Entscheidungsproblem". In: *J. of Math* 58.345-363 (1936), p. 5.

[Str65] Christopher Strachey. "An impossible program". In: *The Computer Journal* 7.4 (1965), pp. 313–313.

[CPW15] Toby S Cubitt, David Perez-Garcia, and Michael M Wolf. "Undecidability of the spectral gap". In: *Nature* 528.7581 (2015), pp. 207–211.

[HS65] Juris Hartmanis and Richard E Stearns. "On the computational complexity of algorithms". In: *Transactions of the American Mathematical Society* 117 (1965), pp. 285–306.

[Fey99] Richard P Feynman. "Simulating physics with computers". In: *Int. J. Theor. Phys* 21.6/7 (1999).

[Deu85] David Deutsch. "Quantum theory, the Church–Turing principle and the universal quantum computer". In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400.1818 (1985), pp. 97–117.

[Deu89] David Elieser Deutsch. "Quantum computational networks". In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 425.1868 (1989), pp. 73–90.

[Kit97] Aleksei Yur'evich Kitaev. "Quantum computations: algorithms and error correction". In: *Uspekhi Matematicheskikh Nauk* 52.6 (1997), pp. 53–112.

[DN05] Christopher M Dawson and Michael A Nielsen. "The solovay-kitaev algorithm". In: *arXiv preprint quant-ph/0505030* (2005).

[Sho94] Peter W Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th annual symposium on foundations of computer science*. Ieee. 1994, pp. 124–134.

[KKR06] Julia Kempe, Alexei Kitaev, and Oded Regev. "The complexity of the local Hamiltonian problem". In: *SIAM Journal on Computing* 35.5 (2006), pp. 1070–1097.

[AL18] Tameem Albash and Daniel A Lidar. "Adiabatic quantum computation". In: *Reviews of Modern Physics* 90.1 (2018), p. 015002.

[Chi] Andrew Childs. "Overview of adiabatic quantum computation". In: ().

[BF28] Max Born and Vladimir Fock. "Beweis des adiabatensatzes". In: *Zeitschrift für Physik* 51.3-4 (1928), pp. 165–180.

[Ami09] Mohammad HS Amin. "Consistency of the adiabatic theorem". In: *Physical review letters* 102.22 (2009), p. 220401.

[EH12] Alexander Elgart and George A Hagedorn. "A note on the switching adiabatic theorem". In: *Journal of Mathematical Physics* 53.10 (2012), p. 102202.

[Far+00] Edward Farhi et al. "Quantum computation by adiabatic evolution". In: *arXiv preprint quant-ph/0001106* (2000).

[Aha+08] Dorit Aharonov et al. "Adiabatic quantum computation is equivalent to standard quantum computation". In: *SIAM review* 50.4 (2008), pp. 755–787.

[Aha+09] Dorit Aharonov et al. "The power of quantum systems on a line". In: *Communications in Mathematical Physics* 287.1 (2009), pp. 41–65.

[BL08] Jacob D Biamonte and Peter J Love. "Realizable Hamiltonians for universal adiabatic quantum computers". In: *Physical Review A* 78.1 (2008), p. 012352.

[FGG14] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. "A quantum approximate optimization algorithm". In: *arXiv preprint arXiv:1411.4028* (2014).

[Zho+18] Leo Zhou et al. "Quantum approximate optimization algorithm: performance, mechanism, and implementation on near-term devices". In: *arXiv preprint arXiv:1812.01041* (2018).

[Ale+19] Yuri Alexeev et al. "Quantum Computer Systems for Scientific Discovery". In: *arXiv preprint arXiv:1912.07577* (2019).

[Alt+19] Ehud Altman et al. "Quantum Simulators: Architectures and Opportunities". In: *arXiv preprint arXiv:1912.06938* (2019).

[HM17] Aram W Harrow and Ashley Montanaro. "Quantum computational supremacy". In: *Nature* 549.7671 (2017), pp. 203–209.

[FH16] Edward Farhi and Aram W Harrow. "Quantum supremacy through the quantum approximate optimization algorithm". In: *arXiv preprint arXiv:1602.07674* (2016).

[BJS11] Michael J Bremner, Richard Jozsa, and Dan J Shepherd. "Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 467.2126 (2011), pp. 459–472.

[AA11] Scott Aaronson and Alex Arkhipov. "The computational complexity of linear optics". In: *Proceedings of the forty-third annual ACM symposium on Theory of computing*. 2011, pp. 333–342.

[Aru+19] Frank Arute et al. "Quantum supremacy using a programmable superconducting processor". In: *Nature* 574.7779 (2019), pp. 505–510.

[AC16] Scott Aaronson and Lijie Chen. "Complexity-theoretic foundations of quantum supremacy experiments". In: *arXiv preprint arXiv:1612.05903* (2016).

[SWM10] Mark Saffman, Thad G Walker, and Klaus Mølmer. "Quantum information with Rydberg atoms". In: *Reviews of modern physics* 82.3 (2010), p. 2313.

[BL16] Antoine Browaeys and Thierry Lahaye. "Interacting cold Rydberg atoms: a toy many-body system". In: *Niels Bohr, 1913-2013*. Springer, 2016, pp. 177–198.

[Jak+00] D Jaksch et al. "Fast quantum gates for neutral atoms". In: *Physical Review Letters* 85.10 (2000), p. 2208.

[Mal+15] KM Maller et al. "Rydberg-blockade controlled-not gate and entanglement in a two-dimensional array of neutral-atom qubits". In: *Physical Review A* 92.2 (2015), p. 022336.

[Gra+19] TM Graham et al. "Rydberg-mediated entanglement in a two-dimensional neutral atom qubit array". In: *Physical Review Letters* 123.23 (2019), p. 230501.

[MM11] Cristopher Moore and Stephan Mertens. *The nature of computation*. OUP Oxford, 2011.

[Roj13] Raúl Rojas. *Neural networks: a systematic introduction*. Springer Science & Business Media, 2013.

[Sta08] Dietrich Stauffer. "Social applications of two-dimensional Ising models". In: *American Journal of Physics* 76.4 (2008), pp. 470–473.

[Cho08] Vicky Choi. "Minor-embedding in adiabatic quantum computation: I. The parameter setting problem". In: *Quantum Information Processing* 7.5 (2008), pp. 193–209.

[CMR14] Jun Cai, William G Macready, and Aidan Roy. "A practical heuristic for finding graph minors". In: *arXiv preprint arXiv:1406.2741* (2014).

[RBL16] Andrea Rocchetto, Simon C Benjamin, and Ying Li. "Stabilizers as a design tool for new forms of the Lechner-Hauke-Zoller annealer". In: *Science advances* 2.10 (2016), e1601246.

[AVL16] Tameem Albash, Walter Vinci, and Daniel A Lidar. "Simulated-quantum-annealing comparison between all-to-all connectivity schemes". In: *Physical Review A* 94.2 (2016), p. 022327.

[Rei+07] A Reinhard et al. "Level shifts of rubidium Rydberg states due to binary interactions". In: *Physical Review A* 75.3 (2007), p. 032712.