

ABSTRACT

Title of Dissertation: Experimental Study of
Quantum Algorithms on
Ion-trap Quantum computers

Daiwei Zhu
Doctor of Philosophy, 2021

Dissertation Directed by: Professor Christopher Monroe
Department of Physics

Quantum computing is likely going to be the most incredible science and technology breakthrough of our generation. Some of its promises, such as factorization of large numbers, require a system holding thousands of error-corrected qubits. Although We cannot build such a system yet, substantial progress has been made since the first-generation systems. Compared with those proof of principle systems holding a few qubits with several percent of error per gate, the latest generation systems can apply more than a hundred gates (with above 99% fidelities) to tens of connected qubits. This thesis focuses on applications of such state-of-the-art ion-trap quantum computers.

The latest ion-trap quantum computers have become complicated enough that their potential cannot be fully realized without automation. We present the full-stack automation scheme implemented on a system at the University of Maryland. With the automation scheme, the system can operate without human interference for a few days.

We present an experimental demonstration of a hybrid circuit-training algorithm for generation modeling. We also present other gate-based digital quantum simulations using hybrid ansatzes or the trotterization method.

Progress in quantum computing will soon enable us to make computations that cannot be simulated via classical computations. We present two different approaches to validate such computations. First, we demonstrate an approach using randomized measurement to compare the results from different quantum computers against each other. We then demonstrate an approach using interactive protocols to classically verify the results produced by quantum computers.

Experimental Study of Quantum Algorithms on Ion-trap Quantum Computers

by

Daiwei Zhu

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2021

Advisory Committee:

Professor Christopher Monroe, Chair/Advisor

professor Mohammad Hafezi

professor Edo Waks Professor Alexey Gorshkov

Professor Lawrence Washington

Professor Norbert Linke

Professor Xiaodi Wu

© Copyright by
Daiwei Zhu
2021

Preface

If needed.

Dedication

To Mom and Dad.

Acknowledgments

First and foremost, I would like to thank my parents. Their support and guidance have always been my most crucial resource to overcome challenges.

I would like to thank my advisor Chris Monroe. He offered excellent guidance. Working in his group, I enjoyed many privileges: extremely talented labmates, abundant funding, close collaborations with world-class researchers globally. His effort in the quantum industry created great career opportunities for researchers in AMO these days. As a slow learner, it took me a long time to find my passion and learn to contribute to research. I am grateful for Chris's patience and support.

I owe many thanks to Norbert Linke, who helped me through my darkest days in grad school. His guidance is crucial for me to overcome the imposter syndrome and become a confident and independent researcher. He is a great mentor and close friend.

Marko Cetina, Kristi Beck, Ken Wright, Kevin Landsman, Michael Goldman, Caroline Figgatt, and Kai Hudeck taught me everything about ion-trap experiments. I enjoyed the time we spent building experiments in the lab and having discussions in front of whiteboards or on the way to lunch. I started to work with Marko in my second year. Marko is one of the smartest physicists I know. Thanks for all the knowledge and experiences you generously shared with me.

Ze-Pei Cian, Tim Hsieh, Sonika Johri, Greg Meyer, Yunseong Nam, Laura Lewis, Marcello Benedetti are the theorists I closely collaborated with. Thank you for sharing exciting theory work with me and guide our experiments. Thank you for being so patient and taking extra time to explain related theories and derivation to me.

I worked closely with Cinthia Huerta Alderete, Nhung Nguyen, Yingyue Zhu, Drew Risinger, Laird Egan, Crystal Noel, Qingfeng Wang, Kate Collins, Bahaa Harraz and Debo Biswas throughout my Ph.D. We collaborated on many exciting projects and had so many discussions that inspired my researches.

I also feel grateful for the time I shared with other members of the Monroe group. Volkan Inlek, Jason Amini, Jiehang Zhang, George Toh, Guido Pagano, William Morong, Ksenia Sosnova, Patrick Becker, Wen Lin Tan, Allison Carter, Arinjoy De, Jameson O'Reilly, Abhishek Menon, Albert Chu, Sophia Scarano, Harvey Kaplan, Marty Lichtman, Micah Hernandez, Clayton Crocker, Steven Moses, and David Wong-Campos, you are the reason why our group is such an amazing team to work with.

Feng Lei joined the lab a year ago. We tried many different restaurants, hiked many different trails during weekends. You are an excellent physicist. Thank you for improving many fundamental aspects of my physicist-ship, including literature review, scientific writing, and math derivations.

Antonis Kyprianidis is my best friend in graduate school. I missed those hikes we took together. Some in the nearby parks, some on-campus just to skip work. Thank you for showing up with that bottle of scotch you said was saved for special

moments when I felt really sad that one night.

ZhangCheng Huang, Jin Yao, and Hang Yang are my best friends that I have been knowing for almost my whole life. We've only seen each other a few times through my Ph.D.. But our friendship is what keeps me encouraged.

I thank my friends outside work Liam Fowl, Andrew Otis, Aaron Doliana, Sohana Nasrin, Haiqing Zhao, Hao wu, Fang li, Zheng Zhang, Kang Zhang, Travis Dietz, Nihal Jhajj, Pancham Patel. The fun we had, either jamming, playing squash, windsurfing, snowboarding, or just fooling around are what keep me surviving.

I would like to thank my Jazz teacher Paul Bangser. Thank you for keeping my dream of becoming a professional musician alive in me.

Table of Contents

Dedication	iii
Acknowledgements	iv
Table of Contents	vii
List of Tables	ix
List of Figures	x
List of Abbreviations	xviii
Chapter 1: Introduction	1
1.1 Quantum Computing	2
1.2 Noisy Quantum Computers	8
1.3 Thesis Outline	10
Chapter 2: Ion Trap Quantum Computer: The Recipe	12
2.1 Trap the ion	12
2.2 Ionization	17
2.3 Doppler Cooling	19
2.4 Optical Pumping	22
2.5 Qubit Readout	23
2.6 Single-qubit gates: Rabi-flopping	23
2.7 Single-qubit gates: Raman transition	29
2.8 Quantum bus: the motional modes	35
2.9 Initialize the bus: Sideband cooling	39
2.10 Address the qubits: Two-qubit gates	40
2.11 Integration	45
Chapter 3: System Automation	48
3.1 Optical alignment	49
3.2 Ion loading	50
3.3 Center scan	52
3.4 Voltage calibration	53
3.4.1 Calibrating X_1 and X_3	54
3.4.2 Calibrating X_2 and X_4	55

3.4.3	Calibrating Q_{XZ}	57
3.4.4	Calibrating Z_1	60
3.4.5	Calibrating Q_{ZY} compensation	61
3.5	Gates calibration	64
3.5.1	Calibrate the gate angle	64
3.5.2	calibrate the Stark shift	65
3.6	Fidelity check	68
3.7	Integration	71
Chapter 4: Ion-Trap Quantum Programming		74
4.1	Rotation of gates	74
4.2	Convert arbitrary gates into ion-native gates	75
4.3	Generating Many-Body Pauli-Gates	77
Chapter 5: Trotterization based quantum simulation		79
5.1	Many-body localization (MBL)	80
5.2	Spectral function	83
5.3	simulation results	87
5.4	Experimental results	87
5.5	Error mitigation	92
Chapter 6: Hybrid Quantum Algorithms		96
6.1	Data Driven Circuit learning for Generative Modeling	97
6.1.1	The variational circuits (ansatzes)	98
6.1.2	The classical optimizer	100
6.1.3	The cost Function	103
6.2	Efficient Thermal State Preparation with the Quantum Approximate Optimization Algorithm (QAOA)	110
6.2.1	Thermofield Double (TFD) State	112
6.2.2	preparing TFD states with QAOA	113
6.2.3	TFD preparation: experimental results	117
6.2.4	prepare the Quantum Critical State at T=0	119
6.2.5	Demonstration of hybrid optimization	122
6.2.6	Cost function for hybrid optimization of TFD states	124
6.2.7	Symmetry Based Error Mitigation	125
6.2.8	Error Simulation	126
Chapter 7: Validating quantum computation		130
7.1	Cross-platform comparison	131
7.2	Interactive protocols for classically-verifiable quantum advantage	131
7.2.1	Learning with Errors (LWE) based protocol	134
7.2.2	Circuit construction for the LWE based protocol	137
7.2.3	$x^2 \bmod N$ based protocol	140
7.2.4	circuit construction of the $x^2 \bmod N$ based protocol	142
7.2.5	experiments	145

List of Tables

6.1	KL divergence (D_{KL} , see Materials and Methods), qBAS score, and entanglement entropy (S) for the state obtained at the end of each of the DDQCL training on hardware, for various circuits and classical optimizers used.	110
-----	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----

List of Figures

1.1	The Bloch state representation of a quantum state. The axes are normally defined in the following way: $ 0\rangle$ is along the positive Z-direction. $ +\rangle = \frac{1}{\sqrt{2}}(0\rangle + 1\rangle)$ is along the positive X-direction	3
1.2	Illustration of some of the single-qubit gates: A Hadamard gate is a rotation round the axis halfway between $ +\rangle$ and $ 0\rangle$ by π . A S gate is a rotation around the Z-axis by π . A T gate is a rotation around the Z-axis by $\pi/2$	4
1.3	symbol and truth table for control-not (CNOT) gates and XX gates .	7
2.1	The simplified diagram of an RF Paul-trap. The two blue electrodes are connected to the ground. The RF potential is applied to the two olive electrodes to provide trapping force in the radial direction. The two end-caps in lime color provide DC confinement along the axial direction	13
2.2	How the pseudo-potential is formed. The RF potential induces an in-phase micro-motion. Because of the micro-motion, the attraction is stronger than the repulsion from the RF. So on average, the ion feels an attraction in the RF potential.	17
2.3	Energy levels used to ionize Yb atoms.	18
2.4	The energy levels of $^{171}\text{Yb}^+$ relevant to the operations needed for our quantum computing scheme.	19
2.5	To Doppler cool ions, we park the laser's frequency at the red-side waist of the absorption spectrum. This is the frequency seen by a stationary ion. An ion moving into the laser will see a blue-shifted laser, which is closer to resonance. Ions moving away from the laser see a red-shifted laser, which is less in resonance. This asymmetry in the absorption cross-section gives ions a higher chance of absorbing counter-moving photons. As a result, the ions loose velocity and cool down eventually.	21
2.6	The energy levels and transitions used for the optical pumping. Because nothing leaves $ 0\rangle$, it eventually accumulates all the population.	22
2.7	Energy levels and transitions used for the qubit-state readout. This is also referred to as the state-dependent fluorescence.	24
2.8	The λ system involved in the Raman transition we use to address the qubit states. We use two lasers to off-resonantly couple both qubit states to an auxiliary state.	30

2.9	Driving Raman transition with two frequency combs. We use two laser beams derived from a single pulsed 355nm laser to drive the Raman transition. AOMs are used to shift the frequency of the combs so that pairs of comb teeth 12.6GHz away from each other drives resonant Raman transition.	34
2.10	To implement sideband cooling, we cycle between RSB transitions and optical pumping. This gradually shift populations to lower phonon number states.	40
2.11	During the MS interaction, different two-qubit states are effectively coupled together via off resonant coupling to motional states. These couplings are similar to the Raman-transition discussed previously. . .	42
2.12	The diagram of components needed to make an ion-trap quantum computer. Bronze arrows stand for lasers. Olive arrows stand for magnetic field. Blue arrows stand for uni- or bi- directional electric signals. Dashed arrows stand for weak signals probed for control purpose.	46
3.1	The schematic of a beam path stabilization system consists of two feedback loops. Each loop consists of a piezo-controlled mirror to steer the beam, a servo, and a position sensor. The first feedback loop stabilizes the position of the beam. The second feedback loop stabilizes the angle of the beam.	50
3.2	PMT-array reading through a typical loading event. At each loading attempt, we try to merge one ion into the ion chain. The reading from the PMT-array changes whenever the number of ions in the chain increase.	51
3.3	During a center scan, we turn on all the individual beams to drive Raman transitions while scanning the center of the whole ions chains along the axial direction. The individual beams' amplitude profiles are sampled as the population transfer from the Raman transitions. This center scan is used to locate the beams and the ions' relative position and count the number of ions in the chain.	52
3.4	The voltage calibration sequence. We calibrate each voltage term in the order shown in the figure. We first finish all the calibrations that use only one ion, then load 15 ions and finish the rest calibrations. .	53
3.5	The X-,Y-,Z- direction are defined as shown. The X-axis is along the axial direction of the ion chain. The Y-axis is parallel to the trap surface.	54
3.6	We use position shift to calibrate X_1 and X_3 . A high X_2 setting is first used to measure the un-shifted position of the ion. Then the X_1 induced position shift is measured with only 1 ion for calibration. We finally use 15 ions to measure the X_3 induced position shift for X_3 calibration.	56

3.7	Population transfer as a function of axial mode frequency. Blue dots represent experimental data and solid yellow curve is a fit to the Gaussian function to determine the center frequency. The difference between the scanned frequency and the calculation (from the X2 setting) is used to calibrate the X2 value.	57
3.8	The vector plot of the electric Field generated by the Q_{XZ} -quadrupole in the XZ-plane. The ion chain is tilted by this electric field.	58
3.9	To calibrate (null) the Q_{XZ} term, we find the frequency of an odd radial mode. We then scan the Q_{XZ} setting while driving a sideband transition on the located odd radial mode. We set the Q_{XZ} to the value that gives the minimum population transfer.	59
3.10	We scan the Z_1 setting while driving a sideband transition with the micro-motion frequency. The Z_1 setting is calibrated to the value that gives the minimum population transfer.	60
3.11	Left: the vector plot of the electric field generated by the Q_{ZY} quadrupole in the ZY-plane. This electric field squeezes one of the two orthogonal sets of radial motional modes and releases the other set. Right: the vector plot of the electric field generated by the RF pseudo-potential in the ZY-plane. This electric field squeezes the two orthogonal sets of radial motional modes together.	62
3.12	Second row: Differences between measured frequencies and the nominal frequencies of all the 15 radial modes we use for two-qubit XX-gates. The measurements are done after voltage calibration, successful or not. First row: same as the second row, but only displays the four modes closest to the frequency we use to drive the XX-gates. . .	63
3.13	The XX-echo sequence is used to measure Stark shift during the XX-gates. We first use a $R_y(\pi/2)$ to rotate both qubits the XX-gate acts into the $ +\rangle$ state, an eigenstate of the XX-gate. Then we repeat $XX(\pi/4)XX(-\pi/4)$ several times. Without Stark shift, the following $R_y(\pi/2)$ will rotate both qubits into $ 1\rangle$. Otherwise, we will have some populations in $ 0\rangle$	67
3.14	The Parity flop we use to measure the coherence (parity scan contrast) of the GHZ state.	70
3.15	The flow-chart of the automation scheme implemented on the Euriqa system at the University of Maryland. The blue arrow leads to operations happening immediately after. The red arrows are triggered by monitoring sub-routine. The green arrows are set by users to repeat with a given frequency.	71
3.16	The fidelity of 11 different XX-gates acting on 13 qubits, recorded through a 3-day-long automated circuit operation. The average of parity and coherence equals the fidelity as given by Eq.3.8.	73

5.1	The circuit used to simulate time-evolution under the Heisenberg model Hamiltonian. (a) After the qubits are prepared into the desired initial state, $m = 6$ Trotter steps are used to evolve the system to time t . All qubits are then measured in the z -basis. (b) Each Trotter step consists of several one- and two-qubit gates, as described by Eq. 5.10. The single-body interactions are implemented as rotations about the X or Z axis (R_x and R_z gates). (c) The two-body interactions $\hat{U}(J\delta)$ are implemented as three XX(Ising) gates sandwiched between single-qubit rotations. This segment of the circuit is equivalent to a sequential application of XX, YY, ZZ gates, which describes evolution under the Heisenberg interaction exactly.	83
5.2	A schematic example of the splitting of spectral lines according to the theoretical model in [1] for a localized system in the limit $J \ll w$	85
5.3	(a) The simulated spectrum for $n = 7$ spins averaged over 100 disorder configurations for fixed $J = 1$, while w is varied. (b) The same plot with the frequency scaled by w	88
5.4	The spectrum of $\hat{\sigma}_i^z$ at different values of J (with $w = 1$) for a 3 site system for two sample disorder realizations (top and bottom). Each panel show both simulation (curves) and experimental (symbols) results. The different colors are for the different sites. The lack of distinguishing characteristics between the spectra at different values of J for individual samples shows the necessity of averaging over several disorder realizations.	90
5.5	(a)-(c) The spectral function of $\hat{\sigma}_i^z$ averaged over position and 24 disorder realizations for different values of J (with $w = 1$) for a 3-site system. Lines show simulation results and circles show experimental results. (d)-(f) The spectral function normalized by its value at $\omega = 0$. At $J = 0.7$, the spectral response increases as ω decreases while at $J = 0.1$, the spectrum is damped at low frequencies. The points at $\omega = 0$ in the top row are discontinuous with the rest of the curves since they arise from the diagonal elements of the observable in the eigenstate basis which have qualitatively different behavior than the off-diagonal ones.	91
5.6	The distribution histogram of linewidths Γ calculated as described in the text at different values of J from data taken on 3 qubits for 24 realizations. The bins are [0-1], [1-2], ... etc. The number of peaks used to generate the distribution is ~ 200 for each value of J . We derive the errorbars shown in the plot by assuming each bin approximately follows a binomial distribution. The inset shows the Pearson's first coefficient of skewness, $Sk1$, and the average linewidth, $\bar{\Gamma}$	94

6.1	Data-driven quantum circuit learning (DDQCL) is a hybrid quantum algorithm scheme that can be used for generative modeling, illustrated here by the example of 2-by-2 Bars and Stripes (BAS) data. From top left, clockwise: A parameterized circuit is initialized at random. Then at each iteration, the circuit is executed on a trapped ion quantum computer. The probability distribution of measurement is compared on a classical computer against the BAS target data set. Next, the quantified difference is used to optimize the parametrized circuit. This learning process is iterated until convergence.	98
6.2	Connectivity graphs and corresponding training circuits. Top: Fully-connected training circuit layer, with layers of rotations (square boxes) and entanglement gates (rounded boxes) between any pair of the four qubits. Bottom: Star-connectivity training circuit layer, with restricted entangling gates. In either case, each rotation (denoted by X or Z) and each entanglement gate (denoted by XX) includes a distinct control parameter, for a total of 18 parameters for the fully-connected circuit layer and 15 parameters for the star-connected circuit layer. We remove the first Z rotation (dashed square box) acting on the initial state $ 0\rangle$, resulting in 14 and 11 parameters, respectively. The connectivity figures on the left define the mapping between the four qubits and the pixels of the BAS images (see Fig.6.1).	99
6.3	Quantum circuit training results with Particle Swarm optimization (PSO), with simulations (orange) and trapped ion quantum computer results (blue). Column (a) corresponds to a circuit with one layer of single qubit rotations (square boxes) and one layer of entanglement gates (rounded boxes) of all-to-all connectivity. The circuit converges well to produce the bars-and-stripes (BAS) distribution. Columns (b) and (c) correspond to a circuit with two and four layers and star-connectivity, respectively. In (b), the simulation shows imperfect convergence with two extra state components (6 and 9), due to the limited connectivity, and the experimental results follow the simulation. In (c), the simulation shows convergence to the BAS distribution, but the experiment fails to converge despite performing 1,400 quantum circuits. The optimization is sensitive to the choice of initialization seeds. To illustrate the convergence behavior, the shaded regions span the 5th-95th percentile range of random seeds (500 for (a) and (b), 1000 for (c), and the orange curve shows the median. The two-layer circuits have 14 and 11 parameters for (a) all-to-all- and (b) star-connectivity, while the (c) star-connectivity circuit with four layers has 26 parameters. The number of PSO particles used is twice the number of parameters, and each training sample is repeated 5000 times. Including circuit compilation, controller-upload time, and classical PSO optimization, each circuit instance takes about 1 min to be processed, in addition to periodic interruptions for the re-calibration of gates.	108

6.4	Quantum circuit training results with Bayesian optimization (BO), with simulations (orange) and trapped ion quantum computer results (blue). Column (a) corresponds to a circuit with two layers of gates and all-to-all connectivity. Columns (b) and (c) correspond to a circuit with two and four layers and star-connectivity, respectively. Convergence is much faster than with PSO (Fig. 6.3). Unlike the PSO results, the four-layer star-connected circuit in (c) is trained successfully, and no prior knowledge enters BO process. As before, the two-layer circuits have 14 and 11 parameters for (a) all-to-all- and (b) star-connectivity, while the (c) star-connectivity circuit with four layers has 26 parameters. We use a batch of 5 circuits per iteration, and each training sample is repeated 5000 times. Including circuit compilation, controller-upload time, and BO classical optimization, each circuit instance takes 2-5 minutes, depending on the amount of accumulated data.	109
6.5	Hybrid quantum-classical optimization with trapped ion qubits to prepare thermal states. The initial Bell-pairs denoted by ribbons connecting qubits 1-4, 2-5, and 3-6 (labeled 1-6 from top to bottom), correspond to the thermofield double state at infinite temperature. Layers of unitaries with independent control parameters are then applied sequentially to cool to the target temperature. The subsystem consisting of the first three qubits is effectively in the thermal (Gibbs) state. The result can be fed into a classical computer which updates the parameters based on a cost-function in a closed loop (see “Full Hybrid Optimization: Preparation of Ground State of TFIM” for details).	115
6.6	Preparation of TFD states of the quantum critical TFIM using two 3-qubit systems. Top row: Comparison between observables of the simulated optimized ansatz circuit and target TFD states (solid lines) for various target temperatures. Bottom row: Comparison between observables of experimentally prepared and target TFD states. Results for all three ion pairs are given at each temperature. The measured correlation functions for different target temperatures are plotted against the theoretical expectations (solid lines) for type (a) Pauli-X (b) Pauli-Y and (c) Pauli-Z. Intra-system correlators in the subsystem-A are: $\langle\sigma_{1,A}\sigma_{2,A}\rangle$, $\langle\sigma_{1,A}\sigma_{3,A}\rangle$, and $\langle\sigma_{2,A}\sigma_{3,A}\rangle$. Cross-system correlators are $\langle\sigma_{1,A}\sigma_{1,B}\rangle$, $\langle\sigma_{2,A}\sigma_{2,B}\rangle$, and $\langle\sigma_{3,A}\sigma_{3,B}\rangle$. Note the experimental data points in the figure have errorbars accounting for statistical errors. Statistical error bars are similar in size or smaller than the symbols used. A symmetry based error mitigation technique is used to post-process the experimental result in (c). The mitigation notably improved the agreement between experiment and theory. Details are given in the SI Appendix and Fig. S2.	118

6.7	Critical TFIM ground state on a 7-qubit system. Top row: Two-point correlations for (a) Pauli-X and (b) Pauli-Z operators as a function of their separation. For a ring of seven spins, there are only three different pairs of ions, which are distinguishable by distance. The three different colors correspond to QAOA protocols with different depth p . The lines denote the theoretical expectations. Bottom row: Energies achieved using full hybrid quantum-classical feedback with increasing gradient descent iteration number for (c) $p = 1$, initialized with random parameter set, and (d) $p = 2$, initialized with theoretically optimal parameters. The line corresponds to the measured energy at each iteration, and the dots correspond to samples taken to evaluate the gradients. Ideally, the lowest energy a $p=1$ protocol can reach is -8.44 . The lowest energy a $p=2$ protocol can reach is -8.62 . The true ground state energy is -8.98 , and the size of the gap is 0.23 . The gap decreases linearly with system size. Statistical error bars in the above figures are of the same size or smaller than the symbols used.	121
6.8	Simulation results with noise for the $p = 1, 2, 3$ QAOA protocols for preparation of the critical ground state of the TFIM. (a) $\lambda = 0$ (no depolarizing noise); (b) $\lambda = 0.1$; (c) $\lambda = 0.22$. Each curve is averaged over 1000 samples. The circles in the figure show what Γ value an experimental result (shown in Fig. 3) predicts for a given set of p and λ . $\lambda = 0.22$ is the point at which we can minimize the variance of the predicted Γ . Note that (a) shows a threshold at $\Gamma = 0.13$ below which higher p give better results.	123
6.9	(a): Comparison between results with and without symmetry based error mitigation. The dots with same hue correspond to the same type of measurement, while dots with brighter color correspond to data corrected with error mitigation.(b): The fraction of data points kept after the symmetry based post-selection.	126
7.1	(a) protocol with implemented with real-time-interaction (b) protocol implemented without real-time-interaction	132
7.2	(a):The flow chart of the interactive protocol based on LWE. The green boxes stands for verifier's actions. The blue boxes stands for prover's actions. The arrows stands for the exchange of information. (b):The circuit diagram of quantum operations the prover would need to perform to answer the verifier's challenge. (c):The experimental results of all the branches of circuits, in relative performance (as defined in eq.7.14). The configurations are arranged in ascending order of two-qubit gates involved in the implementation.The boxes marked with the same index correspond to the same operations. Note if real-time interaction is implemented, the measurement of the output register will be moved to between box 2 and box 4.	135
7.3	Visualization of States in XY Plane of Bloch Sphere	140

7.4	(a):The flow chart of the interactive protocol based on $x^2 \bmod N$. The green boxes stands for verifier's actions. The blue boxes stands for prover's actions. The arrows stands for the exchange of information. (b):The circuit diagram of quantum operations the prover would need to perform to answer the verifier's challenge. (c):The experimental results of all the branches of circuits, in relative performance (as defined in eq.7.14). The boxes marked with the same index correspond to the same operations.	141
7.5	detailed circuits	145
7.6	quantum advantage plot	149

List of Abbreviations

AOM	Acousto-Optic Modulator
EOM	Electro-Optic Modulator
RF	Radio Frequency
NISQ	Noisy Intermediate-Scale Quantum
CNOT	Controlled-Not
RSB	Red Sideband
MS	Mølmer-Sørensen
PMT	Photo-Multiplier Tube
GHZ	Greenberger–Horne–Zeilinger
CW	Continuous Wave
FPGA	Field-Programmable Gate Array
AMO	Atomic, Molecular and Optical
AC	Alternating Current
DC	Direct Current

Chapter 1: Introduction

Classical computing has been one of the most crucial tools in the history of science and technology. It fully realized humanity's potential unlocked by the discovery of classical mechanics and calculus. Classical computing is crucial to many of humanity's most outstanding achievements, like sending astronauts to the moon. However, when we want to use classical computing to address quantum problems, our steps stagger quickly because of the exponentially growing demands of resources.

In a classical system, every component has a definite state. Thus, the resources needed to represent the system grow linearly with the size. However, in a quantum system, every component can be in a superposition state. Thus the resources needed to represent all the superpositions grow exponentially with size.

This exponential scaling of resources motivates us to explore alternatives. In a classical computer, each bit has a definite value of 0 or 1 because that is how a classical object should be: in a definite state. Alternatively, we can let each bit represent the superposition of 0 and 1 simultaneously, just like quantum systems. We call such a bit, capable of being 0 and 1 simultaneously, a qubit. In quantum computing, classical bits are replaced with qubits.

Quantum computing also comes with a different set of operations that are native to quantum systems. These native quantum operations are much more efficient in handling quantum problems.

Is quantum computing only suitable for handling quantum problems? It is a subject actively investigated by a vast community of researchers. A conclusion has yet to be drawn. Shor's factoring algorithm is currently the only quantum computer application that handles a classical problem exponentially more effectively than all the known classical algorithms. However, the existence of an algorithm capable of doing the same on a classical computer is possible. With ongoing research in quantum computing, the number of possible classical applications on quantum computers will grow.

1.1 Quantum Computing

Conceptually, there are two elementary building blocks for any computation scheme: a basic unit to represent states and a universal gate set to perform any necessary operation. For classical computing, classical bits are used to represent states. As for the universal gate set, either NAND or NOR by itself is sufficient. The situation is more complicated for quantum computing. A qubit, unlike its classical counterpart, can be simultaneously in 0 and 1. This, known as superposition, can be written in the wave function form:

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle \quad (1.1)$$

But if we measure it, we will get either 0 or 1, depending on which one the wave function collapses to. The probability of getting 1 is given as:

$$P(1) = |\langle 1|\psi\rangle|^2 = \left| e^{i\phi} \sin\frac{\theta}{2} \right|^2 = \sin^2\frac{\theta}{2}. \quad (1.2)$$

The relative phase $e^{i\phi}$ is one of the most crucial concepts in quantum computing. Though it is irrelevant in the above measurement, we will see its significance throughout the later part of this thesis.

Eq.1.1 suggests that the arguments of a single-qubit wave function can be associated with geometric angles. Because of that, people usually find it convenient to visualize the state of a single qubit with the help of the Bloch sphere, as shown in Fig.1.1

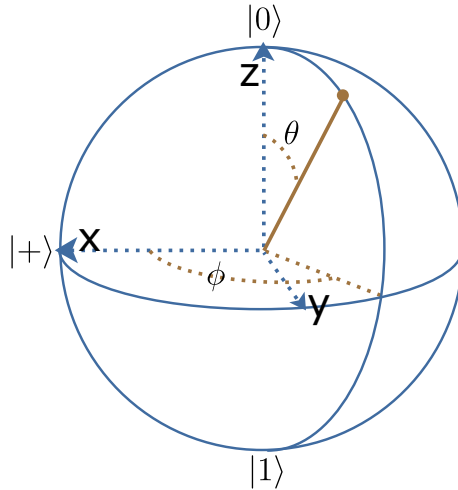


Figure 1.1: The Bloch state representation of a quantum state. The axes are normally defined in the following way: $|0\rangle$ is along the positive Z-direction. $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ is along the positive X-direction

Any single-qubit state described by Eq.1.1 can be mapped to a point on this unit sphere. The Bloch sphere picture provides us with an intuitive representation

of single-qubit quantum operations. Since the states are just points on the Bloch sphere, naturally, we need a set of operations capable of moving any two points on the Bloch sphere into each other. Arbitrary rotations along the Z-axis and the X-axis of the Bloch sphere are a set of such operations. These two types of rotations are the universal set of single-qubit rotations used on ion trap quantum computers, for reasons that will be clear in the next chapter. But mathematically, universal gate sets with as few elements as possible are preferred choices. One example is the set made of the Hadamard gate, T gate, and S gate, as visualized in Fig.1.2: The Hadamard gate is an π rotation along the axis halfway between the x-axis and the z-axis. The S gate is a π rotation along the z-axis. The T gate is a $\pi/2$ rotation along the z-axis. When these gates are combined appropriately, they can efficiently achieve arbitrary rotation, with desired precision, according to the Solovay–Kitaev theorem [2].

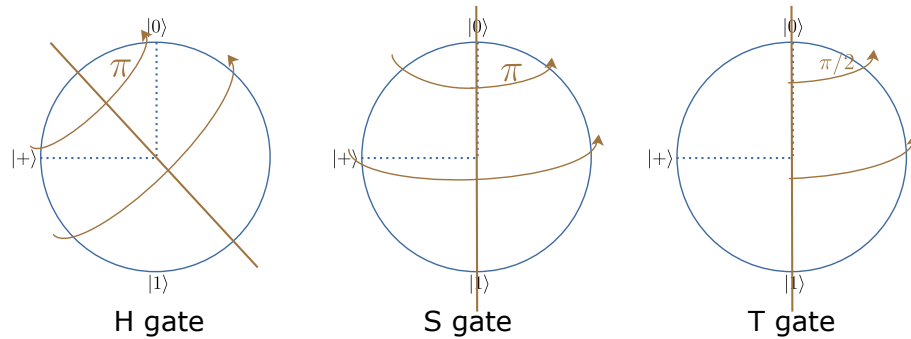


Figure 1.2: Illustration of some of the single-qubit gates: A Hadamard gate is a rotation round the axis halfway between $|+\rangle$ and $|0\rangle$ by π . A S gate is a rotation around the Z-axis by π . A T gate is a rotation around the Z-axis by $\pi/2$

Another advantage of using the Bloch sphere representation is that measurement can be understood intuitively. As described by Eq.1.2, the probability of getting 1 or 0 in each measurement is related to the projection of the state on the

Z-axis. If the state sits on the north pole, the measurement will deterministically yield 0. If the state sits on the south pole, the measurement will yield 1, deterministically. Although the definition of the orientation of the Bloch sphere is arbitrary, we usually align the Z-axis along with the native measurement basis of the physical qubit. We call this measurement basis the computational basis. With the arbitrary single-qubit rotations mentioned above, one can effectively rotate the Bloch sphere and measure in any basis.

Things get exciting when we put multiple qubits together. As mentioned before, because of quantum superposition, the number of resources needed to represent a quantum state grows exponentially with the number of qubits. Let us take a simple three-qubit system as an example:

$$\begin{aligned}
|\psi\rangle &= (a|0\rangle + b|1\rangle)(c|0\rangle + d|1\rangle)(e|0\rangle + f|1\rangle) \\
&= ace|000\rangle + acf|001\rangle + ade|010\rangle + adf|011\rangle + bce|100\rangle + bcf|101\rangle + bde|110\rangle + bdf|111\rangle
\end{aligned}
\tag{1.3}$$

Above is the wave function of three qubits putting together. There are several issues with the above expression. First, we see that the number of arguments needed to represent the states is $6 = 2 * 3$, not $8 = 2^3$. The number of arguments does not grow exponentially, as needed to represent arbitrary states. Second, although we are limited to physical measurement in the computational basis, we can use single-qubit rotations to measure in any basis effectively. If the wave function is written

as Eq.1.3, we can measure each qubit in a rotated basis that the state sits precisely on the north or south pole. If we do so, the measurement will deterministically yield 0 or 1. With only Eq.1.3-like state, quantum computing is reduced to classical computing. What is missing above is Entanglement. Entanglement is equivalent to nontrivial information of the correlation between qubits that cannot be derived from other information. Take the case above as an example. Say the nontrivial information is: the sum of all the qubits' value is an odd number. Then we will have $ace = adf = bcf = bde = 0$. We can verify that all the possible solutions of a , b , and c will render this information trivial. Because all the possible solutions look like $|\psi\rangle = |z_1, z_2, z_3\rangle$, where z_1 , z_2 , and z_3 all have a definite value of either 0 or 1. If we know each qubit's exact value, we know the parity (of the sum) of the qubits trivially.

A state that bears entanglement (entanglement state) simply cannot be written as $|\psi\rangle = (a|0\rangle + b|1\rangle)(c|0\rangle + d|1\rangle)(e|0\rangle + f|1\rangle)$, which people refer to as a product state. One of the states that bear the information "the sum of the value of all the qubits is an odd number" in a nontrivial way is:

$$|\psi\rangle = a|001\rangle + b|010\rangle + c|100\rangle + d|111\rangle. \quad (1.4)$$

The linear space spanned by all the possible product states and entangled states is called the Hilbert space[3]. The Hilbert space scales exponentially with the number of qubits, while the number of product states scales linearly. This tells us that most of the states in a Hilbert space are entangled space. To this point,

it should be clear that entanglement is what sets quantum computing apart from classical computing. All the quantum algorithms use entanglement as a critical element.

With the single-qubit operation introduced above, a state initialized in a product state will continue to be a product state. We need a set of operations that enable us to explore all the possible entangled states in the Hilbert space. A universal single-qubit rotation mentioned above, together with any two-qubit gate that can generate entanglement, suffice as such a set[2]. The most popular choice for such an entanglement gate is the CNOT gate. Fig.1.3 shows the symbol and truth table of the CNOT gate. For ion trap systems, we usually use XX gates, as shown in Fig.1.3, since the physics of our system natively grants us such interaction.

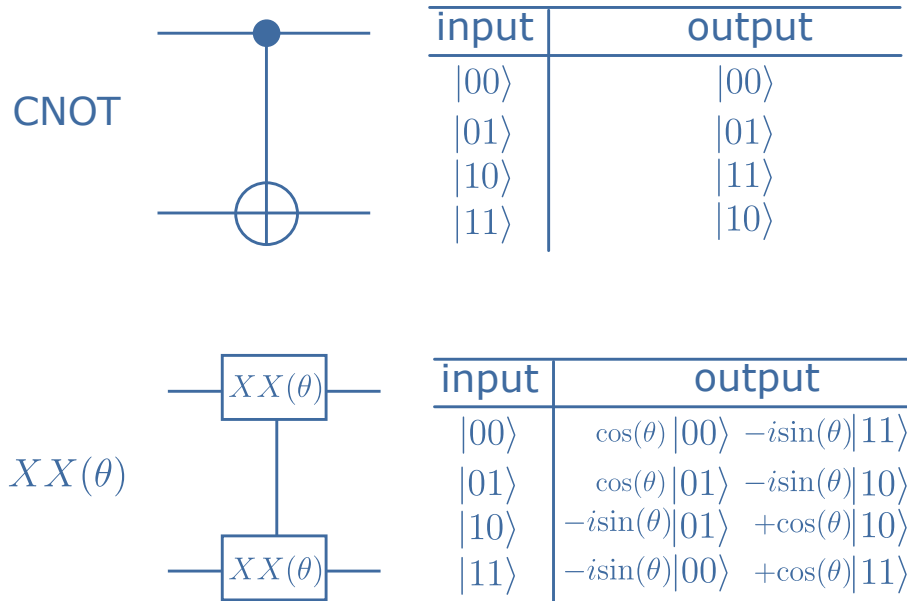


Figure 1.3: symbol and truth table for control-not (CNOT) gates and XX gates

We call a sequence of applications of single-qubit rotations and two-qubit gates a quantum circuit. With circuits built upon a universal quantum gate set, the whole

Hilbert space is now within our reach.

Note all the quantum operations we mentioned above are unitary operations. So are quantum circuits. They are invertible. There is no information loss nor gain through quantum circuits until the measurement, which is not unitary (invertible). This is different from classical computing. For example, the classical "and" operation is not unitary. Having a y does not grant you the knowledge to retrieve the two x_1 and x_2 that give you $y = x_1 \text{ and } x_2$.

The unitary property of quantum circuits is a profound topic. Nevertheless, the intuition is: quantum algorithms do not create new information. It only moves information around the Hilbert space. This is how most quantum algorithms work in a nutshell: 1. encode all the information needed to answer the question into qubits (as superposition) 2. use unitary operations to move this information around the Hilbert space so measurements can extract the useful information and drop the unwanted information.

1.2 Noisy Quantum Computers

Where are we in terms of realizing all the promises of quantum computing? A popular answer frequently quoted by scientists to address this question is the "Noisy Intermediate-Scale Quantum (NISQ)" era. [4] The name summarizes the situation well. Noisy, because we only have imperfect qubits and gate operations. Intermediate-Scale, because we are only able to work with less than or around 50 qubits.

If Tolstoy tried to build quantum computers, we might know his famous quote as "good qubits are all alike; every bad qubit is bad in its own way." Errors can occur in many different ways. To interact with qubits, we need channels to pass information. But information stored in qubits also leaks through such channels. This argument interestingly relates to two ways of looking at errors on the qubits. Errors occur if a qubit loses its information or we lose track of the qubit. If a qubit is easy to interact with, it tends to lose information (decohere). This tends to be the case for superconducting qubits. If a qubit is well isolated, we tend to make mistakes interacting with it. This tends to be the case for ion-trap quantum computers. At the end of this spectrum, there are the topological qubits. They are intrinsically robust to error [5], but too challenging to yet be realized.

Classical computers are not free from this dilemma. All classical devices leak information slowly. In fact, without error correction, current computers would not function properly. It is possible to adapt the concept of error correction for quantum computing. Since Peter Shor proposed the first quantum error correction scheme[6], many different proposals have been made. Quantum error correction is the solution to the intrinsic fragility of qubits. All error correction schemes have thresholds for the operations needed. In the NISQ era, our goal is to improve all the needed operations above the error correction thresholds.

Scaling up is another challenge to be overcome. We need to let two qubits separated apart by hundreds or thousands of other qubits effectively interact with each other. Proposals like the CCD-like scheme[7] and fiber-linked modules[7] have been proposed. It might be tempting to say that shoveling tens more qubits into

what has already been built is not that hard. But holding more qubits makes them harder to interact with, thus limiting the number of operations that could be performed before too much information is lost.

Quantum computing will be used for actual application in the NISQ era. Admittedly, with a NISQ quantum computer, we will not be able to implement a scaled-up Shor’s algorithm [8] to breach RSA encryption[9]. However, as pointed out in [4], the difficulty of using a classical computer to simulate an imperfect intermediate-scale quantum computer is the best evidence of its value. Applications we can do with a noisy intermediate-scale quantum computer are ultimately the focus of my Ph.D. works.

1.3 Thesis Outline

I want to draw a full-stack picture of quantum computing in this thesis, centered at the interface of hardware and quantum algorithms. The discussions in this thesis are based on my works on three different ion-trap quantum computers at the University of Maryland.

Chapter 2 is focused on the physics of ion-trap quantum computers. It explains the concepts behind the building blocks of ion-trap quantum computers. Minimum technical details are included here. Readers interested in such details should refer to theses by other alumnae of the Monroe Group.[10, 11]

Chapter 3 discusses the automation of an ion-trap quantum computer. It involves enough technical details to replicate the automation scheme.

Chapter 4 introduces the fundamental algorithm compiling techniques needed for a "quantum computer programmer" working at the interface of software and hardware.

Chapter 5 discusses a technique for circuit-based digital quantum simulation. A work based on such a technique is presented. The presentation is modified from [\[12\]](#).

Chapter 6 discusses the application of hybrid algorithms on an ion-trap quantum computer. Two applications are presented. The presentations are modified from [\[13, 14\]](#).

Chapter 7 discusses the techniques for verification of quantum computing. Two applications are presented. The presentations are modified from manuscripts not yet published at the time of writing.

Chapter 2: Ion Trap Quantum Computer: The Recipe

Just knowing the recipe is not enough. Don't serve it to your date on the first attempt.

2.1 Trap the ion

To make an ion-trap quantum computer, first and foremost, we need to hunt for ions.

We seek to trap one of the simplest ions made of a single alkali earth atom with one electron stripped away. Thus it becomes positively charged and feels the Coulomb force in the electric field. However, trapping ions with just static Coulomb force is against the fundamental law of physics. In Maxwell equations, we have $\nabla \cdot \vec{E} = 0$. Equivalently, in any point in a charge-free space, if the Coulomb force is pushing in from some direction, it must be pushing out to some other directions. One way to overcome this is to use varying electric fields so that the ions feel a trapping force in all directions on average. To illustrate this point, we shall use a simple four-rod trap as an example, shown in Fig.2.1.

The two rods connected by the RF-source sets up an oscillating quadruple potential in the radio (XY) direction. The potential can be written as:

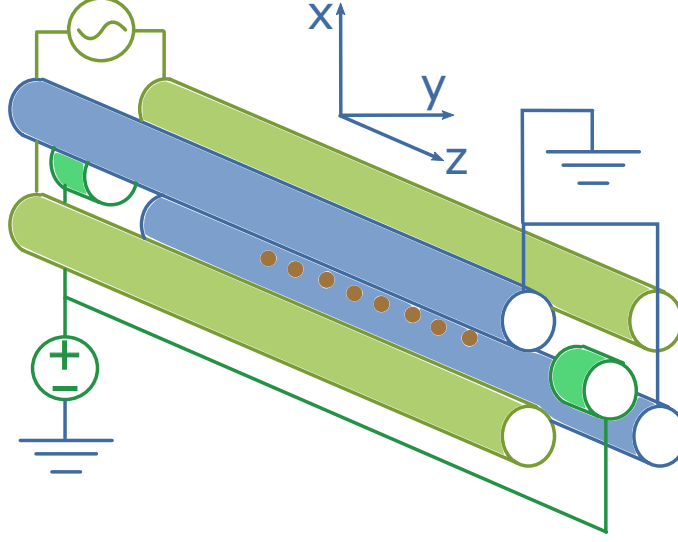


Figure 2.1: The simplified diagram of an RF Paul-trap. The two blue electrodes are connected to the ground. The RF potential is applied to the two olive electrodes to provide trapping force in the radial direction. The two end-caps in lime color provide DC confinement along the axial direction

$$\Phi_{RF}(x, y) = \frac{V_{RF}}{2} \cos(\Omega_{RF}t) \left(1 + \frac{x^2 - y^2}{R^2}\right). \quad (2.1)$$

In this equation, V_{RF} is the amplitude of the RF potential. Ω_{RF} is the driving frequency of the RF field. R is the distance from $z = 0$ (set at the radial center) axis to the rods connected to the RF source. Different geometries of the trap electrodes generally give different R -values.

To confine the ions along the axial (z) direction, we also need to apply voltage to the two end-cap electrodes. This potential generated by these two end caps can be written as

$$\Phi_z(x, y, z) = kV_{DC} \left(z^2 - \frac{x^2 + y^2}{2}\right). \quad (2.2)$$

Here, k is a geometry factor that varies depends on the geometry of the trap elec-

trodes. We can add these two potential together to obtain the total potential

$$\Phi(x, y, z) = \frac{V_{RF}}{2} \cos(\Omega_{RF}t) \left(1 + \frac{x^2 - y^2}{R^2}\right) + kV_{DC} \left(z^2 - \frac{x^2 + y^2}{2}\right). \quad (2.3)$$

Given the potential, we can calculate the Coulomb force applied to the ions:

$$F = -q \nabla \Phi(x, y, z) \quad (2.4)$$

$$= -qV_{RF} \cos(\Omega_{RF}t) \left(\frac{x\hat{x} - y\hat{y}}{R^2}\right) + qkV_{DC}(x\hat{x} + y\hat{y} - 2z\hat{z}). \quad (2.5)$$

Here, q is the charge of the ion. \hat{x} , \hat{y} , and \hat{z} are unit vectors along x , y , and z direction respectively. Using $\frac{\vec{F}}{m} = \ddot{x}\hat{x} + \ddot{y}\hat{y} + \ddot{z}\hat{z}$. We can derive the equation of motion:

$$\ddot{x} + \left[\frac{qV_{RF}}{mR^2\Omega_{RF}^2} \cos(\Omega_{RF}t) - \frac{qkV_{DC}}{m\Omega_{RF}^2}\right] \Omega_{RF}^2 x = 0 \quad (2.6)$$

$$\ddot{y} + \left[-\frac{qV_{RF}}{mR^2\Omega_{RF}^2} \cos(\Omega_{RF}t) - \frac{qkV_{DC}}{m\Omega_{RF}^2}\right] \Omega_{RF}^2 y = 0 \quad (2.7)$$

$$\ddot{z} + \left[\frac{2qkV_{DC}}{m}\right] z = 0 \quad (2.8)$$

The equation of z describes a particle in harmonic well. Thus the ion is trapped in the Z -direction. The equation in the X - and Y - directions can be written in the form of the Mathieu equation. A detailed mathematical analysis can found in [15]. We shall discuss the physical intuition with a non-rigorous approach to solve the

Mathieu equations in the following.

The equation in the X- and Y- directions are very similar. The only difference is that the oscillating term is has a π phase difference. Let us focus on the x-equation and write it as

$$\ddot{x} + [2A\cos(\Omega_{RF}t) + B]\frac{\Omega_{RF}^2}{4}x = 0 \quad (2.9)$$

With $A = \frac{2qV_{RF}}{mR^2\Omega_{RF}^2}$ and $B = -\frac{4qkV_{DC}}{m\Omega_{RF}^2}$. When building the actual physical devices, we would make sure $|A| \ll 1$ and $|B| \ll 1$. We are applying a periodic drive (with $|A| \ll 1$) of frequency Ω_{RF} to a system that has a much slower dynamic $\Omega = B\Omega_{RF} \ll \Omega_{RF}$. However, since B is a negative number, the dynamic of the unperturbed system is unstable. So this is the cool question: how does a periodic drive turn an unstable system into a stable system? We can solve this by separating the two utterly different time scales, where we assume the solutions have the form: $X(t) + \mu(t)$. Here, $X(t)$ is a slower motion with a larger amplitude. $\mu(t)$ is a much faster motion with a much smaller amplitude. We normally call $X(t)$ the secular motion, $\mu(t)$ the micro-motion. Let's first solve for $\mu(t)$. Because $X(t)$ is much slower, we can treat it as a constant. The equation became:

$$\ddot{\mu} + 2A\cos(\Omega_{RF}t)\frac{\Omega_{RF}^2}{4}X = 0 \quad (2.10)$$

Note that we dumped B in Eq.2.10 since it will give a slower dynamic of large amplitude, which should be included in $X(t)$. Solving this gives us the equation of

the micro-motion:

$$x(t) = \frac{AX}{2} \cos(\Omega_{RF}t) \quad (2.11)$$

Plugging this into Eq.2.9, we have:

$$\ddot{X} - \frac{AX\Omega_{RF}^2}{2} \cos(\Omega_{RF}t) + [2A\cos(\Omega_{RF}t) + B] \frac{\Omega_{RF}^2}{4} [X + \frac{AX}{2} \cos(\Omega_{RF}t)] = 0 \quad (2.12)$$

Since we are solving for the slower dynamics here, we take the time average of every term. This gives us:

$$\ddot{X} + \left(\frac{1}{2}A^2 + B\right) \frac{\Omega_{RF}^2}{4} X = 0 \quad (2.13)$$

Now you see if $\frac{A^2}{2} > B$ you will have a equation for harmonic motion of frequency $\frac{1}{2}\Omega_{RF}\sqrt{B + \frac{1}{2}A^2}$. We call this average potential described in Eq.2.13 the pseudo-potential, which lead to the secular motion.

Figure 2.2 presents an intuitive illustration for the pseudo trapping potential arising from a fast-varying RF drive. The micro-motion of the ion described by Eq.2.11 is in phase with the RF-drive. As shown in Fig.2.2, ions are always further away from the equilibrium when the RF is in the "trapping" phase. As a result, although the RF potential oscillates between "trapping" and "escaping", the ion feels more "attraction" on average.

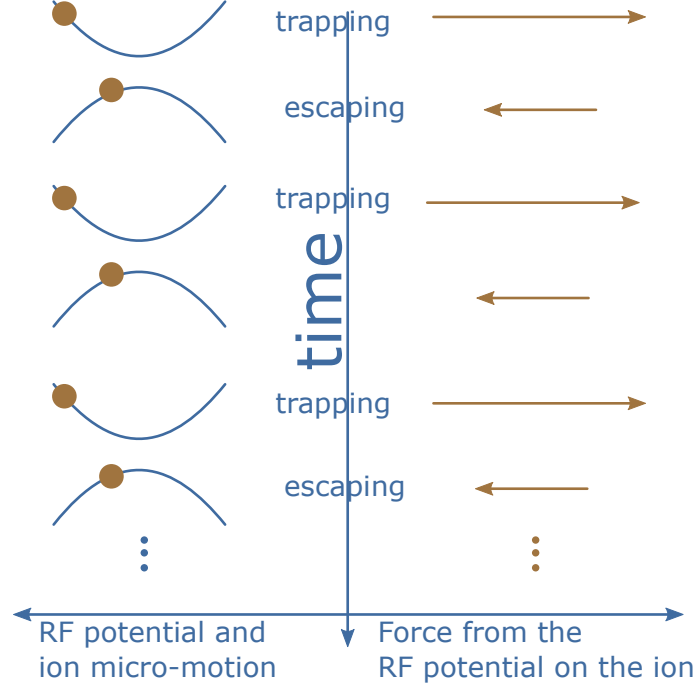


Figure 2.2: How the pseudo-potential is formed. The RF potential induces an in-phase micro-motion. Because of the micro-motion, the attraction is stronger than the repulsion from the RF. So on average, the ion feels an attraction in the RF potential.

Since the difference between the equation of motion in the X-direction (Eq.2.6) and that in the Y-direction (Eq.2.7) is only the sign of A , the solution of Eq.2.7 has the exact same form. Experimentally, we usually set the frequency of RF-drive Ω_{RF} to approximately 30 MHz. The secular frequency ω is usually around 2 MHz.

2.2 Ionization

With the trapping potential ready, we need to get the ions. Though there are many options of ions for the trapped-ion experiments, I exclusively focus on $^{171}\text{Yb}^+$ during my Ph. D. Acquiring ions starting with getting neutral atom vapor in the vacuum. The neutral atoms do not interact with the electric potential when they approach the trapping zone. But when they are in the trapping zone, a set

of ionization lasers will ionize them into $^{171}\text{Yb}^+$ ions. Now the positively charged $^{171}\text{Yb}^+$ will be confined in the trapping zone by the electric potential mentioned above. The two-step ionization scheme we use is illustrated in Fig.2.3.

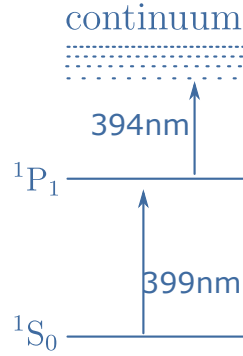


Figure 2.3: Energy levels used to ionize Yb atoms.

We first use a 399nm continuous wave (CW) laser to excite the Yb atom from the $1S_0$ ground state to the $1P_1$ excited state. Then absorbing a 394nm photon, the atom is further excited from the $1P_1$ state to the continuum. Worth mentioning, we can use any light with wavelengths smaller than 394nm for ionization from the $1P_1$ state. We choose to use 399nm laser particularly just for convenience since it is also engaged in other to-be-mentioned purposes.

2.3 Doppler Cooling

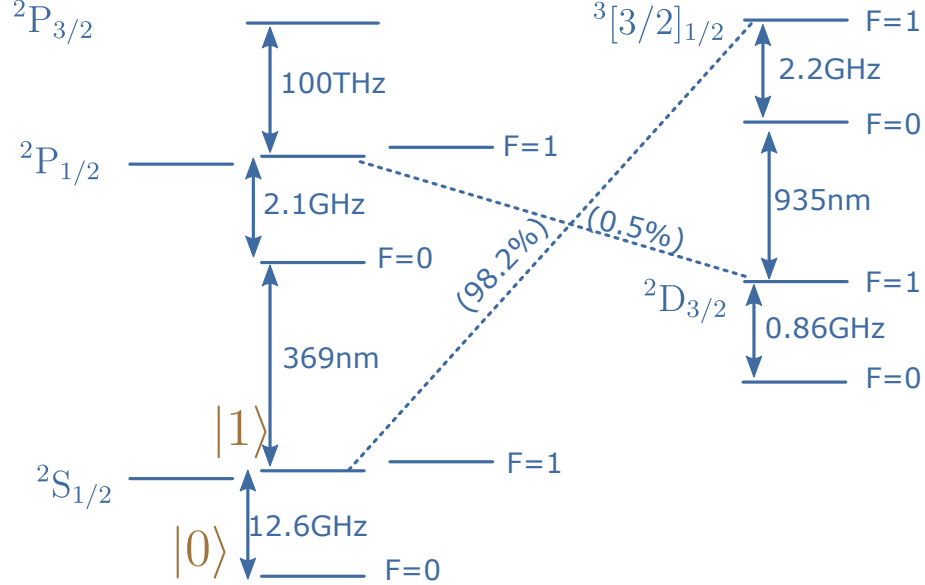


Figure 2.4: The energy levels of $^{171}\text{Yb}^+$ relevant to the operations needed for our quantum computing scheme.

With ions trapped, we have to further cool them before performing any useful operations. We shall first introduce the diagram of the relevant energy levels of $^{171}\text{Yb}^+$ shown in Fig.2.4. We use the $|F=0\rangle$ and $|F=1\rangle$ states of the $^2\text{S}_{1/2}$ manifold as the qubit- $|0\rangle$ and qubit- $|1\rangle$. We choose this pair of internal states of $^{171}\text{Yb}^+$ to be our qubit states for many reasons. First, they are long-lived, magnetic field insensitive states. Second, as we will go through next, via those connected levels shown in Fig.2.4, we can cool the ion, initialize the qubit into $|0\rangle$, and read out the qubit state effectively. Third, all the operations are mostly close-looped, with the only leakages easily recovered by a re-pump laser.

In chapter 1, we learned that qubits are very fragile. When using lasers to manipulate the qubits, we need the laser's frequency and phase to be stable. Fast-

moving ions experience fast-moving lasers. Fast moving lasers can hardly be stable lasers. As a result, our control of the qubits will be stochastic. Besides, the pseudo-potential is conservative, meaning it cannot dissipate the energy of trapped particles. Instead, many things like collision, and RF heating could happen to heat the ion. If we do not efficiently cool the ions down, they will escape our trapping potential. Practically, the atomic vapor is hot enough that newly ionized $^{171}\text{Yb}^+$ will most likely escape right away without efficient cooling.

We implement Doppler cooling for the considerations mentioned above. The intuition behind Doppler cooling is not complicated. To slow an object down, we need to transfer momentum to it against the direction of its velocity. We can excite the internal state of an ion, and as a result, the ion will absorb the photon together with its momentum. If we can convince the ion to absorb more photons moving oppositely to itself, our goal will be achieved. Our trick is the Doppler effect. When an ion is moving against the laser, it will see the photons as blue-shifted, vice versa, as shown in Fig.2.5. Thus when if we park the laser frequency at the red-detune side, as shown in fig2.5, the absorption cross-section of the photons moving into the ions will be larger than that of the photons moving away from the ions. The bigger the difference between the absorption cross-sections of the two cases, the more photon with opposite momentum will the ions absorb. To optimize the Doppler cooling efficiency, we usually park the laser frequency at the waist of the red-detuned side of the absorption spectrum, as where the "stationary" frequency is parked at, in Fig.2.5. Please refer to [16] for a detailed analysis of the Doppler cooling.

We drive all the physically allowed transitions between the $^2\text{S}_{1/2}$ manifold and

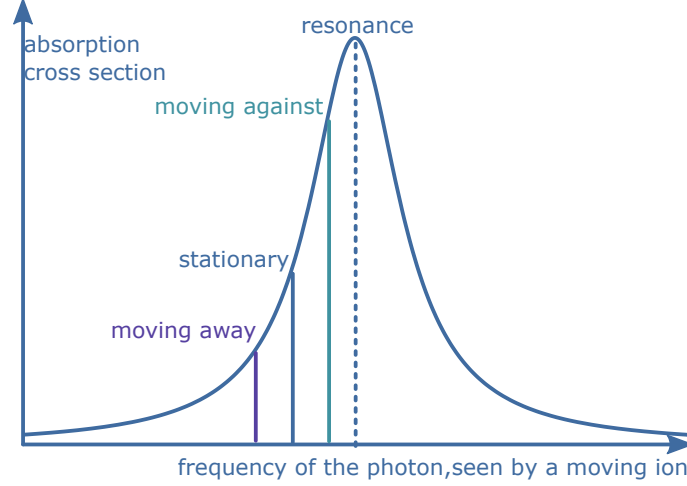


Figure 2.5: To Doppler cool ions, we park the laser's frequency at the red-side waist of the absorption spectrum. This is the frequency seen by a stationary ion. An ion moving into the laser will see a blue-shifted laser, which is closer to resonance. Ions moving away from the laser see a red-shifted laser, which is less in resonance. This asymmetry in the absorption cross-section gives ions a higher chance of absorbing counter-moving photons. As a result, the ions lose velocity and cool down eventually.

the $^2P_{1/2}$ manifold to Doppler cool the ions. This choice is very popular for the following considerations. These are strong transitions that can be directly driven by a single laser, with sidebands generated by electro-optical modulators (EOM). On the other hand, the transitions form an almost closed loop. With the only exception being the 0.5% change of decaying from the $|^2P_{1/2}, F = 1\rangle$ state to the $|^2D_{3/2}, F = 1\rangle$ state (see Fig. 2.4). But this is easily recovered by re-pumping to the $|^3[3/2]_{1/2}, F = 1\rangle$ state, which will then decay back to the cooling loop.

With the intuition mentioned above, one can efficiently cool the ions' motion down to several quanta (of the secular motion). This temperature limit can be estimated assuming the photon recoil (from spontaneous decay) is what keeps the rest of phonons in the secular motion: $\bar{n}\hbar\omega = \hbar\nu$. Here, \bar{n} is the expectation of the phonon number. ω is the frequency of each phonon of the secular motion. ν is the

frequency of the cooling photon.

2.4 Optical Pumping

Now with ion sufficiently cooled, quietly sitting in our trap, it is time to wash away all the information stored in the qubit states, initialize them into $|0\rangle$. This initialization can be achieved using optical pumping.

The intuition behind optical pumping is simple but effective. First, we drive transitions to form a network of states, within which population in one state can be transferred to any other state. Next, we eliminate all the transitions that take any population out of a specific state. Now within this transition network, all the population will gradually get trapped in the transit-in-only state.

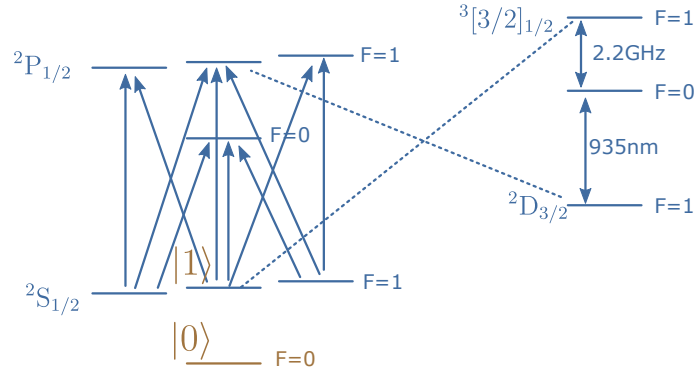


Figure 2.6: The energy levels and transitions used for the optical pumping. Because nothing leaves $|0\rangle$, it eventually accumulates all the population.

In practice, we drive transitions to form a network that includes all the states within the $^2S_{1/2}$ manifold and the $^2P_{1/2}$ manifold. Of course, as shown in Fig.2.6, we still have to use the 935nm laser to recover the leakage population. There is no laser exciting population out of $|^2S_{1/2}, F=0\rangle$. But population decays into $|^2S_{1/2}, F=0\rangle$ from the $|^2P_{1/2}, F=1\rangle$ manifold. Our qubit $|0\rangle$ state will gradually collect all the

populations. With enough power laser power, after 5 μsec of optical pumping, the qubit state $|0\rangle$ can trap $> 99.7\%$ of the population, which is good enough for the experiment. In some cases, to avoid off-resonant excitation of $|0\rangle$, we would decrease the laser power and extend the pumping time to about 100 μsec .

2.5 Qubit Readout

Before discussing how to realize arbitrary single-qubit rotations and two-qubit gates, I will first introduce the qubit states detection for the ions. Assuming all the quantum operations are finished, we then need to measure the results. You want to know whether the wave function of the qubits will collapse to $|0\rangle$ or $|1\rangle$. Forbidden by symmetry, the ions cannot decay from $|^2P_{1/2}, F=0\rangle$ to $|^2S_{1/2}, F=0\rangle$ by emitting photons. But such process is allowed from $|^2P_{1/2}, F=0\rangle$ to $|^2S_{1/2}, F=1\rangle$. So all we need to do is driving all the transitions from the $|^2S_{1/2}, F=1\rangle$ manifold to the $|^2P_{1/2}, F=0\rangle$ state, as shown in Fig.2.7. If the ions are in $|1\rangle$, this process will scatter photons as the ions keep cycling through the transitions. Driving this transition, we can then tell whether the wave function collapsed into $|0\rangle$ or $|1\rangle$, according to the scattering of photons. We call this state-dependent fluorescence[17].

2.6 Single-qubit gates: Rabi-flopping

In ion-trap quantum computing, we can realize arbitrary single-qubit rotations via resonant Rabi-flopping. Before digging into the mathematical derivation of Rabi-flopping of a two-level system, let us take a small step back and discuss the intuition

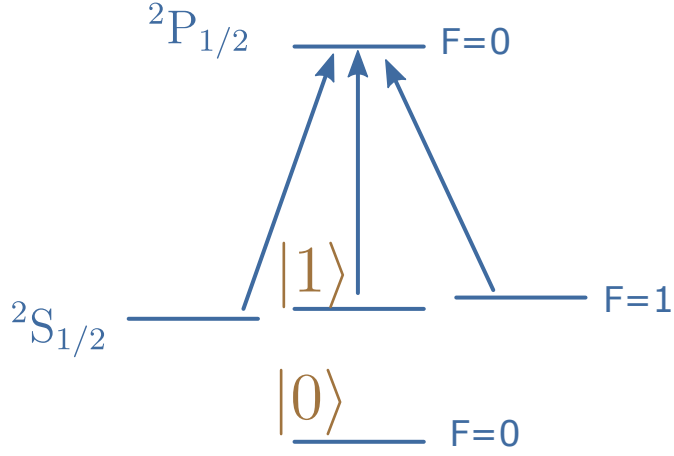


Figure 2.7: Energy levels and transitions used for the qubit-state readout. This is also referred to as the state-dependent fluorescence.

behind the math.

For simplicity, we set $\hbar = 1$ and assume our ions only have two internal states $|1\rangle$ and $|2\rangle$, for which the Hamiltonian has the form:

$$\hat{H}_0 = \begin{bmatrix} \omega_1 & 0 \\ 0 & \omega_2 \end{bmatrix} \quad (2.14)$$

Since this Hamiltonian is already diagonalized, the Schrodinger equation $i \frac{\partial}{\partial t} |\phi\rangle = \hat{H} |\phi\rangle$ has solution of the form:

$$|\phi_0(t)\rangle = c_1 e^{i\omega_1 t} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + c_2 e^{i\omega_2 t} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (2.15)$$

Or equivalently:

$$|\phi_0(t)\rangle = c_1 e^{i\omega_1 t} |1\rangle + c_2 e^{i\omega_2 t} |2\rangle \quad (2.16)$$

Now, we apply lasers to ions. Since the spatial spread of the ion's electron wave function is much smaller than the wavelength of our laser, we can apply dipole approximation to write the perturbation Hamiltonian from the laser as $V(t) = e\hat{\vec{r}} \cdot \vec{E}_0 \cos(\omega t + \theta)$. Here $e\hat{\vec{r}}$ is the dipole operator for the ion. We can write this into Matrix form:

$$\hat{V} = \begin{bmatrix} 0 & \Omega \cos(\omega t + \theta) \\ \Omega^\dagger \cos(\omega t + \theta) & 0 \end{bmatrix} \quad (2.17)$$

Here, $\Omega = \langle 1 | e\hat{\vec{r}} \cdot \vec{E} | 2 \rangle$. Ω^\dagger is the Hermitian conjugate of Ω . Note the diagonal terms are zero because of the symmetry (or to say dipole interaction does not couple states to itself).

We proceed to solve the Schrodinger equation of the driven system with the Hamiltonian $\hat{H} = \hat{H}_0 + \hat{V}$:

$$i \frac{\partial}{\partial t} |\phi(t)\rangle = \begin{bmatrix} \omega_1 & \Omega \cos(\omega t + \theta) \\ \Omega^\dagger \cos(\omega t + \theta) & \omega_2 \end{bmatrix} |\phi(t)\rangle \quad (2.18)$$

Here, we still express $\phi(t)$ as before (because $|1\rangle$ and $|2\rangle$ do form a complete basis).

But now, we know $c_1(t)$ and $c_2(t)$ vary with time.

Now let us do a unitary transformation to bring our perspective into the interaction picture. Instead of solving for $|\phi(t)\rangle$, we solve for $|\phi(t)_I\rangle = e^{i\hat{H}_0 t} |\phi(t)\rangle$. So

Eq.2.18 become:

$$i\frac{\partial}{\partial t}e^{-i\hat{H}_0}|\phi(t)_I\rangle = \begin{bmatrix} \omega_1 & \Omega\cos(\omega t + \theta) \\ \Omega^\dagger\cos(\omega t + \theta) & \omega_2 \end{bmatrix} e^{-i\hat{H}_0}|\phi(t)_I\rangle \quad (2.19)$$

Working out the algebra, we get:

$$i\frac{\partial}{\partial t}|\phi(t)_I\rangle = \begin{bmatrix} 0 & \frac{\Omega}{2}(e^{i(\omega-\Delta)t+\theta} + e^{-i(\omega+\Delta)t+\theta}) \\ \frac{\Omega^\dagger}{2}(e^{-i(\omega-\Delta)t+\theta} + e^{i(\omega+\Delta)t+\theta}) & 0 \end{bmatrix} |\phi(t)_I\rangle \quad (2.20)$$

Here $\Delta = \omega_2 - \omega_1$. Now, if you remember how we handled the Mathieu equation previously, you might recognize here that we are again having two drive (processes) with different time scale: $(\omega + \Delta)t$ and $(\omega - \Delta)t$. This time, the two-time scale approach we used before is applied here again, with a different name: rotating wave approximation. We want to know the "secular", or slow but larger, dynamic. So we can ignore the $(\omega + \Delta)t$ term to get:

$$i\frac{\partial}{\partial t}|\phi(t)_I\rangle = \begin{bmatrix} 0 & \frac{\Omega}{2}e^{i\delta t+\theta} \\ \frac{\Omega^\dagger}{2}e^{-i\delta t+\theta} & 0 \end{bmatrix} |\phi(t)_I\rangle \quad (2.21)$$

Here, $\delta = \omega - \Delta$ is the energy difference (detune) between the driving field and the energy gap between $|1\rangle$ and $|2\rangle$. To solve Eq.2.21, we need to do the unitary

transformation one more time, using $|\phi(t)'_I\rangle = e^{i\hat{U}t} |\phi(t)_I\rangle$. With:

$$\hat{U} = \begin{bmatrix} -\frac{\delta}{2} & 0 \\ 0 & \frac{\delta}{2} \end{bmatrix} \quad (2.22)$$

This turns Eq. 2.22 into:

$$i\frac{\partial}{\partial t} |\phi(t)'_I\rangle = \begin{bmatrix} \frac{\delta}{2} & \frac{\Omega}{2}e^{i\theta} \\ \frac{\Omega^*}{2}e^{-i\theta} & -\frac{\delta}{2} \end{bmatrix} |\phi(t)'_I\rangle \quad (2.23)$$

without time-dependent in the Hamiltonian, we can solve this equation using the diagonalization approach for solving linear ordinary differential equations. The solution has the following form:

$$|\phi(t)'_I\rangle = c_1 e^{i\frac{Wt}{2}} \begin{bmatrix} e^{-i\theta}\Omega \\ -(\delta - W) \end{bmatrix} + c_2 e^{-i\frac{Wt}{2}} \begin{bmatrix} e^{-i\theta}\Omega \\ -(\delta + W) \end{bmatrix} \quad (2.24)$$

Here, c_1 and c_2 depend on the initial state. $\frac{W}{2} = \sqrt{\delta^2 + \Omega^2}/2$ is what we call the Rabi-frequency. We will show later how this is equivalent to an arbitrary rotation on the Bloch sphere. But let us first take a look at the solution for $\delta = 0$ (resonant flop) and $|\phi(t=0)\rangle = |1\rangle$:

$$|\phi(t)'_I\rangle = \cos\left(\frac{Wt}{2}\right) |1\rangle + \sin\left(\frac{Wt}{2}\right) e^{-i\theta} |2\rangle \quad (2.25)$$

This interaction gives us a rotation along an arbitrary axis in the XY-plane

of the Bloch sphere by $\Omega t/2$. Let us see how this rotation picture is universally true for arbitrary initial states. We look for the evolution operator $\hat{U}(t)$ such that $|\phi(t)\rangle = \hat{U}(t) |\phi(0)\rangle$. From the Schrodinger equation, we have $\hat{U}(t) = e^{-i\hat{H}t}$

We can go ahead and write the matrix representation of this operator out in the basis spanned by the eigenvectors:

$$\hat{U}(t) = \begin{bmatrix} e^{iWt/2} & 0 \\ 0 & e^{-iWt/2} \end{bmatrix} \quad (2.26)$$

This in the original basis is:

$$\hat{S}^\dagger \hat{U}(t) \hat{S} = \frac{1}{2W} \begin{bmatrix} -e^{-i\theta} \sqrt{W-\delta} & \sqrt{W+\delta} \\ e^{i\theta} \sqrt{W+\delta} & \sqrt{W-\delta} \end{bmatrix} \begin{bmatrix} e^{iWt/2} & 0 \\ 0 & e^{-iWt/2} \end{bmatrix} \begin{bmatrix} -e^{i\theta} \sqrt{W-\delta} & e^{-i\theta} \sqrt{W+\delta} \\ \sqrt{W+\delta} & \sqrt{W-\delta} \end{bmatrix} \quad (2.27)$$

With quite some math, we can write out the expression of $\hat{U}(t)$ in the original basis as:

$$\hat{U}(t) = \cos\left(\frac{Wt}{2}\right) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - i \sin\left(\frac{Wt}{2}\right) \begin{bmatrix} \frac{\delta}{W} & e^{-i\theta} \frac{\Omega}{W} \\ e^{i\theta} \frac{\Omega}{W} & -\frac{\delta}{W} \end{bmatrix} \quad (2.28)$$

We can write this as:

$$\hat{U}(t) = \cos\left(\frac{Wt}{2}\right)\hat{I} - i\sin\left(\frac{Wt}{2}\right)\vec{n} \cdot \hat{\vec{\sigma}} \quad (2.29)$$

with:

$$\hat{\vec{\sigma}} = \hat{\sigma}_x\vec{x} + \hat{\sigma}_y\vec{y} + \hat{\sigma}_z\vec{z} \quad (2.30)$$

$$\vec{n} = \frac{\Omega}{W}\cos(\theta)\vec{x} + \frac{\Omega}{W}\sin(\theta)\vec{y} + \frac{\delta}{W}\vec{z} \quad (2.31)$$

Here, $\vec{x}, \vec{y},$ and \vec{z} are unit vectors in the x-, y-, and z- direction. $\hat{\sigma}_x, \hat{\sigma}_y, \hat{\sigma}_z$ are the three Pauli matrices.

Now with the evolution written as Eq. 2.29, we can see that the net effect is a rotation around \vec{n} -axis by an angle of Wt [3].

In practice, we set our rotations to be around either the X-axis or the Y-axis of the Bloch sphere by choosing θ properly. As for the rotation along the Z-axis, we equivalently implement it by changing the definition of angle θ in our control software. We call these three operations: $R_x(\theta)$, $R_y(\theta)$, and $R_z(\theta)$.

2.7 Single-qubit gates: Raman transition

There is, however, one problem with our over-simplified version. As we mentioned above, we choose the hyperfine-split states to be our $|0\rangle$ and $|1\rangle$ because they have extremely long lifetimes. One of the most important reasons for their long lifetime is that the forbidden dipole transition between them. The energy difference

between these two states is caused by the magnetic interaction between the nucleus' magnetic momentum and the electron spins. This magnetic interaction does not coupled via electric dipole interaction. As a result, we have $\Omega = 0$ and no Rabi-flop. But, we are not easily giving up on these two excellent long-lived, magnetic field insensitive states. Our work-around uses auxiliary states that couple to these two states via electric dipole interaction to bridge the transition. This is referred to as the Raman transition.

We consider the case of only one auxiliary state here since the effects of multiple auxiliary excited states can be linearly combined. We use two laser, with frequency ω_{L1} , ω_{L2} respectively to couple the $|0\rangle$ and $|1\rangle$ to the auxiliary state $|e\rangle$. Similar to

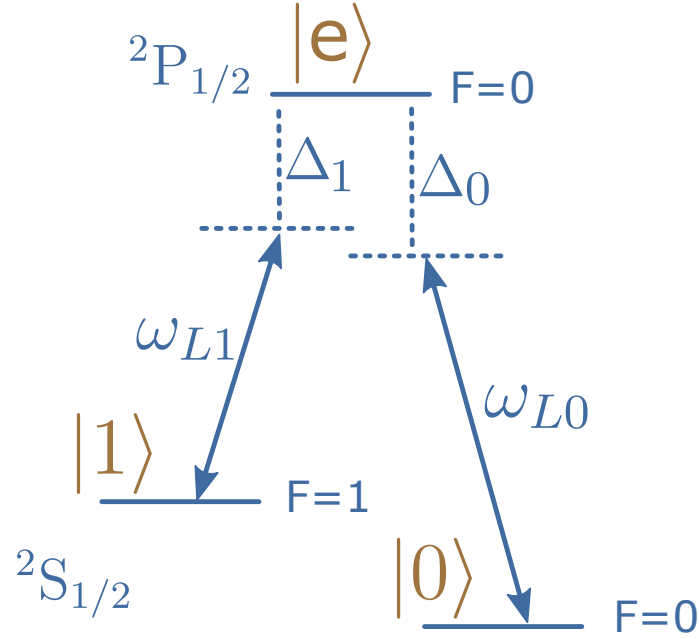


Figure 2.8: The λ system involved in the Raman transition we use to address the qubit states. We use two lasers to off-resonantly couple both qubit states to an auxiliary state.

Eq.2.14, we have the unperturbed Hamiltonian

$$\hat{H}_0 = \begin{bmatrix} \omega_0 & 0 & 0 \\ 0 & \omega_1 & 0 \\ 0 & 0 & \omega_e \end{bmatrix} \quad (2.32)$$

Now with these two lasers, the interaction Hamiltonian is:

$$V(t) = e\hat{\vec{r}} \cdot \vec{E} \quad (2.33)$$

$$= e\hat{\vec{r}} \cdot (\vec{E}_{L0}\cos(\omega_{L0}t + \theta_{L0}) + \vec{E}_{L1}\cos(\omega_{L1}t + \theta_{L1})) \quad (2.34)$$

Similar to Eq.2.18, we have the total hamiltonian $H = H_0 + V(t)$ as:

$$\hat{H} = \begin{bmatrix} \omega_0 & \vec{\mu}_{01}\vec{E} & \vec{\mu}_{0e}\vec{E} \\ \vec{\mu}_{10}\vec{E} & \omega_1 & \vec{\mu}_{1e}\vec{E} \\ \vec{\mu}_{e0}\vec{E} & \vec{\mu}_{e1}\vec{E} & \omega_e \end{bmatrix} \quad (2.35)$$

Here $\vec{\mu}_{ij} = \langle i | e\vec{r} | j \rangle$.

We again try to solve the dynamics in the interaction picture $|\phi(t)_I\rangle = e^{i\hat{H}_0 t} |\phi(t)\rangle$,

in which the Schrodinger equation looks like,

$$i\frac{\partial}{\partial t} |\phi(t)_I\rangle = \begin{bmatrix} 0 & \vec{\mu}_{01}\vec{E}e^{i(\omega_0-\omega_1)t} & \vec{\mu}_{0e}\vec{E}e^{i(\omega_0-\omega_e)t} \\ \vec{\mu}_{10}\vec{E}e^{i(\omega_1-\omega_0)t} & 0 & \vec{\mu}_{1e}\vec{E}e^{i(\omega_1-\omega_e)t} \\ \vec{\mu}_{e0}\vec{E}e^{i(\omega_e-\omega_0)t} & \vec{\mu}_{e1}\vec{E}e^{i(\omega_e-\omega_1)t} & 0 \end{bmatrix} |\phi(t)_I\rangle. \quad (2.36)$$

As illustrated in Fig.2.8, we have the following relations:

$$\vec{E} = E_{L0}\vec{e}^{i\omega_{L0}t} + E_{L0}\vec{e}^{-i\omega_{L0}t} + E_{L1}\vec{e}^{i\omega_{L1}t} + E_{L1}\vec{e}^{-i\omega_{L1}t} \quad (2.37)$$

$$\omega_{L1} + \Delta_1 = \omega_e - \omega_1 \quad (2.38)$$

$$\omega_{L0} + \Delta_0 = \omega_e - \omega_0 \quad (2.39)$$

Note for the frequencies involved here, we set $\Delta_1, \Delta_2 \ll \omega_1 - \omega_0 \ll \omega_{L0} < \omega_{L1}$. Plug these into Eq.2.36 and again apply the multiple time scale method, or "rotating wave approximation" to drop the terms that oscillate faster than Δ_0 or Δ_1 . Eq.2.36 then become:

$$i\frac{\partial}{\partial t} |\phi(t)_I\rangle = \begin{bmatrix} 0 & 0 & g_0 e^{i\Delta_0 t} \\ 0 & 0 & g_1 e^{i\Delta_1 t} \\ g_0^\dagger e^{-i\Delta_0 t} & g_1^\dagger e^{-i\Delta_1 t} & 0 \end{bmatrix} |\phi(t)_I\rangle \quad (2.40)$$

Here, $g_0 = \mu_{0e}^\dagger \vec{E}_0$. $g_1 = \mu_{1e}^\dagger \vec{E}_1$. We again apply the following transformation:

$$|\phi(t)_I\rangle = \begin{bmatrix} e^{i\delta t} & 0 & 0 \\ 0 & e^{-i\delta t} & 0 \\ 0 & 0 & e^{-i\Delta t} \end{bmatrix} |\phi(t)_{I'}\rangle \quad (2.41)$$

Here, $\delta = \frac{\Delta_0 - \Delta_1}{2}$. $\Delta = \frac{\Delta_0 + \Delta_1}{2}$. This transformation is motivated by the photon-dressed picture of the states. With this transformation, the Schrodinger equation become,

$$i \frac{\partial}{\partial t} |\phi(t)_I\rangle = \begin{bmatrix} \delta & 0 & g_0 \\ 0 & -\delta & g_1 \\ g_0^\dagger & g_1^\dagger & \Delta \end{bmatrix} |\phi(t)_I\rangle. \quad (2.42)$$

Eq.2.42 is readily integrable since the time-dependent are eliminated. But let us treat it with the adiabatic elimination to have a more intuitive perspective. First rewrite Eq.2.42 by separate the matrix and vectors into blocks:

$$i \frac{\partial}{\partial t} |\phi(t)_{I'}\rangle_{01} = \begin{bmatrix} \delta & 0 \\ 0 & -\delta \end{bmatrix} |\phi(t)_I\rangle_{01} + \begin{bmatrix} g_0 \\ g_1 \end{bmatrix} |\phi(t)_I\rangle_e \quad (2.43)$$

$$i \frac{\partial}{\partial t} |\phi(t)_{I'}\rangle_e = \begin{bmatrix} g_0^\dagger & g_1^\dagger \end{bmatrix} |\phi(t)_I\rangle_{01} + \Delta |\phi(t)_I\rangle_e \quad (2.44)$$

Here, $|\phi(t)_I\rangle_{01} = c_0(t) |0\rangle + c_1(t) |1\rangle$, $|\phi(t)_I\rangle_e = c_e(t) |e\rangle$. With adiabatic alimation, we look for solutions such that $\frac{\partial}{\partial t} |\phi(t)_{I'}\rangle_e = 0$, indicating that no net population transfer into or out of $|e\rangle$. Since we set the detuning Δ to be much larger than the coupling terms g_1, g_2 , this should be intuitive (recall the rabi-flop discussion above). So Eq.2.44 now become,

$$0 = \begin{bmatrix} g_0^\dagger & g_1^\dagger \end{bmatrix} |\phi(t)_I\rangle_{01} + \Delta |\phi(t)_I\rangle_e \quad (2.45)$$

Plugging this into Eq.2.43 to eliminate $|\phi(t)_I\rangle_e$, we have:

$$i\frac{\partial}{\partial t} |\phi(t)_I\rangle_{01} = \begin{bmatrix} \delta - \frac{|g_0|^2}{\Delta} & \frac{g_0 g_1^\dagger}{\Delta} \\ \frac{g_1 g_0^\dagger}{\Delta} & -\delta - \frac{|g_1|^2}{\Delta} \end{bmatrix} |\phi(t)_I\rangle_{01} \quad (2.46)$$

Now we see this is the form of Eq.2.23: we effectively realized Rabi-flopping between $|0\rangle$ and $|1\rangle$.

Note in Eq.2.46, we see energy of states $|0\rangle$ and $|1\rangle$ are shifted by $\frac{|g_0|^2}{\Delta}$ and $\frac{|g_1|^2}{\Delta}$, respectively. This is exactly the AC-Stark shift [18] caused by laser \vec{E}_{L0} coupling $|0\rangle$ and $|e\rangle$ with detuning $\Delta_0 \approx \Delta$, and laser \vec{E}_{L1} coupling $|1\rangle$ and $|e\rangle$ with detuning $\Delta_1 \approx \Delta$. Also, note the effective coupling strength (Rabi-frequency) between $|0\rangle$ and $|1\rangle$ is given by $\Omega = \left| \frac{g_0 g_1^\dagger}{\Delta} \right| \sim \left| \vec{E}_{L0} \right| \left| \vec{E}_{L1} \right|$. We can usually reach around 2 MHz Rabi-frequency in practice.

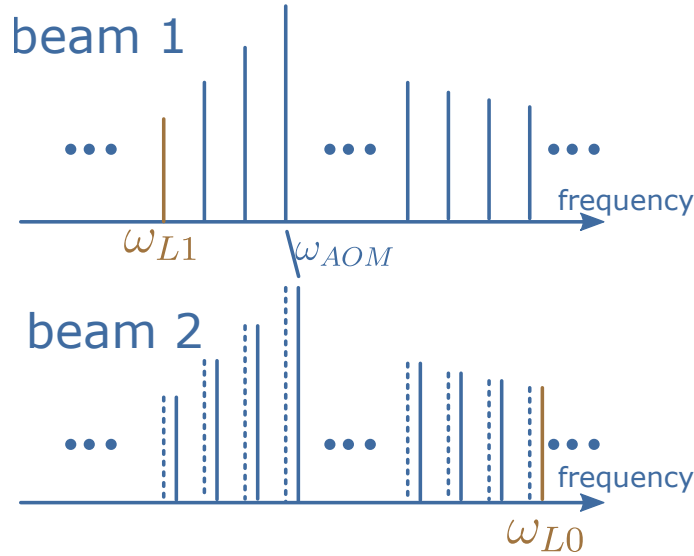


Figure 2.9: Driving Raman transition with two frequency combs. We use two laser beams derived from a single pulsed 355nm laser to drive the Raman transition. AOMs are used to shift the frequency of the combs so that pairs of comb teeth 12.6GHz away from each other drives resonant Raman transition.

Experimentally, we derive the two lasers \vec{E}_0 and \vec{E}_1 from a same 355 nm pulse laser, with pico-seconds pulse duration. The two pulse lasers, as illustrated in Fig.2.9, are combs of equally spaced frequency components. We want to find pairs of comb teeth precisely separated by 12.6 GHz, as needed for resonant Raman transitions. With the off-the-shelf laser, it is impossible to have stable pairs of the exact 12.6 GHz frequency difference. In practice, we use acousto-optic modulators (AOM) to actively shift the frequency of ω_{L0} frequency by ω_{AOM} . By actively measuring the spacing between the teeth of the frequency comb, we feedforward to ω_{AOM} so $\omega_{L0} - \omega_{L1} = 12.6$ GHz is stabilized. A considerable advantage of this scheme is that the frequency stabilization is implemented on the RF elements that drive the AOMs instead of the optical elements. RF stabilization is much easier than 355nm laser stabilization because of the collections of highly reliable tools related to its broad industrial applications.

2.8 Quantum bus: the motional modes

To implement two-qubit gates, we need a quantum bus to interact with information stored on different qubits. Theoretically, anything that couples to all the qubits can be used as the quantum bus. Naturally, the optimal choice should have the strongest coupling. For ion-trap quantum computers, this optimal choice is the motional modes of ion's collaborative motion in the pseudo-potential formed by the RF and DC fields. An intuitive understanding is that the coupling between the qubits and motional modes is equivalently the Coulomb interaction between an ion

and all the other ions. As far as we know, the Coulomb interaction is the strongest interaction beyond the atomic scale. We now introduce the interaction between this quantum bus (motional modes) and the qubits (internal state of ions).

We will again base our discussion on a simple rabi-flop. In Eq.2.17, we ignored the spatial variation of the electric field because it is much larger than the spread of wave function of the ions. However, the coupling between the motional states and the internal states of ions is exactly caused by this spatial variation. Taking the spatial variation into consideration, we can write Eq.2.17 as,

$$\hat{V} = \begin{bmatrix} 0 & \Omega \cos(\vec{k} \cdot \hat{\vec{x}} + \omega t + \theta) \\ \Omega^\dagger \cos(\vec{k} \cdot \hat{\vec{x}} + \omega t + \theta) & 0 \end{bmatrix}. \quad (2.47)$$

Since the RF-pseudo-potential is effectively a harmonic potential, it is convenient to continue our derivation using the creation \hat{a}^\dagger and annihilation operator \hat{a} [3] of the phonon states. We can write the position operator of the ions in terms of the creation and annihilation operators as:

$$\hat{x}_i = \sum_j p_{ij} \sqrt{\frac{\hbar}{2m\omega_j}} (\hat{a}_j + \hat{a}_j^\dagger) \quad (2.48)$$

where m is the mass of the ion. ω_j is the normal frequency of the j-th motional mode. p_{ij} is an entry of the participation matrix, quantifying how much the vibration of the j-th mode is affecting the position of the i-th ion. Since the physics stays the same, regardless of the number of ions or modes, for simplicity, let us only focus on the case of a single ion and a single mode in one dimension.

Thus, Eq.2.47 become:

$$\hat{V} = \begin{bmatrix} 0 & \Omega \cos(\eta(\hat{a} + \hat{a}^\dagger) + \omega t + \theta) \\ \Omega \cos(\eta(\hat{a} + \hat{a}^\dagger) + \omega t + \theta) & 0 \end{bmatrix} \quad (2.49)$$

Here η is the Lamb-Dicke parameter, which quantifies the comparison between the amplitude of harmonic motion of the ions and the wavelength of the electric field.

When $\eta \ll 1$, we can apply Taylor expansion to Eq.2.49 and get

$$\hat{V} = \begin{bmatrix} 0 & \Omega(1 + i\eta(\hat{a} + \hat{a}^\dagger))e^{i(\omega t + \theta)t} + c.c \\ \Omega(1 - i\eta(\hat{a} + \hat{a}^\dagger))e^{-i(\omega t + \theta)t} + c.c & 0 \end{bmatrix} \quad (2.50)$$

Because we expressed the motional state into phonon number states, we have to write the operator $\hat{V}(t)$ in a Hilbert space spanned by $|1n\rangle$ and $|0n\rangle$, where n is the phonon number of the motional state. Since this Hilbert space is no longer finite, we will temporarily switch away from matrix representation for convenience:

$$\hat{V} = i\Omega\eta e^{i(\omega t + \theta)}\hat{a}\hat{\sigma} + i\Omega\eta e^{i(\omega t + \theta)}\hat{a}^\dagger\hat{\sigma} + (-i)\Omega\eta e^{-i(\omega t + \theta)}\hat{a}\hat{\sigma}^\dagger + (-i)\Omega\eta e^{-i(\omega t + \theta)}\hat{a}^\dagger\hat{\sigma}^\dagger + c.c \quad (2.51)$$

where $\sigma = |0\rangle\langle 1|$. The 0-th order terms of the Taylor expansion correspond to the resonant rabi-flopping discussed above. Since we are now focusing on the coupling to phonons, we can drop it. As we will see later that they can be eliminated using the rotating-wave approximation. From Eq.2.51, we can already see that the motion

and internal states got coupled together. Let us take one more step forward to see how would the laser frequency ω affect the dynamics. We apply our favorite tool, transforming into the interaction picture, with the unitary $|\phi(t)_I\rangle = e^{i\hat{H}_0} |\phi(t)\rangle$. Here, we have:

$$e^{i\hat{H}_0} = e^{i\omega_q t} |1\rangle \langle 1| + e^{i\omega_p \hat{a}^\dagger \hat{a} t} \quad (2.52)$$

Here, ω_q is the energy difference between $|0\rangle$ and $|1\rangle$. ω_p is the energy of a phonon in our motional mode. Note $\hat{a}^\dagger \hat{a} = \hat{N}$ is the phonon number operator. With this transformation, the Schrodinger equation becomes:

$$\hat{V}_I = e^{-i\hat{H}_0} \hat{V} e^{i\hat{H}_0} \quad (2.53)$$

$$\begin{aligned} &= i\Omega\eta e^{i(\omega t - (\omega_q + \omega_p)t + \theta)} \hat{a} \hat{\sigma} + i\Omega\eta e^{i(\omega t - (\omega_q - \omega_p)t + \theta)} \hat{a}^\dagger \hat{\sigma} \\ &+ (-i)\Omega\eta e^{-i(\omega t + (\omega_q - \omega_p)t + \theta)} \hat{a} \hat{\sigma}^\dagger + (-i)\Omega\eta e^{-i(\omega t + (\omega_q + \omega_p)t + \theta)} \hat{a}^\dagger \hat{\sigma}^\dagger + c.c. \end{aligned} \quad (2.54)$$

Depending on the frequency of the driving laser ω , we would have two different Schrodinger equations after applying the rotating-wave approximation (keeping only non-oscillating terms).

When $\omega = \omega_q + \omega_p$ we have:

$$i \frac{\partial}{\partial t} |\phi(t)_I\rangle = \hat{V}_I |\phi(t)_I\rangle = (i\Omega\eta e^{i\theta} \hat{a} \hat{\sigma} - i\Omega\eta e^{-i\theta} \hat{a}^\dagger \hat{\sigma}^\dagger + c.c) |\phi(t)_I\rangle \quad (2.55)$$

We see the coupling is only between $|0\rangle_q |n\rangle_p$ and $|1\rangle_q |n+1\rangle_p$. The increase of phonon number is accompanied by the increase of internal state energy, vice versa. This is the blue-sideband coupling.

When $\omega = \omega_q - \omega_p$ we have:

$$i\frac{\partial}{\partial t} |\phi(t)_I\rangle = \hat{V}_I |\phi(t)_I\rangle = (i\Omega\eta e^{i\theta} \hat{a}^\dagger \hat{\sigma} - i\Omega\eta e^{-i\theta} \hat{a} \hat{\sigma}^\dagger + c.c) |\phi(t)_I\rangle \quad (2.56)$$

We see the coupling is only between $|1\rangle_q |n\rangle_p$ and $|0\rangle_q |n+1\rangle_p$. The increase of phonon number is accompanied by the decrease of internal state energy, vice versa.

This is the red-sideband coupling.

Importantly, because $\hat{a} |n+1\rangle_p = \sqrt{n+1} |n\rangle_p$ and $\hat{a}^\dagger |n\rangle_p = \sqrt{n+1} |n+1\rangle_p$, the coupling strength(rabi-frequency) between $|n\rangle_p$ and $|n+1\rangle_p$ is $\sqrt{n+1}$ times stronger than that between $|0\rangle_p$ (phonon) and $|1\rangle_p$ (phonon).

2.9 Initialize the bus: Sideband cooling

To use the quantum bus (motional modes) for two-qubit gates, we need to cool the ion chain down below the Doppler limit (several phonons). The reason will be clear in the next section. We next explain how to use sideband cooling to cool the ions down to ~ 0.1 phonons.

Fig.2.10 shows the cooling scheme. We repeatedly cycle between applying the red-sideband transition and the optical pumping. The red-sideband transition moves population between $|0\rangle_q |n\rangle_p$ and $|1\rangle_q |n-1\rangle_p$. The red-sideband couples all the state except the $|0\rangle_q |0\rangle_p$ state, because there is no $|1\rangle_q |-1\rangle_p$ state. The optical pumping, on the other hand, moves population from $|1\rangle_q |n\rangle_p$ to $|0\rangle_q |n\rangle_p$. In combination, as we cycle between these two processes, populations on all the states progressively step down these ladders into the $|0\rangle_q |0\rangle_p$ state.

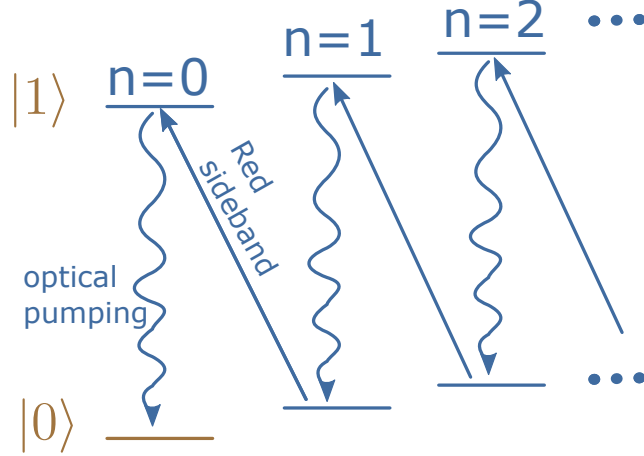


Figure 2.10: To implement sideband cooling, we cycle between RSB transitions and optical pumping. This gradually shift populations to lower phonon number states.

One caveat for experimentalists is mentioned at the end of the section 2.8. The coupling strength of the red-sideband is proportional to the phonon state \sqrt{n} . As the populations gradually gather into the states with lower phonon numbers, the time needed for each red-sideband cycle to optimally transfer the population from $|n\rangle$ to $|n-1\rangle$ should decrease, accordingly. Of course, this does not matter in an ideal world, since it is an incoherent process. Repeating the cycles sufficiently many times will eventually trap all the populations in $|0\rangle_q |0\rangle_p$. However, practically, we are competing with many experimental imperfections that heat the ion thus, it is advantageous to optimize the duration of red sideband transition in each cycle.

2.10 Address the qubits: Two-qubit gates

The final component of our quantum computing carnival is the two-qubit gates. As mentioned in the introduction chapter, with single-qubit rotations ready, any two-qubit gates that generates entanglement could complete the universal gates set. Naturally, the choice depends on the specific properties of the physical system.

For ion-trap quantum computers, the first two-qubit gate proposed is the Cirac-Zoller gate[19]. This gate uses the collective motion of ions as a cache to realize a controlled-phase gate. There is an issue of using the motion as a cache instead of a bus. Cache stores information in it while the bus does not. We tend not to use the Cirac-Zoller gates because the motional states are not exactly a good cache. Representing the collective Coulomb interaction of all the ions, the motional modes interact strongly with the environment, thus suffer from significant information loss.

Using the collective motion as only a bus to mediate the interaction between ions, we can realize two-qubit gates via the Mølmer-Sørensen (MS) interaction[20]. Such two-qubit gates are known as the MS-gates, or alternatively, XX-gate. Because no information is ever stored in the quantum bus (motion), we don't need to worry too much about information loss via the Coulomb coupling.

The matrix representation of the MS-gate is:

$$XX_{ij}(\chi) = \exp[i\chi\hat{\sigma}_x^{(i)}\hat{\sigma}_x^{(j)}] = \begin{bmatrix} \cos(\chi) & 0 & 0 & -i\sin(\chi) \\ 0 & \cos(\chi) & -i\sin(\chi) & 0 \\ 0 & -i\sin(\chi) & \cos(\chi) & 0 \\ -i\sin(\chi) & 0 & 0 & \cos(\chi) \end{bmatrix} \quad (2.57)$$

To implement this gate, we put two frequency components to the laser: one red-detuned and another one symmetrically blue-detuned. We use lasers with these two components to drive two ions. The coupling in the interaction picture described

by Eq.2.54 will become (with rotating wave approximation):

$$\hat{V}_I = \sum_j i\Omega_j (\eta_j e^{i(-\mu t + \theta_b)} \hat{a} \hat{\sigma}^j + \eta_j e^{i(\mu t + \theta_r)} \hat{a}^\dagger \hat{\sigma}^j - \eta_j e^{i(-\mu t - \theta_r)} \hat{a} \hat{\sigma}^{\dagger j} - \eta_j e^{i(\mu t - \theta_b)} \hat{a}^\dagger \hat{\sigma}^{\dagger j}) + c.c \quad (2.58)$$

where μ is the frequency difference between the red-detuned component and the red-sideband (equivalently, the frequency difference between the blue-detuned component and the blue-sideband). θ_r and θ_b are the phase of the red-detuned laser and the blue-detuned laser, respectively. j is the indices of ions. Note the number of motional modes always equals the number of ions. We considered only one mode here for simplicity. But since the dynamics of multiple modes is just a linear superposition of the dynamics of each individual mode, our derivation does not lose any generality.

From here, we can take two approaches. The first one, using our favorite adiabatic-elimination approximation, is more intuitive but only works with large μ .

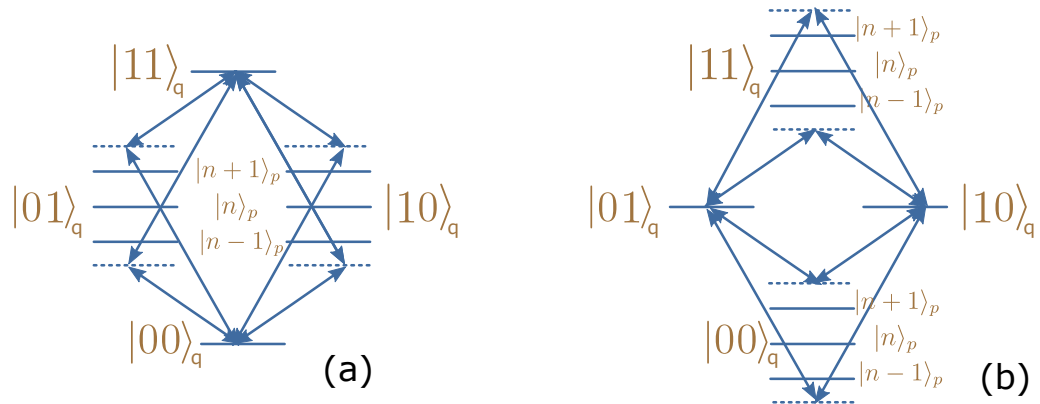


Figure 2.11: During the MS interaction, different two-qubit states are effectively coupled together via off resonant coupling to motional states. These couplings are similar to the Raman-transition discussed previously.

According to the adiabatic-elimination argument we used previously in deriving the Raman transition, there is no first-order population transfer when μ is large. We will only have second-order population transfer. This is illustrated from two different perspective in Fig.2.11 (a) and (b). The second-order effect, is two separate effective Rabi-flopping :one between $|00\rangle_q$ and $|11\rangle_q$, another between $|01\rangle_q$ and $|10\rangle_q$. Exactly as the definition of MS-gate (Eq.2.57) suggests. The actual math will be an 8-dimensional version of derivation starting from Eq.2.40. *The exact calculation will be left as an exercise for readers (CLASSIC)*

The above approach is not valid when the detuning μ is small enough to excite the phonon states substantially. This region is where we usually operate our experiment because that the stronger coupling grants us with faster two-qubit gates. In this region, we use Magnus expansion to solve the Schrodinger equation of interaction described by Eq.2.58.

With some algebra work, we re-write Eq.2.58 as:

$$\hat{V}_I = \sum_j \Omega_j \eta_j (e^{i(\mu t - \theta_m)} \hat{a} + e^{-i(\mu t - \theta_m)} \hat{a}^\dagger) \hat{\sigma}_{\theta_s - \frac{\pi}{2}}^{(j)} + c.c \quad (2.59)$$

where $\theta_m = (\theta_b - \theta_r)/2$, $\theta_s = (\theta_b + \theta_r)/2$. $\hat{\sigma}_{\theta_s - \frac{\pi}{2}}^j = \cos(\theta_s - \frac{\pi}{2}) \hat{\sigma}_x^j + \sin(\theta_s - \frac{\pi}{2}) \hat{\sigma}_y^j$.

To get this form, we used the relation regarding our spin-flip operator $\hat{\sigma}^\dagger = \hat{\sigma}_x + i\hat{\sigma}_y$.

For simplicity, we usually set $\theta_s = \frac{\pi}{2}$, so $\hat{\sigma}_{\theta_s - \frac{\pi}{2}}^{(j)} = \hat{\sigma}_x^{(j)}$

To solve Eq.2.59, we can write out the time evolution operator of our system $\hat{U}(t)$ using the Magnus expansion [21]:

$$U(\hat{t}) = \exp[-i \int_0^\tau dt V_I(\hat{t}) - \frac{1}{2} \int_0^\tau dt_2 \int_0^{t_2} dt_1 [\hat{V}(t_2), \hat{V}(t_1)]] \quad (2.60)$$

The higher order terms of the Magnus expansion vanish because the commutator $[\hat{a}, \hat{a}^\dagger] = 1$ commute with any operator. If we evaluate Eq.2.59, we will have the time-evolution operator in the following form:

$$U(\hat{t}) = \prod_j \hat{D}(\alpha(t)) \hat{\sigma}_x^{(j)} \prod_{j,k} \exp[i\chi_{jk}(t) \hat{\sigma}_x^{(j)} \hat{\sigma}_x^{(k)}] \quad (2.61)$$

Here $\hat{D}(\alpha(t)) = \exp[\alpha_j(t)\hat{a}^\dagger + \alpha_j(t)^\dagger\hat{a}]$ is the displacement operator of the phonon mode. The term $\hat{D}(\alpha(t))\hat{\sigma}_x^j$ entangle the qubits with the motional mode (quantum bus). When we finish our two-qubit gate at t , $\alpha_j(t) = 0$, we need to make sure that none of the qubit information is left in the quantum bus. As mentioned before, in reality, when we use n ions, there will be n different motional modes, which the qubits couples to. So we need to make sure all the $\alpha(t)$ terms vanish at the end of the gate. This is a classical linear control problem. The solution is to vary the laser field have a time-dependent $\Omega(t)_j$. In fact, for a linear controllable system (like ours), there are solutions guaranteed to reach any state we want[22]. Ref.[23] contains more detailed explanation of the control scheme currently implemented in all the ion-trap quantum computers at the University of Maryland. If we null the entanglement between the phonons and the qubits, we will be left with the following

$\hat{U}(t)$:

$$\hat{U}(t) = \prod_{j,k} \exp[i\chi_{jk}(t)\hat{\sigma}_x^{(j)}\hat{\sigma}_x^{(k)}] \quad (2.62)$$

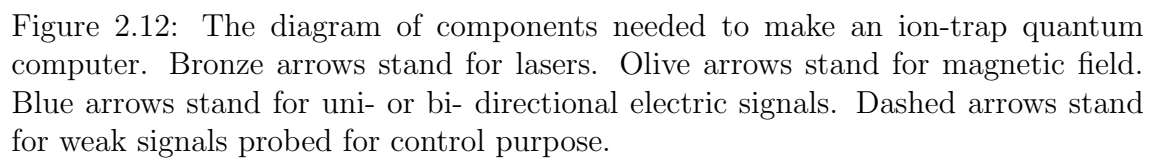
For simplicity, we only implement this Molmer-Sorensen interaction between two ions at any given time, so Eq.2.62 is simply:

$$\hat{U}(t) = \exp[i\chi(t)\hat{\sigma}_x^{(j)}\hat{\sigma}_x^{(k)}] \quad (2.63)$$

MS-gate, exactly as we want. According to our derivation, as a second-order coupling, the strength of the MS coupling is proportional to $|\Omega_i\Omega_j|$. In our experiment, since the Raman-transition we use to couple the qubit states is already a second-order coupling, the strength is $|\Omega_i\Omega_j| = |E_{i1}E_{i2}E_{j1}E_{j2}|$. The indices 1 and 2 correspond to the two lasers that drive Raman-transition on ion-i and ion-j. We adjust the laser power to scale the angle χ of the XX-gate. See [23] for the exact math.

2.11 Integration

Before we finish this chapter, let's step back and take another high-level look at how all the components are integrated into an ion-trap quantum computer. Fig.2.12 illustrate the integration of all the components. The trap, Yb sources are enclosed in a vacuum chamber. We have CW lasers for cooling, detection, optical pumping, and ion loading. We use a 355nm pulsed laser to drive coherent operation gate opera-



tions. Magnetic fields are generated by coils to establish the proper quantum states. RF and DC potentials are applied to the trap to shape the trapping potential. A PMT array collects the photons scattered by the ions through detection and cooling to read out the qubit states or determine the number of ions. All these components and operations are orchestrated by a set of FPGA (field-programmable gate array). We use control interfaces to interact with the FPGA and control everything. All the lasers and RF sources are actively stabilized by feedback loops.

Chapter 3: System Automation

Make your experiment do your researches for you. It is the capitalist's approach to a Ph.D. degree.

We aim to build the latest generation of ion-trap quantum computers at the University of Maryland to function as a block box. Ideally, users worldwide can submit their circuits via a user-friendly interface agnostic the system tune-up. Full-stack system automation is required to reach this goal. From an immediate perspective, a quantum computer, even in its early stage, is already substantially complicated, with numerous tuning knobs drifting slowly. Without appropriate automation, the researchers operating the machine will never get the machine to work with all the knobs tuned simultaneously. Moreover, we see more and more applications involving the evaluation of hundreds or thousands of circuits. Without automation, we will be overwhelmed by unnecessary operational errors, as well as physical fatigue.

The definition of automation can become philosophical. To simplify the discussion, we define that the automation discussed here refers to extra efforts beyond what people would take for a typical atomic, molecular, and optical (AMO) experiment. In the following, we shall go through the elements of our automation.

3.1 Optical alignment

The first experimental element that needs automation is the alignment of the free-space optics for the 355nm pulse-laser addressing $^{171}\text{Yb}^+$. The free space optics are very susceptible to mechanical vibration, temperature and humidity fluctuation, airflow, and dust. Generally, to improve stability, fibers are preferred over free-space optics. Unfortunately, the laser wavelength of 355 nm is not suitable for fiber optics because of the high absorption from the glass. Besides, high absorption induced short- and long-term damage on the glass tip further rule out the possibility to use fiber at high laser power. [24]. At the power of our 355nm laser, we observe the development of damages to the waveplates, mirrors, beam splitters even in our free-space optical setup over a few months. Nevertheless, many ongoing studies are exploring the future possibilities of fiber-based solutions.

As mentioned in the previous chapter, the laser phase and intensity need to be tuned and stabilized for gate operations. The laser intensity is usually stabilized through an AOM by deflecting the excessive power away. When the laser power drops, we compensate for the drop by decreasing the deflected power. In addition to the laser itself, the laser phase and intensity, from the ions' perspective, also depend on the beam alignment. The impact from alignment fluctuation on the phase can be partially mitigated by reconfigure the gate scheme [25]. But the ultimate solution would be physically stabilizing the beam alignment, as the scheme in Fig.3.1.

We use two beam-position sensors (quadrant photo-diodes) to sample the beam's position at two locations apart with a distance much larger than the Rayleigh

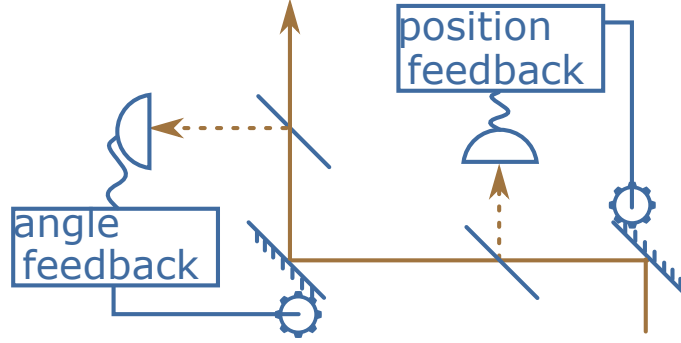


Figure 3.1: The schematic of a beam path stabilization system consists of two feedback loops. Each loop consists of a piezo-controlled mirror to steer the beam, a servo, and a position sensor. The first feedback loop stabilizes the position of the beam. The second feedback loop stabilizes the angle of the beam.

range. This way, the first and second sensors approximately monitor the two orthogonal degrees of freedom of the beam. We feedback the error signal from the sensor to the motors controlling the mirrors. Since the beam is stabilized at these two locations, it is stabilized throughout the whole optical path.

Right before the laser meets the ion, we have another feedback loop steering the beam. For this one, we directly use the Rabi-flopping rate as the signal to compensate for the drift between each measurement. We refer to this last feedback loop as the pointing lock.

3.2 Ion loading

Deterministic loading is necessary for full-stack automation. According to our discussion in chapter.2, if we blast atom vapors in our trap with lasers, we would end up with a random number of ions. Our solution is to split the trap potential into two spatially separated regions: one hosts the loading potential, and the other provides holding potential. The loading potential sitting in the ionization beam is

shaped only to hold one ion. The holding potential can hold multiple ions but does not see any ionization beam. We repetitively try to load in the loading potential and merge it with the holding potential. Every time the two potentials merge, the number of ions in the holding potential would either increase by one or zero (if the loading potential fails to catch one ion through this attempt). We repeat this procedure until there is a desired number of ions in the holding potential.

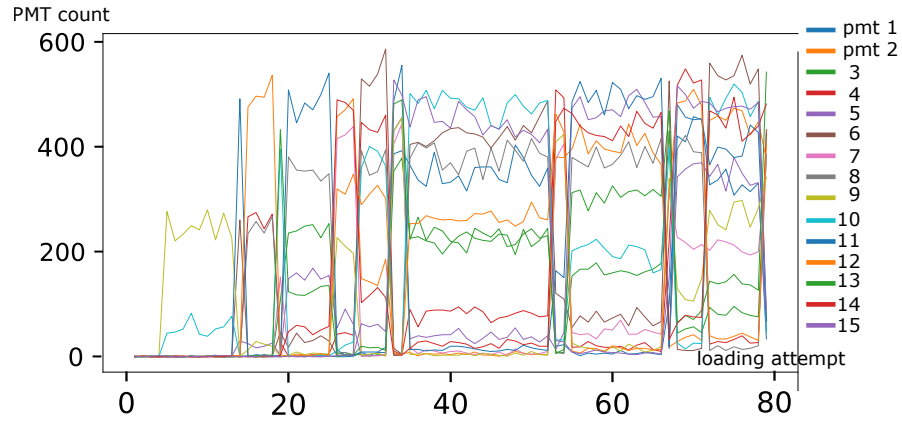


Figure 3.2: PMT-array reading through a typical loading event. At each loading attempt, we try to merge one ion into the ion chain. The reading from the PMT-array changes whenever the number of ions in the chain increase.

How do we determine the number of ions in the holding zone? We always start a loading procedure by dumping everything in the "holding" zone, so the number always starts at 0. When we load one more ion, the photon counts registered by the PMT array change. We organize the PMT array counts into a vector and compute the change of the vector's orientation after each merge. If the orientation change is big enough, we count one more ion in the "holding" potential. Like Fig.3.2 shows. The PMT counts change 15 times through an attempt to load 15 ions.

3.3 Center scan

Another critical building block of full-stack automation is the center scan. We use the center scan to locate the ions' relative position to the individual beams and check the number of the ions in the chain.

The center scan operation is illustrated in fig.3.3. We shift the center of the voltage in the axial direction, so the whole ion chain shift relative to the array of the individual beams we use to drive Raman transition. If we turn all the beams on during the scan, the individual beams' amplitude profiles will be sampled as the ions' population transfer. The result looks like those shown in Fig.3.6 (c). We optimize the voltage so that the ion spacing, except that for the two edge ions, is uniform across the chain, as shown in Fig.3.3. Thus there are two individual beam amplitude profiles shifted from the rest in fig.3.6 (c).

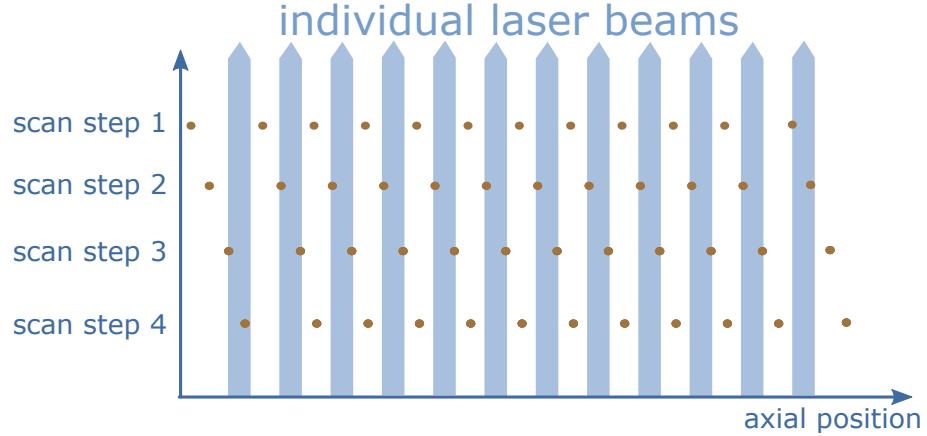


Figure 3.3: During a center scan, we turn on all the individual beams to drive Raman transitions while scanning the center of the whole ions chains along the axial direction. The individual beams' amplitude profiles are sampled as the population transfer from the Raman transitions. This center scan is used to locate the beams and the ions' relative position and count the number of ions in the chain.

3.4 Voltage calibration

As discussed in the chapter.2, our two-qubit XX gates are mediated by the phonon modes. The design of the gate solution depends on the phonon modes spectrum of the ion chain. The voltage configuration for the ion-chain determines the spectrum of the phonon modes. So to make the two-qubit gate operations work as designed, we need to calibrate the voltage configuration.

The automation is implemented on a High-Optical-Access(HOA)2.0 surface trap[26]. The surface trap has an array of independent electrodes that allows individual adjustment of many different voltage terms. We use the voltage calibration routine summarized in Fig.3.4 to automatically handles the calibrations of these voltage terms in a designated order.

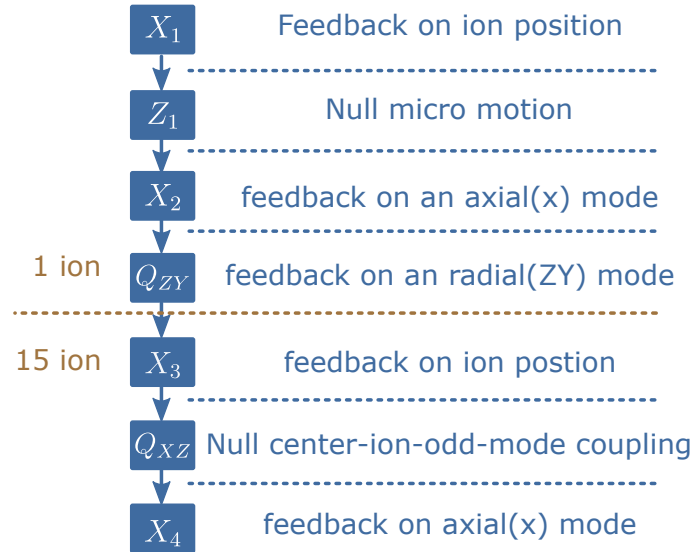


Figure 3.4: The voltage calibration sequence. We calibrate each voltage term in the order shown in the figure. We first finish all the calibrations that use only one ion, then load 15 ions and finish the rest calibrations.

We use the coordinate system illustrated in Fig.3.5.

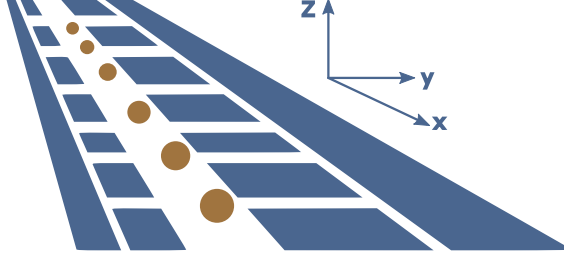


Figure 3.5: The X-,Y-,Z- direction are defined as shown. The X-axis is along the axial direction of the ion chain. The Y-axis is parallel to the trap surface.

For the terms in the X-direction, we we expand the DC potential in Taylor series, and consider up to the fourth-order

$$V(x) = X_1x + X_2x^2 + X_3x^3 + X_4x^4 + O(x^5). \quad (3.1)$$

Here $X_n = \frac{\partial^n V(x)}{\partial x^n}$ is the n-th order derivate of $V(x)$ with respect to x . In an ideal case, only the X_2 and X_4 term should have non-zero values. However, trap imperfections such as exposed dielectric material can cause undesired charging due to the photovoltaic effect. Thus all four terms need to be adjusted. As for higher-order terms ($n > 4$), the symmetry required to alter them is usually not satisfied by the trap's geometry configuration.

3.4.1 Calibrating X_1 and X_3

The existence of X_1 within the potential applies a constant push on the ions along the X-direction. So we can use ion position as an indicator to monitor this term. But X_3 term can also cause the ion to move horizontally along the X-direction. The difference is that the push from this term $F = e \cdot \frac{\partial(X_3x^3)}{\partial x}$ is small near the origin, while the push from the X_1 term is constant along the X-direction. During X_1

calibration, we use a single ion trapped near the origin to minimize the effect of the X_3 term. The X_2 attracts the ion to its minimum, But the X_1 pushes it away. If we set the X_2 to a high value, the push from X_1 will be negligible. We can then scan the ion position to find the minimum of the X_2 (the origin), as shown in Fig.3.6(a). Then we switch the X_2 to a low value. Now, the ion position will be shifted from the push from X_1 unless $X_1 = 0$, as shown in Fig.3.6(b). We scan the ion position and use this shift as the feedback signal to adjust the X_1 term. We repeat this position scan with a low X_2 setting until the ion stays at the origin. After nulling the X_1 term, we can use the same technique to calibrate X_3 . Since a single ion is not sensitive to X_3 term, we need to use 15 ions, as shown in Fig.3.6(c).

3.4.2 Calibrating X_2 and X_4

X_2 and X_4 are both confining terms along the X-direction. When these two terms change, both the phonon mode spectrum in the axial direction and that in the radial direction will change accordingly. Since the axial phonon mode spectrum is more sensitive to the axial potential, we use the axial mode for the calibration. Similar to the discussion in the previous section, near the origin, the recovering force from the X_4 terms $F = e \cdot \frac{\partial(X_4 x^4)}{\partial x}$ is weaker than that from the X_2 term. Using a single ion, with X_1 calibrated, we only need to consider the axial mode spectrum change caused by the X_2 term. It is straightforward to calculate the mode frequency from the X_2 . We repeatedly measure the axial mode frequency and compare it with the calculated value. The difference is again used as a feedback signal to adjust the

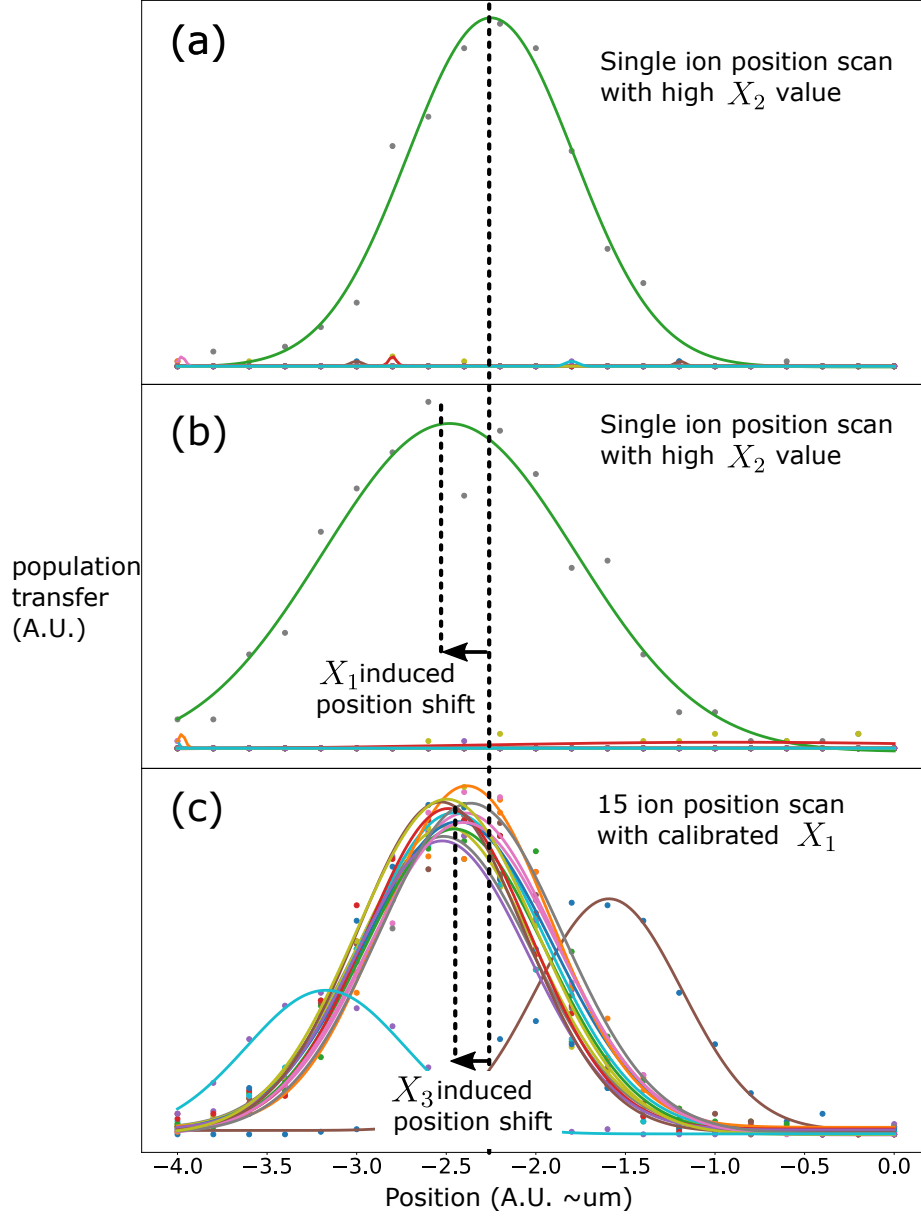


Figure 3.6: We use position shift to calibrate X_1 and X_3 . A high X_2 setting is first used to measure the un-shifted position of the ion. Then the X_1 induced position shift is measured with only 1 ion for calibration. We finally use 15 ions to measure the X_3 induced position shift for X_3 calibration.

X_2 until the difference is below measurement precision, as shown in Fig.3.7

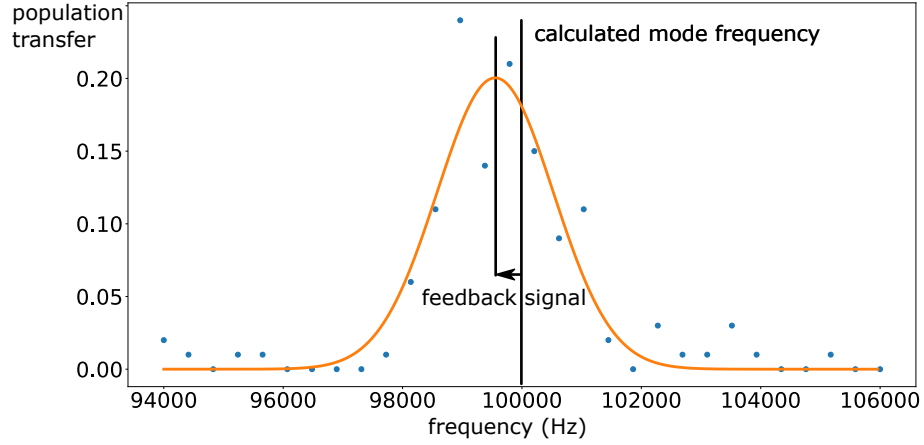


Figure 3.7: Population transfer as a function of axial mode frequency. Blue dots represent experimental data and solid yellow curve is a fit to the Gaussian function to determine the center frequency. The difference between the scanned frequency and the calculation (from the X_2 setting) is used to calibrate the X_2 value.

In our experiment, we stop the feedback process when the difference is below 600Hz. This value is chosen empirically. The calibration should be as accurate as possible but not exceed the precision of the frequency scan. After we are done with the X_2 term, we can calibrate the X_4 term with 15 ions in the chain. There are 15 axial modes in the spectrum for a chain of 15. We monitor one of the 15 axial modes to calibrate the X_4 term, with the same feedback technique discussed above. The choice of the mode to use for the calibration is determined empirically. We use a mode that is sensitive to the change of X_4 and couples strongly to the laser.

3.4.3 Calibrating Q_{XZ}

In the X-direction, Except for the four terms discussed above, we also consider a quadrupole Q_{XZ} terms. The Coulomb force derived from this term is illustrated in Fig.3.8. This Coulomb force tilts the ion chain and breaks the reflection symmetry.

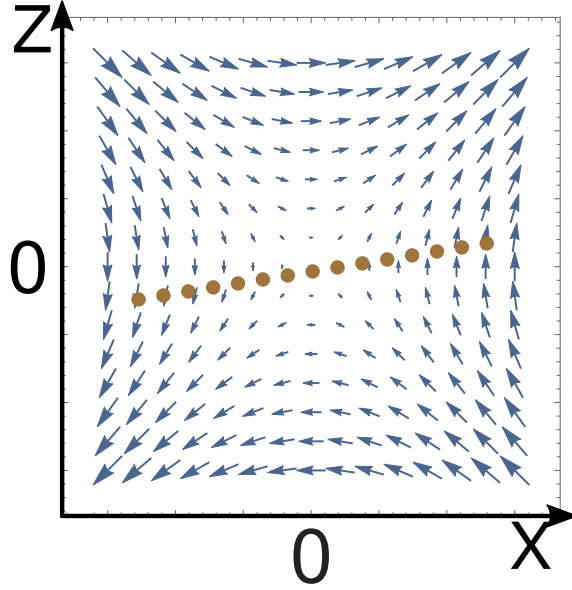


Figure 3.8: The vector plot of the electric Field generated by the Q_{XZ} -quadrupole in the XZ-plane. The ion chain is tilted by this electric field.

We use this symmetry to calibrate this term. Among the 15 different radial modes, some are even modes, and the rest are odd modes. The odd modes have the following symmetry:

$$p_j(t) = -p_{-j}(t) \quad (3.2)$$

$$q_j(t) = -q_{-j}(t) \quad (3.3)$$

Here p and q are the momentum and position of ions. j is the integer index of the ions, ranging from -7 to 7 . We can see from that $p_0(t) = q_0(t) = 0$, the center ion does not couple to odd modes. Thus if we drive the sideband transition on the center ion on the odd modes, we will see no population transfer. However, if

a non-zero Q_{XZ} breaks the mirror symmetry, the center ion will couple to the odd modes. We use this as a feature to calibrate Q_{XZ} . First, as shown in Fig.3.9(a), we scan the frequency of the sideband transition (within a known range) to identify the frequency of the odd mode. Then we drive the sideband transition using the identified frequency and scan the Q_{XZ} setting. The minimum population transfer corresponds to the optimal Q_{XZ} that preserves the reflection symmetry mentioned above.

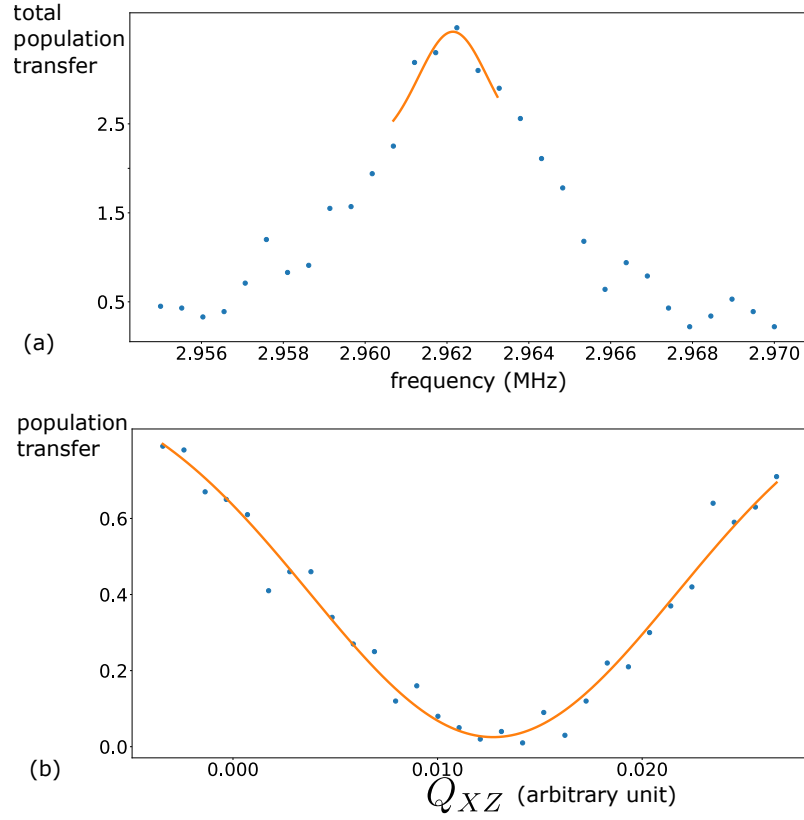


Figure 3.9: To calibrate (null) the Q_{XZ} term, we find the frequency of an odd radial mode. We then scan the Q_{XZ} setting while driving a sideband transition on the located odd radial mode. We set the Q_{XZ} to the value that gives the minimum population transfer.

3.4.4 Calibrating Z_1

In the YZ (radial) direction, we consider the first-order term Z_1 term of the voltage expansion in the Z direction. This term pushes the ions away from the minimum of the RF pseudo-potential (see discussion in chapter 2). If the ion gets too far away from the minimum of the RF pseudo-potential, it will be modulated by the RF-induced micro-motion. We use this micro-motion as a feature to calibrate the Z_1 term. We drive a sideband transition with the micro motion frequency and scan the setting of Z_1 . We set Z_1 to the value that has minimum population transfer via the micro-motion sideband transition. This is illustrated in Fig.3.10.

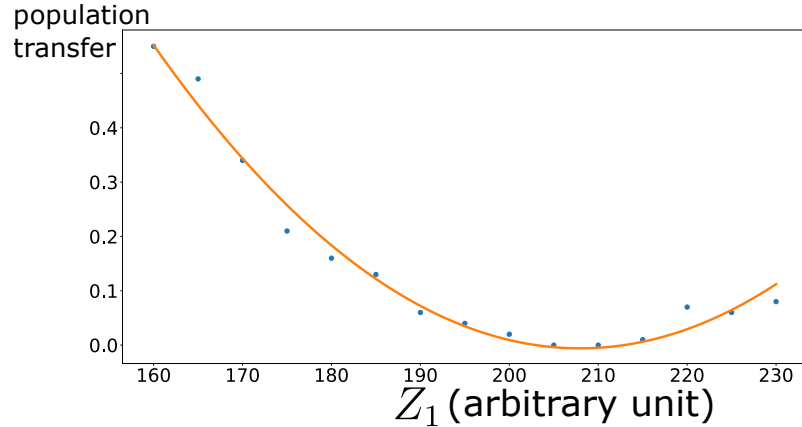


Figure 3.10: We scan the Z_1 setting while driving a sideband transition with the micro-motion frequency. The Z_1 setting is calibrated to the value that gives the minimum population transfer.

Luckily, we do not need to worry about the Y_1 term nor the micro-motion in the Y direction. Because of the symmetry of the surface trap, the electric noise can only contribute to Y_1 term weakly.

3.4.5 Calibrating Q_{ZY} compensation

In the radio direction, we also calibrate the quadrupole potential Q_{ZY} . The Coulomb force generated by this term is identical to what is shown in Fig.3.11. As shown in the figure, in the radial direction, there are two sets of modes. We use one of them to drive the two-qubit gates. The Coulomb force derived from the RF pseudo-potential squeezes or relaxes the ion symmetrically in all directions, affecting the two sets of modes identically. But the Coulomb force derived from the Q_{ZY} potential acts oppositely on the two sets of radial modes, squeeze one, relax the other, or vice versa. For a single ion, this can be summarized into the following equation:

$$\omega^2 = \Omega_{RF}^2 - \Omega_x^2 \pm \sqrt{Q_{ZZ}^2 + Q_{ZY}^2} \quad (3.4)$$

Here, ω is the frequency of the two orthogonal radial modes. Ω_{RF} is the natural frequency of the harmonic oscillation of the ion in the radial direction resulted from the RF pseudo-potential. Ω_x is the natural frequency of the harmonic oscillation of the ion in the axial direction resulted from the axial DC potential (for one ion, dominated by the X_2 term).

By adjusting this Q_{ZY} term, the impact of RF pseudo-potential (amplitude) fluctuation on the set of radial modes used to mediate two-qubit gates can be compensated. Note from Eq.3.4 and Fig.3.11 this will sacrifice the other set of modes,

but it is fine since we only care about the modes used for two-qubit gates. Experimentally, we scan the sideband transition frequency in an anticipated range and find the frequency of this radial mode we care about. We compare the difference between the theoretical prediction and the measured value. The difference is used to feedback and adjust the Q_{ZY} . This procedure is identical to that illustrated in Fig.3.7.

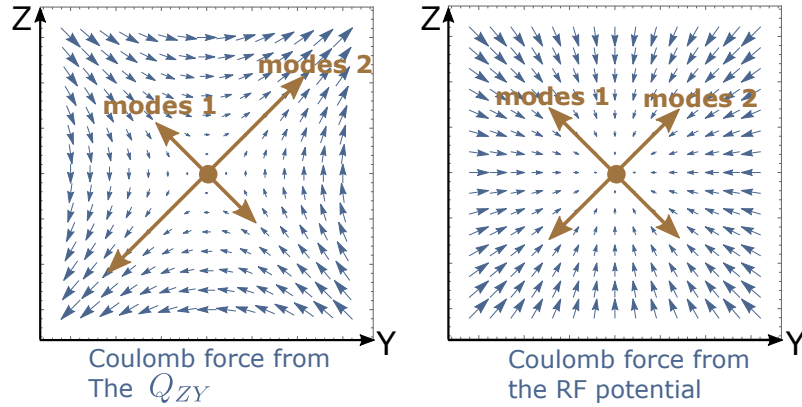


Figure 3.11: Left: the vector plot of the electric field generated by the Q_{ZY} quadrupole in the ZY-plane. This electric field squeezes one of the two orthogonal sets of radial motional modes and releases the other set. Right: the vector plot of the electric field generated by the RF pseudo-potential in the ZY-plane. This electric field squeezes the two orthogonal sets of radial motional modes together.

We calibrate all these terms in the order illustrated in Fig.3.4. The order in which we handle these terms is based on two considerations. First, we want to minimize loading time since loading is a relatively slow operation. Loading 15 ions usually takes about 5 minutes. Thus, we finish all the calibrations that use a single ion before doing anything that requires 15 ions. Another consideration is the interference between each of these terms. For example, as explained previously, X_3 calibration will be affected by X_1 calibration. But if we use one ion, the effect of X_3 could be excluded when calibrating the X_1 .

Fig.3.12 shows the effectiveness of this voltage stabilization routine. It shows the differences between the measured frequency after calibration and the calculated frequency of all the 15 modes mediating our two-qubit XX-gates. The differences are recorded after every voltage calibration routine implemented between June 2020 and April 2021. The first row and the second row of Fig.3.12 are identical, except the first row only displays the four closest modes to where we drive the gates, which matters most. The modes are stabilized to within 500 Hz. Our gate fidelity is only sensitive to the mode drift beyond 1 kHz, as justified by scan the laser frequency while applying the gates and measure the fidelity.

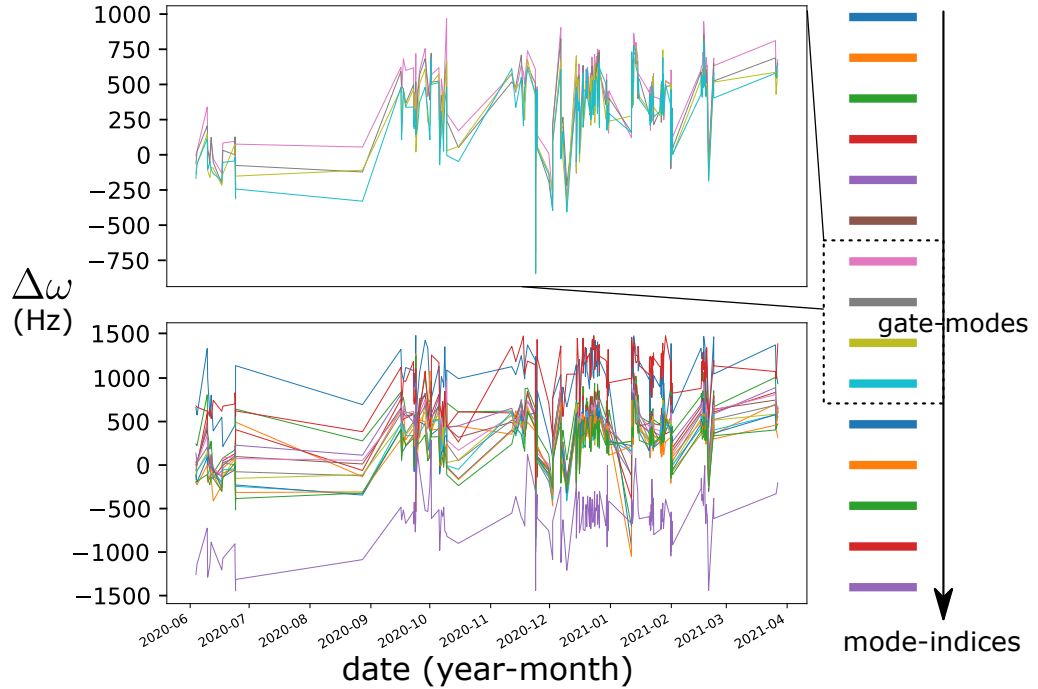


Figure 3.12: Second row: Differences between measured frequencies and the nominal frequencies of all the 15 radial modes we use for two-qubit XX-gates. The measurements are done after voltage calibration, successful or not. First row: same as the second row, but only displays the four modes closest to the frequency we use to drive the XX-gates.

3.5 Gates calibration

3.5.1 Calibrate the gate angle

With the voltage configuration calibrated, the Molmer-Sorensen gate will work as described by Eq.2.63. But there are still two issues to address. The first is to calibrate the gate angle. In chapter 2, we mentioned that the gate angle is proportional to the intensity (amplitude square) product of the two beams driving the XX-gate. This can be written as

$$\chi \propto k^2 |E_{L1}| |E_{L2}| \quad (3.5)$$

Here, k is the amplitude scale we apply to one of the two lasers to adjust the χ . Experimentally, we can not directly measure χ . To calibrate χ , we directly apply the gate defined in Eq.2.57 to the qubits initialized in $|00\rangle$. Then we measure the state. It is straight forward to derive that for both qubits:

$$P_1 = \cos^2(k^2 |E_{L1}| |E_{L2}|) \quad (3.6)$$

Here P_1 is the probability of getting $|1\rangle$. We try to find the nominal scale k_n with which we get the gate angle $\pi/4$. To do so, we measure the average population transfer P_1 of both qubits with whatever the scale k_0 that is currently used. Then

we calculate for k_n using:

$$k_n = k_0 \sqrt{\frac{\pi/4}{\arccos(\sqrt{P_1})}} \quad (3.7)$$

Now with this k_n , we can scale χ to any desired value. During this experiment, we also collect the statistics of $P_{00} + P_{11}$, the probability of measuring $|00\rangle$ or $|11\rangle$. We call this parity. It is used to derive the gate fidelity, as we will introduce later.

3.5.2 calibrate the Stark shift

We first mentioned Stark shift in Eq.2.46. When driving a transition, the population will oscillate between the two coupled levels. It is convenient to study the new eigenstates of the driven system in the dressed state basis [27]. When the drive is off-resonant, the two dressed states will be very similar to the two original states but with shifted energy. We call this (AC) stark shift. When we drive the off-resonant MS-interaction, there are many states off-resonantly coupled together. As a result, it changes the qubit frequency and effectively rotates our qubit state around the Bloch sphere during gates. If we do not correct this phase drift, we will lose our definition of the Bloch sphere, or equivalently, lose our definition of following gates. For example, what we call $R_x(\theta)$ might be turned into $R_y(\theta)$ because of the Stark shift. We use two knobs to correct the Stark shift. The first knob is what we call sideband-imbalance. Recall when we started the derivation of the MS-interaction from Eq.2.58, we assumed the blue-detuned component of our laser is balanced with

the red-detuned component of our laser. Applying imbalanced sidebands has two effects. First, it breaks the MS-interaction very minutely. Second, it generates an extra Stark shift. From Eq.2.46, it is clear that the Stark shift depends on the sign of the detuning. The blue and the red sidebands have the exact opposite detuning, so they should mostly cancel each other. Thus when we change the balance, we will generate an extra Stark shift. As the second knob, we can add extra $R_z(\theta)$ gates to cancel the Stark shift at the software level.

Interestingly, the two knobs correspond to two conjugating perspectives. Using the sideband imbalance, we physically handle the phase shift of the qubit states. Using the $R_z(\theta)$, we only change our definition of the Bloch sphere's orientation, thus the definition of the gates. Both of the two techniques work since the Bloch sphere orientation is defined relative to gates operation. The key difference between these two knobs is that the Stark shift correction made with sideband-imbalance scales with the χ during XX-gate, while that made with $R_z(\theta)$ does not.

We calibrate the Stark shift in practice with an XX-echo protocol. This protocol is illustrated in Fig.3.13. Note we are calibrating the XX-gate, which is a two-qubit gate. But it is really hard to visualize the multi-qubit Bloch sphere. According to Eq.2.63, the xx-gate is equivalently rotating both of the target qubits around the X-axis. Also, we are not generating any entanglement in this protocol, so it is fine to only look at the Bloch sphere of one of the two qubits.

We first apply $R_y(\pi/2)$ gates to both qubits. This will rotate both qubits into the X-axis of the Bloch sphere, as shown in Fig.3.13(b). We then repeatedly apply $XX(\pi/4)$ and $XX(-\pi/4)$ to both qubits. Without Stark shift, the qubit states

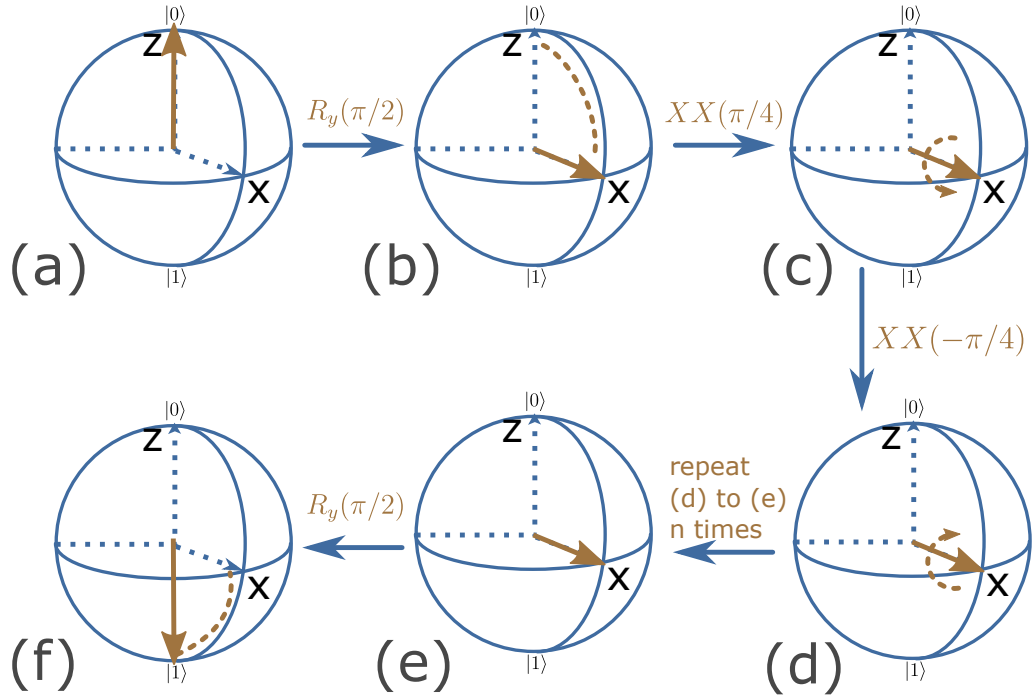


Figure 3.13: The XX-echo sequence is used to measure Stark shift during the XX-gates. We first use a $R_y(\pi/2)$ to rotate both qubits the XX-gate acts into the $|+\rangle$ state, an eigenstate of the XX-gate. Then we repeat $XX(\pi/4)XX(-\pi/4)$ several times. Without Stark shift, the following $R_y(\pi/2)$ will rotate both qubits into $|1\rangle$. Otherwise, we will have some populations in $|0\rangle$.

should stay aligned with the X-axis because rotations around X-axis do not affect states on X-axis. Thus when we eventually apply another $R_y(\pi/2)$ to both qubits, as shown in fig 3.13(f), the state of both qubits will be rotated into $|1\rangle$. Of course, this is assuming zero Stark shift. With Stark shift, the qubit state will rotate around Z-axis, during step (b) to step (e) in Fig. 3.13. As a result, the last $R_y(\pi/2)$ will not fully bring the states of both qubits into $|1\rangle$.

We repeat this XX-echo experiment, first scanning the value of the sideband imbalance. We set the sideband imbalance to the value that maximizes the probability of measuring $|1\rangle$, for the reason explained above. Next, we do the same thing, but scanning the value of the angle θ of the $R_z(\theta)$ we additionally apply to correct the Stark shift. As a result, the majority of the Stark shift will be corrected by sideband imbalance. Whatever is left will be addressed by the additional $R_z(\theta)$ gates.

3.6 Fidelity check

The gate fidelity is conventionally defined as the fidelity between the unitary experimentally implemented as the gate and the unitary this gate is supposed to be. A quantum process tomography operation is normally needed to measure this fidelity. The tomography operations are very costly to implement. So we usually give our definition of fidelity a little twist for convenience. Ideally, if we apply $XX(\pi/4)$ to the initial state $|00\rangle$, we get the GHZ state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. How well we can prepare this state is defined as the fidelity of the XX gate. This is a valid

and useful definition of fidelity because the point of two-qubit gates is to generate entanglement. Since the GHZ state is a maximally entangled state, how well we can generate it indeed represent how useful our two-qubit gates are.

To fully characterize a quantum state, the standard way, again, is to use quantum state tomography. But for the GHZ state, we can use a special witness called parity contrast. The overlap we want to measure, can be easily written as $F = \text{Tr}[\rho_i \rho]$. Here ρ is the density matrix of the state we prepared. $\rho_i = \frac{1}{2}(|00\rangle + |11\rangle)(\langle 00| + \langle 11|)$ is the density matrix of the ideal state. We also have $F = \frac{1}{2}(\langle 00| + \langle 11|)\rho(|00\rangle + |11\rangle) = \text{Tr}[\rho_i \rho]$, if this looks more intuitive. After figuring out the math, we have

$$F = \frac{1}{2}(\rho_{00,00} + \rho_{11,11} + \rho_{00,11} + \rho_{11,00}) \quad (3.8)$$

Here $\rho_{00,00}$ is the entry of the density matrix that correspond to the $|00\rangle \langle 00|$. The other terms are defined in the same way. The first two terms $\rho_{00,00} + \rho_{11,11}$ correspond to the probability of measuring 00 or 11, $P_{00} + P_{11}$. As we mentioned in the gate amplitude calibration section, the value for this two terms is readily obtained through the calibration. To measure the last two term $\rho_{00,11} + \rho_{11,00}$, we use a routine called parity scan. During a parity scan, we rotate the state by applying a $R_z(\theta)$ followed by a $R_x(\pi/2)$ to each qubit. We then measure the parity of the rotated states: $\Pi(\theta) = P_{00} + P_{11} - P_{01} - P_{10}$

Again, with some math, we can show:

$$\Pi(\theta) = 2|\rho_{00,11}| \cos(2\theta + \phi_0) \quad (3.9)$$

Here, ϕ_0 is a phase offset derived from the relative phase between $|00\rangle$ and $|11\rangle$. We also assumed $\rho_{00,11} = \rho_{11,00}$ as normally the entries of a density matrices should be. Now, we just need to scan θ and measure the amplitude of the oscillation. As shown in Fig.3.14.

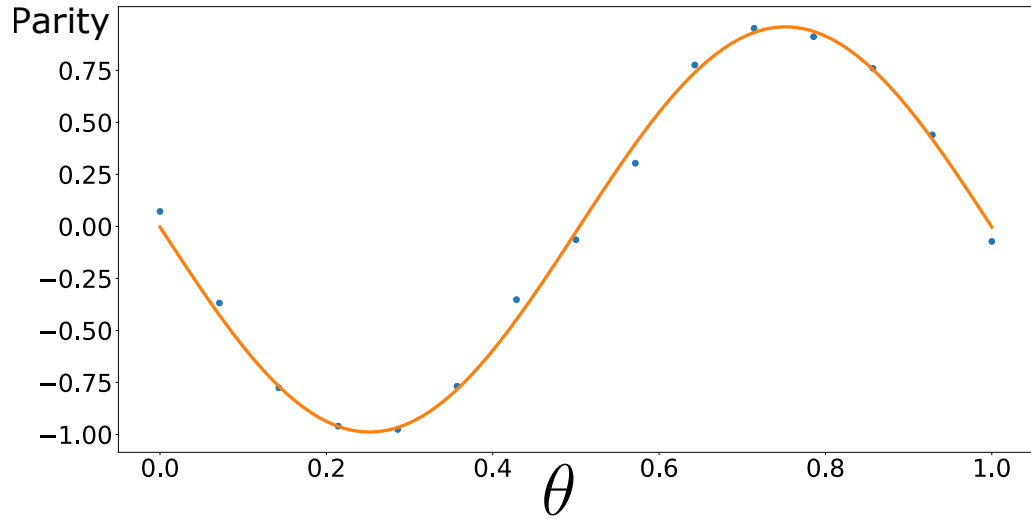


Figure 3.14: The Parity flop we use to measure the coherence (parity scan contrast) of the GHZ state.

We call the amplitude of this oscillation the parity contrast. It gives $\rho_{00,11} = \rho_{11,00}$. Because these two terms also represent the coherence between $|00\rangle$ and $|11\rangle$, I will also call it the coherence throughout this thesis. As given by Eq.3.8, the average of $P_{00} + P_{11}$ (sometimes referred to as the parity) and the coherence is what we use to quantify the fidelity of our XX-gate.

3.7 Integration

We implement all the calibration procedures as individual modules. The modules are tested separately to verify the reliability and tune for parameters. With the ARTIQ control software, we can conveniently integrate all the tuned modules together.

The modules are integrated into the automation routine illustrated in Fig. 3.15.

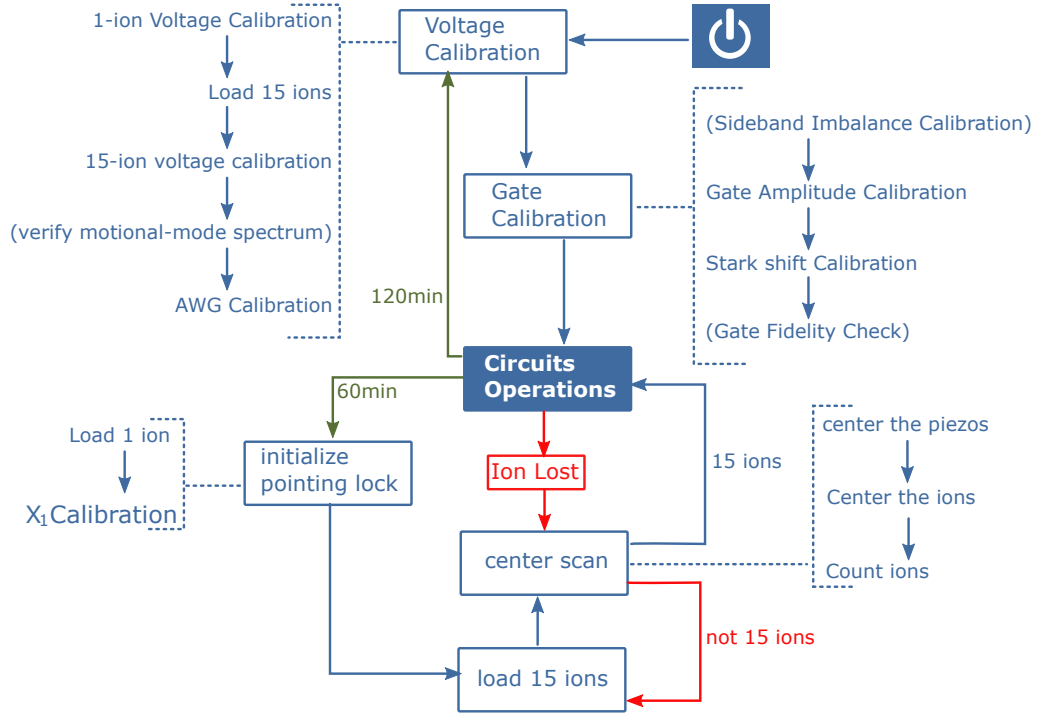


Figure 3.15: The flow-chart of the automation scheme implemented on the Euriqa system at the University of Maryland. The blue arrow leads to operations happening immediately after. The red arrows are triggered by monitoring sub-routine. The green arrows are set by users to repeat with a given frequency.

With the highly integrated routines, after the user hits the start button, the system goes through the voltage calibration steps, then calibrates each gate needed in the circuits. If the measured fidelities of all the gates are within the normal range (usually 96% ~ 99%), the system will be ready for circuit operations. The system

re-engages voltage calibration and gate calibration 2 hours (an empirical choice) to correct the system’s drifts. We also re-initialize the pointing lock every 1 hour to correct the walk-offs, mostly caused by the position shift induced by X_1 drift. During circuit operations (and voltage calibration procedures that use 15 ions), a subroutine monitors the scattered light from the Doppler cooling. When this cooling count fluctuates, it triggers a position scan and pauses all the other operations. The position scan (center scan, as illustrated in Fig.3.6(c)) verifies whether we have 15 ions. If not, it activates a loading procedure to load 15 ions. If we do have 15 ions, the system continues all the paused operations. A loading procedure always comes with a position scan to verify the number of loaded ions. Users usually submit the circuits and other non-circuit-based experiments from a separate computer, which is totally agnostic to this automation routine.

With the automation routine, the system can achieve a user-interference-free run of multiple days before some other random errors kick in, requiring human intervention. Fig.3.16 shows the record of gate fidelity through a 3-day-long free run session. Coherence and parity (average to fidelity) of 11 different XX-gates involving 13 qubits are recorded here. During this session, there is still one laser lock failure that required human attention. But other than that, Fig.3.16 shows that the automation routine maintained normal circuit operations through the session.

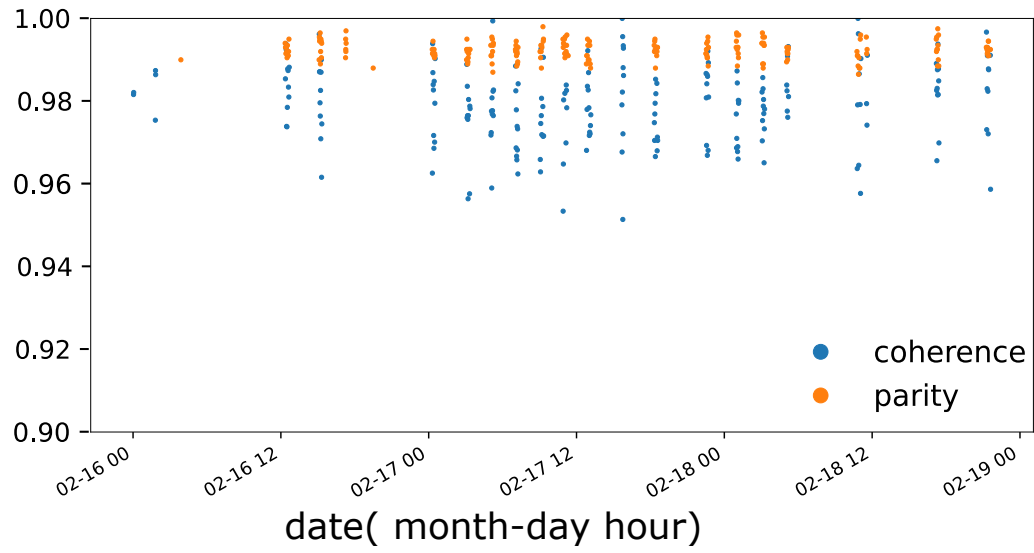


Figure 3.16: The fidelity of 11 different XX-gates acting on 13 qubits, recorded through a 3-day-long automated circuit operation. The average of parity and coherence equals the fidelity as given by Eq.3.8.

Chapter 4: Ion-Trap Quantum Programming

4.1 Rotation of gates

When compiling quantum circuits, we frequently need to rotate a gate. This can be conveniently done in the following way.

We want to implement an the following unitary operation

$$\hat{U} = U(\hat{u}_1, \hat{u}_2, \hat{u}_3 \dots) \quad (4.1)$$

where, \hat{u}_i 's are single-qubit operators acting on qubit-i. But what we can readily implement is

$$\hat{U}' = U(\hat{u}'_1, \hat{u}'_2, \hat{u}'_3 \dots) \quad (4.2)$$

To convert Eq.4.1 into Eq.4.2, we can do the following transformation

$$\hat{U} = \hat{R}_1 \hat{R}_2 \hat{R}_3 \dots U(\hat{u}'_1, \hat{u}'_2, \hat{u}'_3 \dots) \hat{R}_1^\dagger \hat{R}_2^\dagger \hat{R}_3^\dagger \dots \quad (4.3)$$

where \hat{R}_i 's are the rotation operations such that $\hat{u}_i = \hat{R}_i \hat{u}'_i$.

As an example, when doing gate-based quantum simulation, we very frequently

need to implement YY-gate, defined as $\hat{Y}\hat{Y}_{jk}(\chi) = \exp[i\chi\hat{\sigma}_y^{(j)}\hat{\sigma}_y^{(k)}]$. We can implement this by rotating XX-gate in the following way

$$\hat{Y}\hat{Y}_{jk}(\chi) = \hat{R}z_j(\pi/2)\hat{R}z_k(\pi/2)\hat{X}\hat{X}(\chi)\hat{R}z_j(-\pi/2)\hat{R}z_k(-\pi/2) \quad (4.4)$$

where we assume gates act on quantum states from left to right, like how matrices act on vectors.

4.2 Convert arbitrary gates into ion-native gates

A lot of gate-based quantum simulation algorithms naturally use Ising-type gates: XX, YY, or ZZ. But most commonly, algorithms are designed with gates like controlled-not, controlled-phase, and controlled-something because these gates are defined in the computational basis, which is more closely related to classical computing.

We can convert them into Ising type gates using the following fact. First, the value of a qubit in the computational basis (0 or 1) can be expressed with Pauli-operator as

$$\frac{1 - \hat{\sigma}_z}{2} |\phi\rangle = \text{value of } |\phi\rangle \text{ in the computational basis (0 or 1)} \quad (4.5)$$

Second, a bit flip in computational basis is a Pauli-X operator: $\hat{\sigma}_x$.

Whenever we have a gate defined in computational-basis, we can use these two facts to convert it into Pauli-basis and write it using Ising gates.

We use the Toffoli-gate as an example to demonstrate this procedure. The Toffoli-gate is also known as the controlled-controlled-not gate. This gate flips a target-qubit if and only if the two control-qubits both have the value 1 (in computational basis). We can write the Boolean expression for the two controlled bit (j,k) as

$$B = \left(\frac{1 - \hat{\sigma}_z^{(j)}}{2}\right)\left(\frac{1 - \hat{\sigma}_z^{(k)}}{2}\right) |\phi\rangle \quad (4.6)$$

B is one if and only if both of the control-qubits are 1.

Now flip the target qubit (l) can be achieved by an x-rotation of π , that is:

$$\overline{|\phi\rangle} = R_x(\pi) |\phi\rangle = \exp(-i\frac{\pi}{2}\hat{\sigma}_x^{(l)}) |\phi\rangle \quad (4.7)$$

Next, we need to put the Boolean value into the rotation angle to condition for rotate-or-not. This gives us the Pauli-expression for the Toffoli-gate

$$\exp[-iB\frac{\pi}{2}\hat{\sigma}_x^{(l)}] = \exp[-i(\frac{1 - \hat{\sigma}_z^{(j)}}{2})(\frac{1 - \hat{\sigma}_z^{(k)}}{2})\frac{\pi}{2}\hat{\sigma}_x^{(l)}] \quad (4.8)$$

$$= \exp[-i(\frac{\pi}{8})\hat{\sigma}_x^{(l)}]\exp[-i(-\frac{\pi}{8})\hat{\sigma}_z^{(j)}\hat{\sigma}_x^{(l)}]\exp[-i(-\frac{\pi}{8})\hat{\sigma}_z^{(k)}\hat{\sigma}_x^{(l)}] \quad (4.9)$$

$$\times \exp[-i(\frac{\pi}{8})\hat{\sigma}_z^{(j)}\hat{\sigma}_z^{(k)}\hat{\sigma}_x^{(l)}] \quad (4.10)$$

The Toffoli-gate can be decomposed into a R_x gate, two ZX gates, and a ZZX gate. We see from the discussion in the last section, ZX and ZZX can be converted into XX and XXX gates. We will see how to make many-body Pauli-gates with our

native two-body XX-gate in the next section.

4.3 Generating Many-Body Pauli-Gates

Using the Baker-Campbell-Hausdorff (BCH) formula [28], one can prove the following relation

$$e^{i\hat{\sigma}_j\hat{\sigma}_k\theta} = \cos(\theta)\hat{I} + i\sin(\theta)\hat{\sigma}_j\hat{\sigma}_k \quad (4.11)$$

Using this relation, working out some math, we can show the following decomposition

$$\exp[-i\frac{\pi}{4}\sigma_y^{(i)}\sigma_y^{(j)}]\exp[i\theta\sigma_x^{(j)}\sigma_x^{(k)}]\exp[i\frac{\pi}{4}\sigma_y^{(i)}\sigma_y^{(j)}] = \exp[-i\theta\sigma_y^{(i)}\sigma_z^{(j)}\sigma_x^{(k)}] \quad (4.12)$$

This decomposition can be cascaded to generate many-body terms in the following way

$$\exp[-i\frac{\pi}{4}\sigma_y^{(i)}\sigma_y^{(j)}]\exp[i\theta\sigma_x^{(j)}\sigma_x^{(k)}\sigma_x^{(l)}]\exp[i\frac{\pi}{4}\sigma_y^{(i)}\sigma_y^{(j)}] = \exp[-i\theta\sigma_y^{(i)}\sigma_z^{(j)}\sigma_x^{(k)}\sigma_x^{(l)}] \quad (4.13)$$

To generate the $\exp[i\theta\sigma_x^{(j)}\sigma_x^{(k)}\sigma_x^{(l)}]$ in Eq.4.13, we just need to use the gate-rotation trick introduced above to convert what we obtained in Eq.4.12.

In summary, using this technique, we can generate arbitrary many-body Pauli gates. It is easy to calculate the number of two-qubit gates needed to generate an n -body Pauli-gate is $2 \times n - 1$. The cost could be further optimized if we need to

apply many commutable n-body Pauli-gate together. We will see this in chapter [7.2.4](#).

Chapter 5: Trotterization based quantum simulation

Any Hamiltonian $\hat{H}(t)$ describing a quantum system can be written into the following form:

$$\hat{H}(t) = \hat{H}_1(t) + \hat{H}_2(t) + \hat{H}_3(t) + \hat{H}_4(t) + \dots \quad (5.1)$$

Here each of the $\hat{H}_i(t)$'s is easy to implement separately. Or more relevant to our gate-based (digital) quantum computers: each of the $\hat{H}_i(t)$'s can be individually implemented efficiently with a universal gate set. But implementing them all together is costly if not impossible. Is there a method that we can universally apply to study the dynamics described by this Hamiltonian?

One such method is called the trotterization, based on the Suzuki–Trotter expansion:

$$e^{\hat{A}+\hat{B}} = \lim_{n \rightarrow \infty} (e^{\frac{\hat{A}}{n}} + e^{\frac{\hat{B}}{n}})^n \quad (5.2)$$

This is easy to derive using the Baker–Campbell–Hausdorff formula:

$$e^{\hat{A}}e^{\hat{B}} = e^{\hat{A}+\hat{B}+\frac{1}{2}[\hat{A},\hat{B}]+\dots} \quad (5.3)$$

The time evolution operator of a Hamiltonian can be written as $\hat{U} = e^{-\frac{i\hat{H}}{\hbar}}$. Using eq.5.2, we can approximate the time evolution operator as:

$$\hat{U}(t) \approx \prod_{k=1 \sim n} e^{-\frac{i\hat{H}_1(k*t/n)\frac{t}{n}}{\hbar}} e^{-\frac{i\hat{H}_2(k*t/n)\frac{t}{n}}{\hbar}} e^{-\frac{i\hat{H}_3(k*t/n)\frac{t}{n}}{\hbar}} \dots \quad (5.4)$$

Note although we used the time evolution operator written with a time-independent operator, we can apply the trotterization method to a time-dependent Hamiltonian as shown above.

For the rest of this chapter, we present a study of many-body localization based on trotterization method.

5.1 Many-body localization (MBL)

Many-body localization (MBL) is a phenomenon which emerges in quantum systems with both interactions and disorder. At large values of disorder, a many-body system can fail to thermalize even at high temperatures causing it to exhibit properties like long-term memory retention, logarithmic entanglement growth in time, and area-law entanglement scaling [29, 30]. The many-body localization-

delocalization transition, which occurs at a critical disorder strength, is a dynamical phase transition. This necessitates the study of excited states, rather than just the ground state of the system. The study of this phenomenon in spin systems via full diagonalization exhausts classical computational power for a system of about 20 spins [31]. Specialized approximate schemes such as tensor network methods can in principle handle larger system sizes but tend to only work well for short-range interacting systems in one-dimension away from the phase transition [32, 33]. Many open questions still abound regarding the effects of symmetry, topology, dimensionality, long-range interactions, thermal inclusions, and the universality class of the disorder, especially near the phase transition. Better simulations of this phenomenon would also lead to a deeper understanding of fundamental concepts in quantum thermodynamics such as the eigenstate thermalization hypothesis. Thus, the study of a many-body localized system has been proposed as a benchmark for showing the utility of near-term quantum computers capable of noisy but classically unapproachable computations. [34].

Experimental efforts to probe MBL include quantum simulators consisting of thousands of cold atoms [35, 36, 37, 38, 39, 40] and a Hamiltonian whose disorder arises from the superposition of lattice potentials with incommensurate wavelengths. Another set of leading examples are experiments on trapped ions with tens of spins, which investigate the role of disorder in long-range Ising chains [41, 42]. Finally, up to three interacting photons in an array of transmons with random on-site energies have been studied [43, 44]. A limitation of all of these experiments is that they are specialized to a particular class of Hamiltonians that are native to the system and

therefore cannot address many open questions about MBL. The only simulation of MBL on a quantum computer operated in a universal fashion was limited to a 2-spin system realized with transmon qubits [45]. Additionally, the energy statistics and entanglement entropy studied in [43, 44] take exponentially longer to measure as the number of interacting particles increases. Another problem arises from the noise in near-term quantum computers, which manifests itself as a thermal bath coupled to the system. Since diagnostics like level statistics and entanglement growth have been shown to revert to thermal behavior on even weak coupling to a thermal bath [46], they are particularly unsuitable for the study of localization on such near-term devices.

Here, we introduce a technique for studying MBL on universal quantum computers by measuring the spectral functions of local operators. These carry signatures of localization that are known to survive coupling to a thermal bath as long as it is weaker than the characteristic energy scales of the model [46]. We measure spectra for the Heisenberg model with disordered magnetic fields along two directions implemented by 3 qubits on an ion trap quantum computer. In the many-body localized phase, the spectral functions exhibit a discrete nature, and after averaging over disorder, display a suppression of amplitude or "soft gap" at low frequencies, compared to the thermalized phase. In addition to the natural robustness to noise of our chosen observables, we also design an error mitigation scheme specific to the study of disorder-averaged spectral functions.

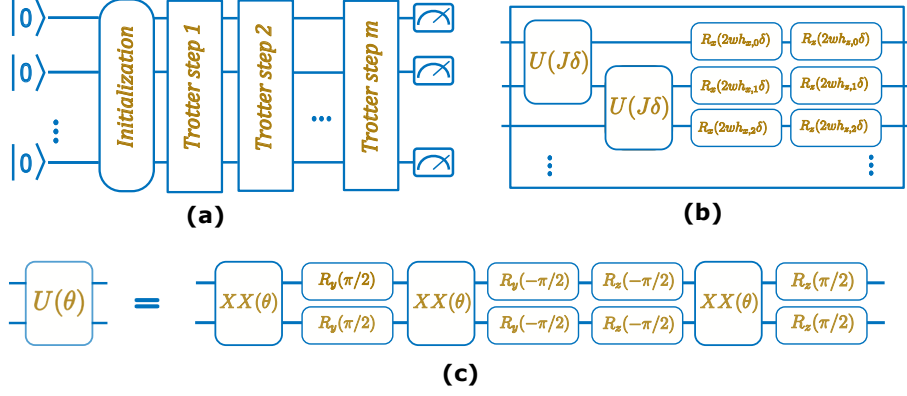


Figure 5.1: The circuit used to simulate time-evolution under the Heisenberg model Hamiltonian. (a) After the qubits are prepared into the desired initial state, $m = 6$ Trotter steps are used to evolve the system to time t . All qubits are then measured in the z -basis. (b) Each Trotter step consists of several one- and two-qubit gates, as described by Eq. 5.10. The single-body interactions are implemented as rotations about the X or Z axis (R_x and R_z gates). (c) The two-body interactions $\hat{U}(J\delta)$ are implemented as three XX (Ising) gates sandwiched between single-qubit rotations. This segment of the circuit is equivalent to a sequential application of XX , YY , ZZ gates, which describes evolution under the Heisenberg interaction exactly.

5.2 Spectral function

For a given Hamiltonian H with eigenstates $|\phi_m\rangle$ and corresponding eigenenergies E_m , the spectral function of an operator \hat{a} is defined as

$$A(\omega) = \sum_{l,k} |\langle \phi_l | \hat{a} | \phi_k \rangle| \delta_{\omega, E_k - E_l}, \quad (5.5)$$

where δ is the Kronecker delta, ω is the frequency. We take $\hbar = 1$ through out this study.

For our study, we choose the one-dimensional Heisenberg model with random

fields along two axes which for n spins has the Hamiltonian

$$H = J \sum_{i=1}^{n-1} \hat{\vec{\sigma}}_i \cdot \hat{\vec{\sigma}}_{i+1} + w \left(\sum_{i=1}^n h_i^x \hat{\sigma}_i^x + \sum_{i=1}^n h_i^z \hat{\sigma}_i^z \right). \quad (5.6)$$

Here, $\hat{\vec{\sigma}}_i = (\hat{\sigma}_i^x, \hat{\sigma}_i^y, \hat{\sigma}_i^z)$ are the Pauli operators. J determines the nearest neighbor coupling strength. w is the a coefficient that determines the global strength of the external fields. The disorder in the model comes from the fields h_i^x and h_i^z , which are random variables chosen from a uniform probability distribution between -1 and 1. In the limit $w/J \rightarrow 0$, the system is in the thermalized phase and for $w/J \rightarrow \infty$, it is in the localized phase. This model is known to have a phase transition at $w/J \sim 6$ [47]. We set $w = 1$.

For a local operator \hat{a} such as a single spin Pauli operator, the spectral function $A(\omega)$ for n spins at $J = 0$ will consist of $2n$ delta functions at $\pm 2w\sqrt{h_i^{x2} + h_i^{z2}}$. The average spacing between the peaks is $\sim w/n$. For $0 < J \ll w$, each peak of the non-interacting spectrum will split into a cluster of delta functions with a hierarchy of energy gaps [1, 46]. The full width of the cluster is $J \exp(-1/\xi)$, where ξ is the localization length which is an increasing function of J/w . Fig. 5.2 shows a schematic example of this. When J is large enough, no discrete structure will remain and the spectrum will be continuous, indicating a transition to thermalization. In our experiment, we do not sample at enough points in the time-evolution to resolve the hierarchical structure of the energy gaps but we can measure the total broadening of the original spectral line.

When the system is coupled to a thermal bath, the spectral lines broaden

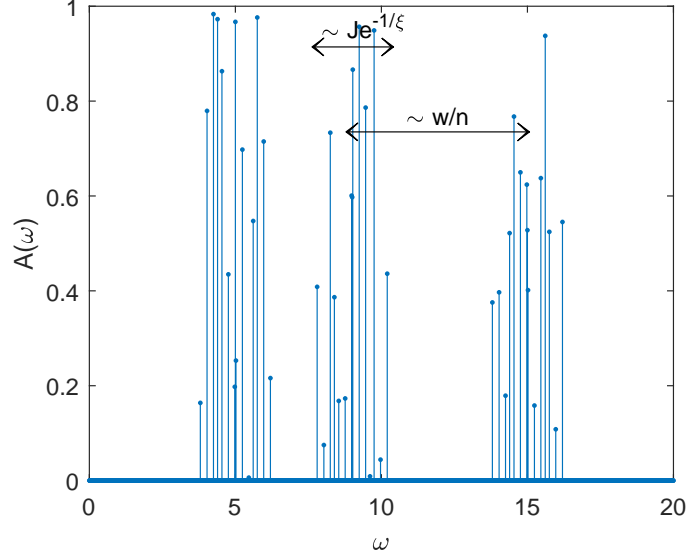


Figure 5.2: A schematic example of the splitting of spectral lines according to the theoretical model in [1] for a localized system in the limit $J \ll w$.

and the discrete structure gradually vanishes as the coupling strength increases. It disappears only when the coupling becomes comparable to J . In contrast, in the thermal phase, $A(\omega)$ is expected to become an increasingly smooth function of energy as n increases. Here we construct the probability distribution of the widths of these clusters from the linewidths Γ of the peaks in the spectrum.

After averaging over spin locations and disorder realizations, the ratio of the averaged spectrum of the localized phase to that of the thermalized phase should go to zero as $\omega \rightarrow 0$ [1]. This implies that in the localized phase, local operators are less likely to connect nearby energy eigenstates, instead mixing them and giving rise to level repulsion. The width of the resulting spectral soft-gap is a function of w and remains finite in the thermodynamic limit. In contrast, in the thermalized phase, the spectral function decays as ω increases for $\omega < J$ [29].

As we now show, the spectral functions can be approximated on a quantum

computer by Hamiltonian time evolution, followed by measurement of the expectation value of the local operator and a Fourier transform of the resulting time series data. At $t = 0$, let the system be in the state

$$|\Psi(t = 0)\rangle = \sum_k c_k |\phi_k\rangle, \quad (5.7)$$

where $|k\rangle$ are the eigenstates of the system. The expectation value of operator \hat{a} at time t is

$$\langle \hat{a}(t) \rangle = \sum_{k,l} c_k c_l^* a_{kl} e^{-i(E_k - E_l)t}, \quad (5.8)$$

where $a_{kl} = \langle \phi_l | \hat{a} | \phi_k \rangle$. The absolute value of the Fourier transform of the above expression gives

$$\mathcal{F}\{\langle \hat{a} \rangle\} = \sum_{k,l} |c_k c_l^* a_{kl}| \delta_{\omega, E_k - E_l}. \quad (5.9)$$

Note the similarity to the spectrum of \hat{a} from Eq. 5.5, especially when the initial state (Eq. 5.8) is spread over the eigenstates of the system. In the experiment we use $\hat{a} = \hat{\sigma}_i^z$, and initialize the qubits in the $|+\rangle$ state, which is an equal superposition of the two eigenstates of \hat{a} . We measure in the z basis at the end, in order to extract the spectral function corresponding to $\hat{\sigma}_i^z$ for qubit i . When discussing the experimental measurement of $\overline{A}(\omega)$, we are referring to the expression in Eq. 5.9 after disorder-averaging.

5.3 simulation results

We first introduce the simulation results to give an ideal picture of what we are trying to observe.

Fig. ?? (a) shows simulations for a 7 spin system for which $J = 1$ and w is varied using 500 Trotter steps for each sample time and averaging over 100 disorder realizations. As w increases, the maximum of the spectrum shifts right while its magnitude at low frequencies goes down. The high-frequency regime represents one-body physics for which the energy scale is set by w . Therefore, in Fig. ?? (b) which shows the same spectra on a plot where the frequency ω has been scaled by the disorder magnitude w , the curves now lie on top of each other at high-frequencies. Note that this plot is equivalent to fixing $w = 1$ while varying J , and plotting the spectrum versus ω as is done for the data in Fig. ?? of the paper. At low frequencies, there is a suppression of the spectral function as the ratio w/J increases, consistent with the results presented in the main text for a 3 spin system.

5.4 Experimental results

The study of disordered systems requires averaging over many disorder realizations. We run the experiment for different values of the coupling strength: $J=0.1$, $J=0.3$ and $J=0.7$, using 24 circuits each to sample instances of disorder. The choice of sample size for each coupling strength can, according to simulation, guarantee low enough statistical uncertainty. The circuits are generated beforehand and fed

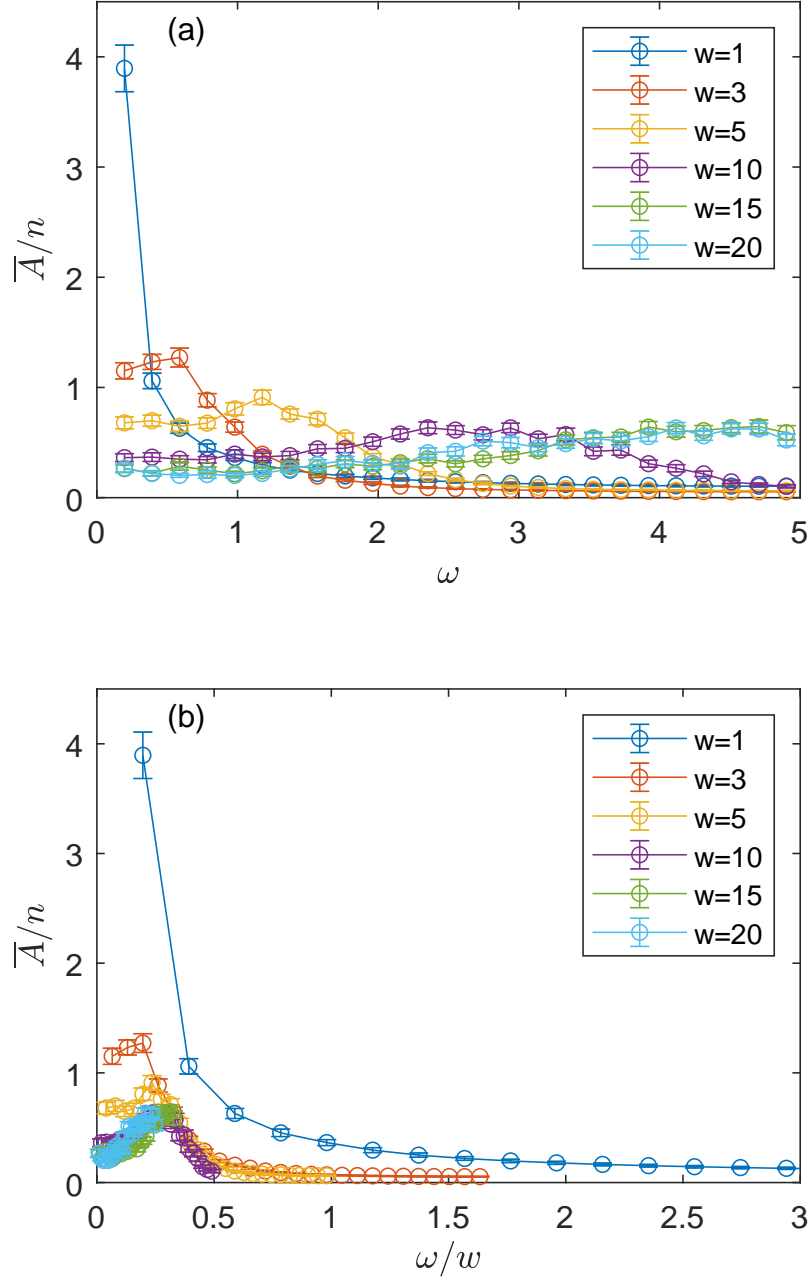


Figure 5.3: (a) The simulated spectrum for $n = 7$ spins averaged over 100 disorder configurations for fixed $J = 1$, while w is varied. (b) The same plot with the frequency scaled by w .

to the experiment control computer in batches. As long as the ions stay trapped, the system automatically executes the circuits sequentially. The quantum circuit corresponding to the time evolution under H is shown in Fig. 5.1. The Hamiltonian evolution cannot be implemented exactly on digital quantum computers (with finite number of gates). We use Trotterization to decompose it into one and two-qubit gates (Fig. 5.1 (b)). The two-qubit interaction is exactly captured by the unitary U since XX , YY and ZZ terms commute with each other. Each Trotter step achieves the following unitary:

$$\hat{U}_H = \prod_{k=1}^n (e^{-ih_{z,k}\hat{\sigma}_k^z\delta} e^{-ih_{x,k}\hat{\sigma}_k^x\delta}) \prod_{k=1}^{n-1} (e^{-iJ\hat{\sigma}_k^z\hat{\sigma}_{k+1}^z\delta} \times e^{-iJ\hat{\sigma}_k^y\hat{\sigma}_{k+1}^y\delta} e^{-iJ\hat{\sigma}_k^x\hat{\sigma}_{k+1}^x\delta}). \quad (5.10)$$

The total evolution time is given by $t = m\delta$. It is straightforward to extend the circuit to an arbitrary number of qubits.

The time evolution is sampled at 10 different equally spaced intervals between $0 < t \leq 10$. The expectation value of $\hat{\sigma}_z$ at $t = 0$ is trivially known to be zero. We use a constant number $m = 6$ Trotter steps for each sample time making the Trotter angle $\delta = t/6$. This is in contrast to the more widely-used method of Trotterization where δ stays fixed and the number of Trotter steps increases with time. Since the number of Trotter steps is constant no matter the time, the magnitude of experimental error is the same in every circuit [48]. We will see that this becomes critical to the error mitigation technique we introduce below.

Each circuit is measured 2400 times to sufficiently reduce the statistical error.

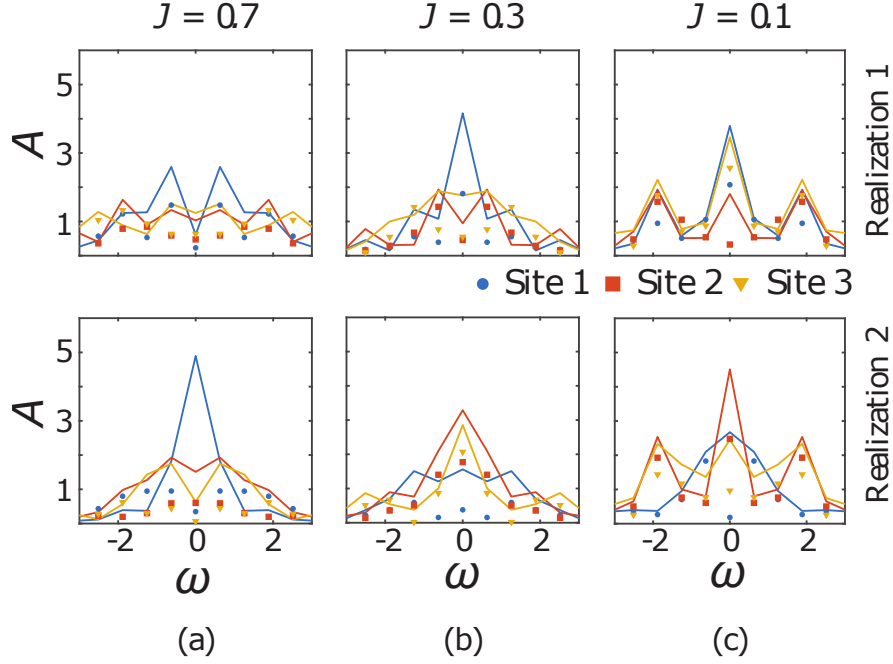


Figure 5.4: The spectrum of $\hat{\sigma}_i^z$ at different values of J (with $w = 1$) for a 3 site system for two sample disorder realizations (top and bottom). Each panel show both simulation (curves) and experimental (symbols) results. The different colors are for the different sites. The lack of distinguishing characteristics between the spectra at different values of J for individual samples shows the necessity of averaging over several disorder realizations.

We initialize all qubits into $|+\rangle$ states with Hadamard gates. Because the $|+\rangle^{\otimes n}$ state is an eigenstate of the \hat{U} operator, we can skip the application of the first set of \hat{U} gates on all qubits. Each circuit thus consists of 30 two-qubit gates and 116 single-qubit gates. We run a total of 792 circuits to obtain the data in this paper.

A discrete Fourier transform is then applied to the time series for each instance to obtain the spectrum. In the thermodynamic limit, $J = 0.1$ lies in the localized phase, $J = 0.7$ in the thermalized and $J = 0.3$ near the phase transition. For a small system, there is no sharp phase transition but we expect to see a change from thermalized to localized behavior as we lower the value of J .

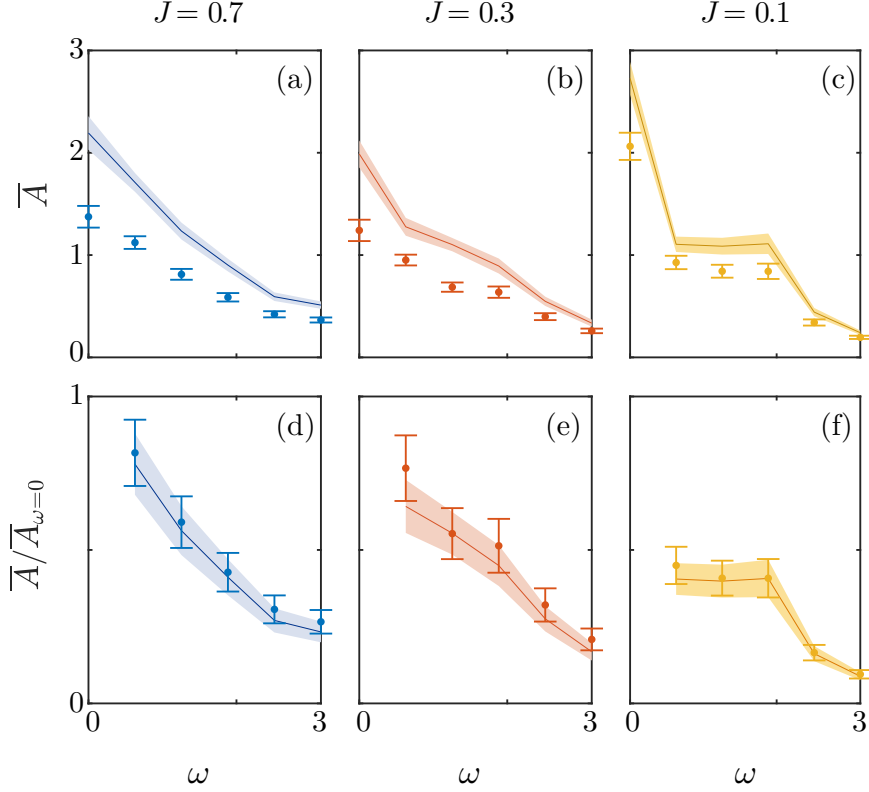


Figure 5.5: (a)-(c) The spectral function of $\hat{\sigma}_i^z$ averaged over position and 24 disorder realizations for different values of J (with $w = 1$) for a 3-site system. Lines show simulation results and circles show experimental results. (d)-(f) The spectral function normalized by its value at $\omega = 0$. At $J = 0.7$, the spectral response increases as ω decreases while at $J = 0.1$, the spectrum is damped at low frequencies. The points at $\omega = 0$ in the top row are discontinuous with the rest of the curves since they arise from the diagonal elements of the observable in the eigenstate basis which have qualitatively different behavior than the off-diagonal ones.

Fig. 5.4 shows several instances of the measured spectrum for $\hat{\sigma}_i^z$. The spectrum is symmetric about $\omega = 0$. We note that the experimental data is significantly damped compared to the simulation. The figure also shows the necessity of averaging over several realizations in the study of disordered systems since the behavior of the system in the thermodynamic limit cannot be determined from the behavior of a finite-size individual disorder realization.

We next average the spectral functions over lattice sites and disorder configu-

rations. The results are shown in Fig. 5.5 (a)-(c). The value at $\omega = 0$ arises from the diagonal elements and is related to the equilibrium value of the observable at infinite temperature, whereas the behavior at $\omega > 0$ gives the dynamical response of the system. The simulation curves show that the value of the spectral function at low-frequencies drops as J decreases. Simulation results for a larger system size show similar behavior (Appendix B).

5.5 Error mitigation

While the experimental data follow the trend of the simulation for each value of J , the error obscures the difference between the spectra at different values of J . To address this, we now introduce an error mitigation technique.

It has been shown that the error in the mean value of an observable measured after the application of a set of random circuits with the same structure can be well-approximated by a depolarizing error model, whatever the origin of the noise [49]. Therefore, the mean density matrix after the application of the unitaries $\{\hat{U}_H\}$ in Eq. 5.10 to an initial state $|\Psi_0\rangle$ is

$$\bar{\rho} = \epsilon_m \overline{\hat{U}_H |\Psi_0\rangle \langle \Psi_0| \hat{U}_H^\dagger} + (1 - \epsilon_m) \frac{I}{D}, \quad (5.11)$$

where I is the identity matrix and $D = 2^n$. $\epsilon_m = p^m$, where p is the disorder-averaged depolarization fidelity per Trotter step. The expectation value of \hat{a} at time

t is

$$\overline{\langle \hat{a} \rangle}(t) = \text{Tr}(\bar{\rho}(t))\hat{a} = p^m \text{Tr}\left(\overline{U_H(t) |\Psi_0\rangle \langle \Psi_0| U_H^\dagger(t) \hat{a}}\right). \quad (5.12)$$

Since the same number of Trotter steps m is used for measuring at all times, the corresponding spectrum obtained by Fourier transform becomes

$$\overline{A}(\omega) = p_c^m \int \text{Tr}\left(\overline{U_H(t) |\Psi_0\rangle \langle \Psi_0| U_H^\dagger(t) \hat{a}}\right) e^{-i\omega t} dt. \quad (5.13)$$

If we now divide by the zero-frequency component,

$$\frac{\overline{A}(\omega)}{\overline{A}(0)} = \frac{\int \text{Tr}\left(\overline{U_H(t) |\Psi_0\rangle \langle \Psi_0| U_H^\dagger(t) \hat{a}}\right) e^{-i\omega t} dt}{\int \text{Tr}\left(\overline{U_H(t) |\Psi_0\rangle \langle \Psi_0| U_H^\dagger(t) \hat{a}}\right) dt}, \quad (5.14)$$

p^m is canceled. We should thus essentially get a noiseless signal after the normalization. Fig. 5.5 (d)-(f) shows the normalized spectra. We see that the data match the normalized curves within the statistical uncertainty, especially at $J = 0.1$ and $J = 0.7$ which are deep in the localized and thermalized phase respectively. Note that the estimated fidelity of the quantum computation obtained by multiplying the fidelities of the individual gates is only 54%, making the experimental reproduction of the theoretical curves in Fig. 5.5(b) remarkable.

We next test the discreteness of the distribution by studying the linewidths Γ of the peaks in the spectrum. We expect the peaks in the localized phase to be narrower than in the thermalized phase on average. As shown in the Appendix of Ref [46], the distribution should be skewed to larger linewidths, indicating the

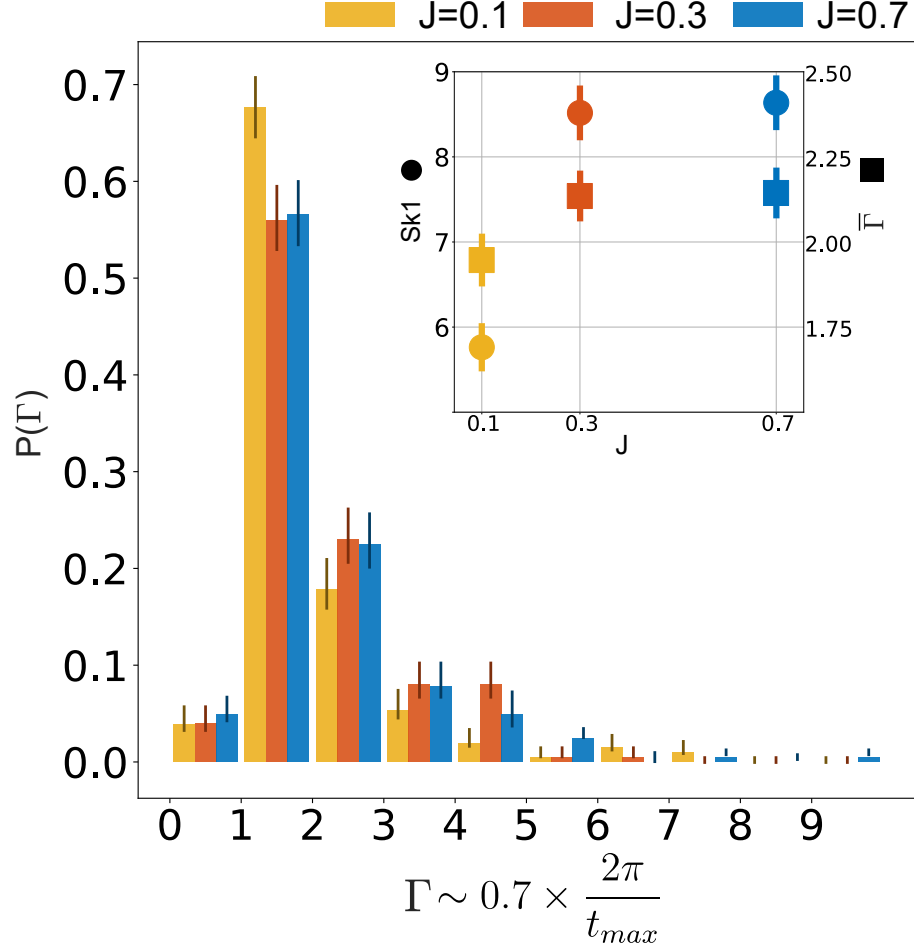


Figure 5.6: The distribution histogram of linewidths Γ calculated as described in the text at different values of J from data taken on 3 qubits for 24 realizations. The bins are $[0-1]$, $[1-2]$, ... etc. The number of peaks used to generate the distribution is ~ 200 for each value of J . We derive the errorbars shown in the plot by assuming each bin approximately follows a binomial distribution. The inset shows the Pearson's first coefficient of skewness, $Sk1$, and the average linewidth, $\bar{\Gamma}$.

presence of resonant clusters of spins.

We use the following procedure to obtain the probability distribution of the linewidths:

1. Fit individual spectra (such as those in Fig. 5.4) with an interpolating polynomial and find the peaks.
2. For each peak, find the width corresponding to half the prominence of the peak.
3. Plot a normalized histogram corresponding to the probability distribution of the line-widths thus obtained.

The probability distributions $P(\Gamma)$ are shown in Fig. 5.6 for different values of J . As expected, they are skewed to the right. In the inset of Fig. 5.6, we show that both the average linewidth as well as the skewness, which measures the probability of resonant clusters, are smaller at $J = 0.1$ than at larger values of J .

Chapter 6: Hybrid Quantum Algorithms

Hybrid quantum algorithms [50] use both classical and quantum resources to solve potentially difficult problems. This approach is particularly promising for current quantum computers of limited size and power [4]. Several variants of hybrid quantum algorithms have recently been demonstrated, such as the Variational Quantum Eigensolver (VQE) for quantum chemistry and related applications [51, 52, 53, 54, 55], and the Quantum Approximate Optimization Algorithm (QAOA) for graph or other optimization problems [56, 57].

All hybrid algorithms share a similar intuition: using variational circuits to answer questions. These variational circuits, also known as the ansatzes, give the correct answers with some specific set of parameters. We cannot directly derive the correct parameters based on our understanding of the questions. But we can use a cost function to quantify how good the answers given by a specific set of parameters are. Thus, we can repeatedly implement an optimization loop to look for the proper parameters that give us good answers.

In this chapter, we will present two types of hybrid algorithms with their applications: a data-driven circuit learning algorithm for generative modeling and (QAOA) quantum approximate optimization algorithm for efficient quantum state

preparation

6.1 Data Driven Circuit learning for Generative Modeling

Generative modeling aim to learn representations of data in order to make subsequent tasks easier. Applications of generative modeling include computer vision [58], speech synthesis [59], the inference of missing text [60], de-noising of images [61], and chemical design [62]. Here, we apply a hybrid quantum learning scheme on a trapped ion quantum computer [10] to accomplish a generative modeling task.

Data-driven quantum circuit learning (DDQCL) is a hybrid framework for generative modeling of classical data where the model consists of a parameterized quantum circuit [63]. The model is trained by sampling the output of a quantum computer and updating the circuit parameters using a classical optimizer. After convergence, the optimal circuit produces a quantum state that captures the correlations in the training data sets. Hence the trained circuit serves as a generative model for the training data. Theoretical results suggest that such generative models have more expressive power than widely used classical neural networks [64, 65]. This is because instantaneous quantum polynomial circuits – special cases of the parameterized quantum circuits used for generative modeling – cannot be efficiently simulated by classical means.

The Bars-and-Stripes (BAS) data set is a canonical body of synthetic data for generative modeling [66]. It can be easily visualized in terms of images containing horizontal bars or vertical stripes, where each pixel represents a qubit. Here, we

use the uniformly distributed 2-by-2 BAS shown in Fig.6.1 in a proof-of-principle generative modeling task. We compare the performance of different classical optimization algorithms and conclude that Bayesian optimization shows significant advantages over Particle Swarm Optimization for this task.

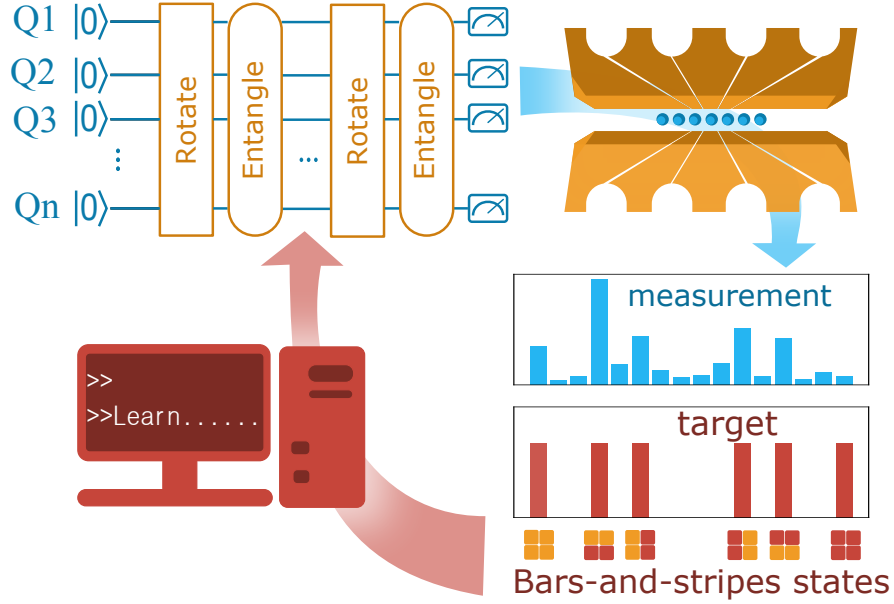


Figure 6.1: Data-driven quantum circuit learning (DDQCL) is a hybrid quantum algorithm scheme that can be used for generative modeling, illustrated here by the example of 2-by-2 Bars and Stripes (BAS) data. From top left, clockwise: A parameterized circuit is initialized at random. Then at each iteration, the circuit is executed on a trapped ion quantum computer. The probability distribution of measurement is compared on a classical computer against the BAS target data set. Next, the quantified difference is used to optimize the parametrized circuit. This learning process is iterated until convergence.

6.1.1 The variational circuits (ansatzes)

The quantum circuits are structured as layers of parameterized gates. We use two types of layers, involving single-qubit rotations and two-qubit entangling gates. A single-qubit layer sandwiches an X-rotation between two Z-rotations on

each qubit i , or $R_z^{(i)}(\alpha_i)R_x^{(i)}(\beta_i)R_z^{(i)}(\gamma_i)$, involving twelve rotation parameters for the four qubits (see Fig. 6.2). An entangling layer applies Ising or XX gates between all pairs of qubits according to any imposed connectivity graph. This is expressed as a sequence of $XX^{i,j}(\chi_{i,j})$ operations as shown in Fig. 6.2), with up to six entangling parameters [10] for four qubits. Due to the universality of this gate set, a sufficiently long sequence of layers of these two types can produce arbitrary unitaries.

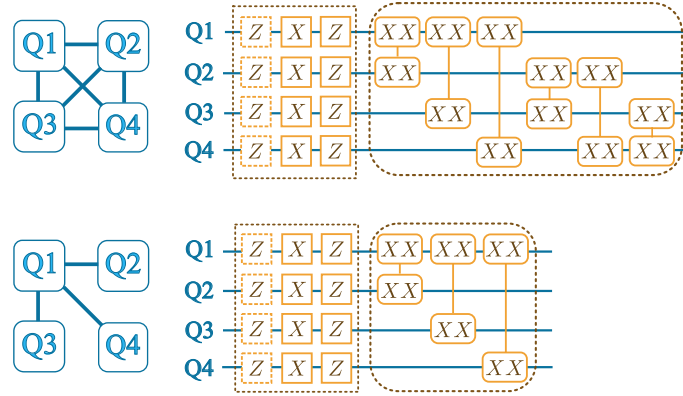


Figure 6.2: Connectivity graphs and corresponding training circuits. Top: Fully-connected training circuit layer, with layers of rotations (square boxes) and entanglement gates (rounded boxes) between any pair of the four qubits. Bottom: Star-connectivity training circuit layer, with restricted entangling gates. In either case, each rotation (denoted by X or Z) and each entanglement gate (denoted by XX) includes a distinct control parameter, for a total of 18 parameters for the fully-connected circuit layer and 15 parameters for the star-connected circuit layer. We remove the first Z rotation (dashed square box) acting on the initial state $|0\rangle$, resulting in 14 and 11 parameters, respectively. The connectivity figures on the left define the mapping between the four qubits and the pixels of the BAS images (see Fig.6.1).

At the start of DDQCL, all the rotation and entangling parameters are initialized with random values. Next the circuit is repeatedly executed on the trapped ion quantum computer in order to reconstruct the state distribution. A classical computer then compares the measured distribution with the target distribution and

quantifies the difference using a cost function (see Method for details). A classical optimization algorithm then varies the parameters. We iterate the entire process until convergence.

We impose two distinct connectivity graphs in a four-qubit circuit: all-to-all and star, as shown in Fig. 6.2. With star connectivity, entanglement between certain qubit-pairs cannot occur within a single gate layer, which means more layers are necessary for certain target distributions. Comparing the training process between circuits of different connectivity provides insight into the performance of DDQCL algorithms on platforms with more limited interaction graphs.

For each connectivity graph, we add layers until the goal of reproducing the BAS data with the trained model is achieved. The match between training data and model is limited by noise, experimental throughput rate (how fast the system can process circuits), and sampling errors. The cost function used in optimization scores the result, but a successful training process must be able to generate data that can be qualitatively recognized as a BAS pattern to ensure that the system provides usable results in the spirit of generative modeling in machine learning [67].

6.1.2 The classical optimizer

We now describe the classical optimization strategies for the training algorithm. Although gradient-based approaches were recently proposed for DDQCL [68], we employ gradient-free optimization schemes that appear less sensitive to noise and experimental throughput. We explore two such schemes: Particle Swarm Optimiza-

tion (PSO) [69] and Bayesian Optimization (BO) [70]. We explore two different classical optimizer in this study: Particle Swarm Optimization(PSO) and Bayesian Optimization(BO).

PSO is a gradient-free optimization method inspired by the social behaviour of some animals. Each particle represents a candidate solution and moves within the solution space according to its current performance and the performance of the swarm. Three hyper-parameters control the dynamics of the swarm: a cognition coefficient c_1 , a social coefficient c_2 , and an inertia coefficient w [69].

Concretely, each particle consists of a position vector θ_i and a velocity vector v_i . At iteration t of the algorithm, the velocity of particle i for the coordinate d is updated as

$$v_{i,d}^{(t+1)} = wv_{i,d}^{(t)} + c_1r_{1,d}^{(t)}(p_{i,d}^{(t)} - \theta_{i,d}^{(t)}) + c_2r_{2,d}^{(t)}(g_d^{(t)} - \theta_{i,d}^{(t)}), \quad (6.1)$$

where $r_{1,d}^{(t)}$ and $r_{2,d}^{(t)}$ are random numbers sampled from the uniform distribution in $[0,1]$ for every dimension and every iteration, $p_i^{(t)}$ is the particle's best position, $g^{(t)}$ is the swarm's best position. The position is then updated as

$$\theta_i^{(t+1)} = \theta_i^{(t)} + v_i^{(t)}, \quad (6.2)$$

In our problem, each particle corresponds to a point in parameter space of the quantum circuit. For example, in the fully connected circuit with two layers, each particle consists of an instance of the 14 parameters. Recall, however, that parameters are angles and are therefore periodic; We customized the PSO updates

above to use this information. In Eq. (6.1), $p_{i,d}^{(t)}$ and $\theta_{i,d}^{(t)}$ can be thought of as two points on a circle. Instead of using the standard displacement $p_{i,d}^{(t)} - \theta_{i,d}^{(t)}$, we use the angular displacement, that is the signed length of the minor arc on the unit circle. We use the same definition of displacement for the swarm’s best position $g_{i,d}^{(t)}$. Finally, in Eq. (6.2), we make sure to express angles always using their principal values.

In our experiments, we set the number of particles to twice the number of parameters of the circuit. Position and velocity vectors of each particle are initialized from the uniform distribution. For the coefficients we use $c_1 = c_2 = 1$ and $w = 0.5$.

Bayesian Optimisation is a powerful global optimisation paradigm. It is best suited to finding optima of multi-modal objective functions that are expensive to evaluate. There are two main features that characterize the a BO process: the surrogate model and an acquisition function.

The surrogate model is non-parametric model of the objective function. At each iteration, the surrogate model is updated using the sampled points in parameter space. The package used in this study is OPTaaS by MindFoundry. It implements the surrogate model as regression using Gaussian Process[71]. A kernel (or correlation function) characterizes the Gaussian process, we use a Matern 5/2 as it provides the most flexibility.

The acquisition function is computed from the surrogate model. It is used to select points for evaluation during the optimization. It trades off exploration against exploitation. The acquisition function of a point has a high value if the cost function is expected to give a significant improvement over historically sampled points, or if

the uncertainty of the point is high, according to the surrogate model. A simple and well known acquisition function, Expected Improvement[72], is employed here.

In our case, OPTaaS also leverages the cyclic symmetry of the angles by embedding the parameter space into a metric space with the appropriate topology, effectively allowing the Gaussian Process surrogate model to be placed over a hyper-torus, rather than a hyper-cube. This greatly alleviates the so-called curse of dimensionality[73], and allows for much more efficient use of samples of the objective function.

It is key in Bayesian Optimisation to adequately optimise the acquisition function during each iteration. OPTaaS puts considerable computational resources towards this non-convex optimisation problem.

There are two major reasons why the BO out performs PSO in our specific case. First, PSO spends significant amount of computation resource exploring trajectories far from optimal, while BO mitigates it by the use of acquisition function. Second, the maintenance of the surrogate model enable us to make much better use of the information from the historical exploration of the parameter space.

6.1.3 The cost Function

We use a cost function to quantify the difference between the target BAS distribution and the experimental measurements of the circuit. The cost functions used to implement the training are variants of the original Kullback-Leibler Divergence

(D_{KL}) [74]:

$$D_{KL}(p, q) = - \sum_i p(i) \log \frac{q(i)}{p(i)}. \quad (6.3)$$

Here p and q are two distributions.

$D_{KL}(p, q)$ is an information theoretic measure of how two probability distribution differ. If base 2 for the logarithm is used, it quantifies the expected number of extra bits required to store samples from p when an optimal code designed for q is used instead. It can be shown that $D_{KL}(p, q)$ is non-negative, and is zero if and only if $p=q$. However, it is asymmetric in the arguments and does not satisfy the triangle inequality. Therefore $D_{KL}(p, q)$ is not a metric.

The KL divergence is a very general measure, but it is not always well-defined, e.g. if an element of the domain is supported by p and not by q , the measure will diverge. This problem may occur quite often if $D_{KL}(p, q)$ is estimated from samples and if the dimensionality of the domain is large. For PSO, we use the clipped negative log-likelihood cost function [63],

$$C_{nll} = - \sum_i p(i) \log \{\max[\epsilon, q(i)]\}. \quad (6.4)$$

Here we set p as the target distribution. Thus Eq.6.4 is equivalent to Eq.6.3 up to a constant offset, so the optimization of these two functions is equivalent. ϵ is a small number (0.0001 here) used to avoid a numerical singularity when $q(i)$ is measured to be zero.

For BO, we use the clipped symmetrized Kullback-Leibler (KL) divergence as

the cost function

$$\tilde{D}_{KL}(p, q) = D_{KL}[\max(\epsilon, p), \max(\epsilon, q)] + D_{KL}[\max(\epsilon, q), \max(\epsilon, p)]. \quad (6.5)$$

This is found to be the most reliable variant of D_{KL} for BO.

Experimental Results

Results from PSO optimization are shown in Fig. 6.3. We first simulate the training procedure using a classical simulator in place of the quantum processor (orange plots in Fig. 6.3). Since the PSO method is sensitive to the initial "seed" values of the particles, we simulate the convergence for many different random seeds (see Fig.6.3). We choose a seed that converges quickly and reliably under simulated sampling error to start the training procedure on the trapped ion quantum computer illustrated in Fig.6.1. We iterate the training until it converges (blue plots in Fig.6.3). In practice, which seeds are successful is unknown, and different seeds need to be tried experimentally until a good model is obtained. This incurs an additional cost in the form of multiple independent DDQCL training rounds.

For all-to-all connectivity, we find that a circuit with one rotation gate layer and one entangling gate layer is able to produce the desired BAS distribution (Fig. 6.3a). This is not the case for the star-connected circuit, with the closest state having two additional components in the superposition (states 6 and 9 in Fig. 6.3b). With two additional layers, the star-connected circuit is able to model the BAS distribution (orange plots of Fig. 6.3c). In the experiment however (blue plots in

Fig. 6.3c), the PSO is unable to converge to an acceptable solution even using the best pre-screened seed value and sufficient sample statistics. We conclude that PSO fails because the throughput rate is too low for effectively training the circuit in the face of gate imperfections.

For these reasons, we instead employ a Bayesian optimization scheme for the circuit training procedure. We find that all circuits experimentally converge in agreement with the simulations, as shown in Fig. 6.4. Moreover, even the star-connected circuit with four layers now produces a recognizable BAS distribution (Fig. 6.4c). In contrast to PSO, BO dramatically reduces the number of samples needed for training and does not require any pre-selection of random seeds or other prior knowledge of the cost-function landscape.

BO updates the surrogate model using the experimental result of every iteration. Therefore, the classical part of each BO iteration consumes more time than with PSO, where the time cost on the classical optimizer is negligible. However, the BO procedure converges faster to the desired BAS distribution. More generally, these examples highlight the need to balance quantum and classical resources in order to produce acceptable performance and run time in a hybrid quantum algorithm.

As a measure of the performance of the various training procedures, we compute the Kullback-Leibler (KL) divergence [74] and the qBAS score (an alternative performance measure suggested in [63]) of the experimental results at the end of each DDQCL training run, shown in Table 6.1. We also compute the entanglement entropy (S) averaged over all two plus two qubit partitions assuming a pure state [75],

estimated via simulation of the quantum state from the trained circuits. The entanglement entropy quantifies the level of entanglement of a state, thus indicates how difficult it is to produce such state. This metric shows that the successfully trained circuits generate states that are consistent with a high level of entanglement. As a reference, the entanglement entropy of a GHZ state over any partition is $S = 1$.

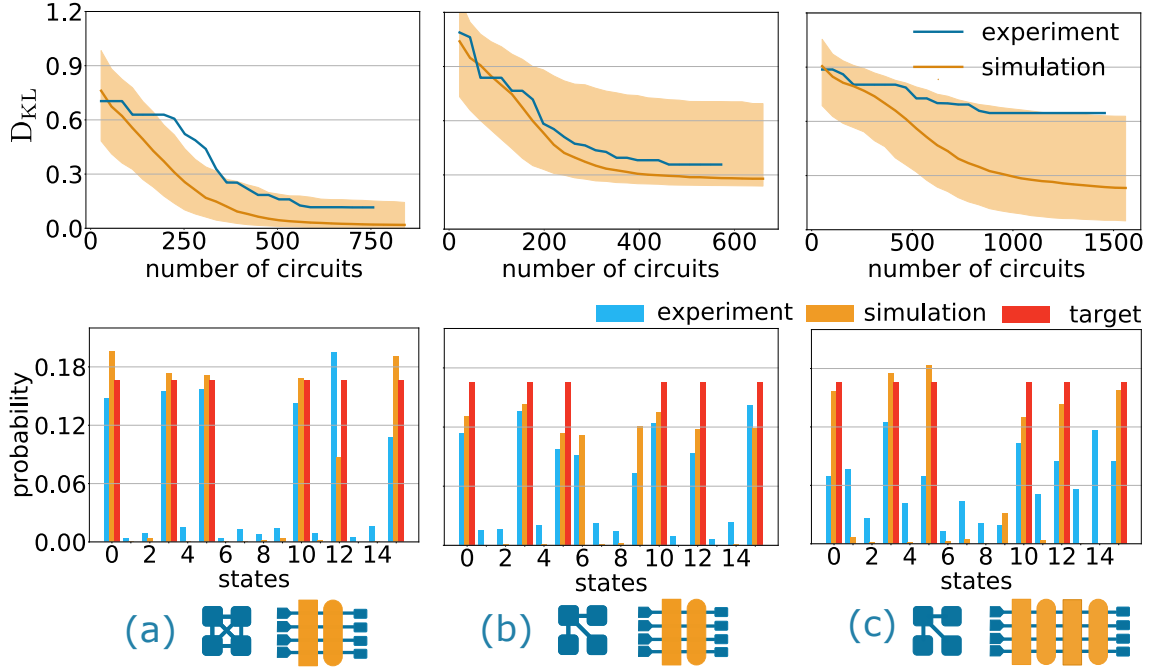


Figure 6.3: Quantum circuit training results with Particle Swarm optimization (PSO), with simulations (orange) and trapped ion quantum computer results (blue). Column (a) corresponds to a circuit with one layer of single qubit rotations (square boxes) and one layer of entanglement gates (rounded boxes) of all-to-all connectivity. The circuit converges well to produce the bars-and-stripes (BAS) distribution. Columns (b) and (c) correspond to a circuit with two and four layers and star-connectivity, respectively. In (b), the simulation shows imperfect convergence with two extra state components (6 and 9), due to the limited connectivity, and the experimental results follow the simulation. In (c), the simulation shows convergence to the BAS distribution, but the experiment fails to converge despite performing 1,400 quantum circuits. The optimization is sensitive to the choice of initialization seeds. To illustrate the convergence behavior, the shaded regions span the 5th-95th percentile range of random seeds (500 for (a) and (b), 1000 for (c), and the orange curve shows the median. The two-layer circuits have 14 and 11 parameters for (a) all-to-all- and (b) star-connectivity, while the (c) star-connectivity circuit with four layers has 26 parameters. The number of PSO particles used is twice the number of parameters, and each training sample is repeated 5000 times. Including circuit compilation, controller-upload time, and classical PSO optimization, each circuit instance takes about 1 min to be processed, in addition to periodic interruptions for the re-calibration of gates.

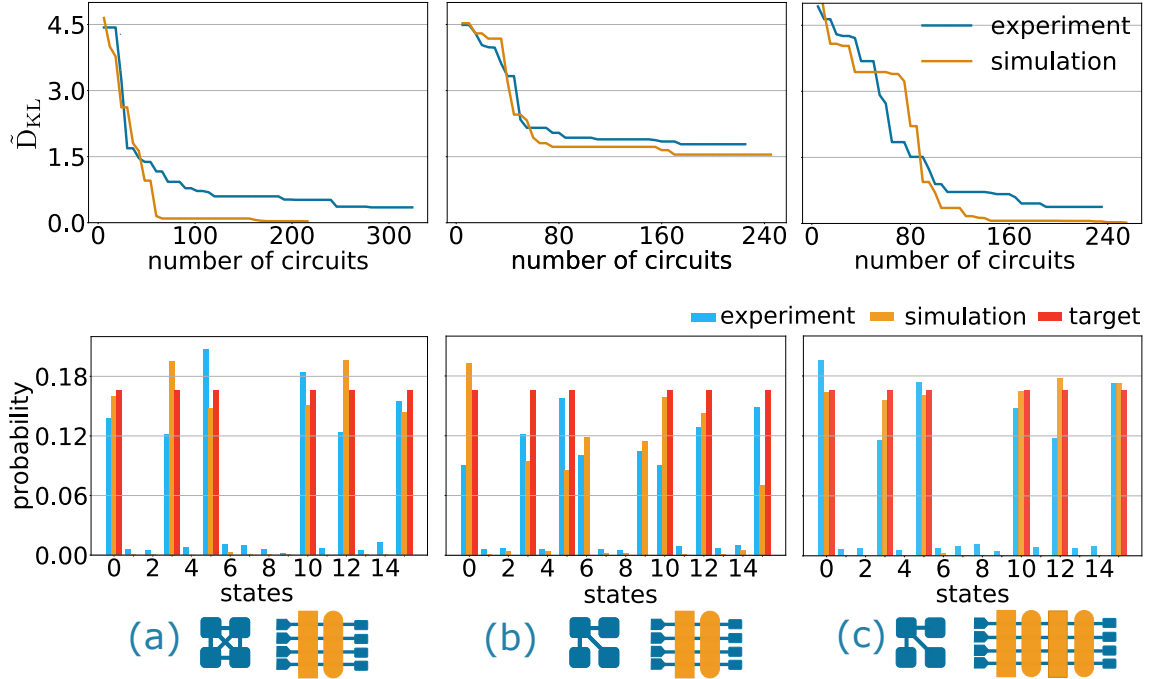


Figure 6.4: Quantum circuit training results with Bayesian optimization (BO), with simulations (orange) and trapped ion quantum computer results (blue). Column (a) corresponds to a circuit with two layers of gates and all-to-all connectivity. Columns (b) and (c) correspond to a circuit with two and four layers and star-connectivity, respectively. Convergence is much faster than with PSO (Fig. 6.3). Unlike the PSO results, the four-layer star-connected circuit in (c) is trained successfully, and no prior knowledge enters BO process. As before, the two-layer circuits have 14 and 11 parameters for (a) all-to-all- and (b) star-connectivity, while the (c) star-connectivity circuit with four layers has 26 parameters. We use a batch of 5 circuits per iteration, and each training sample is repeated 5000 times. Including circuit compilation, controller-upload time, and BO classical optimization, each circuit instance takes 2-5 minutes, depending on the amount of accumulated data.




circuits	optimizer	D_{KL}	qBAS score	S
	PSO	0.116	0.91	1.628
	BO	0.094	0.91	1.659
	PSO	0.357	0.74	0.9950
	BO	0.328	0.77	0.9999
	PSO	0.646	0.59	0.8867
	BO	0.100	0.91	1.709

Table 6.1: KL divergence (D_{KL} , see Materials and Methods), qBAS score, and entanglement entropy (S) for the state obtained at the end of each of the DDQCL training on hardware, for various circuits and classical optimizers used.

6.2 Efficient Thermal State Preparation with the Quantum Approximate Optimization Algorithm (QAOA)

Progress in the control of synthetic quantum systems such as superconducting qubits [76] and trapped ions [77] has enabled continual advances in the depth of quantum computer circuits and the complexity of quantum simulations. As the number of qubits and their coherence times increase, such systems have the potential to simulate highly non-trivial macroscopic quantum phenomena.

While there has been progress in the preparation of entangled quantum states such as squeezed or “cat” states [78, 79], much less attention has been paid to generating thermal (Gibbs) states of a many-body Hamiltonian, even though these states underpin phenomena ranging from high temperature superconductivity [80] to quark confinement in quantum chromodynamics [81].

The simulation of many-body thermal states challenges currently available quantum platforms, owing to the required level of control over both the many-

body interactions and the effective coupling to the thermal bath. Proposed schemes [82, 83, 84] to generate many-body thermal states involve subroutines like quantum phase estimation, which are difficult to implement on near-term devices, or require engineered dissipative couplings [85]. Experimental platforms such as optical lattices of ultracold atoms have enabled finite temperature simulation [86, 87], but these are specific to particular (Hubbard) models, and cooling to low effective temperatures remains a major obstacle.

Here we show how to use protocols based on the quantum approximate optimization algorithm (QAOA) [56] to efficiently prepare such thermal states. The method can be easily applied to the case of pure state, which can be equivalently seen as thermal states with zero temperature.

The adiabatic theorem [88] explains the intuition behind QAOA pretty well. Say we prepared the ground state $|\phi\rangle_0$ of a Hamiltonian \hat{H}_0 . We can adiabatically evolve $|\phi\rangle_0$ into the ground state $|\phi\rangle$ of another more complicated Hamiltonian $\hat{H} = \hat{H}_0 + \hat{H}_I$. All we need to do is evolving $|\phi\rangle_0$ with the Hamiltonian $\hat{H}' = \hat{H} = \hat{H}_0 + g\hat{H}_I$ and slowly crank up g from 0 to 1. The adiabatic theorem says if we do this slow enough, $|\phi\rangle_0$ will be effectively evolved into $|\phi\rangle$. It should be intuitive to see that the digitized version of operations described above can be described as:

$$|\phi\rangle = \prod_{j=1 \sim n} (\exp[-i\frac{\hat{H}_0\delta}{\hbar}]\exp[-i\frac{g_j\hat{H}_I\delta}{\hbar}]) |\phi\rangle_0 \quad (6.6)$$

Here, δ is a sufficiently small time incremental unit. g_j very slowly change

from 0 to 1, as j change from 1 to n . This digitized application of adiabatic theorem takes huge amount of gates operation to implement. But the QAOA tell us, we can modify eq.6.6 into the following form:

$$|\phi\rangle = \prod_{j=1 \sim m} (\exp[-i\frac{\hat{H}_0\alpha_j}{\hbar}]\exp[-i\frac{\hat{H}_I\beta_j}{\hbar}]) |\phi\rangle_0 \quad (6.7)$$

Here α 's and β 's are all independent parameters. If we can find the right set of parameters, we can greatly decrease the value of m . This is the power of QAOA. This shouldn't be surprise to us, at least, if not intuitive. From the perspective of control theory, since the system is controllable (as eq.6.6 shows), optimize the naive control scheme(into eq.6.7) could greatly increase its efficiency.

6.2.1 Thermofield Double (TFD) State

Thermofield double (TFD) states [89] are pure quantum states entangled between two systems, with the property that when either system is considered independently by tracing over the other, the TFD reduces to a thermal mixed state at a specified temperature.

Consider two identical Hilbert spaces A and B consisting of qubits labeled by an index j . The Pauli spin operators on qubit j are labeled X_j , Y_j , and Z_j [90]. Let H_A be a Hamiltonian with eigenstates $|n\rangle_A$ and corresponding energies E_n . A thermofield double state corresponding to inverse temperature β is defined on the

joint system A and B as

$$|TFD(\beta)\rangle = \frac{1}{\sqrt{Z(\beta)}} \sum_n e^{-\beta E_n/2} |n\rangle_A |n'\rangle_B, \quad (6.8)$$

where $Z(\beta)$ is a normalization factor. In general, the set $\{|n'\rangle_B\}$ can be any orthonormal basis spanning B , and we will make the choice $|n'\rangle = U|n\rangle$ where $U = \otimes_j Y_j$. This choice is consistent with the infinite temperature TFD defined below. Tracing out the auxiliary system B results in the thermal (Gibbs) state of system A $\rho_A = e^{-\beta H_A}/Z$; in this sense, realizing the TFD allows one to simulate the thermal Gibbs state in a subsystem A with the effective bath B .

TFD states are purifications of thermal Gibbs states and have played a key role in the holographic correspondence relating a quantum field theory to a gravitational theory in one higher dimension. In this correspondence, TFD states are dual to wormholes on the gravity side [91, 92] and enable teleportation (“traversable wormholes”) [93, 94]. The simulation of these concepts has motivated several approaches for preparing TFD states.

6.2.2 preparing TFD states with QAOA

The protocol [95] starts with an initial state $|\psi_0\rangle$ that is a product of Bell-pair singlets $\frac{1}{\sqrt{2}}(|0\rangle_A |1\rangle_B - |1\rangle_A |0\rangle_B)$ between pairs of A and B qubits. This is an infinite temperature TFD since ρ_A is maximally mixed. Note that the two components of a Bell-pair singlet are related up to a phase by $Y|0\rangle = |1\rangle$ and $Y|1\rangle = -|0\rangle$, which justifies our choice of basis above. One then alternates between time evolution

with the inter-system coupling $H_{AB} = \sum_i X_{i,A}X_{i,B} + Z_{i,A}Z_{i,B}$ and the intra-system Hamiltonians $H_A + H_B$, where H_B is the rotated version of H_A (UH_AU^\dagger) acting on the B qubits. H_{AB} is chosen based on the fact that its ground state is $|\psi_0\rangle$, allowing for an adiabatic limit of our protocol described below. As in QAOA, each timestep is a variational parameter, and after p layers of alternation, the resulting variational wavefunction is

$$|\psi(\vec{\alpha}, \vec{\gamma})\rangle_p = \prod_{j=1}^p e^{i\alpha_j H_{AB}} e^{i\gamma_j (H_A + H_B)/2} |\psi_0\rangle. \quad (6.9)$$

The variational parameters $\vec{\alpha}, \vec{\gamma}$ are chosen to maximize the fidelity with the target TFD state: $F_p(\vec{\alpha}, \vec{\gamma}) \equiv |\langle \text{TFD}(\beta) | \psi(\vec{\alpha}, \vec{\gamma}) \rangle_p|^2$. As detailed in [95], this protocol is guaranteed to target the zero temperature TFD in the limit of large p because in that limit it subsumes the adiabatic algorithm; the intuition, verified through several examples [95], is that the finite temperature TFD is easier to prepare than zero temperature ground state because the thermal correlation length is generally finite.

In the holographic correspondence, TFDs of conformal field theories describing gapless quantum matter are particularly interesting because they correspond to wormholes on the gravity side. Their preparation is also useful to condensed matter physics because they enable investigation of finite-temperature properties of systems near a critical point by tracing over one of the systems in the double. Hence, our first objective is to prepare thermofield double states of the transverse field Ising model (TFIM) at its quantum critical point. Defined on a one-dimensional ring of

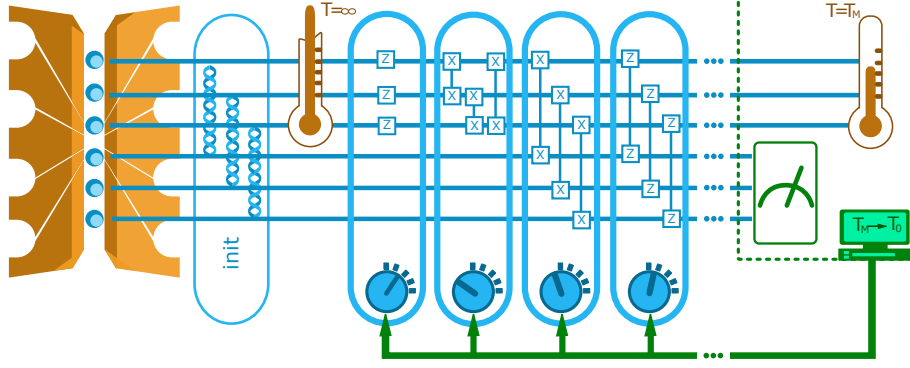


Figure 6.5: Hybrid quantum-classical optimization with trapped ion qubits to prepare thermal states. The initial Bell-pairs denoted by ribbons connecting qubits 1-4, 2-5, and 3-6 (labeled 1-6 from top to bottom), correspond to the thermofield double state at infinite temperature. Layers of unitaries with independent control parameters are then applied sequentially to cool to the target temperature. The subsystem consisting of the first three qubits is effectively in the thermal (Gibbs) state. The result can be fed into a classical computer which updates the parameters based on a cost-function in a closed loop (see “Full Hybrid Optimization: Preparation of Ground State of TFIM” for details).

L qubits, the TFIM Hamiltonian is

$$H_{TFIM} = \sum_{i=1}^L X_i X_{i+1} + g \sum_{i=1}^L Z_i \equiv H_{XX} + gH_Z. \quad (6.10)$$

Here g is the strength of the transverse field. When $g = 1$, the ground state is a critical point between anti-ferromagnetic and paramagnetic quantum phases and has several interesting properties, including correlations between two spins decaying as a power of their separation and entanglement entropy scaling logarithmically with the size of the subsystem.

To prepare the TFD of the quantum critical TFIM, we tailor the general protocol above (Eq. 6.9) to the capabilities of an experimental system with six trapped ions. The initial state is the product state of three spin-singlet Bell pairs

formed between pairs of A and B spins. Ideally following the general protocol, we would like to evolve sequentially with $H_A = H_{XX} + H_Z$ (in addition to H_B), followed by

$$H_{AB} = \sum_i Z_{i,A} Z_{i,B} + \sum_i X_{i,A} X_{i,B} \equiv H_{ABZ} + H_{ABX}. \quad (6.11)$$

Since H_{ABZ} and H_{ABX} commute, this step can be simply decomposed into evolution with each piece separately. However, time evolution with H_A in general requires a Trotter decomposition which could require many steps beyond the capabilities of current experimental systems. Moreover, here H_B introduces additional gates which we find are not essential for achieving high fidelity. In particular, in the $p = 1$ ansatz, H_A and H_B act directly on the maximally entangled state $|\psi_0\rangle$, which has the property that $H_A|\psi_0\rangle = H_B|\psi_0\rangle$; thus, H_B is redundant in this case. Hence, we instead use a minimal variational ansatz for the TFD consisting of four pieces:

$$|\psi(\alpha_1, \alpha_2, \gamma_1, \gamma_2)\rangle = \exp(iH_{ABZ}\alpha_2) \exp(iH_{ABX}\alpha_1) \quad (6.12)$$

$$\times \exp(iH_{XX}\gamma_2) \exp(iH_Z\gamma_1) |\psi_0\rangle \quad (6.13)$$

The first two operations represent a minimal Trotterization of time evolution with H_A . The optimal parameters are determined (on a classical computer) by maximizing the fidelity with the target TFD. In this case, the optimal fidelities are extremely good, ranging from 0.93 for the zero temperature TFD to 1 for the infinite temperature TFD. These can be further improved by adding additional iterations

of this sequence of unitaries in the protocol. The single-body observables and two point correlation functions of the optimized ansatz compare well with those of the target TFD, as evident in Fig.6.6. We note that the general protocol preparing the TFD of the classical ($g = 0$) Ising model achieves perfect fidelity for $p = L/2$ layers [95].

6.2.3 TFD preparation: experimental results

To confirm the preparation of the TFD state, we measure both intra-system observables (single and two body correlation functions within system A) and inter-system correlators between corresponding sites from the A and B systems. The purpose of the intra-system measurements is to verify physical properties of the thermal Gibbs state. In the phase diagram parameterized by temperature T and transverse field g , there is a regime $|g-1| \ll T \ll 1$ called the quantum critical fan [96], whose properties are dictated by the continuum theory of the critical point. For instance, this regime exhibits exponentially decaying correlations with correlation length proportional to inverse temperature in this case. Our intra-system measurements could verify this phenomena and other features of the quantum critical fan for larger system sizes. The purpose of the inter-system measurements is to observe how correlations and entanglement between the two systems decreases as one lowers the target temperature and thereby the thermal entropy (which in the TFD is the entanglement entropy between the systems).

As shown in Fig.6.6, the results agree well with those expected from the TFD

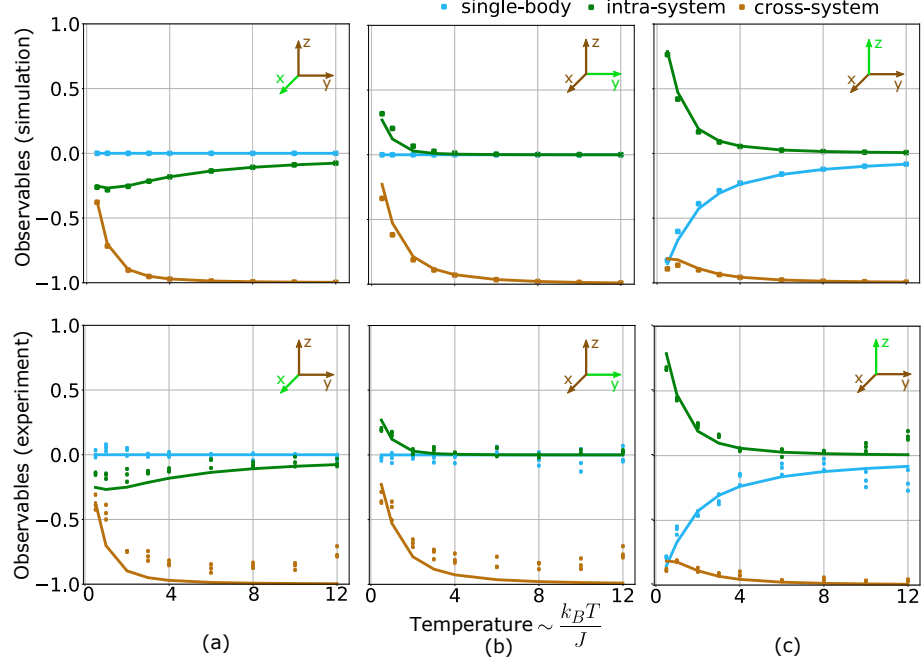


Figure 6.6: Preparation of TFD states of the quantum critical TFIM using two 3-qubit systems. Top row: Comparison between observables of the simulated optimized ansatz circuit and target TFD states (solid lines) for various target temperatures. Bottom row: Comparison between observables of experimentally prepared and target TFD states. Results for all three ion pairs are given at each temperature. The measured correlation functions for different target temperatures are plotted against the theoretical expectations (solid lines) for type (a) Pauli-X (b) Pauli-Y and (c) Pauli-Z. Intra-system correlators in the subsystem-A are: $\langle \sigma_{1,A} \sigma_{2,A} \rangle$, $\langle \sigma_{1,A} \sigma_{3,A} \rangle$, and $\langle \sigma_{2,A} \sigma_{3,A} \rangle$. Cross-system correlators are $\langle \sigma_{1,A} \sigma_{1,B} \rangle$, $\langle \sigma_{2,A} \sigma_{2,B} \rangle$, and $\langle \sigma_{3,A} \sigma_{3,B} \rangle$. Note the experimental data points in the figure have errorbars accounting for statistical errors. Statistical error bars are similar in size or smaller than the symbols used. A symmetry based error mitigation technique is used to post-process the experimental result in (c). The mitigation notably improved the agreement between experiment and theory. Details are given in the SI Appendix and Fig. S2.

states, with some reduction in correlations caused by imperfect entangling operations. We note that at high temperature, there is a slight increase in error arising from an artifact of there being many sets of parameters that yield very good fidelities, and the optimal angles found are large enough to cause the observed errors (see SI Appendix, Fig. S2). In fact, for such high temperatures, the initial set of Bell pairs is already a very good approximation to the target TFD, and it would be better to avoid using any gates.

6.2.4 prepare the Quantum Critical State at $T=0$

To prepare the zero-temperature critical TFIM (pure) state, one does not require a purifying auxiliary system and thus a larger system A can be accessed experimentally. However, the long-range correlations and relatively high entanglement of the critical state pose challenges for preparation. Because a finite depth circuit consisting of local gates can only produce a state with finite correlation length, to generate critical states one needs a quantum circuit (of local gates) with depth scaling with system size. With non-local gates, long range correlated states can be prepared with fewer steps [97]; however, tailoring the effective power-law decaying interactions in trapped ion systems to target an arbitrary critical state is in general a difficult problem. One method for generating such critical states is the adiabatic algorithm, which requires tuning g adiabatically. On a digital quantum platform, this would require a compilation such as Trotterization into discrete gates, and the resulting deep circuit would be very susceptible to errors.

An alternative is the QAOA-motivated variational approach detailed in [98]. One begins with the product ground state of H_Z , which we denote $|0\rangle$, and then evolves with H_{XX}, H_Z in an alternating fashion:

$$|\psi(\vec{\alpha}, \vec{\gamma})\rangle_p = e^{-i\alpha_p H_Z} e^{-i\gamma_p H_{XX}} \dots e^{-i\alpha_1 H_Z} e^{-i\gamma_1 H_{XX}} |0\rangle \quad (6.14)$$

Again, p denotes how many pairs of iterations are used, and in the hybrid quantum-classical optimization, $(\vec{\gamma}, \vec{\beta})$ are variational parameters to be optimized to achieve the ground state of $-H_{XX} - H_Z$; in this section, we target the critical ground state of the ferromagnetic transverse field Ising model.

Trotterizing the adiabatic approach for preparing the critical state would lead to a unitary sequence of the above form, with $(\vec{\gamma}, \vec{\beta})$ infinitesimal; this implies that for sufficiently large numbers of layers p , there is guaranteed to exist a set of parameters $(\vec{\gamma}, \vec{\beta})$ for which the ansatz converges to the target state. However, the key question is how well the above ansatz performs for finite p . Remarkably, it has been observed that for a system size L , the above protocol can prepare the target critical state (and any state in the TFIM phase diagram) with perfect fidelity given $p = L/2$ layers [98].

For a trapped ion system of seven qubits, a $p = 3$ protocol can generate the desired ground state with perfect fidelity, and we find the optimal angles $(\vec{\alpha}, \vec{\gamma})$ on a classical computer to maximize the many-body overlap $|\langle\psi_t|\psi_p\rangle|^2$ of the ansatz $|\psi_p\rangle$ and the target state $|\psi_t\rangle$. While $p = 3$ layers exactly prepares the critical state, $p = 1, 2$ yield theoretical fidelities of 0.76 and 0.88, respectively.

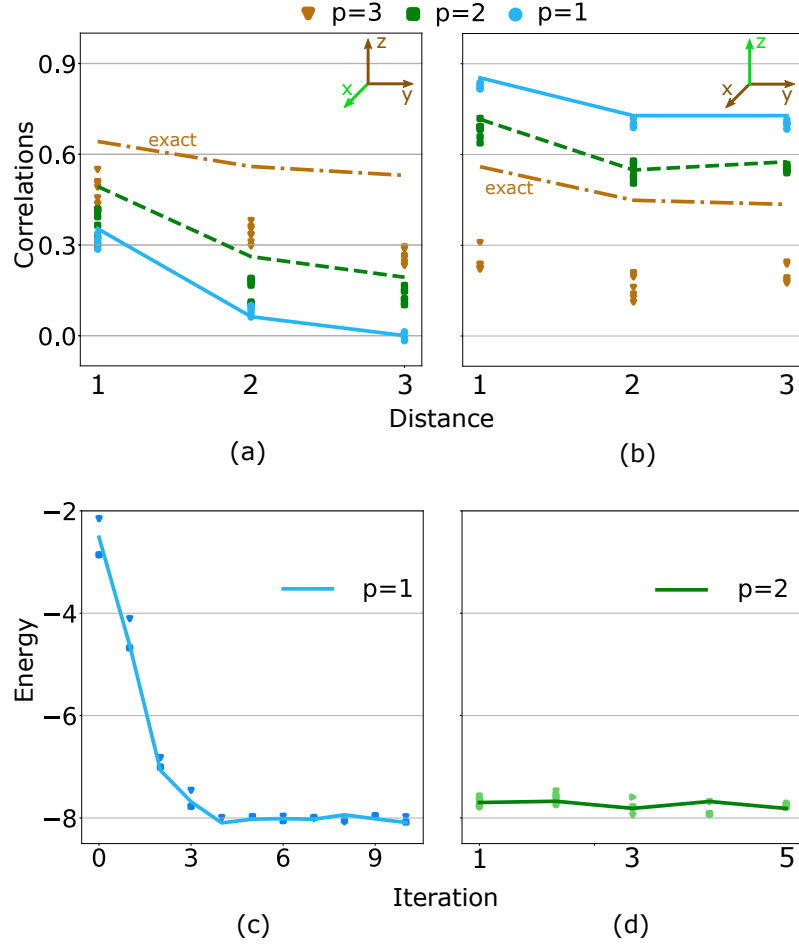


Figure 6.7: Critical TFIM ground state on a 7-qubit system. Top row: Two-point correlations for (a) Pauli-X and (b) Pauli-Z operators as a function of their separation. For a ring of seven spins, there are only three different pairs of ions, which are distinguishable by distance. The three different colors correspond to QAOA protocols with different depth p . The lines denote the theoretical expectations. Bottom row: Energies achieved using full hybrid quantum-classical feedback with increasing gradient descent iteration number for (c) $p = 1$, initialized with random parameter set, and (d) $p = 2$, initialized with theoretically optimal parameters. The line corresponds to the measured energy at each iteration, and the dots correspond to samples taken to evaluate the gradients. Ideally, the lowest energy a $p=1$ protocol can reach is -8.44 . The lowest energy a $p=2$ protocol can reach is -8.62 . The true ground state energy is -8.98 , and the size of the gap is 0.23 . The gap decreases linearly with system size. Statistical error bars in the above figures are of the same size or smaller than the symbols used.

For each number of layers p , we run the protocol with optimal angles on the trapped ion system and again measure two body correlation functions for Pauli Z and X operators (Fig. 6.7(a)(b)). The theoretical and experimental values agree well for the $p = 1$ protocol, but deviate for $p = 2, 3$, as errors accumulate in the deeper circuit. The data shows that larger p protocols are more effective at generating long range correlation along the X direction, but have more error in the Z observable, resulting in less accurate energy. In particular, in the experiment the $p=3,2,1$ protocols attain energies $-5.46 \pm 0.097, -7.74 \pm 0.095$, and -8.02 ± 0.043 (respectively). In the simulation, the corresponding numbers are -8.98 for $p=3$, -8.62 for $p=2$ and -8.44 for $p=1$. Fig. 6.8 (a) provides a visual comparison. We find that the QAOA protocol with the least number of steps produces the state with the lowest energy, though theoretically it should be the worst. This reflects the level of noise in the experimental system which we will discuss later.

6.2.5 Demonstration of hybrid optimization

Determining the optimal angles using classical simulation is feasible for current system sizes. For larger systems and higher p , however, one would need extrapolation based on patterns in the control parameters of QAOA protocols [99, 100]. Therefore, a hybrid approach which involves a feedback loop between a quantum simulator and a classical computer has to be employed. As depicted in Fig. 1, one first carries out the unitary circuit for a given set of parameters and measures the energy cost function $E_p(\vec{\alpha}, \vec{\gamma}) = \langle \psi(\vec{\alpha}, \vec{\gamma}) | -H_{XX} - H_Z | \psi(\vec{\alpha}, \vec{\gamma}) \rangle_p$. The lower the energy, the

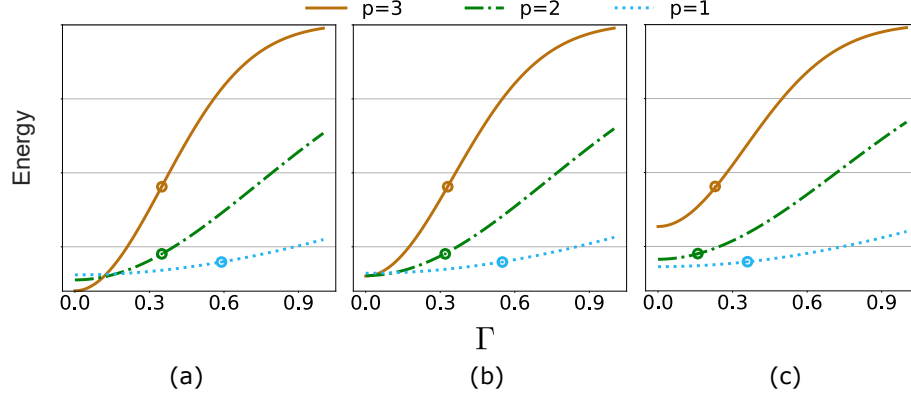


Figure 6.8: Simulation results with noise for the $p = 1, 2, 3$ QAOA protocols for preparation of the critical ground state of the TFIM. (a) $\lambda = 0$ (no depolarizing noise); (b) $\lambda = 0.1$; (c) $\lambda = 0.22$. Each curve is averaged over 1000 samples. The circles in the figure show what Γ value an experimental result (shown in Fig. 3) predicts for a given set of p and λ . $\lambda = 0.22$ is the point at which we can minimize the variance of the predicted Γ . Note that (a) shows a threshold at $\Gamma = 0.13$ below which higher p give better results.

better this ansatz can approximate the critical ground state of $-H_{XX} - H_Z$. One then uses classical optimization to vary the parameters to lower the cost function until convergence is reached. One benefit of this hybrid scheme is that systematic errors from the quantum device are reduced.

We implement the full QAOA hybrid algorithm using standard gradient descent as the classical optimization strategy. To obtain an estimate of the partial derivatives, we change each parameter separately by a small amount and measure the corresponding energy difference. We then take a small (proportional to the gradient, with coefficient adjusted according to simulation) step along the gradient with all parameters. We target the critical TFIM ground state for $p = 1$ starting from a random set of initial parameters. Results are shown in figure 6.7(c). The optimization converges to a set of parameters that is different from the simulated result, but the measured energy matches the theoretical prediction for $p = 1$.

To examine whether significant systematic errors play a role for deeper circuits in our experiment, we implement the hybrid optimization for $p = 2$. This time, we initialize the process with the optimum values obtained from numerical simulation. A drop in the cost function would indicate that systematic errors shift the system away from the optimal state. The results in figure 6.7(d) show that this is not the case in our system.

6.2.6 Cost function for hybrid optimization of TFD states

We demonstrated the hybrid optimization for the preparation of zero temperature state, for which energy can directly be used as the cost function. This is not the case for the finite temperature TFD states. The experiment results we obtained for the finite temperature TFD states are based on simulated parameters. The cost function used to simulate for the variational parameters $\vec{\alpha}, \vec{\gamma}$ is the fidelity with the target TFD state: $F_p(\vec{\alpha}, \vec{\gamma}) \equiv |\langle \text{TFD}(\beta) | \psi(\vec{\alpha}, \vec{\gamma}) \rangle_p|^2$. However, this quantity cannot be calculated efficiently for a state prepared on a quantum computer. Alternatively, we know that the thermal state at temperature T minimizes the cost function $F(\rho) = E(\rho) - TS(\rho)$, where ρ is the partial density matrix of the state, E is the expectation value of the Hamiltonian, and S is the von Neumann entropy. Therefore, a possible cost function for the TFD state could be $F_A + F_B$, where F_A is the free-energy of subsystem A and F_B of subsystem B. Here too, while $E_A = \text{Tr}(\rho_A H)$ can be easily calculated, calculating $S_A = -\text{Tr}(\rho_A \log(\rho_A))$ is not straightforward. Therefore, we are still missing a cost function that could be ef-

ficiently calculated, which would make it possible to run a true variational TFD preparation protocol on a hybrid quantum-classical computing system.

The QAOA-inspired technique demonstrated in this paper may give a hint to a simple expression for the cost function. We know that for preparing the ground state of a given Hamiltonian, the bang-bang protocol of QAOA breaks the Hamiltonian into two parts, and applies them alternately for varying times. Since the bang-bang protocol in this paper consists of alternately applying $H_A + H_B$ and H_{AB} , we can hypothesize that the cost function for the TFD state may be of the form $H_{\text{TFD}} = (H_A + H_B) + f(T, L)H_{AB}$, where $f(T)$ is an increasing function of the temperature that is zero at $T = 0$. Indeed, on inspection, we see that such a cost function is in fact of the form $F_A + F_B$. The expectation value of the first term of H_{TFD} gives $E(\rho_A) + E(\rho_B)$, while the second term gives the correlations between subsystems A and B which is a measure of their entropy. In future work, we hope to design a simple and approximately correct cost function that will allow variational preparation of TFD states in a quantum-classical hybrid fashion.

6.2.7 Symmetry Based Error Mitigation

The transverse field Ising chain defined as $H = \sum_i (X_i X_{i+1} + g Z_i)$ has a Z_2 symmetry, i.e. the Hamiltonian commutes with the operator $\prod_i Z_i$. The TFD state $|\Psi\rangle = \frac{1}{Z(\beta)} \sum_n \exp(-\beta E_n/2) |n\rangle_A |n'\rangle_B$ is a superposition of states in which subsystem B has a time-reversed copy of the eigenstate of H in A. Therefore, $Z_{1A} Z_{2A} Z_{3A} = -Z_{1B} Z_{2B} Z_{3B}$, and any measurement that does not satisfy this should

be discarded.

This symmetry based error mitigation is applied to measurements in the Z direction. Fig. 6.9 (a) visualizes the difference between the corrected and uncorrected data. Notable improvement can be seen in the cross-system correlators. Fig.6.9 (b) visualizes the selection rate (the proportion of data kept) at each temperature. It can be observed from Fig.6.9 (b) that the selection rate drops as the temperature rise. This agrees with the trend observed in Fig.2 (manuscript) that the error is larger at high temperature.

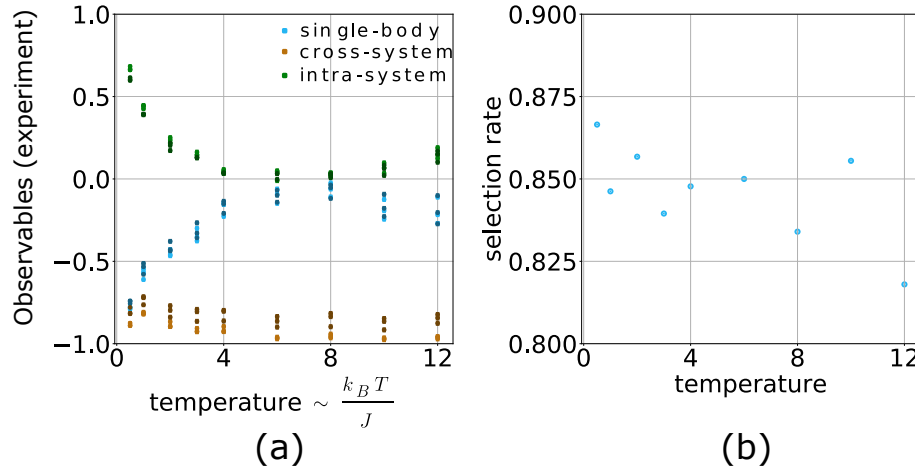


Figure 6.9: (a): Comparison between results with and without symmetry based error mitigation. The dots with same hue correspond to the same type of measurement, while dots with brighter color correspond to data corrected with error mitigation.(b): The fraction of data points kept after the symmetry based post-selection.

6.2.8 Error Simulation

We simulate the QAOA protocol in the presence of noise for different numbers of layers p , analyzing the trade-off between theoretical and experimental errors. The two-qubit XX gates are the main source of error in the experiment, likely limited by laser beam intensity fluctuation δI on the trapped ion qubits. Because the angle of

the XX gate evolution depends on the square of the laser intensity I , the fractional error in the XX gate angle is $\Gamma = 2\delta I/I$.

We model this error with a Monte Carlo simulation by setting the angle of the two qubit gate to be $\theta = \theta_0(1 + \Gamma r)$, where θ_0 is the nominal gate angle, r is a Gaussian-distributed random number with mean 0 and standard deviation 1, and we average over 1000 samples. Fig. 6.8(a) shows the results for the variation of the measured energy versus Γ . The 3 points marked in the figure indicate the experimentally measured values for the $p = 1, 2, 3$ protocols. The value of the noise parameter Γ inferred from this error model is consistent between $p = 2$ and $p = 3$.

As seen in Fig. 6.8(a), for $\Gamma \leq 0.13$, the higher-depth circuit produces a better outcome, and for higher levels of Γ , the lower depth circuit is preferable. This implies a type of threshold noise behavior, where the optimization protocol converges to near-optimal solutions as long as the noise is below a critical value. The threshold can be explained by observing that the accuracy of QAOA for preparing ground states of Hamiltonians with unweighted terms is likely to increase exponentially with p [99], while the experimental accuracy on average decreases exponentially with p .

Generically, we also expect the two-qubit gates to include some depolarizing error on the qubits involved in the gate. We simulate this error channel by averaging over rotations around a random axis after every XX gate. To see this, first consider

the effect of a depolarizing channel on the density matrix for a single qubit:

$$\rho \xrightarrow{\text{depol}} \left(1 - \frac{3p}{4}\right)\rho + \frac{p}{4}(\sigma_X\rho\sigma_X + \sigma_Y\rho\sigma_Y + \sigma_Z\rho\sigma_Z), \quad (6.15)$$

where $\sigma_{X/Y/Z}$ are the Pauli matrices.

Instead, rotating by an angle ϕ around an axis \hat{n} would give:

$$\begin{aligned} \rho &\rightarrow \exp\left(i\frac{\phi}{2}\hat{n}\cdot\vec{\sigma}\right)\rho\exp\left(-i\frac{\phi}{2}\hat{n}\cdot\vec{\sigma}\right) \\ &= \left(\cos\left(\frac{\phi}{2}\right)I + i\sin\left(\frac{\phi}{2}\right)\hat{n}\cdot\vec{\sigma}\right)\rho\left(\cos\left(\frac{\phi}{2}\right)I - i\sin\left(\frac{\phi}{2}\right)\hat{n}\cdot\vec{\sigma}\right) \end{aligned} \quad (6.16)$$

Here $\hat{n}\cdot\vec{\sigma} = n_X\sigma_X + n_Y\sigma_Y + n_Z\sigma_Z$. Let ϕ be a random variable with distribution $P(\phi)$ that is Gaussian with mean 0 and standard-deviation λ . Averaging over samples with different values of ϕ and \hat{n} is equivalent to integrating $\int d\hat{n} \int_{-\infty}^{\infty} P(\phi)\rho d\phi$, where $P(\phi)$ is the distribution over ϕ . On integration, all the terms containing one $\sin(\phi)$ term will disappear since they are odd functions of ϕ . On integrating over $n_{X/Y/Z}$, the only terms that remain are the ones that contained $n_{x/y/z}^2$ and so are of the form $\sigma_X\rho\sigma_X$. So finally this procedure returns the single qubit depolarizing channel in Eq. 6.15 where p is a function of λ . This treatment can be straightforwardly extended to a depolarizing channel on two qubits by averaging over rotations around random axes on both qubits after every XX gate.

In our experiment, we calculate this error from a measurement of the state populations [101]. To relate λ to the depolarizing error rate measured in experiment,

we can write Eq. 6.16 explicitly as:

$$\rho \rightarrow \int_{-\infty}^{\infty} \frac{1}{\lambda\sqrt{\pi}} \exp\left(-\frac{\theta^2}{\lambda^2}\right) \cos^2(\theta) d\theta \rho + \int_{-\infty}^{\infty} \frac{1}{\lambda\sqrt{\pi}} \exp\left(-\frac{\theta^2}{\lambda^2}\right) \sin^2(\theta) d\theta \hat{\sigma} \rho \hat{\sigma} \quad (6.17)$$

Here the second integral gives the depolarizing error rate. Different error sources give different weights to $\hat{\sigma}_x \rho \hat{\sigma}_x$, $\hat{\sigma}_y \rho \hat{\sigma}_y$, and $\hat{\sigma}_z \rho \hat{\sigma}_z$. For the Molmer-Sorensen interaction [20], we expect the $\hat{\sigma}_x \hat{I} \hat{\sigma}_x$ term to have the biggest weight. The variance in Γ is calculated for several values of λ between 0.1 and 0.3. Fig. 6.8 (b) and (c) show results for different values of λ , with $\lambda = 0.22$ being the point at which we can minimize the variance of the predicted Γ . This value corresponds to a two-qubit depolarizing error of 2.37%. This agrees well with the typical experimentally measured error rate of 1.5% \sim 2.5%, as described in the SI Appendix. Note the threshold for Γ described above appears for $\lambda < 0.1$, corresponding to a depolarizing error rate $< 0.5\%$.

Chapter 7: Validating quantum computation

When the size of our system reaches around 50 qubits, with the capability to handle circuits of more than 1000 two-qubit gates, we will no longer be able to predict its behavior with classical simulation. At this size, quantum computers will start to generate practical value by "giving us something classical computers cannot give". However, if quantum computers are the only type of devices that can address a challenge, how can we be assured that such challenges are appropriately addressed.

There are approaches to address this issue, along with many directions. For example, one can take an induction-like approach and claim that if every simulatable sub-sections of a quantum computing process is verified by classical simulation, then the entire process is equivalently verified.

One can also repeat the same computation independently on multiple different quantum computers and compare them against each other. If the independent quantum computers all give the same answer, one can be confident that the computation is correct. This approach is philosophically related to beating atomic clocks against each other to establish their precision.

Alternatively, one can challenge quantum computers with some particular tasks, which are hard to address but easy to verify. If the quantum computers'

solutions to these particular tasks are verified, one can be confident about the quantum computers’ capability to address other non-verifiable tasks. This approach is related to the concept of quantum advantages.

In this final chapter, we will focus on the latter two approaches.

7.1 Cross-platform comparison

Still writing the first draft

7.2 Interactive protocols for classically-verifiable quantum advantage

A truly loophole-free demonstration of quantum advantage will require protocols that are *efficiently classically verifiable*. To accomplish this and related tasks, a number of recent works have proposed protocols that leverage *cryptography* for verification. Via constructions similar to those used for classical authentication and secure communication, protocols have been developed that achieve certifiable randomness generation [102], remote-state preparation [103], self-testing of single quantum devices [104], and even classical verification of arbitrary quantum computations [105]. Specifically, many of these schemes take advantage of the idea of the *interactive protocol*, in which multiple rounds of data are exchanged between the quantum “prover” and a classical “verifier”, as shown in Fig.7.1 (a)

The power of interactive protocols stems from the notion of *commitment*. When the prover sends data to the verifier, they are committing to answer future queries based on that data, even though they don’t know what the next query, or

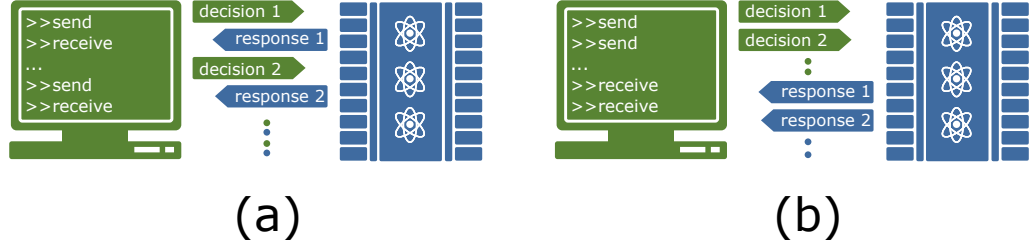


Figure 7.1: (a) protocol with implemented with real-time-interaction (b) protocol implemented without real-time-interaction

“challenge,” will be. While it may be possible for the prover to choose a commitment that makes it easy to answer a specific future challenge, the randomness in the verifier’s choices implies that the prover must be prepared to answer *any* challenge. This feature is particularly useful in the quantum setting: if the verifier’s follow-up challenge corresponds to a request for quantum measurement in a particular basis, a quantum prover can demonstrate that they *would* yield correct results in any of several possible bases, even though in practice only one measurement is possible due to wavefunction collapse. This idea has formed the foundation of a broad range of quantum cryptographic protocols, including those experimentally demonstrated in this work. Generally, the first step of such a protocol has the prover generate a quantum state that would be hard to compute classically, and the remainder of the protocol allows the verifier to specify how to measure that state.

In our case, for the first step, both protocols make use of a cryptographic primitive called the *Trapdoor Claw-Free Function* (TCF). This type of function is 2-to-1, such that each output corresponds to a pair of inputs. It also has the special property of being “claw-free:” it is cryptographically hard to find any such pair of inputs that both evaluate to the same output. Despite the claw-free property, a

quantum computer can efficiently generate a *superposition* of two such inputs by evaluating the function on a superposition of the entire domain, and then collapsing the wavefunction onto a single output through projective measurement. In this way a quantum prover can generate the state $\sum(|x_1\rangle + |x_2\rangle)|w\rangle$, with $w = f(x_1) = f(x_2)$ known classically after the measurement. This is a fundamentally quantum process, and this technique comprises the first round of the protocols we demonstrate in this paper.

Crucially, the state $\sum(|x_1\rangle + |x_2\rangle)|w\rangle$ is hard to compute classically because x_1 and x_2 are hard to find, but it is simple to *describe* classically once those values are known. This fact allows the verifier to determine the prover’s state exactly, by using another feature of the TCF called the “trapdoor.” When the verifier chooses a specific function f for the protocol, they also generate some secret data t , which makes it possible to efficiently invert f for *any* value w .

With perfect knowledge of the prover’s supposed state, the verifier’s remaining challenge is to confirm that the prover actually holds the correct superposition. As alluded to earlier, this is done by randomly choosing between a number of possible measurements, such that all of the measurement results *combined* would yield data that is cryptographically protected. We emphasize that the prover can only make one meaningful measurement due to wavefunction collapse. But because the prover does not know the measurement beforehand, the verifier can effectively ensure the correctness of *all* of the possible measurements.

In this study, we investigate two related protocols, one using a TCF based on the Learning with Errors (LWE) problem and another based off of “Rabin’s

function” $x^2 \bmod N$. A specific cryptographic property of the Learning with Errors TCF allows a simpler measurement scheme than in the $x^2 \bmod N$ case; we describe the TCFs and their corresponding measurement schemes in the following sections.

At the time of writing, we only implemented the non-interactive version of the protocols, as shown in Fig.7.1(b). All the decisions are pre-made and send to the prover. To compensate for the lack of interaction, we demonstrated all the possible responses the prover could give to address the verifier’s decision. Just like the interaction, this stops the prover from cheating. But using interaction to protect against cheating is scalable, while demonstrating all the possible responses is not.

We are actively working on implementing the protocols with real-time-feedback, as shown in Fig.7.1(a)

7.2.1 Learning with Errors (LWE) based protocol

The first variation is based on the Learning with Errors (LWE) problem. The motivation behind this protocol is mainly from Mahadev’s protocol for classical verification of quantum computation which utilized TCFs based on LWE in order to achieve this [105]. Other works have explored the potential of TCFs based on LWE, in particular, Vidick and Gheorghiu’s work on an improvement of Mahadev’s original protocol [103]. This variation of the protocol ensures additional cryptographic properties and is also more modular, allowing it to be incorporated into other protocols easily. For instance, a notable application is the use of these same TCFs to achieve certifiable randomness, another task that a classical computer cannot

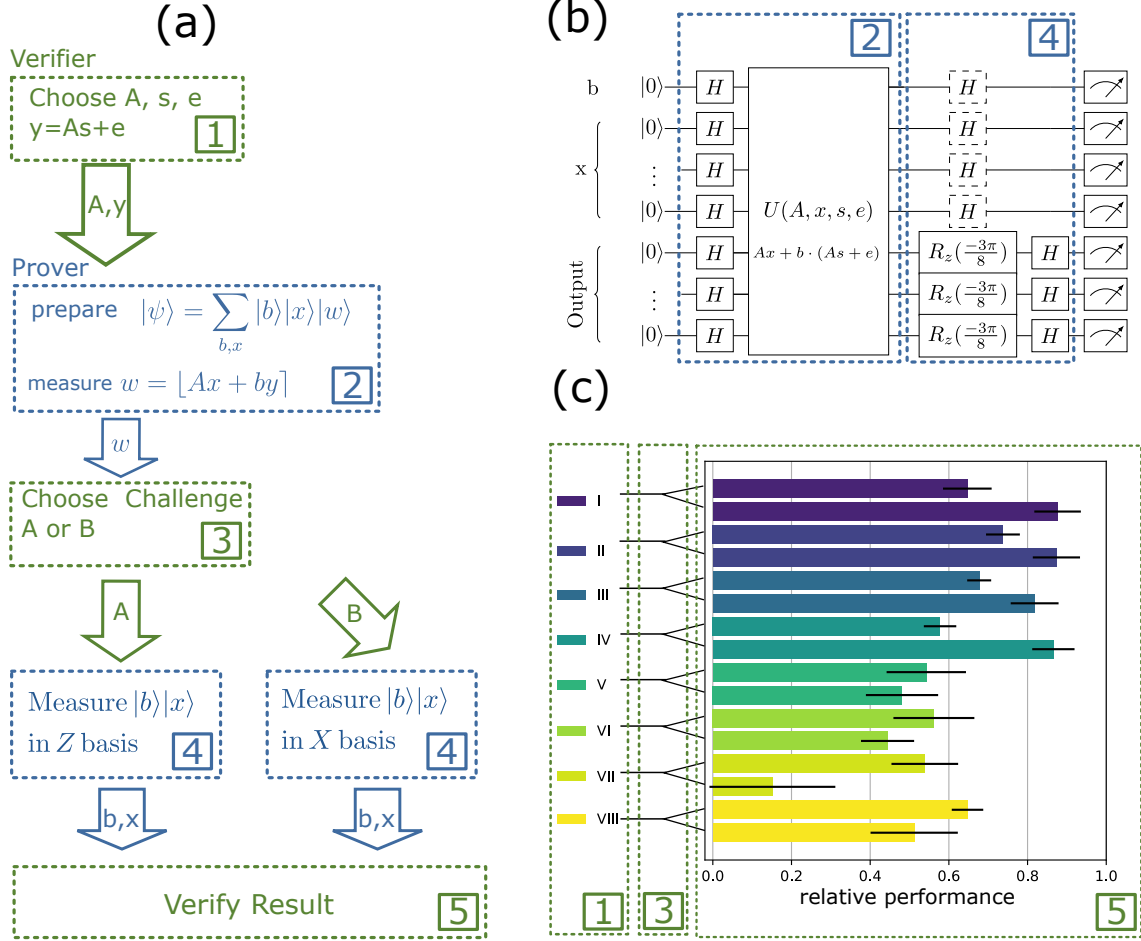


Figure 7.2: (a):The flow chart of the interactive protocol based on LWE. The green boxes stands for verifier's actions. The blue boxes stands for prover's actions. The arrows stands for the exchange of information. (b):The circuit diagram of quantum operations the prover would need to perform to answer the verifier's challenge. (c):The experimental results of all the branches of circuits, in relative performance (as defined in eq.7.14). The configurations are arranged in ascending order of two-qubit gates involved in the implementation. The boxes marked with the same index correspond to the same operations. Note if real-time interaction is implemented, the measurement of the output register will be moved to between box 2 and box 4.

achieve with high probability [102].

Thus, in this case, we follow the protocol presented by Gheorghiu and Vidick [103], with a TCF of the form $f(b, x) = \lfloor Ax + b \cdot (As + e) \rfloor$. Here A is a matrix. x, s , and e are vectors of appropriate dimensions. The $\lfloor \cdot \rfloor$ indicates a rounding function

which takes the most significant bit of the argument given. As the experimental protocol in Fig.7.2 (b) shows, we first use Hadamard gates to prepare all the qubits into $|+\rangle$ state. On one hand, this generates a superposition over all of the input bit string (b,x) in computational (Z) basis. On the other hand, these Hadamard gates also initialize the phase of output qubits, which we use to evaluate the function $f(b,x)$. In our approach, we reduce the number of qubits and depth of the circuit by encoding the evaluation of $f(b,x)$ before rounding into the phase of these output qubits. After the initialization, we use a unitary $U(A,x,s,e)$ to evolve the qubits, advancing the phase of the output qubits according to the input qubits, which is detailed more in Section ?? . In order to recover the most significant bit (i.e. the rounded value of the function), we measure the output qubits along the axis that is $3\pi/8$ away from the x-axis in the xy-plane. Following this measurement, obtaining the result bitstring w , the state collapses to $1/\sqrt{2}(|0\rangle |X_1\rangle + |1\rangle |X_2\rangle) |w = f(0, X_1) = f(1, X_2)\rangle$.

With the measurement result w reported to the verifier, the verifier then uses the trapdoor to compute the preimages corresponding to w , i.e. $(0, X_1)$ and $(1, X_2)$ such that $w = f(0, X_1) = f(1, X_2)$. Then, the prover will be asked to either measure all the input qubits in the b and x registers directly in the computational basis or in the Hadamard basis. If the verifier chooses the first challenge, the prover obtains $(0, X_1)$ or $(1, X_2)$, from the measurement. The verifier can then efficiently check that this is a valid pre-image by comparing with the pre-images computed previously with the trapdoor, certifying that the prover indeed evaluated the $f(x)$ properly. In the alternative challenge, if the prover measures in Hadamard basis, they will obtain bitstrings b' and d . If the prover was acting honestly, these bitstrings

should be related via the equation $b' = d \cdot (X_1 \oplus X_2)$.

These two challenges are referred to as Branch A and Branch B, respectively, in the figures. As mentioned previously, this simple measurement scheme is the result of a cryptographic property of the LWE based TCF called the *adaptive hardcore bit* [102]. This property states that a classical prover cannot succeed in both Branch A and B at the same time with high probability. Here, in the interactive protocol, by forcing the prover to commit to a specific w bitstring in the first measurement before the verifier decides which challenge to issue, we guarantee that the prover must have quantum capabilities. Otherwise, in order to succeed in the verifier's challenge, a classical prover must have been able to succeed in both branches simultaneously, violating the adaptive hardcore bit property.

7.2.2 Circuit construction for the LWE based protocol

Here we detail the procedure to implement the circuit given at a high level in Fig.???. First, the verifier samples $A \in Z_q^{m \times n}$, $s \in \{0, 1\}^n$, and $e \in Z_q^m$, the LWE instance as described in [106]. The verifier then sends A and $y = As + e \in Z_q^m$ to the prover. In order to evaluate the function $f(b, x) = \lfloor Ax + b \cdot y \rfloor$ in superposition, the prover first applies a layer of Hadamard gates to all qubits to put the state into equal superposition of all states. Then the major effort is to create the state

$$\sum_{b \in \{0, 1\}} \sum_{x \in Z_q^n} \frac{1}{\sqrt{2^N}} |b\rangle |x\rangle |\lfloor Ax + by \rfloor\rangle \quad (7.1)$$

where $N = 1 + n \log(q) + m$ is the total number of qubits required. This is the operation depicted by the $U(A, x, s, e)$ in Fig.???. In the instance for this experiment, we choose $m = 6, n = 2, q = 8$, so $N = 13$. Naturally, the circuit is consist of three registers: two for the inputs to the function b and x , and one for storing the result of evaluating the function itself.

The first register contains $|b\rangle$ which requires only one qubit, as the b input to the function is a single bit. In the second register, the vector $\mathbf{x} = (x_0, x_1)$ in the modulo 8 space is encoded into the binary representation with six qubits as $|\mathbf{x}\rangle = |x_{00}, x_{01}, x_{02}, x_{10}, x_{11}, x_{12}\rangle$. Lastly, in the third register, we encode $\theta = \langle A_i, x \rangle + b \cdot y_i$ into the phase of the i th qubit, where A_i denotes the i th row of A and y_i denotes the i th entry of the y vector. Further note that $\langle A_i, x \rangle$ is the inner product modulo $q = 8$. In this way, when measured in a specific basis, we can obtain the rounded result $\lfloor Ax + b \cdot y \rfloor$ with high probability. This requires another six qubits.

Since θ is in the space Z_q^m , to store it in the phase, the 2π plane is evenly divided by q parts, so that the phases in the third register is encoded as

$$\bigotimes_{j=1}^m \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i\frac{2\pi}{q}\theta_j} |1\rangle \right) \quad (7.2)$$

As a concrete example, for our circuits, A is a 6×2 matrix, so that the first qubit of the third register encodes the information $\theta_0 = A_{00}(2^2x_{00} + 2^1x_{01} + 2^0x_{02}) + A_{01}(2^2x_{10} + 2^1x_{11} + 2^0x_{12}) + by_0$. Thus, encoding this in the phase of the first qubit,

this qubit is in the state

$$\begin{aligned}
& \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i\frac{\pi}{4}[A_{00}(2^2x_{02}+2^1x_{01}+2^0x_{00})+A_{01}(2^2x_{12}+2^1x_{11}+2^0x_{10})+by_0]} |1\rangle \right) \\
&= \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i\frac{\pi}{4}[A_{00}(2^2x_{02})]} e^{i\frac{\pi}{4}[A_{00}(2^1x_{01})]} e^{i\frac{\pi}{4}[A_{00}(2^0x_{00})]} e^{i\frac{\pi}{4}[A_{01}(2^2x_{12})]} e^{i\frac{\pi}{4}[A_{01}(2^1x_{11})]} e^{i\frac{\pi}{4}[A_{01}(2^0x_{10})]} e^{i\frac{\pi}{4}[by_0]} |1\rangle \right)
\end{aligned} \tag{7.3}$$

From this equation, we obtain this phase of the first qubit in the third register by executing controlled rotations by suitable rotation angles, where these rotations are controlled on the first and second registers. For example, to obtain the phase $e^{i\frac{\pi}{4}[A_{00}(2^2x_{02})]}$ we apply $\text{Rz}(\frac{\pi}{4}[2^2A_{02}])$ controlled on the qubit encoding x_{02} , which is the third qubit in the second register. The control sequence for other qubits can also be derived in the similar way.

Finally, we measure the third register in a specific basis in order to recover the rounded result of $\lfloor Ax + b \cdot y \rfloor$. In order to achieve this, we measure against the axis $3\pi/8$ away from the X axis. This corresponds to applying the gates $\text{Rz}(-3\pi/8)$ followed by Hadamard gates to all of the qubits in the third register.

The prover will then measure the first and second registers in either Z basis or X basis according to the challenge issued by the verifier. Should the verifier choose to measure in X basis, the prover applies Hadamard gates on all qubits in the first and the second register before measuring in the computational basis.

To specify further about these qubits in the third register and how we recover the most significant bit, this is based on a quantum random access code [107]. Each

of the m qubits are put into the state

$$|+\theta\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta}|1\rangle), \quad \theta = \frac{2\pi}{q}\langle A_i, x \rangle + b \cdot y_i$$

using controlled Z rotations, where recall again that A_i denotes the i th row of A and y_i denotes the i th entry of y . For the case of $q = 8$, there are 8 possible such $|+\theta\rangle$ states, as depicted in Fig.7.3 below. The bits above each $|+\theta\rangle$ state correspond to the binary representation of $\langle A_i, x \rangle + b \cdot y_i$.

(0,0) circle (2.5cm); (-3.5, 0) – (3.5, 0); (0, -3.5) – (0, 3.5); (-2.5, -2.5) – (2.5, 2.5); (-2.5, 2.5) – (2.5, -2.5); [thick, blue]

Figure 7.3: Visualization of States in XY Plane of Bloch Sphere

Here, notice that the four states corresponding to most significant bit 0 are on the right side of the dark blue line, while those corresponding to 1 are on the left side. Thus, if we perform a measurement along the axis orthogonal to the dark axis, i.e. the green line, then we will recover the correct most significant bit with high probability. From the figure, we see that measuring along the green axis corresponds to measuring in the basis $\{|+_{3\pi/8}\rangle, |-_{3\pi/8}\rangle\}$. This is the reason why we measure against the axis $3\pi/8$ away from the X axis.

7.2.3 $x^2 \bmod N$ based protocol

The second variation uses the TCF $f(x) = x^2 \bmod N$. This function does not have the extra cryptographic property of the adaptive hardcore bit, so we use a scheme that is on a computational equivalent of the Bell inequality or CHSH game [108, 109]. Like in the LWE protocol, in this case, the prover prepare the

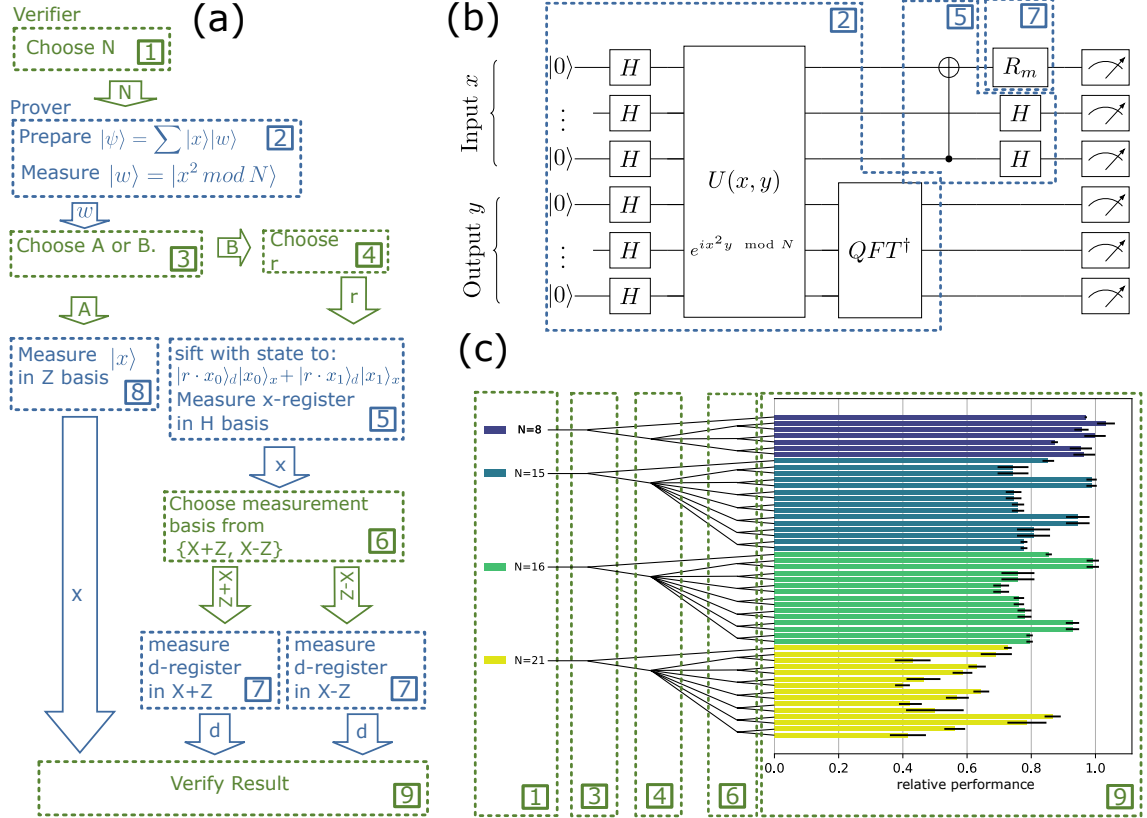


Figure 7.4: (a):The flow chart of the interactive protocol based on $x^2 \bmod N$. The green boxes stands for verifier's actions. The blue boxes stands for prover's actions. The arrows stands for the exchange of information. (b):The circuit diagram of quantum operations the prover would need to perform to answer the verifier's challenge. (c):The experimental results of all the branches of circuits, in relative performance (as defined in eq. 7.14). The boxes marked with the same index correspond to the same operations.

superposition according to the configuration determined by the verifier. Then in branch A the verifier requests a simple computational basis measurement of the x -register (input), yielding a value x . But in the B branch, the information contained in the x values is “condensed” into a single qubit. This is accomplished by performing a series of CNOT gates from a random subset r of the qubits holding x , where r is chosen by the verifier. After the CNOTs are performed, all qubits other than the target of the CNOTs are measured in the Hadamard basis. The prover reports this measurement to the verifier. Ultimately, the remaining qubit is in one of the states $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$; it depends on the cryptographic secret in the same way the state of one qubit in a Bell pair depends on the measurement basis and outcome of the other qubit. This qubit is measured in an “intermediate” basis $X + Z$ or $X - Z$ randomly chosen by the verifier. Ultimately, the correlation between this measurement result and the cryptographic secret constitutes the proof of quantumness, in the same way that correlations between Bell measurement outcomes constitute a proof of entanglement.

7.2.4 circuit construction of the $x^2 \bmod N$ based protocol

As illustrated in Fig.??, the circuit consists of several functional blocks. The major parts of the circuit is to prepare the following state:

$$\sum_{0 \leq x \leq N/2} \frac{1}{\sqrt{2^{N/2}}} |x\rangle |f(x) = x^2 \bmod N\rangle \quad (7.4)$$

To achieve this, we first apply Hadamard gates to all to prepare the following state:

$$\sum_{0 \leq x \leq N/2, 0 \leq y \leq N} \alpha |x\rangle |y\rangle \quad (7.5)$$

,where α is the normalization factor.

Next, we evolve the state with the unitary $U(x, y) = e^{2\pi i \frac{x^2 y}{N}}$. Since the phase naturally has period 2π , the unitary is equivalent to $U(x, y) = e^{2\pi i \frac{x^2 y \bmod N}{N}}$. So after applying the unitary, we will have the state

$$\alpha \sum_{0 \leq x \leq N/2} |x\rangle \sum_{0 \leq y \leq N} e^{2\pi i \frac{x^2 y \bmod N}{N}} |y\rangle \quad (7.6)$$

Next, the QFT^\dagger block applies inverse quantum Fourier transformation to y-register will give us:

$$\alpha \sum_{0 \leq x \leq N/2} |x\rangle |x^2 \bmod N\rangle \quad (7.7)$$

Once we get this state, the rest operations are relatively straightforward. To calculate the sifted parity, we apply CNOT gates from the selected qubits in x-register to another qubit in the x-register now re-purposed as the ancilla.

The rest of the protocol then can be straight forwardly implemented using

standard single-qubit rotation.

We now further explain how to efficiently implement $U(x, y) = e^{2\pi i \frac{x^2 y}{N}}$ on ion trap quantum computers.

First, note the multiplication in the phase can be expressed as a sum of bit-wise multiplication

$$U(x, y) = \prod_{i,j,k} \exp(2\pi i \frac{2^{i+j+k}}{N} x_i x_j y_k) \quad (7.8)$$

This operation, expressed with Pauli operator is:

$$\prod_{i,j,k} \exp(2\pi i \frac{2^{i+j+k-3}}{N} (1 - \sigma_z^{(i)})(1 - \sigma_z^{(j)})(1 - \sigma_z^{(k)})) \quad (7.9)$$

.

We can organize the terms into:

$$\prod_{i,j,k} \exp(\alpha_{i,j,k} \sigma_z^{(i)} \sigma_z^{(j)} \sigma_z^{(k)}) \prod_{i,j} \exp(\beta_{i,j} \sigma_z^{(i)} \sigma_z^{(j)}) \prod_i \exp(\gamma_i \sigma_z^{(i)}) \quad (7.10)$$

.

The third term, single-qubit z-rotation, are implemented efficiently as software-phase-advances. The second term, zz-interactions, are efficiently implemented as XX-gate sandwiched between single qubit rotations. The third term, zzz-interactions

can be decompose into ZZ-interactions using the following relation:

$$\exp(-\pi/4i\sigma_y^{(i)}\sigma_y^{(j)})\exp(i\theta\sigma_x^{(j)}\sigma_x^{(k)})\exp(i\pi/4\sigma_y^{(i)}\sigma_y^{(j)}) = \exp(-i\theta\sigma_y^{(i)}\sigma_z^{(j)}\sigma_x^{(k)}) \quad (7.11)$$

With this decomposition, we cascade a chain of zzz-interaction in the following way, which has linear scaling of gates count.

$$\exp(-i\theta\sigma_y^{(a)}\sigma_z^{(b)}\sigma_x^{(1)})\exp(-i\theta\sigma_y^{(a)}\sigma_z^{(b)}\sigma_x^{(2)})\dots = \quad (7.12)$$

$$\exp(-\pi/4i\sigma_y^{(a)}\sigma_y^{(b)})\exp(i\theta\sigma_x^{(b)}\sigma_x^{(1)}).\exp(i\theta\sigma_x^{(b)}\sigma_x^{(2)}).\dots\exp(i\pi/4\sigma_y^{(a)}\sigma_y^{(b)}) \quad (7.13)$$

Sandwiching these operation within single-qubit rotations, we efficiently implement the third term in eq.7.10 as shown in Fig.7.5.

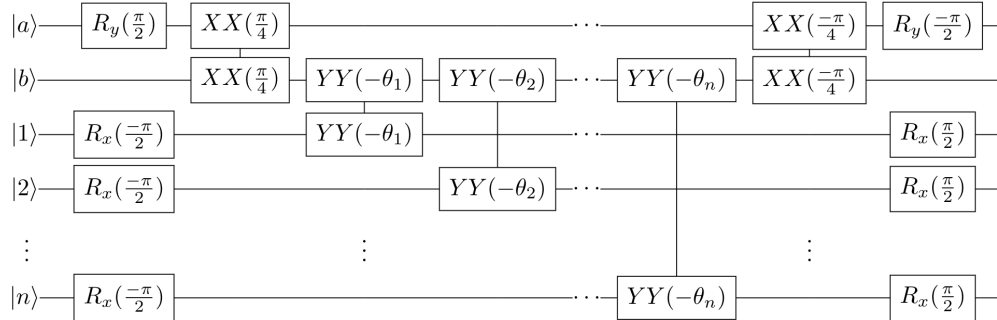


Figure 7.5: detailed circuits

7.2.5 experiments

We implement the protocols on an ion-trap quantum computer with 13 individual addressed qubits. For experiments of this scale, the verifier need not sample

randomly to generate their queries—instead, the total number of possible challenges is small enough that we can simply enumerate all of them, and demonstrate the device’s overall performance directly for every possible choice the verifier could make. This exhaustive enumeration brings the added benefit of allowing all measurements to be deferred to the end of the protocol, since it serves the same purpose as the “commitment” of the prover in the protocols. We call each of the verifier’s possible choices a “branch” of the protocol; each is an instance of the circuits shown in Fig. 7.2 (b) and Fig. 7.4 (b). (See appendix for details of circuit construction using operations native to ion-trap quantum computers). We generate a separate quantum circuit for each branch, and then apply an automated optimization routine to process each one. The optimization routine uses several circuits reduction and cancellation techniques to reduce the number of two-qubit gates, as well as the total number of gates [110]. Once the circuits are fully optimized they are executed on the trapped-ion quantum computer.

For the LWE based protocol, we explore eight different configurations, corresponding to eight different choices of the matrix A and vectors s and e . For each configuration we generate two circuits, corresponding to the verifier’s two choices of branches: asking for the pre-image (A) or asking for the Hadamard-basis measurement (B) (see Fig. 7.2 for more details). All the $2 * 8 = 16$ circuits studied here involve 13 qubits: 1 in the b-register, 6 in the x-register, and 6 in the w-register. The number of two-qubit gates involved in the optimized circuits of different configurations ranges from 21 to 31. We cycled through all 16 circuits six times for a total of 96 separate runs, with 2000 shots in each attempt.

For each circuit, we compare the probability of a prover passing the verifier's test using the experimental results ($p_{\text{experiment}}$) against the probability of a prover doing so with random guessing (p_{guess}). We quantify this comparison using a value defined as:

$$\text{relative performance} = \frac{p_{\text{experiment}} - p_{\text{guess}}}{p_{\text{ideal}} - p_{\text{guess}}}. \quad (7.14)$$

p_{ideal} is the value of $p_{\text{experiment}}$ in the error-free limit. So in each case, relative performance will reach one with an ideal quantum computer, and zero if your computer is no better than a random guess, i.e. completely decohered. The results are shown in Fig. 7.4. Using the output of our experiments, the prover can outperform a random guesser reliably in all but one cases.

For the protocol based on $x^2 \bmod N$, we explore four different configurations corresponding to $N \in \{8, 15, 16, 21\}$. According to the protocol, the value of x ranges from 0 to $N/2$. The value of $w = x^2 \bmod N$ ranges from 0 to N . So to represent them properly with the x -register and w -register, we use $2 + 3 = 5$, $3 + 4 = 7$, and $3 + 4 = 7$ qubits for $N = 8, 15, 16$ respectively. For the case of $N = 21$, we have to decrease the number of qubit in x -register from 4 to 3 because of a hardware limit on the total number of gates. With this compromise, we miss some colliding input pairs. Consequently, errors occur in branch B if an input is missing from an expected colliding pair.

For branch A of each configuration, we generate one circuit in which the the

pre-image is measured together with w . For branch B, we generate two separate circuits for every non-zero choice of the “sifting” set r , and each of the two measurement bases for that final single-qubit measurement: X+Z or X-Z. For $N = 8$, this yields 7 different circuits; for $N = 15$ and $N = 16$ it yields 15 circuits. In the $N = 21$ case there are ostensibly 31 possible circuits, but since we reduce the size of the x-register by one, ultimately yielding 15 total circuits. We cycle through all the circuits on the ion trap quantum computer three times, with 2000 shots in each run. The circuit results are again evaluated using the relative performance defined in Eq. 7.14. As Fig. 7.4(c) shows, using the experiment outputs, the prover consistently outperforms random guessing. Configurations that requires more qubits also involve more gate operations. Thus, due to accumulation of operation errors, the performance of circuits with $N = 8$ is better than that of $N = 15$ and $N = 16$. For $N = 21$, a major part of performance degradation is due to the insufficient number of qubits in the x-register. We note that the verifier bins statistics from all of the different choices of r together, yielding very good statistical uncertainty despite taking fewer shots of each individual circuit when compared to the LWE case.

should rewrite this next couple paragraphs when we have the LWE framing nailed down, so that the story is consistent

Due to the mathematically possible existence of a better classical strategy, beating the random guesser in every possible branches, as we’ve shown here, is not enough to claim quantum advantage. As derived in [citing Greg’s paper], no classical strategy can have $P_x + 4P_m > 4$. Otherwise, this classical strategy can be

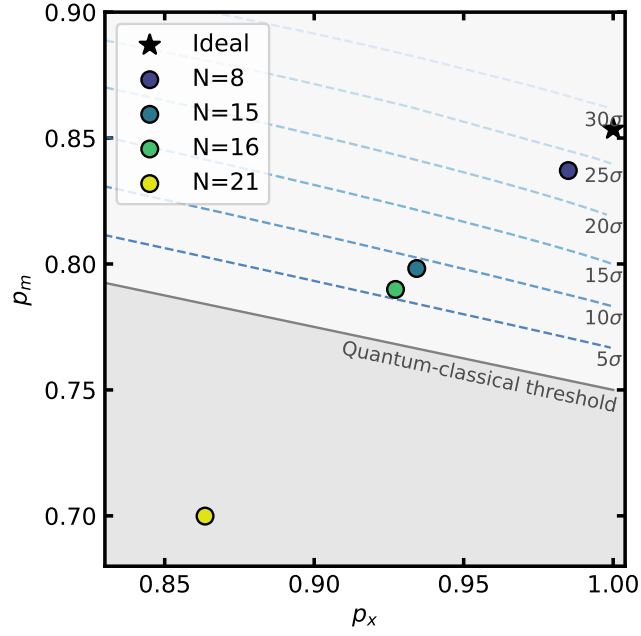


Figure 7.6: quantum advantage plot

used to drive an algorithm that factorizes large number on a classical computer. We used our results from all the possible branches to calculate our strict advantage over classical prover. The results are shown in Fig. 7.6.

As seen in Fig. 7.6, our results for $N=8$, $N=15$, and $N=16$ configurations exceed the quantum-classical threshold with more than 5σ . The $N=21$ configuration didn't pass the threshold, again, largely due to the approximation involved in the implementation of relevant circuits.

Bibliography

- [1] Rahul Nandkishore, Sarang Gopalakrishnan, and David A. Huse. Spectral features of a many-body-localized system weakly coupled to a bath. *Phys. Rev. B*, 90:064203, Aug 2014.
- [2] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
- [3] J. J. Sakurai. *Modern Quantum Mechanics (Revised Edition)*. Addison Wesley, 1 edition, September 1993.
- [4] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.
- [5] Chetan Nayak, Steven H Simon, Ady Stern, Michael Freedman, and Sankar Das Sarma. Non-abelian anyons and topological quantum computation. *Reviews of Modern Physics*, 80(3):1083, 2008.
- [6] Peter W Shor. Scheme for reducing decoherence in quantum computer memory. *Physical review A*, 52(4):R2493, 1995.
- [7] Christopher Monroe and Jungsang Kim. Scaling the ion trap quantum processor. *Science*, 339(6124):1164–1169, 2013.
- [8] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.
- [9] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
- [10] Shantanu Debnath, Norbert M Linke, Caroline Figgatt, Kevin A Landsman, Kevin Wright, and Christopher Monroe. Demonstration of a small programmable quantum computer with atomic qubits. *Nature*, 536(7614):63, 2016.

- [11] Timothy Andrew Manning. *Quantum information processing with trapped ion chains*. PhD thesis, 2014.
- [12] D Zhu, S Johri, NH Nguyen, C Huerta Alderete, KA Landsman, NM Linke, C Monroe, and AY Matsuura. Probing many-body localization on a noisy quantum computer. *Physical Review A*, 103(3):032606, 2021.
- [13] D Zhu, NM Linke, M Benedetti, KA Landsman, NH Nguyen, CH Alderete, A Perdomo-Ortiz, N Korda, A Garfoot, C Brecque, et al. Training of quantum circuits on a hybrid quantum computer. *arXiv preprint arXiv:1812.08862*, 2018.
- [14] D Zhu, S Johri, NM Linke, KA Landsman, C Huerta Alderete, NH Nguyen, AY Matsuura, TH Hsieh, and C Monroe. Generation of thermofield double states and critical ground states with a quantum computer. *Proceedings of the National Academy of Sciences*, 117(41):25402–25406, 2020.
- [15] Felix Medland Arscott. *Periodic differential equations: an introduction to Mathieu, Lamé, and allied functions*, volume 66. Elsevier, 2014.
- [16] J Harold. Metcalf and peter van der straten, laser cooling and trapping. *Springer*, 1999.
- [17] S. Olmschenk, K. C. Younge, D. L. Moehring, D. N. Matsukevich, P. Maunz, and C. Monroe. Manipulation and detection of a trapped yb^+ hyperfine qubit. *Phys. Rev. A*, 76:052314, Nov 2007.
- [18] Christopher J Foot et al. *Atomic physics*, volume 7. Oxford University Press, 2005.
- [19] Juan I Cirac and Peter Zoller. Quantum computations with cold trapped ions. *Physical review letters*, 74(20):4091, 1995.
- [20] Klaus Mølmer and Anders Sørensen. Multiparticle entanglement of hot trapped ions. *Phys. Rev. Lett.*, 82:1835–1838, Mar 1999.
- [21] Sergio Blanes, Fernando Casas, Jose-Angel Oteo, and José Ros. The magnus expansion and some of its applications. *Physics reports*, 470(5-6):151–238, 2009.
- [22] Harry L Trentelman, Anton A Stoorvogel, and Malo Hautus. *Control theory for linear systems*. Springer Science & Business Media, 2012.
- [23] Shantanu Debnath. *A programmable five qubit quantum computer using trapped atomic ions*. PhD thesis, 2016.
- [24] SW Allison, GT Gillies, DW Magnuson, and TS Pagano. Pulsed laser damage to optical fibers. *Applied optics*, 24(19):3140–3145, 1985.

- [25] Patricia J Lee, Kathy-Anne Brickman, Louis Deslauriers, Paul C Haljan, Lu-Ming Duan, and Christopher Monroe. Phase control of trapped ion quantum gates. *Journal of Optics B: Quantum and Semiclassical Optics*, 7(10):S371, 2005.
- [26] Peter Lukas Wilhelm Maunz. High optical access trap 2.0. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2016.
- [27] Rodney Loudon. *The quantum theory of light*. OUP Oxford, 2000.
- [28] R Gilmore. Baker-campbell-hausdorff formulas. *Journal of Mathematical Physics*, 15(12):2090–2092, 1974.
- [29] Dmitry A. Abanin, Ehud Altman, Immanuel Bloch, and Maksym Serbyn. Colloquium: Many-body localization, thermalization, and entanglement. *Rev. Mod. Phys.*, 91:021001, May 2019.
- [30] Rahul Nandkishore and David A. Huse. Many-body localization and thermalization in quantum statistical mechanics. *Annual Review of Condensed Matter Physics*, 6(1):15–38, 2015.
- [31] Fabien Alet and Nicolas Laflorencie. Many-body localization: An introduction and selected topics. *Comptes Rendus Physique*, 19(6):498 – 525, 2018. Quantum simulation / Simulation quantique.
- [32] Thorsten B. Wahl, Arijeet Pal, and Steven H. Simon. Efficient representation of fully many-body localized systems using tensor networks. *Phys. Rev. X*, 7:021018, May 2017.
- [33] A. Chandran, J. Carrasquilla, I. H. Kim, D. A. Abanin, and G. Vidal. Spectral tensor networks for many-body localization. *Phys. Rev. B*, 92:024201, Jul 2015.
- [34] Andrew M. Childs, Dmitri Maslov, Yunseong Nam, Neil J. Ross, and Yuan Su. Toward the first quantum simulation with quantum speedup. *Proceedings of the National Academy of Sciences*, 115(38):9456–9461, 2018.
- [35] Henrik P. Lüschen, Pranjal Bordia, Sebastian Scherg, Fabien Alet, Ehud Altman, Ulrich Schneider, and Immanuel Bloch. Observation of slow dynamics near the many-body localization transition in one-dimensional quasiperiodic systems. *Phys. Rev. Lett.*, 119:260401, Dec 2017.
- [36] Henrik P. Lüschen, Pranjal Bordia, Sean S. Hodgman, Michael Schreiber, Saubhik Sarkar, Andrew J. Daley, Mark H. Fischer, Ehud Altman, Immanuel Bloch, and Ulrich Schneider. Signatures of many-body localization in a controlled open quantum system. *Phys. Rev. X*, 7:011034, Mar 2017.
- [37] Pranjal Bordia, Henrik Lüschen, Sebastian Scherg, Sarang Gopalakrishnan, Michael Knap, Ulrich Schneider, and Immanuel Bloch. Probing slow relaxation and many-body localization in two-dimensional quasiperiodic systems. *Phys. Rev. X*, 7:041047, Nov 2017.

- [38] Pranjal Bordia, Henrik P. Lüschen, Sean S. Hodgman, Michael Schreiber, Immanuel Bloch, and Ulrich Schneider. Coupling identical one-dimensional many-body localized systems. *Phys. Rev. Lett.*, 116:140401, Apr 2016.
- [39] Antonio Rubio-Abadal, Jae-yoon Choi, Johannes Zeiher, Simon Hollerith, Jun Rui, Immanuel Bloch, and Christian Gross. Many-body delocalization in the presence of a quantum bath. *Phys. Rev. X*, 9:041014, Oct 2019.
- [40] Thomas Kohlert, Sebastian Scherg, Xiao Li, Henrik P. Lüschen, Sankar Das Sarma, Immanuel Bloch, and Monika Aidelsburger. Observation of many-body localization in a one-dimensional system with a single-particle mobility edge. *Phys. Rev. Lett.*, 122:170403, May 2019.
- [41] J. Smith, A. Lee, P. Richerme, B. Neyenhuis, P. W. Hess, P. Hauke, M. Heyl, D. A. Huse, and C. Monroe. Many-body localization in a quantum simulator with programmable random disorder. *Nature Physics*, 12(10):907–911, 2016.
- [42] PW Hess, P Becker, HB Kaplan, A Kyprianidis, AC Lee, B Neyenhuis, G Pagano, Phil Richerme, C Senko, J Smith, et al. Non-thermalization in trapped atomic ion spin chains. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 375(2108):20170107, 2017.
- [43] P. Roushan, C. Neill, J. Tangpanitanon, V. M. Bastidas, A. Megrant, R. Barends, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. Fowler, B. Foxen, M. Giustina, E. Jeffrey, J. Kelly, E. Lucero, J. Mutus, M. Neeley, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. White, H. Neven, D. G. Angelakis, and J. Martinis. Spectroscopic signatures of localization with interacting photons in superconducting qubits. *Science*, 358(6367):1175–1179, 2017.
- [44] B. Chiaro, C. Neill, A. Bohrdt, M. Filippone, F. Arute, K. Arya, R. Babbush, D. Bacon, J. Bardin, R. Barends, S. Boixo, D. Buell, B. Burkett, Y. Chen, Z. Chen, R. Collins, A. Dunsworth, E. Farhi, A. Fowler, B. Foxen, C. Gidney, M. Giustina, M. Harrigan, T. Huang, S. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. Klimov, A. Korotkov, F. Kostitsa, D. Landhuis, E. Lucero, J. McClean, X. Mi, A. Megrant, M. Mohseni, J. Mutus, M. McEwen, O. Naaman, M. Neeley, M. Niu, A. Petukhov, C. Quintana, N. Rubin, D. Sank, K. Satzinger, A. Vainsencher, T. White, Z. Yao, P. Yeh, A. Zalcman, V. Smelyanskiy, H. Neven, S. Gopalakrishnan, D. Abanin, M. Knap, J. Martinis, and P. Roushan. Growth and preservation of entanglement in a many-body localized system, 2019.
- [45] X. Zou, S. P. Premaratne, M. A. Rol, S. Johri, V. Ostroukh, D. J. Michalak, R. Caudillo, J. S. Clarke, L. DiCarlo, and A. Y. Matsuura. Enhancing a near-term quantum accelerator’s instruction set architecture for materials science applications. *IEEE Transactions on Quantum Engineering*, 1:1–7, 2020.

- [46] Sonika Johri, Rahul Nandkishore, and R. N. Bhatt. Many-body localization in imperfectly isolated quantum systems. *Phys. Rev. Lett.*, 114:117401, Mar 2015.
- [47] S D Geraedts, N Regnault, and R M Nandkishore. Characterizing the many-body localization transition using the entanglement spectrum. *New Journal of Physics*, 19(11):113021, nov 2017.
- [48] N. M. Linke, S. Johri, C. Figgatt, K. A. Landsman, A. Y. Matsuura, and C. Monroe. Measuring the rényi entropy of a two-site fermi-hubbard model on a trapped ion quantum computer. *Phys. Rev. A*, 98:052334, Nov 2018.
- [49] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandra, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.
- [50] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, 2016.
- [51] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671):242, 2017.
- [52] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O’Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5:4213, 2014.

- [53] Cornelius Hempel, Christine Maier, Jonathan Romero, Jarrod McClean, Thomas Monz, Heng Shen, Petar Jurcevic, Ben P. Lanyon, Peter Love, Ryan Babbush, Alán Aspuru-Guzik, Rainer Blatt, and Christian F. Roos. Quantum chemistry calculations on a trapped-ion quantum simulator. *Phys. Rev. X*, 8:031022, Jul 2018.
- [54] PJJ O’Malley, Ryan Babbush, ID Kivlichan, Jonathan Romero, JR McClean, Rami Barends, Julian Kelly, Pedram Roushan, Andrew Tranter, Nan Ding, et al. Scalable quantum simulation of molecular energies. *Physical Review X*, 6(3):031007, 2016.
- [55] C Kokail, C Maier, R van Bijnen, T Brydges, MK Joshi, P Jurcevic, CA Muschik, P Silvi, R Blatt, CF Roos, et al. Self-verifying variational quantum simulation of lattice models. *Nature*, 569(7756):355, 2019.
- [56] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *MIT-CTP/4610*, 2014.
- [57] JS Otterbach, R Manenti, N Alidoust, A Bestwick, M Block, B Bloom, S Caldwell, N Didier, E Schuyler Fried, S Hong, et al. Unsupervised machine learning on a hybrid quantum computer. *arXiv preprint arXiv:1712.05771*, 2017.
- [58] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017.
- [59] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR abs/1609.03499*, 2016.
- [60] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *SIGNLL Conference on Computational Natural Language Learning (CONLL)*, 2016, 2016.
- [61] Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In *Advances in Neural Information Processing Systems*, pages 899–907, 2013.
- [62] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.
- [63] Marcello Benedetti, Delfina Garcia-Pintos, Oscar Perdomo, Vicente Leyton-Ortega, Yunseong Nam, and Alejandro Perdomo-Ortiz. A generative modeling

- approach for benchmarking and training shallow quantum circuits. *npj Quantum Information*, 5(1):45, 2019.
- [64] Yuxuan Du, Min-Hsiu Hsieh, Tongliang Liu, and Dacheng Tao. The expressive power of parameterized quantum circuits. *arXiv preprint arXiv:1810.11922*, 2018.
 - [65] X. Gao, Z.-Y. Zhang, and L.-M. Duan. A quantum machine learning algorithm based on generative models. *Science Advances*, 4(12), 2018.
 - [66] David JC MacKay and David JC Mac Kay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.
 - [67] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.
 - [68] Jin-Guo Liu and Lei Wang. Differentiable learning of quantum circuit born machine. *arXiv preprint arXiv:1804.04168*, 2018.
 - [69] J Kennedy and R Eberhart. Particle swarm optimization (pso). In *Proc. IEEE International Conference on Neural Networks, Perth, Australia*, pages 1942–1948, 1995.
 - [70] Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.
 - [71] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.
 - [72] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.
 - [73] Richard E Bellman. *Adaptive control processes: a guided tour*, volume 2045. Princeton university press, 2015.
 - [74] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.
 - [75] A Higuchi and A Sudbery. How entangled can two couples get? *Physics Letters A*, 273(4):213–217, 2000.
 - [76] M. H. Devoret and R. J. Schoelkopf. Superconducting circuits for quantum information: An outlook. *Science*, 339(6124):1169–1174, 2013.
 - [77] C. Monroe and J. Kim. Scaling the ion trap quantum processor. *Science*, 339:1164, 2013.

- [78] Serge Haroche. Nobel lecture: Controlling photons in a box and exploring the quantum to classical boundary. *Rev. Mod. Phys.*, 85:1083–1102, Jul 2013.
- [79] Onur Hosten, Nils J. Engelsen, Rajiv Krishnakumar, and Mark A. Kasevich. Measurement noise 100 times lower than the quantum-projection limit using entangled atoms. *Nature*, 529:505–508, 2016.
- [80] Patrick A Lee, Naoto Nagaosa, and Xiao-Gang Wen. Doping a mott insulator: Physics of high-temperature superconductivity. *Reviews of modern physics*, 78(1):17, 2006.
- [81] David J. Gross, Robert D. Pisarski, and Laurence G. Yaffe. Qcd and instantons at finite temperature. *Rev. Mod. Phys.*, 53:43–80, Jan 1981.
- [82] Kristan Temme, Tobias J Osborne, Karl G Vollbrecht, David Poulin, and Frank Verstraete. Quantum metropolis sampling. *Nature*, 471(7336):87, 2011.
- [83] Barbara M. Terhal and David P. DiVincenzo. Problem of equilibration and the computation of correlation functions on a quantum computer. *Phys. Rev. A*, 61:022301, Jan 2000.
- [84] David Poulin and Pawel Wocjan. Sampling from the thermal quantum gibbs state and evaluating partition functions with a quantum computer. *Phys. Rev. Lett.*, 103:220502, Nov 2009.
- [85] F. Brandao and M. Kastoryano. Finite correlation length implies efficient preparation of quantum thermal states . *ArXiv e-prints*, September 2016.
- [86] Robert Jördens, Niels Strohmaier, Kenneth Günter, Henning Moritz, and Tilman Esslinger. A mott insulator of fermionic atoms in an optical lattice. *Nature*, 455(7210):204, 2008.
- [87] Christian Gross and Immanuel Bloch. Quantum simulations with ultracold atoms in optical lattices. *Science*, 357(6355):995–1001, 2017.
- [88] Tosio Kato. On the adiabatic theorem of quantum mechanics. *Journal of the Physical Society of Japan*, 5(6):435–439, 1950.
- [89] Yasushi Takahashi and Hiroomi Umezawa. Thermo field dynamics. *International Journal of Modern Physics B*, 10(13n14):1755–1805, 1996.
- [90] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, UK, 2000.
- [91] W. Israel. Thermo-field dynamics of black holes. *Physics Letters A*, 57(2):107–110, 1976.
- [92] Juan Maldacena. Eternal black holes in anti-de sitter. *Journal of High Energy Physics*, 2003(04):021–021, apr 2003.

- [93] Ping Gao, Daniel Louis Jafferis, and Aron C. Wall. Traversable wormholes via a double trace deformation. *Journal of High Energy Physics*, 2017(12):151, Dec 2017.
- [94] Juan Maldacena, Douglas Stanford, and Zhenbin Yang. Diving into traversable wormholes. *Fortschritte der Physik*, 65(5):1700034, 2017.
- [95] Jingxiang Wu and Timothy H Hsieh. Variational thermal quantum simulation via thermofield double states. *arXiv preprint arXiv:1811.11756*, 2018.
- [96] Subir Sachdev. *Quantum Phase Transitions*. Cambridge University Press, 1999.
- [97] Wen Wei Ho, Cheryne Jonay, and Timothy H. Hsieh. Ultrafast variational simulation of nontrivial quantum states with long-range interactions. *Phys. Rev. A*, 99:052332, May 2019.
- [98] Wen Wei Ho and Timothy H. Hsieh. Efficient variational simulation of non-trivial quantum states. *SciPost Phys.*, 6:29, 2019.
- [99] Leo Zhou, Sheng-Tao Wang, Soonwon Choi, Hannes Pichler, and Mikhail D Lukin. Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. *arXiv preprint arXiv:1812.01041*, 2018.
- [100] Fernando GSL Brandao, Michael Broughton, Edward Farhi, Sam Gutmann, and Hartmut Neven. For fixed control parameters the quantum approximate optimization algorithm’s objective function value concentrates for typical instances. *arXiv preprint arXiv:1812.04170*, 2018.
- [101] T. Choi, S. Debnath, T. A. Manning, C. Figgatt, Z.-X. Gong, L.-M. Duan, and C. Monroe. Optimal quantum control of multimode couplings between trapped ion qubits for scalable entanglement. *Phys. Rev. Lett.*, 112:190502, May 2014.
- [102] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 320–331. IEEE, 2018.
- [103] Alexandru Gheorghiu and Thomas Vidick. Computationally-secure and composable remote state preparation. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1024–1033. IEEE, 2019.
- [104] Tony Metger and Thomas Vidick. Self-testing of a single quantum device under computational assumptions. *arXiv preprint arXiv:2001.09161*, 2020.

- [105] Urmila Mahadev. Classical verification of quantum computations. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 259–267. IEEE, 2018.
- [106] Oded Regev. The learning with errors problem. *Invited survey in CCC*, 7:30, 2010.
- [107] Andris Ambainis, Ashwin Nayak, Amnon Ta-Shma, and Umesh Vazirani. Dense quantum coding and a lower bound for 1-way quantum automata. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 376–383, 1999.
- [108] J. S. Bell. On the Einstein Podolsky Rosen paradox. *Physics Physique Fizika*, 1(3):195–200, November 1964. Publisher: American Physical Society.
- [109] John F. Clauser, Michael A. Horne, Abner Shimony, and Richard A. Holt. Proposed Experiment to Test Local Hidden-Variable Theories. *Physical Review Letters*, 23(15):880–884, October 1969. Publisher: American Physical Society.
- [110] Yunseong Nam, Neil J Ross, Yuan Su, Andrew M Childs, and Dmitri Maslov. Automated optimization of large quantum circuits with continuous parameters. *npj Quantum Information*, 4(1):1–12, 2018.