

박사학위논문  
Ph.D. Dissertation

양자 정보 처리를 위한 리드버그 원자 배열의 생성  
및 제어

Formation and Control of Rydberg Atom Arrays for Quantum  
Information Processing

2019

이우준 (李宇峻 Lee, Woojun)

한국과학기술원

Korea Advanced Institute of Science and Technology

박사학위논문

양자 정보 처리를 위한 리드버그 원자 배열의 생성  
및 제어

2019

이우준

한국과학기술원

물리학과

# 양자 정보 처리를 위한 리드버그 원자 배열의 생성 및 제어

이 우 준

위 논문은 한국과학기술원 박사학위논문으로  
학위논문 심사위원회의 심사를 통과하였음

2018년 11월 12일

심사위원장 안재욱 (인)

심사위원 조용훈 (인)

심사위원 서민교 (인)

심사위원 최형순 (인)

심사위원 최재윤 (인)

# Formation and Control of Rydberg Atom Arrays for Quantum Information Processing

Woojun Lee

Advisor: Jaewook Ahn

A dissertation submitted to the faculty of  
Korea Advanced Institute of Science and Technology in  
partial fulfillment of the requirements for the degree of  
Doctor of Philosophy in Physics

Daejeon, Korea  
November 12, 2018

Approved by

---

Jaewook Ahn  
Professor of Department of Physics

The study was conducted in accordance with Code of Research Ethics<sup>1</sup>.

<sup>1</sup> Declaration of Ethical Conduct in Research: I, as a graduate student of Korea Advanced Institute of Science and Technology, hereby declare that I have not committed any act that may damage the credibility of my research. This includes, but is not limited to, falsification, thesis written by someone else, distortion of research findings, and plagiarism. I confirm that my thesis contains honest conclusions based on my own careful research under the guidance of my advisor.

DPH  
20138080

이우준. 양자 정보 처리를 위한 리드버그 원자 배열의 생성 및 제어. 물리학과 . 2019년. 110+v 쪽. 지도교수: 안재욱. (영문 논문)

Woojun Lee. Formation and Control of Rydberg Atom Arrays for Quantum Information Processing. Department of Physics . 2019. 110+v pages. Advisor: Jaewook Ahn. (Text in English)

### 초 록

중성 원자 배열을 이용한 양자 정보 처리에 기반이 되는 실험적 부분들 및 그에 대한 분석을 다루었다. 포획된 단일 중성원자 배열에서 개별적 원자의 전송 및 배열의 재배치를 구현하였으며, 이를 이용하여 본질상 부분적으로만 채워지는 단일 원자 배열로부터 확정적 원자 배열을 얻을 수 있음을 실험적으로 보였다. 또한 원자 사이의 리드버그 차단을 이용한 양자적 얽힘을 최대  $N=5$ 까지의 포획된 원자에 대해 구현하였고, 이러한 시스템의 역학에 큰 영향을 주는 환경적 요인들을 고려하여  $N=3-5$ 개의 리드버그 원자의 역학에 대한 정확한 모델링을 제시하였다.

핵심 낱말 원자 물리, 광학, 단일 원자, 리드버그 원자, 양자정보, 양자컴퓨터

### Abstract

Fundamental experiments and analysis on quantum information processing with neutral atom arrays are presented. Individual transport of trapped single atoms and reconfiguration of single atom arrays were realized, and assembly of deterministic arrays from single atom arrays that are only partially filled by nature, with the single atom transport method, was demonstrated. Also, quantum entanglement of up to  $N=5$  trapped single atoms via Rydberg blockade was demonstrated and a well-fitting model of the dynamics of the system is suggested considering the environmental condition that considerably affects the dynamics.

Keywords atomic physics, optics, single atoms, Rydberg atoms, quantum information, quantum computer



# Contents

Contents . . . . .	i
List of Tables . . . . .	iv
List of Figures . . . . .	v
<b>Chapter 1. Introduction</b>	<b>1</b>
1.1 Neutral atom qubits . . . . .	1
1.2 Single- and two-qubit operation . . . . .	2
1.3 Dipole traps for neutral atoms . . . . .	2
1.4 Rydberg atoms and Rydberg blockade . . . . .	3
<b>Chapter 2. Magneto-optical traps for cold atom loading</b>	<b>4</b>
2.1 Vacuum . . . . .	4
2.2 Magnetic field gradient by anti-Helmholtz coil . . . . .	7
2.3 Cooling laser . . . . .	7
2.4 Repump laser . . . . .	8
2.5 Earth's magnetic field compensation . . . . .	8
<b>Chapter 3. Transport of single atoms in three-dimensional space</b>	<b>10</b>
3.1 Experimental requirement . . . . .	11
3.1.1 Single atom trapping with dipole traps . . . . .	11
3.1.2 Imaging . . . . .	12
3.1.3 Computer control . . . . .	12
3.2 Procedure . . . . .	13
3.2.1 MOT loading . . . . .	13
3.2.2 Single atom loading . . . . .	13
3.3 SLM phase calculation . . . . .	14
3.4 Results . . . . .	15
3.4.1 Transport in 2D . . . . .	15
3.4.2 Transport in 3D . . . . .	17
3.4.3 Efficiency and loss analysis . . . . .	18
<b>Chapter 4. Defect-free atom array formation</b>	<b>23</b>
4.1 Closed-loop feedback system . . . . .	25
4.1.1 Optical and computational layout . . . . .	25
4.1.2 GPU-accelerated phase calculation . . . . .	25

4.1.3	MATLAB code for analysing the array status and assigning path . . . . .	37
4.1.4	MATLAB code of the Hungarian algorithm . . . . .	39
4.2	Flicker-free phase solution for dynamic holographic traps . . . .	44
4.3	Path planning . . . . .	47
4.3.1	Heuristic move . . . . .	49
4.3.2	Backpropagation method . . . . .	49
4.3.3	Atom-target site matching (source indexing) . . . . .	49
4.3.4	Hungarian matching algorithm . . . . .	53
4.4	Collision-free property of matching with Hungarian algorithm .	54
4.5	Experiment . . . . .	55
4.5.1	Procedure . . . . .	55
4.5.2	Result . . . . .	55
4.6	Conclusion . . . . .	57
<b>Chapter 5.</b>	<b>N-atom interacting system with Rydberg blockade</b>	<b>64</b>
5.1	Introduction . . . . .	64
5.2	Experimental setup . . . . .	65
5.2.1	780 nm laser and 480 nm laser . . . . .	65
5.2.2	Quantization Axis . . . . .	66
5.3	Experiment procedure . . . . .	66
5.3.1	Atom Trap and Optical Pumping . . . . .	66
5.3.2	Rydberg Excitation . . . . .	67
5.3.3	State measurement by recapture method . . . . .	67
5.4	Rydberg blockade and collective excitation . . . . .	67
5.5	Dynamics of $N = 3 \sim 5$ atoms with environmental errors . . . .	68
5.5.1	Main results . . . . .	68
5.5.2	Detailed implementation . . . . .	72
5.5.3	Conclusion . . . . .	83
5.6	Machine learning of N-atom dynamics . . . . .	84
5.6.1	Curve fitting vs. machine learning: computational complexity . . . . .	84
5.6.2	N-atom dynamics . . . . .	84
5.6.3	Artificial neural network . . . . .	85
5.6.4	Program codes . . . . .	86
5.6.5	Results and discussion . . . . .	94
<b>Chapter 6.</b>	<b>Conclusion and outlook</b>	<b>95</b>

<b>Bibliography</b>	<b>97</b>
<b>Acknowledgments in Korean</b>	<b>106</b>
<b>Curriculum Vitae</b>	<b>107</b>

## List of Tables

4.1	Comparison of phase generating algorithms. . . . .	44
4.2	Comparison of path planning algorithms. . . . .	48
4.3	Comparison of atom-site matching algorithms . . . . .	50
5.1	Fitting and fixed parameters in simulation . . . . .	74
5.2	Individual dephasing sources. . . . .	81
5.3	Collective dephasing sources . . . . .	82
5.4	Projection measurement error sources . . . . .	83

## List of Figures

2.1	Principle of mageneto-optical traps. The schematic picture is adopted from Ref. [31]	5
2.2	Design of the vaccum chamber. . . . .	6
2.3	Pressure inside the vacuum chamber while baking . . . . .	6
2.4	Photo of the vaccum chamber and surrounding optics. . . . .	9
3.1	Technical image of the objective lens. . . . .	11
3.2	Schematic illustration of the experimental setup. Optical microtraps were programmed with the LCOS-SLM to capture single atoms from the pre-cooled $^{87}\text{Rb}$ ensemble in the MOT. The information of the trapped single-atom configuration obtained with the EMCCD was sent back to the SLM computer for the feedback control. . . . .	12
3.3	(a) Transverse displacement of focal spots. (b) Axial displacement. (c) Phase pattern synthesis. . . . .	15
3.4	Selected results of demonstration. (a) Rotation of a 3-by-3 array as a collective control. (b) 2D vacancy filling and (c) ‘Worm running’ as individual atom controls. (d-f) ‘Rightward alignment’ as feedback controls of atom arrays. The leftmost column presents the schematic diagram of each operation scenario. In each column, the initial and in-between photos are followed by the final photos [36].	16
3.5	(a) Trapping and imaging of a 3D single-atom array, where the image plane was shifted by translating the EMCCD. Each image corresponded to the EMCCD position at $z = -10, -5, 0, 5, 10$ mm and was accumulated from 100 single-event images that were captured right after the atoms were trapped in each acquisition. (b) Individual transport of single atoms in 3D, where each demonstration spanned 90 SLM frames and each step of atom moving was an equal division of the entire path. 370-time accumulated sequential images and selected single-event sequential images are displayed. The lattice constant of the 2D array was $d = 4.5 \mu\text{m}$ and the depth of the axial travels for the both atoms were $4 \mu\text{m}$ [36].	18
3.6	(a) Schematic illustration for phase jump which causes frame-to-frame flicker of traps. (b) Destructive interference of phase-jump area and no-phase-jump area. (c) Experimental result of flicker-induce atom loss during transport for various values of R [61]. . . . .	19
3.7	Experimental single step loss calculated from experimental total loss for various R values, compared with simulation with two temperatures, $T = U/10, U/13$ [61].	20

3.8	(a) Single-atom transport efficiency vs. the step distance in a 40-step (41 frames) transport operation, where the values for moving in positive and negative directions are drawn with dashed and dash-dot black lines, respectively, and the total average with the blue line with ‘star’ marks. The given transport (only) efficiency is the actually measured probability divided by the probability without moving. (b) Single-atom transport efficiency vs. the travel distance (with a fixed step distance of 250 nm) for various frame refresh rates and directions. (c), (d) The single-step loss vs. the step distance and travel distance, calculated from the main data in (a) and (b), respectively [36]. . . . .	21
4.1	Array reconfiguration from an initial probabilistic array into a filled $3 \times 3$ array. Defect-free arrays are achieved by filling vacancies with nearby reservoir atoms. . . . .	23
4.2	A road map of calculated transport efficiency for $5 \mu\text{m}$ travel by single atom collisional loss rate and single atom moving loss per neighboring site. The transport efficiency has been improved by a quick matching algorithm, faster SLM, turning off the imaging beam, and reducing the atom temperature. . . . .	24
4.3	Structure and procedure of GPU-accelerated phase calculation. . . . .	25
4.4	Computational time vs. the number of initial sites. The computational time of atom-matching to target sites using the brute-force, heuristic shortest move, and Hungarian algorithms, when the numbers of target sites, atoms and initial sites are given by $N : N_A : N_i = 1 : 2 : 4$ , respectively. Each errorbar represents the standard deviation [105]. . . . .	51
4.5	Visualization of move solutions from (a) the 7-by-7 initial array, by (b) shortest-move matching algorithm, and (c-d) Hungarian algorithm matching with $\alpha = 1$ and 2, respectively, to the central 3-by-3 target array. The orange dotted circles show overlapping of the paths and trespassing of atom sites. . . . .	58
4.6	(a) “Nearly trespassing path” and (b) “Relaying path” for a configuration with two atoms and two targets in a 1-by-L lattice. . . . .	59
4.7	Deterministic circular chain arrays. Several trial stages of vacancy filling with experimental parameter sequence and selected captured atom images are shown, each followed by atom number histogram in the final target array. . . . .	60
4.8	Single step loss for GS, Random Distribution, Superposition (SP), and theory. . . . .	61
4.9	Experimental examples of the formation of a 3-by-3 atom array from a partially filled 7-by-7 array using the Hungarian matching algorithm, where $P_s$ is the success probability of achieving a filled target lattice. Histograms of the atom number in the target lattice are shown below from a total of 250 events [105]. . . . .	62

4.10	Success rate comparison for shortest-move and Hungarian algorithm matching with various $\alpha$ values in the 7-by-7 lattice for the target 3-by-3 lattice in the central region. The circles and errorbars correspond to the experimental data and the dotted lines show the simulation results [105]. . . . .	63
4.11	Examples of defect-free atomic array formation: (a) a rectangular ring, (b) a triple X, and (c) the capital letters of the word “atom” with upper and lower images showing the initial and final configurations, respectively [105]. . . . .	63
5.1	Schematic diagram of Rydberg atom experiment setup. . . . .	65
5.2	Energy level diagram for excitation of the atoms. The atoms initially in $ g\rangle =  5S_{1/2}\rangle$ state are excited to $ r\rangle =  67S_{1/2}\rangle$ Rydberg state by non-resonant two-photon transition through $ 5P_{3/2}\rangle$ state. . . . .	66
5.3	Evidence of atom-atom Rydberg entanglement in our experiment. One, two, and three atoms were excited respectively, with all atoms in the blockade radius in each set. $ \uparrow\rangle =  r\rangle =  67S_{1/2}\rangle$ and $ \downarrow\rangle =  g\rangle =  5S_{1/2}\rangle$ . . . . .	68
5.4	The Power spectral density of frequency noise $S_\nu(f)$ from PDH error signal of the (a) 780 nm and (b) 480 nm laser obtained by a spectrum analyzer from PDH locking electronics. . . . .	71
5.5	(a) Geometry of $N=3$ atoms in linear configuration. (b) Scan of $R^2$ between simulation and experimental data of time-sampled projection measurement with two free parameters $\Delta S$ and $\Omega$ in simulation. The local maximum existed (red dashed circle). The curves in the points of other $\Delta S$ values (gray dashed circles) were also examined in (c) for comparison. (c) Experimental data (red circles) and simulation (red curves) with the best fitting parameters, for symmetric basis states, along with non-best fit simulation with difference in $\Delta S$ by $\pm 3.1$ dB. “0” and “1” indicate $ g\rangle$ and $ r\rangle$ states, respectively. . . . .	73
5.6	(a) Geometry of the atoms, (b) Scan of $R^2$ between simulation and experimental data, and (c) Experimental data (red circles) and simulation (red curves) with the best fitting parameters, for $N = 3$ atoms in triangle configuration . . . . .	74
5.7	(a) Geometry of the atoms, (b) Scan of $R^2$ between simulation and experimental data, and (c) Experimental data (red circles) and simulation (red curves) with the best fitting parameters, for $N = 4$ atoms in linear configuration. . . . .	75
5.8	(a) Geometry of the atoms, (b) Scan of $R^2$ between simulation and experimental data, and (c) Experimental data (red circles) and simulation (red curves) with the best fitting parameters, for $N = 4$ atoms in zig-zag configuration. . . . .	76
5.9	(a) Geometry of the atoms, (b) Scan of $R^2$ between simulation and experimental data, and (c) Experimental data (red circles) and simulation (red curves) with the best fitting parameters, for $N = 5$ atoms in linear configuration. . . . .	77

5.10	(a) Geometry of the atoms, (b) Scan of $R^2$ between simulation and experimental data, and (c) Experimental data (red circles) and simulation (red curves) with the best fitting parameters, for $N = 5$ atoms in zig-zag configuration. . . . .	78
5.11	(a) Schematic illustration of the experimental setup. (b) Timeline of the experimental procedure. . . . .	79
5.12	The computational complexity of computing N-atom dynamics. . . . .	85
5.13	The concept of dynamics mapping on ANN. . . . .	86
5.14	(a) The scheme of machine learning of multi-qubit dynamics. In the experimental setup, a physical multi-qubit system evolves and is measured in time series. The measured data is sent into the input layer of the neural network to determine the system parameters. The neural network is pre-trained with numerically generated data. (b) Result of learning in $N = 2$ qubit system. A set of data and the corresponding reconstructed dynamics from the output of the neural network. “0” indicates the probability in the state with zero excited atoms, and “1” indicates the average of the probabilities of the one-atom excited states. (c) Result for $N = 3$ qubit triangular system. . . . .	87

# Chapter 1. Introduction

A system of atoms has been one of the most important platforms for exploring quantum physics, standards such as atomic clocks, and quantum computation. Especially, an atomic qubit has been a promising candidate platform of a qubit thanks to its sharp linewidth, or the long coherence time, and the long history of controlling technique on atoms in atomic physics and optics fields. Recently, techniques for single atom manipulation, such as dipole trapping technique of single neutral atoms and atom-atom interaction via Rydberg blockade, have been of much interests for quantum information processing. Here, brief introductions to neutral atom qubits, single and two qubit operation, dipole traps for neutral atoms and experiments with Rydberg atoms are presented.

## 1.1 Neutral atom qubits

Trapped neutral atoms have intensively studied as physical implementation of qubits for quantum information processing, along with photons [1], trapped ions [2, 3, 4], and superconductors [5, 6]. As a qubit, a neutral atom, as well as a trapped ion, has a clear advantage that it has a well-defined set of quantum states with sharp linewidth, which makes it a well-defined qubit with long coherence time. Besides, the history of controlling atoms, or optical spectroscopy, with many kinds of modulated lights makes it more accessible. Also, neutral atom qubits have high scalability [7], in which they are more favorable than trapped ions, as their trap geometries are easily created by a range of light modulation techniques [8]. On the other hand, achieving high gate fidelity on neutral atoms for quantum computing still remains as a challenge yet.

Any two states can be used for a qubit in neutral atoms among many states that have variety of physical characteristics such as linewidth and dipole moments. For long coherence time of qubits, any two hyperfine states in the ground state are favored as a qubit [9]. The clock transition state pair in alkali atoms, for example, is a good choice as a qubit for the purpose. The states with high principal quantum numbers, called Rydberg states, on the other hand, can be exploited for atom-atom interaction since they can create a considerable overlap among atoms which results in dipole-dipole interaction. To this day, many valuable suggestions and demonstrations have been known for quantum computing [10, 11, 12] and quantum simulation [13, 14, 15].

## 1.2 Single- and two-qubit operation

Study of single-qubit operation of trapped neutral atoms basically shares its principle with the traditional field of optical spectroscopy and quantum control. Well-known Rabi oscillation [16, 17, 18] is one of the representative ways for longitudinal transition of a qubit in the Bloch sphere, and also phase evolution, or z-rotation in the Bloch sphere can be realized by phenomena such as ac stark shift with cw lasers and ultrafast pulses [19]. Implementing a set of universal quantum gates, which is any set of gates where an unitary operation can be reduced to another operation consisting of elements of the set only, is one of the main interests in the field. A typical set of universal gates includes a few single-qubit gates and a two-qubit gate such as a C-NOT gate.

Researches on control of trapped neutral atoms which focus more on the basis of quantum information have been presented recently, in depth. Single-qubit operation conducted on hyperfine qubits has been reported with microwaves [20, 21], and two-frequency Raman lights [22, 23]. Also, randomized benchmarking has been recently performed on large atom arrays for Clifford gates giving high fidelity [7, 24]. Two-qubit gates, in forms of quantum entanglement [9] and a C-NOT gate [25], have been demonstrated via a way of atom-atom interaction called the Rydberg blockade, in recent days. The Rydberg blockade will be discussed in Chapter 5 in detail.

## 1.3 Dipole traps for neutral atoms

Suggestions and realizations of trapping neutral atoms in dipole traps have inspired the use of neutral atoms as qubits in quantum computation and investigating them as physical quanta. The physical model is as followings [26].

An atom in a laser light field has a dipole moment  $\mathbf{p}$  induced by an electric field  $\mathbf{E}$  of the field. Then the atom is involved in an interaction potential given by,

$$U = -\frac{1}{2}\langle\mathbf{p}\mathbf{E}\rangle = -\frac{1}{2\epsilon_0 c}Re(\alpha)I. \quad (1.1)$$

where  $\alpha$  is the complex polarizability of the atom and  $I$  is the intensity of the laser field. It is known that when we apply the driven oscillator model for atom polarizability, we get an expression of the dipole trap potential,

$$U(\mathbf{r}) = -\frac{3\pi c^2}{2\omega^3}\left(\frac{\Gamma}{\omega_0 + \omega} + \frac{\Gamma}{\omega_0 - \omega}\right)I(\mathbf{r}) \quad (1.2)$$

where  $\omega$  is the laser field frequency,  $\omega_0$  is the frequency of the involved atom transition levels, and  $\Gamma$  is the linewidth of the corresponding transition.

After demonstration and analysis of single neutral atom trapping [27], the technique was further developed, for example, employing beam shaping methods to prepare neutral atom arrays in arbitrary geometries [8], along with single atom addressing techniques for quantum

gates [7]. Meanwhile, efforts to enhance the loading probability to a dipole trap has been also made [28].

## 1.4 Rydberg atoms and Rydberg blockade

Recently, Rydberg atoms, atoms in high principle quantum states, have been intensively studied as a kind of quantum computation platform, due to their strong interactions suitable for gate operations, sharp linewidth and long coherence time for preservation of quantum states [29]. Atom-atom interaction can be experimentally realized by a phenomenon known as the Rydberg blockade, which is basically the dipole-dipole blockade between Rydberg-state-populated atoms. Two neighboring atoms with sufficient polarizability in the Rydberg state have their energy shifted in the simultaneous excitation state. The energy shift can be calculated by perturbation theory, between a two-atom state  $|rr\rangle$  and its neighboring state  $|r'r''\rangle$ , with a Hamiltonian,

$$H = \begin{pmatrix} 0 & \frac{C}{R^3} \\ \frac{C}{R^3} & \delta_F \end{pmatrix}, \quad (1.3)$$

where  $C$  is a constant proportional to the dipole moments of the atoms and  $\delta_F$  is the energy difference between the two states, called the Förster defect [30].  $R$  is the distance between the two atoms. The eigenvalues of the Hamiltonian are,

$$V_{int}(R) = \frac{\delta_F}{2} \pm \frac{1}{2} \sqrt{\delta_F^2 + \left(\frac{2C}{R^3}\right)^2}. \quad (1.4)$$

In van der Waals interaction case, where  $\frac{C}{R^3} \ll \delta_F$ , the Taylor expansion of the square root term, in terms of  $\frac{2C}{R^3}$  yields,

$$V_{int}(R) = \frac{\delta_F}{2} \pm \frac{1}{2} \left[ \delta_F + \frac{2C^2}{\delta_F R^6} + \dots \right] \quad (1.5)$$

Thus, the energy shift is, without the unperturbed energies,

$$\Delta V_{int}(R) = \pm \frac{C^2}{\delta_F R^6}, \quad (1.6)$$

which will be denoted simply as  $C_6/R^6$ . For instance, for a Rydberg state  $|67S_{1/2}\rangle$  and an interatomic distance of  $6 \mu m$ , the energy (in frequency unit) of the Rydberg interaction is about  $2\pi \times 10$  MHz. This energy shift can successfully prevent the simultaneous excitation of the both atoms in the condition that the single atom Rabi frequency is sufficiently smaller than the energy shift, say, about 1 MHz, which makes the Rydberg blockade an useful way of atom-atom interaction in quantum information processing.

## Chapter 2. Magneto-optical traps for cold atom loading

Since its development [31, 32, 33, 34, 35], the magneto-optical trap has been a standard experimental apparatus for various quantum experiments with cold and dense neutral atom ensembles, including Bose-Einstein condensates and quantum control of optically dipole-trapped single neutral atoms.

A magneto-optical trap consists of a vacuum chamber, an anti-Helmholtz coil, and lasers for cooling. The atoms are shot by cooling lasers from six directions (Fig. 2.1). The cooling lasers are red-detuned so that only those atoms moving towards the direction where a beam comes from are Doppler-cooled. However, in order to gather and trap atoms at the center of the region where the laser beams intersect, the Doppler cooling has to be parity-dependent. This is achieved by a combination of quadrupole magnetic field generated by an anti-Helmholtz coil and oppositely-polarized counter-propagating cooling lasers in each of the three dimensions. The gradient of the magnetic field at the center of the quadrupole geometry causes a gradient of energy shift for Zeeman magnetic states of an atom. The direction of the energy shift depends on the sign of the magnetic quantum number of the state. For example, in a geometry of quadrupole magnetic field where the direction of the field on the  $z$ -axis is toward the center ( $z = 0$ ), in the  $z > 0$  side,  $\sigma^-$ -polarized lights coming to the center become resonant for  $m = -1$  state among the excited levels, while  $\sigma^+$ -polarized lights become resonant for  $m = 1$  state in the opposite side ( $z < 0$ ). By this, atoms moving outwards lose momentum by resonant photons, causing cooling and trapping of atoms. Typically, the temperatures of the cold atoms are  $< 100 \mu\text{K}$ , and the density of an atom cloud is  $\sim 10^{10} / \text{cm}^3$ .

### 2.1 Vacuum

For isolation of trapped cold atoms, as low as  $< 100 \mu\text{K}$ , from the room temperature air, atoms were trapped in a vacuum chamber. To secure optical windows for further atom control such as single atom transport and Rydberg state control, as well as for magneto-optical trapping, we used a custom-made glass-bodied vacuum chamber made with borosilicate (Precision Glassblowing). The design of the chamber is shown in Fig. 2.2. Four-fold surrounding glass walls ( $100 \text{ mm} \times 40 \text{ mm}$ ,  $3.5 \text{ mm}$  thick) of the middle part and both-end 2.75-inch flanges in the perpendicular direction provided sufficient space for three-dimensional magneto-optical trapping and additional optical controls. The geometry of the middle part was to make single atom trapping and imaging possible with a high-NA lens. The  $^{87}\text{Rb}$  atom dispenser (SAES Getters) attached between the middle part and an end flange was heated by electrical current (a few amperes) to supply background atoms for atom trapping. The window walls were AR-coated in the wavelength range of  $400 \text{ nm} - 800 \text{ nm}$  for trapping and control of atoms.

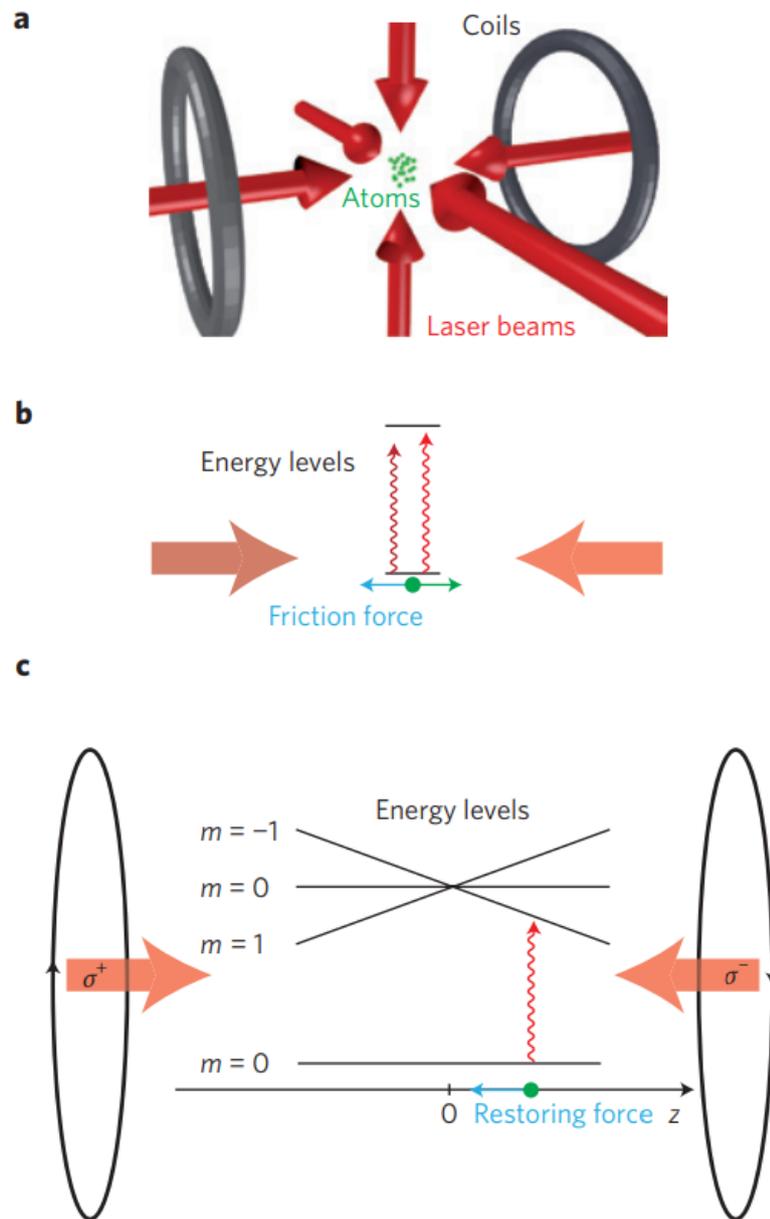


Figure 2.1: Principle of magneto-optical traps. The schematic picture is adopted from Ref. [31]

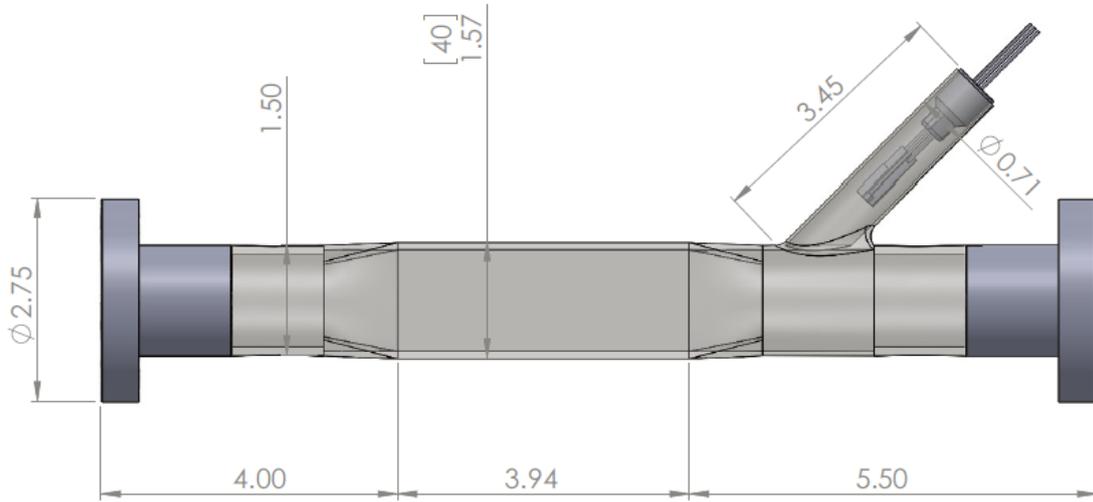


Figure 2.2: Design of the vacuum chamber.

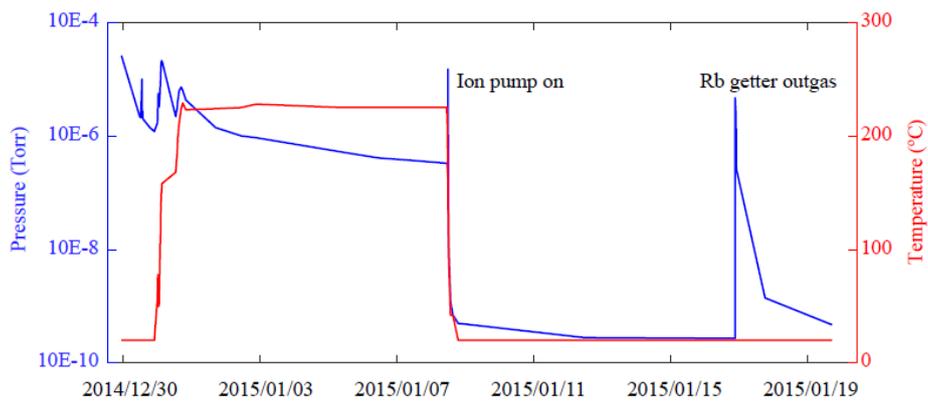


Figure 2.3: Pressure inside the vacuum chamber while baking

For achieving ultra-high vacuum, the pressure inside the vacuum was decreased through three stages. In the first stage, the air in the chamber was mechanically pumped out by a turbomolecular pump (Varian V-70). With the pump, the pressure went down to under  $10^{-5}$  Torr within hours. After then, the vacuum chamber entered a bake-out stage for removal of water vapor and other contaminants and acceleration of artificial outgassing which enabled a higher vacuum. The chamber was first wound by some heating tapes. Then, it was covered by aluminum foils for even distribution and gradual change of heats. The temperature of the vacuum chamber was increased and reached a temperature slightly less than  $250\text{ }^{\circ}\text{C}$  avoiding the chamber damage threshold. This stage took about a week. The temperature and pressure plot during the whole vacuum-generating period is shown in Fig. 2.3. The pressure was decreased a bit further after bake-out stage. Finally, ion pump was used to reach  $10^{-10}$  Torr order of pressure. Pressure of  $10^{-9}$  Torr was obtained in a few hours and it converged to  $3 \times 10^{-10}$  Torr afterwards in days. The accidental diverging of pressure in the plot was due to outgassing of Rubidium dispenser and corresponding additional bake-out of it. The pressure entered a stable phase after treatments.

## 2.2 Magnetic field gradient by anti-Helmholtz coil

For atom cooling and trapping, parity-dependent absorption of photons by atoms is required which could be realized by a magnetic field gradient at the trapping region along with oppositely circular-polarized counter-propagating cooling beams. The gradient of magnetic field is formed by an anti-Helmholtz coil connected to a current supply. The magnitude of the magnetic field is controlled by the amount of current. Between the coil and the current supply there is a MOSFET switch to turn on or off the magnetic field gradient in the procedure of single atom trapping and control.

## 2.3 Cooling laser

The cooling laser for the magneto-optical trap is locked to the frequency red-detuned to the  $|5S_{1/2}, F = 2\rangle$  state to  $|5P_{3/2}, F' = 3\rangle$  state, which is around 780 nm wavelength. The amount of the detuning is around 3 times the decay rate of the transition which might be optimized for the MOT density. For three-dimensional cooling and trapping the laser is splitted into three ways and delivered through optical fibers to shine the trapped atoms in the vacuum chamber from one vertical axis and two horizontal axis. Each beam is reflected after passing through the chamber, in configuration of two-way Doppler cooling of the atoms along with the original passing beam. After each output of fibers are half-wave plate - polarizer - quarter-wave plate to ensure the cooling laser has circular polarization to react to the magnetic field generated by the anti-Helmholtz coil for cooling. A reflected beam passes a quarter-wave plate twice to reverse the polarization ( $\sigma^+ \leftrightarrow \sigma^-$ ). The power of each beam is around 2.6 mW to exceed the

saturation intensity of an atom.

## 2.4 Repump laser

The repump laser is locked to the resonance of  $|5S_{1/2}, F = 1\rangle$  to  $|5P_{3/2}, F' = 2\rangle$  transition, in order to pump the leaked population of  $|5S_{1/2}, F = 1\rangle$  back to the cooling cyclic transition. For better efficiency of optical pumping the in later Rydberg atom experiment, the linear polarization parallel to the quantization axis (B-field axis, or 'z'-axis), exerted from the direction perpendicular to the axis, is chosen, rather than coming along the same path as the cooling beam, although the polarization does not heavily affect the performance of cooling and trapping. The power of the beam is around 3 mW and the cooling performance is not much sensitive to the alignment.

## 2.5 Earth's magnetic field compensation

The Earth's magnetic field has non-negligible effect on the magnetic gradient geometry, thus affecting the efficiency of atom cooling and trapping. The field is compensated by Helmholtz coils wound around the cage outside the chamber. The coils create uniform B-field in each of three orthogonal axes. Enamelled wires are connected to current supplies to control the magnitude B-fields in three axes ( $\sqrt{2}\mu_0 I/\pi a$  for current  $I$  and coil dimension  $a$ ). In each axis, a pair of 20-loop coils creates 0.8 G/A. The earth B-field of several hundred miliGauss in magnitude can be compensated by this configuration of coils with current less than or around an Ampere. Tapes are attached at edges of the cage under wires to avoid peeling of the wires which causes electric leakage and short circuit.

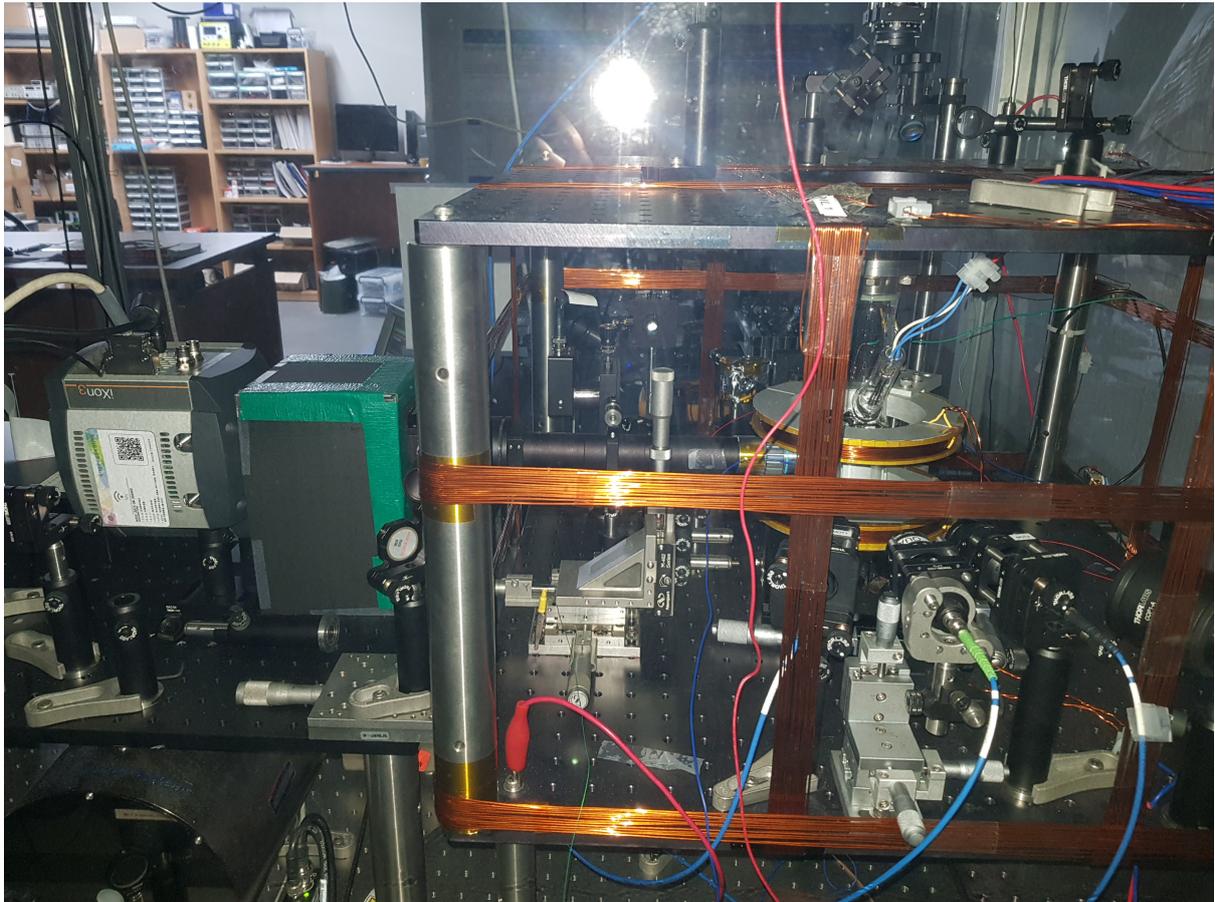


Figure 2.4: Photo of the vacuum chamber and surrounding optics.

## Chapter 3. Transport of single atoms in three-dimensional space

Optical dipole trapping is a simple and powerful technique for holding and steering atoms in space [36, 37, 38, 39, 40, 41, 42]. This technique, recently developed to far off-resonant optical trapping (FORT), utilizes the electric dipole interaction force exerted by light to manipulate the external degrees of freedom of quantum objects. From a pre-cooled atom ensemble, focused laser beams can capture and isolate single atoms without inducing optical transitions, so their internal states are well preserved in the electronic ground state up to several seconds, which makes the optically trapped single atoms a promising candidate for storing and processing quantum information [43]. Currently there is a strong interest in using the FORT in engineering scalable quantum platforms [8], [44]-[52], because the manipulation of  $N$  single atoms in a synthetic structure of a few  $\mu\text{m}$  size is a crucial necessity for the study of quantum computation, quantum simulation, and quantum many-body physics [53]-[56].

Optical dipole microtraps and optical lattices [56] are the well-known tools for atom arrays. In the context of the manipulation and control of individual atoms in an array, optical microtraps are considered to be a versatile tool, having many control parameters. The optical microtraps have been achieved by various methods, including micro lens arrays [57], diffractive optical elements such as Dammann grating [58], spatial light modulators (SLM) [8], [45]-[46], optical standing waves [50], and dynamic light deflectors [45]. With these methods, adiabatic transport of atoms in one and two dimensions [47]-[48], atom sorting with a cross junction [50], collisional blockade mechanism [52], controlled collisions for near-deterministic atom loading [59], atom array rotations [46], and single-qubit gate arrays [53] have been demonstrated. These impressive achievements are currently being geared towards a deterministically-loaded high-dimensional arbitrary architecture of  $N$  single atoms in which the the internal and external degrees of freedom of the atoms are freely controllable.

Holographic methods of using a programmable SLM in the Fourier domain have been a work horse in the construction and manipulation of various forms of two-dimensional (2D) microtrap arrays [8]. The SLM phase pattern generation for  $N_{tr} \sim 2N$  optical microtraps is often performed with iterative Fourier transform algorithms (IFTA). However, because of the frame-to-frame intensity flickering [60], dynamic and simultaneous control of  $N$  single atoms with a series of IFTA-generated patterns has remained difficult. Recently we devised a simple method for flicker-free atom controls with optical microtraps [61]. In this section, we extend this method further to achieve 3D rearrangements of  $N$  single atoms. We experimentally demonstrate holographic methods for various 2D and 3D transformations of single atoms in an array.

In the remaining sections, we first describe in Sec. 2.2 the principle of the dynamic and simultaneous displacement of  $N_{tr}$  optical microtraps in 3D. The transverse and axial displace-

## G Plan Apo 50X

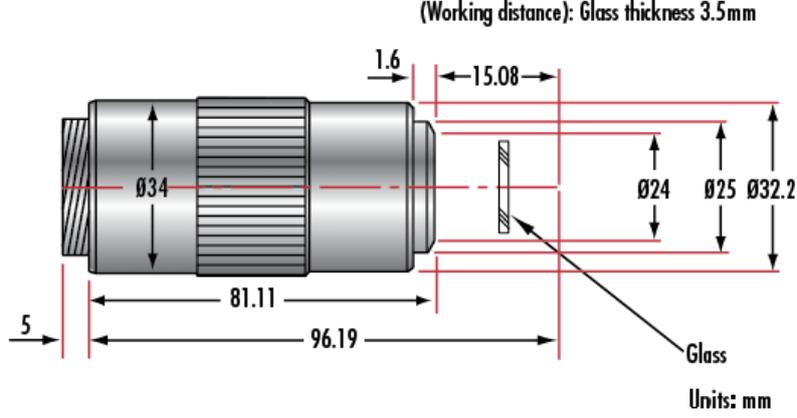


Figure 3.1: Technical image of the objective lens.

ments are programmed with linear and quadratic phase gradients of the SLM pixels, respectively. In Sec. 2.3 we explain the experimental setup and the control and imaging procedure. The experimental demonstrations of various 2D and 3D atom-array rearrangements are presented in Sec. 2.4, before the conclusion in Sec. 2.5.

### 3.1 Experimental requirement

Trapping of single atoms starts with trapping of a cold atom cloud which is described in Sec. 2. Additional experimental settings required for single atom trapping are explained here.

#### 3.1.1 Single atom trapping with dipole traps

The light source for the FORT was a continuous-wave Gaussian beam from a Ti:sapphire laser (SolsTiS from M-Squared Lasers) tuned at  $\lambda = 820$  nm. The beam was programmed and reflected by an LCOS-SLM (Liquid Crystal on Silicon-Spatial Light Modulator, phase-only, Holoeye PLUTO,  $1920 \times 1080$  pixels, 60 Hz frame rate) to generate  $N$  sub-beams.  $N$  optical microtraps were formed at the focal plane of an objective lens (Mitutoyo G Plan,  $50\times$ ,  $NA = 0.5$ , working distance 13.89 mm,  $f = 4$  mm, infinity-corrected, 3.5-mm-glass compensation, technical drawing in Fig. 3.1), where the laser power was 0.55 W after the SLM and the beam diameter entering the objective lens was  $1/e^2 = 4$  mm, limited by the aperture size. Each sub-beam from the SLM was delivered onto the objective exit pupil by a pair of relay lenses with the same focal length of  $f = 200$  mm in a  $4-f$  geometry. The focal diameter of the sub-beam in the vacuum chamber was  $2w_o = 2.28 \mu\text{m}$  estimated by a separate beam profile measurement (not shown).

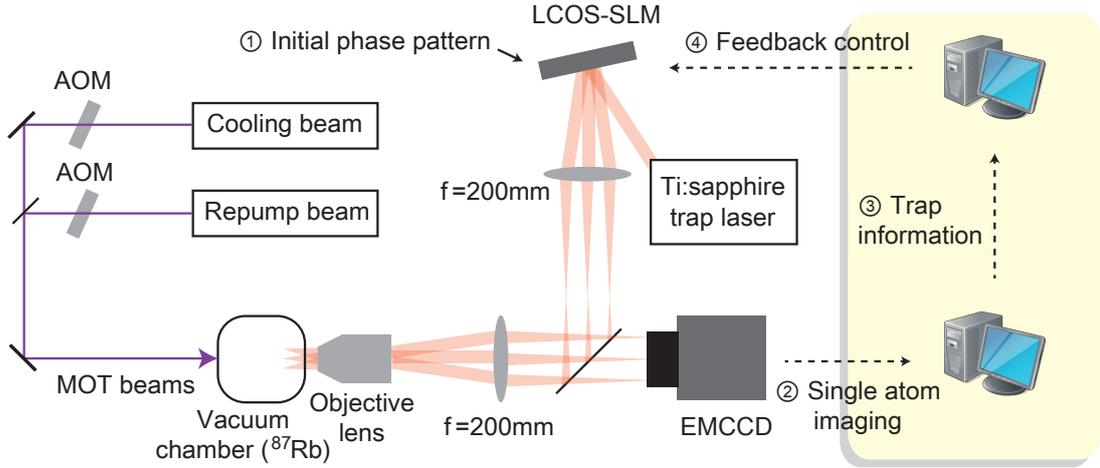


Figure 3.2: Schematic illustration of the experimental setup. Optical microtraps were programmed with the LCOS-SLM to capture single atoms from the pre-cooled  $^{87}\text{Rb}$  ensemble in the MOT. The information of the trapped single-atom configuration obtained with the EMCCD was sent back to the SLM computer for the feedback control.

### 3.1.2 Imaging

To verify transport of single atoms, one must be able to observe single atoms. This can be done by obtaining images of fluorescence from the single atoms which are excited by cooling laser in a detuned frequency with an electron-multiplying charge-coupled device (EMCCD, Andor iXon3). The cooling laser is red-shifted by about 30 MHz from the frequency when MOT operates, in order to avoid heating of the atoms by the cooling (imaging) laser (the procedure is described in Sec. 3.2). The fluorescent light is collected by the same objective lens used for dipole trap generation.

### 3.1.3 Computer control

The experiment requires real-time control of devices, a whole period in a second, concerning that typical trap lifetime is in order of tens of seconds. For this, the setup is controlled by a programmed procedure. The software part of the computer control was made and run by Lab-view. The software program controlled the devices through the hardware part of the computer control system, NI-PCIe series and NI-PCI series. These devices delivered the programmed procedure from the software (digital outputs and analog outputs) to the experimental devices in the setup to control whole sequence from MOT generation through single atom trapping to imaging of the single atoms.

## 3.2 Procedure

### 3.2.1 MOT loading

Before trapping of single atom, a cold atom cloud was prepared as a source for single atoms. In our experiment, we used cold rubidium atoms ( $^{87}\text{Rb}$ ) in a magneto-optical trap (MOT) and single atoms were captured and controlled with optical microtraps (see Fig. 3.2). The atoms were first pre-cooled in the MOT with the D2 line ( $F=2 \rightarrow F'=3$ ) with a density of  $10^{10}$  atoms/cm<sup>3</sup> in the vacuum chamber with a pressure of  $3.0 \times 10^{-10}$  Torr. A detailed explanation about a magneto-optical trap is provided in Sec. 2

### 3.2.2 Single atom loading

After the atoms were loaded in the MOT, the microtrap beams were turned on and the MOT cooling beam was red-shifted by 45 MHz for polarization gradient cooling (PGC) and atom imaging at the same time. After 200 ms of optical molasses, the magnetic field of the MOT was turned off to keep background atoms from further gathering. The PGC beam was temporarily turned off during axial transport of atoms in order to reduce heating. The trap depth of the microtraps was  $U = (3\pi c^2 \Gamma / 2\omega_o^3 \Delta) I > 1.4$  mK, where  $\Gamma / (2\pi) = 5.75$  MHz was the natural line width of the  $^{87}\text{Rb}$  D1 line,  $\omega_o$  the transition frequency,  $\Delta / (2\pi) = -1.16 \times 10^{13}$  Hz the detuning, and  $I = 2 \times 10^9$  W/m<sup>2</sup> the intensity of the the beam. The FORT-induced heating rate was  $7 \mu\text{K/s}$  assuring that the atoms remained in the cooled temperature. The trap frequency was  $(2/w_0) \sqrt{U/m} \sim (2\pi)70$  kHz, where  $m$  is the mass of an atom. The fluorescence of the atoms was detected with the EMCCD. The EMCCD detected about 200 photons per atom during an exposure time of 50 ms. Note that the PGC scattering rate, with  $I = 27$  mW/cm<sup>2</sup> and  $\Delta \approx -17\Gamma$ , was  $2.9 \times 10^5$  s<sup>-1</sup> and about 6.7% of emitted photons were collected with the objective lens of  $NA = 0.5$ . Our experiment was performed in the collisional blockade regime [52] with a single-atom trapping probability of 50.9%, measured by fluorescence histogram. The trap lifetime was  $\tau > 13$  s, given from the decay of the remaining atom probability (when the trap was stationary). The decay process was dominated by background gas collisions [1].

Dynamic manipulation of the single atoms was achieved by applying a sequence of phase patterns to the SLM. We used an active area of  $800 \times 800$  pixels around the SLM center to take full advantage of the hardware frame update rate of 60 fps (frames per second), because the displaying rate from the computer to the SLM was relatively slow to use the full frame. Also, we used two personal computers (PCs) to accelerate the whole operation. The phase patterns were first diagnosed with the first computer and transferred to the SLM through a display port (DVI or HDMI). In “adaptive” mode, where the set of the trap trajectories depended on the initial configuration of the atoms, the EMCCD operated by the second computer took the image of the initial atom configuration and sent the information to the first PC (see Fig. 3.2). The PC then loaded a 30-frame movie depending upon the initial state. For this, we prepared a look-up

library having all possible trajectories between the initial and final atom configurations. The computer memory required for the look-up library in our experiment was 10 GB (gigabytes). Because each trap site was occupied probabilistically, there were total  $2^{N_{tr}}$  initial configurations for  $N_{tr}$  trap sites. The memory for a single frame of  $800 \times 800$  pixels with an 8-bit gray level was 640 KB (kilobytes) and there were 30 frames for each movie. So, with  $N_{tr} = 9$  sites,  $2^9 \times 30 \times 40$  KB was about 10 GB. During the 30-frame movie was being played on the SLM, the EMCCD captured sequential images of the atom array. The number of the movie frames was limited by the lifetime  $\tau$  of the atoms.

### 3.3 SLM phase calculation

The flicker-free frame-to-frame evolution of microtraps in 3D is implemented with phase patterns programmed in the Fourier domain. The transverse displacement  $(\Delta x, \Delta y)$  is implemented by a linear phase gradient [61] and the axial displacement  $\Delta z$  by a Fresnel lens phase [62]. The linear phase  $e^{i(k_X X + k_Y Y)}$ , where  $(X, Y)$  is the position in the Fourier domain and  $(k_X, k_Y)$  the transverse wave vector, makes the focal spot of a beam shifted by  $(\Delta x, \Delta y) = (k_X f_o, k_Y f_o)/k$  from the optic axis (the  $z$  axis), where  $k$  is the wavenumber of the beam, when the beam is focused by an objective lens with focal length  $f_o$  [see Fig. 3.3(a)]. Also, when the beam passes through a Fresnel lens with focal length  $f_F$  and then focused by the same  $f_o$  lens which is  $f_o$  apart from the Fresnel lens, after relaying 4-f geometry [see Fig. 3.3(b)], the axial displacement is given by  $\Delta z = -f_o^2/f_F$ , where the thin lens formula  $1/f_o = 1/(-f_F + f_o) + 1/(f_o + \Delta z)$  is used. Therefore, the 3D displacement of the focal spot

$$(\Delta x, \Delta y, \Delta z) = \left( \frac{k_X f_o}{k}, \frac{k_Y f_o}{k}, -\frac{f_o^2}{f_F} \right) \quad (3.1)$$

is implemented with the phase pattern  $\phi(X, Y)$  on the SLM given by

$$\phi(X, Y) = \text{mod} \left( k_X X + k_Y Y + \frac{k}{2f_F} (X^2 + Y^2) + \pi, 2\pi \right), \quad (3.2)$$

where  $2\pi$  is the phase modulation depth of the SLM and the constant  $\pi$  phase is for less phase jumping at the center.

Scaling the number of focal spots, or producing an array of optical microtraps, is achieved by generating more than one such spots simultaneously [see Fig. 3.3(c)]. The simplest way is to divide the SLM plane spatially and implement phase patterns onto each region; however, this scheme changes the profile of the focal spot because it loses some spatial frequencies. Alternatively, we can mix the fundamental phase patterns into a single phase pattern; each of them is randomly distributed over the entire SLM space with equal appearance. To mix  $N_{tr}$  phase patterns  $\phi_m(i, j)$ , where  $m \in [1, N_{tr}]$  and  $i, j$  are the SLM pixel indexes, into a combined phase pattern  $\phi_{\text{mixed}}(i, j)$ , we use an addressing matrix  $S(i, j)$  of random integers between 1 and  $N_{tr}$  with an equal probability. The mixed phase pattern  $\phi_{\text{mixed}}(i, j)$  is then given by

$$\phi_{\text{mixed}}(i, j) = \phi_{S(i, j)}(i, j). \quad (3.3)$$

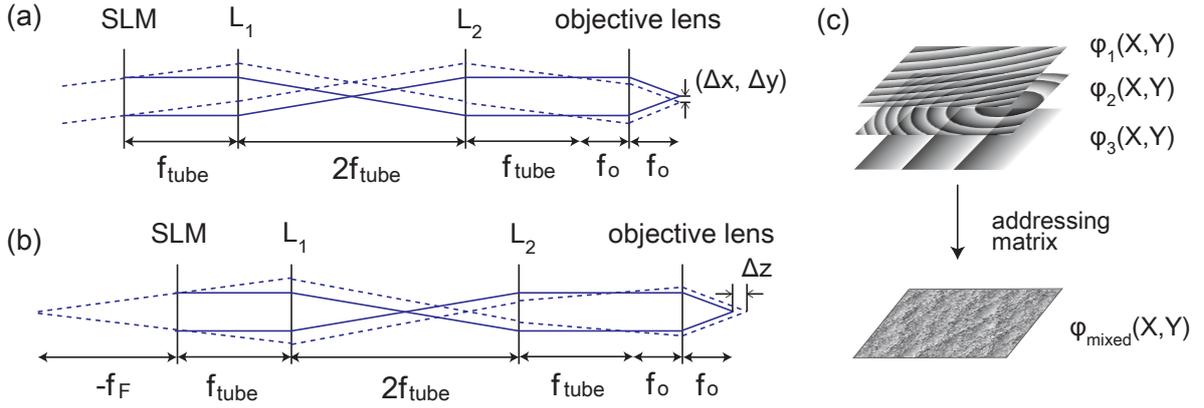


Figure 3.3: (a) Transverse displacement of focal spots. (b) Axial displacement. (c) Phase pattern synthesis.

In this way, each focal spot can be controlled independently while the spatial profile of the original spot is well preserved. The independent control of each focal spot guarantees the flicker-free frame-to-frame evolution of  $N_{tr}$  focal spots [61]. Note that a periodic addressing, the way in which  $m$  is chosen in a periodic manner, rather than randomly, also works, but an unwanted interference could occur.

## 3.4 Results

### 3.4.1 Transport in 2D

Figure 3.4 shows selected demonstrations of our creation and reconfiguration of 2D single-atom arrays. In every experiment, a phase movie moved  $N_{tr}$  microtraps along each pre-defined trajectory. The phase movie frame rate was 20~60 fps and the time interval between the captured atom images was 60 ms for the overall demonstration. The nearest neighbor spacing in arrays was  $d = 4.5 \mu\text{m}$  for every scheme of operation. In the first set of experiments, (a) rotation, (b) 2D vacancy filling, and (c) worm running in Fig. 3.4, the atoms were guided under a fixed scenario of placing and rearranging the  $N_{tr}$  optical microtraps. The number of trapped atoms was smaller than the number of optical microtraps, so not all optical microtraps guided atoms. So, in the second set of experiments, (d) Fall to the right: case 1, (e) Fall to the right: case 2, and (f) Fall to the right: case 3, an appropriate scenario to move the atoms was chosen from the look-up library, according to the initial configuration of the trapped atoms. Then, the microtraps occupied by atoms were guided along the chosen trajectory, while the unoccupied ones were dragged outward.

(a) *Array rotation*: Rotation of a 3-by-3 single atom array as a whole is presented as a simple operation on arrays. Six atoms were initially trapped out of nine optical microtrap sites and the trapping of all atoms was maintained until the end of the  $150^\circ$  rotation. The outermost atoms traveled a distance of  $14.8 \mu\text{m}$  during the 25 operational frames which spanned 1.2

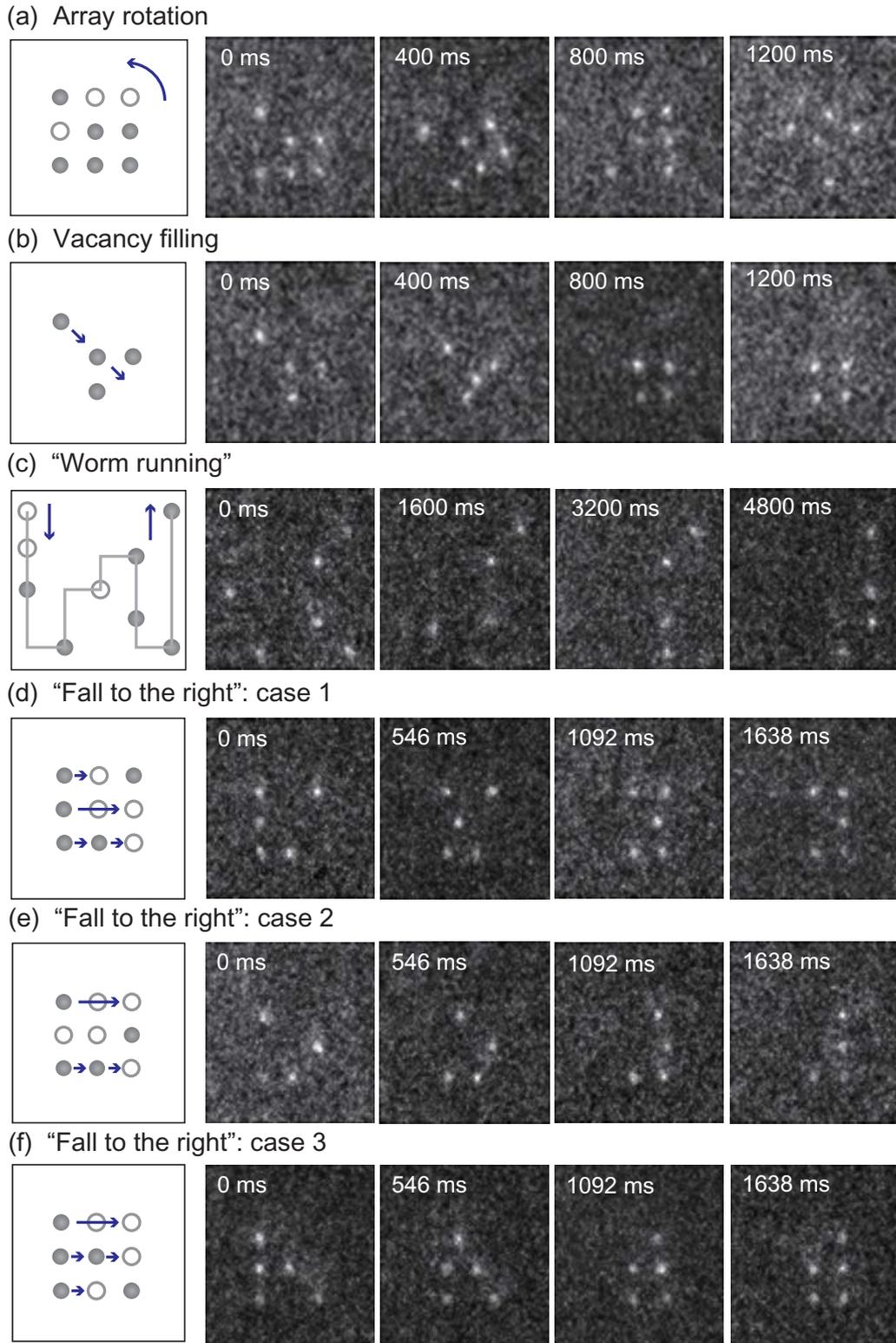


Figure 3.4: Selected results of demonstration. (a) Rotation of a 3-by-3 array as a collective control. (b) 2D vacancy filling and (c) 'Worm running' as individual atom controls. (d-f) 'Rightward alignment' as feedback controls of atom arrays. The leftmost column presents the schematic diagram of each operation scenario. In each column, the initial and in-between photos are followed by the final photos [36].

seconds. The frame-to-frame step rotation angle was  $6^\circ$ , so the step distance for the outermost atoms was  $0.67 \mu\text{m}$ . The atoms were individually controlled in this demonstration, but rotating the phase pattern itself also worked well in our trial demonstration. Note that rotating the phase pattern rotates the focal plane profile on the axis of zeroth order diffraction beam of the SLM.

(b) *Vacancy filling*: A more explicit evidence of an individual atom control is demonstrated, which also could be a candidate for the vacancy filling scheme in atom arrays. Initially four atoms were captured and the atoms in diagonal sites proceeded to the next diagonal sites while the rest of the atoms stayed in their positions, ending in a complete 2-by-2 single-atom array as a whole. Despite the simplicity of the operation, it was impossible to achieve the same performance by frames generated by Gerchberg-Saxton algorithm [8], one of the widely used IFTAs, losing all the atoms only after a few frames. The traveling distance of the atoms in the diagonal sites was equal to the next nearest neighbor spacing,  $6.3 \mu\text{m}$ , so the step distance in this 23-frame operation was  $0.30 \mu\text{m}$ .

(c) *Worm running*: For another demonstration, which may be called *snake crawling*, nine atoms were initially trapped at arbitrary chosen positions and started to follow through a designated path in a line while being *pushed* by their precedent atoms (including vacancies). This scenario clearly shows a full degree of freedom in controlling the positions of individual atoms, where some atoms moved in horizontal directions and some moved in vertical directions, while the other were stationary, in a simultaneous manner. The triggering atom travels by  $45 \mu\text{m}$  during 225 operational frames in 4.8 seconds. Some atoms were lost during the operation either by background gas collision or by moving loss.

(d)-(f) *Fall to the right - adaptive operation*: An initial 3-by-3 trap array, as in (a), trapped atoms with a probability of around 0.5 for each trap site. Since there were  $N_{tr} = 9$  trap sites in the array, there were  $2^9$  cases of initial trap conditions. The scenario was to detect the initial positions of the atoms and move them to the right to fill the array from the left, as if they were under a gravity directing to the right. The phase pattern movie for every initial case of condition was retrieved from the look-up library. Demonstrations for three different initial condition cases are shown in (d)-(f). The feedback control, or adaptive control, worked well albeit the stationary and moving loss of the atoms. The longest travel possible was twice the nearest neighbor spacing,  $2d = 9 \mu\text{m}$ , with a step distance of  $0.30 \mu\text{m}$  in the 30 operational frames.

### 3.4.2 Transport in 3D

Finally, we present the proof-of-principle demonstration of the 3D transportation of single atoms. Fig. 3.5(a) shows a set of layered 2D images of a 3D atom array to check the validity of our 3D transport scheme. The 3D atom array consisted of two layers of total eight atoms. The first layer had four atoms in the square configuration and the second had also four atoms but in the diamond configuration. The layers were separated by  $1 \mu\text{m}$  in the axial direction. The

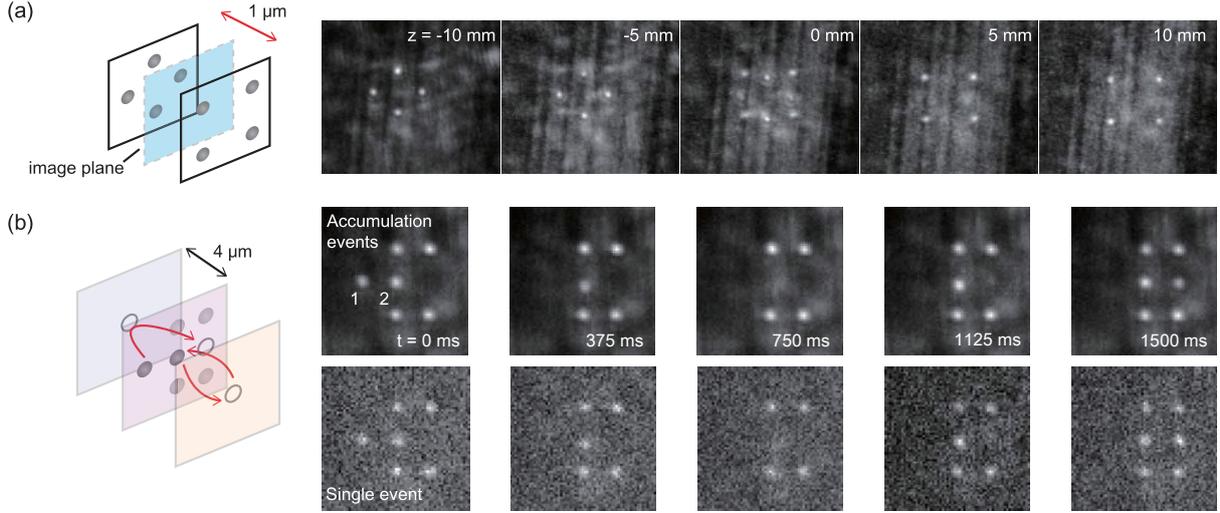


Figure 3.5: (a) Trapping and imaging of a 3D single-atom array, where the image plane was shifted by translating the EMCCD. Each image corresponded to the EMCCD position at  $z = -10, -5, 0, 5, 10$  mm and was accumulated from 100 single-event images that were captured right after the atoms were trapped in each acquisition. (b) Individual transport of single atoms in 3D, where each demonstration spanned 90 SLM frames and each step of atom moving was an equal division of the entire path. 370-time accumulated sequential images and selected single-event sequential images are displayed. The lattice constant of the 2D array was  $d = 4.5 \mu\text{m}$  and the depth of the axial travels for the both atoms were  $4 \mu\text{m}$  [36].

result in Fig. 3.5(a) shows the  $z$ -scan imaging of the 3D atom array conducted by translating the EMCCD in the axial direction, which confirms the creation of the 3D atom array as designed. Figure 3.5(b) shows a trial demonstration scheme for the 3D individual transport of single-atoms along with the captured images during the demonstration. The trial atoms moved along the designated paths out of the image plane while the other atoms stayed in position. The leftmost atom (marked as “1”) was programmed to bypass the neighboring atom (marked as “2”) in the axial direction to fill the vacancy in the right column. At the same time, the atom 2 traveled in the opposite axial direction. The rest atoms remained in the position during the operation. Atom 1 followed an elliptical path with the given dimensions. The SLM operated at 60 fps.

### 3.4.3 Efficiency and loss analysis

Even using “flicker-free” method of phase generation, there exists certain amount of flicker occurring due to finite range of phase modulation, or phase stroke, which inevitably causes phase jump. For example, when the phase of a pixel changes from  $1.99\pi$  to  $2.01\pi$ , and if the phase stroke is  $[0, 2\pi]$ , the phase has to be actually change from  $1.99\pi$  to  $0.01\pi$ . This occurs in a portion of the entire SLM area and ruins the overall frame-to-frame continuity, which causes some flicker. Fig. 3.6 (a) graphically shows the phase jump when the phase pattern changes from  $k_1x$  (blue) to  $k_2x$  (red). The gray shaded area indicates the part of normal phase change while

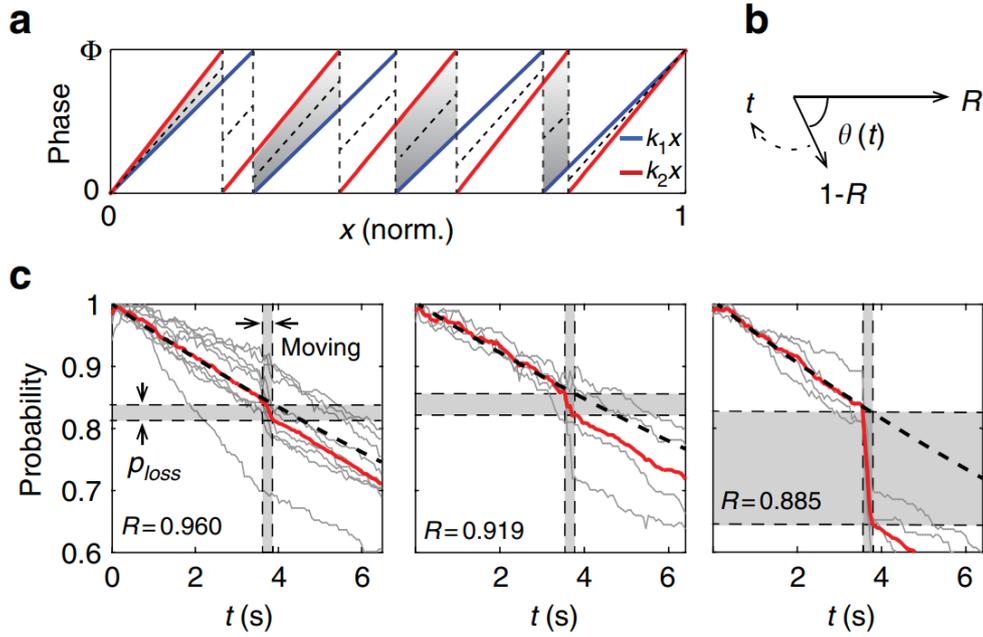


Figure 3.6: (a) Schematic illustration for phase jump which causes frame-to-frame flicker of traps. (b) Destructive interference of phase-jump area and no-phase-jump area. (c) Experimental result of flicker-induce atom loss during transport for various values of  $R$  [61].

the rest indicates phase jump area. If we denote the portion of the former  $R$  and the rest,  $1 - R$ , the  $1 - R$  part would destructively contribute to the trap depth during frame-to-frame evolution (Fig. 3.6). Fig. 3.6 (c) shows experimental result of atom loss by flicker during transport along  $5 \mu\text{m}$  travel distance, with step distance  $\Delta x = 180, 360, 520$  nm, respectively. The averaged  $R$  value for each is 0.960, 0.919, 0.885. The dashed decreasing line is the stationary loss without transport. The atoms were dragged in time that is depicted in the shaded areas. The red line shows averaged atom remaining probability of individual probabilities in gray plots. The loss probability due to transport,  $p_{loss}$  is depicted in vertical shades.

Single step loss  $P_1$  can be calculated by  $P_1 = 1 - (1 - p_{loss})^{1/n}$ , assuming the total remaining probability is repetitive multiplication of single step remaining probability. The calculated result is shown Fig. 3.7 for  $R > 0.84$  which is well laid between simulation result of loss for two temperatures  $T = U/10, U/13$ .

The measured transport efficiency of the 3D transport is shown in Fig. 3.8. Among  $N_{tr} = 8$  trap sites initially created on the original image plane ( $z = 0$ ), four sites were programmed to move in the positive axial direction ( $\Delta z > 0$ ) and come back, while the other four were moved in the negative axial direction ( $\Delta z < 0$ ) and came back. When the efficiency was defined as the ratio of the number of remaining atoms after an operation to the number of initially trapped atoms, Fig. 3.8(a) shows the measured efficiency for various step distances. The step distance is defined by the displacement between frames and the travel distance is the total displacement sum of the entire round-trip travel. Figure 3.8(b) shows the transport efficiency when the travel

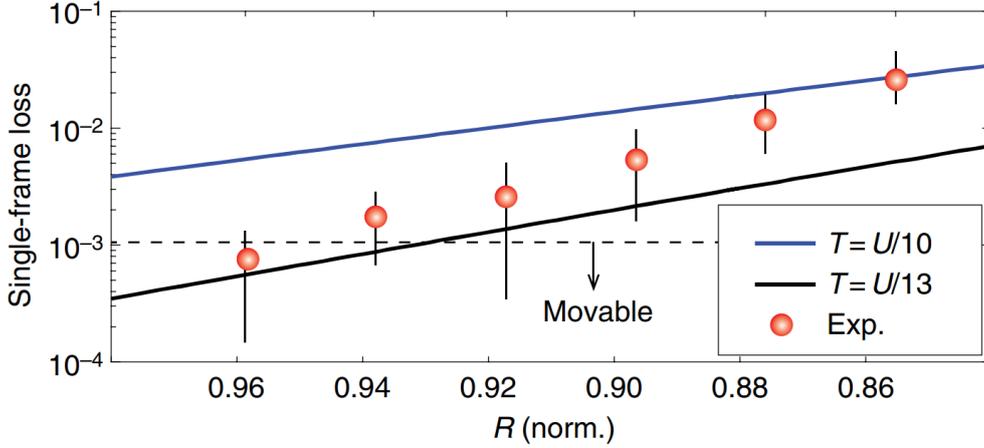


Figure 3.7: Experimental single step loss calculated from experimental total loss for various  $R$  values, compared with simulation with two temperatures,  $T = U/10, U/13$  [61].

distance was varied and the step distance was fixed at 250 nm. The result shows that the transport efficiency is little dependent on the rate at which the trap is refreshed. The transport loss in a single frame-to-frame operation is calculated from Figs. 3.8(a) and 3.8(b) and the results are displayed in Figs. 3.8(c) and 3.8(d), respectively. When being calibrated with the passive loss values, the single-step transport loss increases as the step distance increases; the farther the trap moves during a frame-to-frame operation, the more the intensity flickering occurs, as expected. On the other hand, the travel distance did not increase the single step loss significantly, when the step distance was fixed, in our experiment. The results indicates that within the practical travel range of 20  $\mu\text{m}$  from the initial focal plane, the traps are created in a reliable manner and, therefore, the atoms are safely transported.

We note that the main mechanism of loss from the trap is the frame-to-frame intensity flickering (degrading) which is caused by the phase jump at the boundaries of the finite SLM phase modulation range. For example, provided that the phase modulation range is  $[0, 2\pi]$ , a pixel evolving from  $1.99\pi$  to  $2.01\pi$  actually evolves from  $1.99\pi$  to  $0.01\pi$ , so the phase value in between has no defined value at certain time interval, failing to contribute to the trap. Note that heating and acceleration play minor roles because the scattering rate for heating is estimated to be smaller than the cooling rate, and the frame-to-frame moving speed of traps ( $\sim 0.1$  mm/s), determined by the relaxation time of the liquid crystals on the SLM, is adiabatically slow compared to the motional speed of the atoms in the traps ( $\sim 1$  m/s).

Our demonstration is currently limited by the laser power (currently up to  $N = 9$  atoms), the trap lifetime ( $\sim 13.4$  s), the SLM update rate (60 fps), the moving loss, and the computer data communication speed, some of which can be readily improved by a new integrated system design. An operation with  $N_{tr} = 25$  sites, for example, currently requires a look-up library of 660 TB (terabytes) in memory, simply exceeding the conventional PC memory capacity.

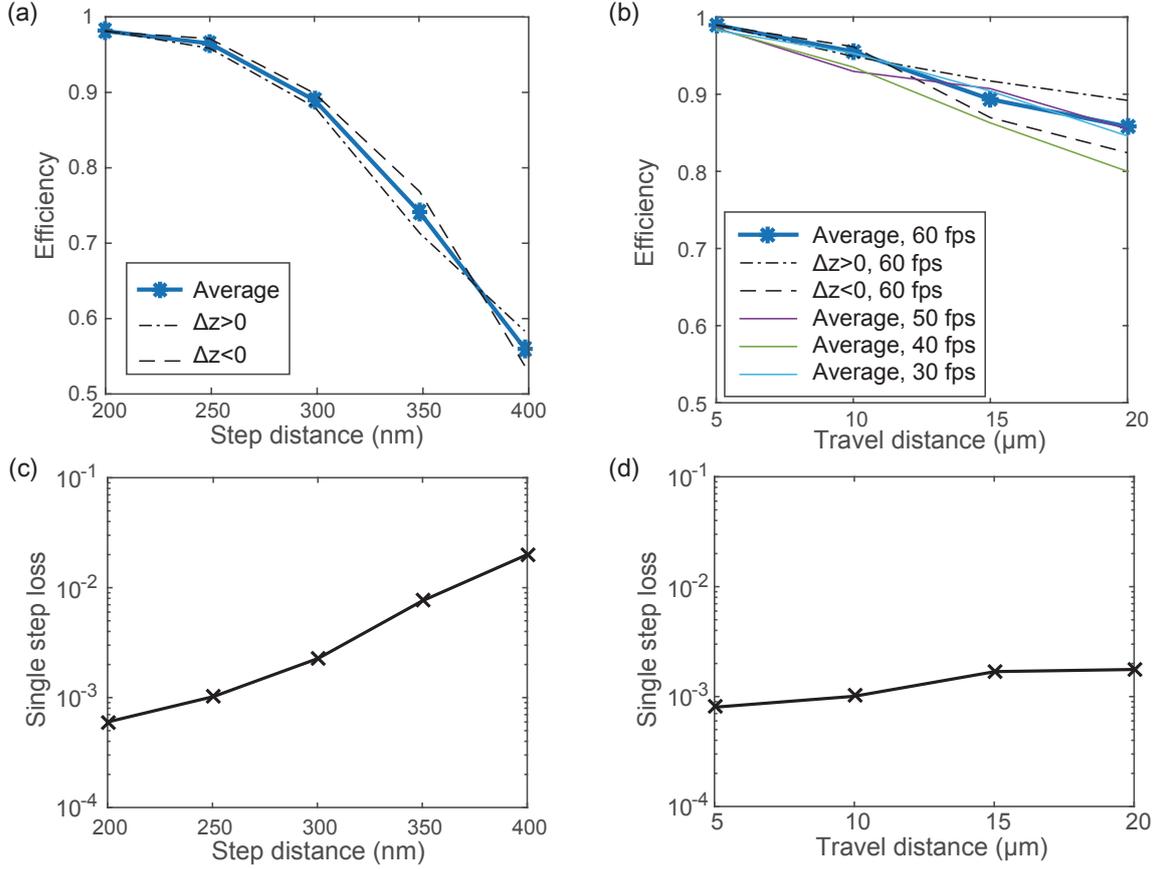


Figure 3.8: (a) Single-atom transport efficiency vs. the step distance in a 40-step (41 frames) transport operation, where the values for moving in positive and negative directions are drawn with dashed and dash-dot black lines, respectively, and the total average with the blue line with ‘star’ marks. The given transport (only) efficiency is the actually measured probability divided by the probability without moving. (b) Single-atom transport efficiency vs. the travel distance (with a fixed step distance of 250 nm) for various frame refresh rates and directions. (c), (d) The single-step loss vs. the step distance and travel distance, calculated from the main data in (a) and (b), respectively [36].

Thus, a large number of atoms may be better processed with a real-time feedback generation of phase patterns, rather than using memories, which is plausible with modest graphic processing systems [63]. Also, 3D imaging of atoms in real time could be executed in a compact way with rapid 3D microscopy with tunable lenses [64].

## Chapter 4. Defect-free atom array formation

Since single atom trapping probability per site is around one half [52] so far, the filling factor and the configuration of the entire array is, in consequence, probabilistic. Assuming the trap probability per site is 0.5, the probability of filling the entire array with  $N$  atoms equals  $1/2^N$ , which is far from unity. There has been improvement on increasing the site loading rate [65], but the loading probability of defect-free arrays still remains distant from unity, especially when a large number of atoms are to be used.

Here, we discuss a method of achieving defect-free arrays at a high probability by filling vacancies with nearby reservoir atoms, as shown in fig 4.1 [66, 67]. In this scheme, without atom loss, the probability of unity filling is not anymore  $1/2^N$ , but is  $P(N|M)$ , where  $P(a|b)$  denotes the probability of trapping more than or equal to  $a$  atoms out of the initial trap array with  $b$  traps. For this, the experimental system has to be a closed-loop feedback system to check the initial array status and relocate the trapped atoms accordingly. The operating sequence would be: initial array trap, array imaging, identify the array status, path planning for individual atoms, SLM phase calculation according to the path plan, perform the individual atom transport, and return to the array imaging stage.

When atom loss is considered, trapping atoms more than or equal to the required number of atoms does not guarantee unity filling anymore, so the loss rate is now to be taken into account. Atom loss in this scheme can be specified into two kinds; one is collisional loss, which occurs by colliding of a background atom which has high kinetic energy, and the other is moving loss, or flicker loss, which is due to lowered intensity by frame-to-frame intensity flickering of traps. A probability map to achieve unity filling of  $3 \times 3$  square array from  $7 \times 7$  initial array is shown in Fig. 4.2 with moving loss and collisional loss as parameters. The success rate can be increased by faster path planning, faster SLM operation, and some enhancement in experimental conditions. The experimental regime in past and present is depicted on the map in a red dot. The success rate of unity filling in such a scheme will be discussed in this chapter.

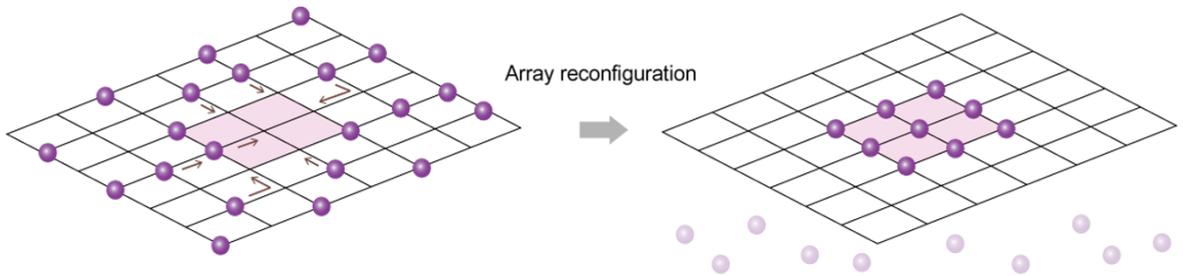


Figure 4.1: Array reconfiguration from an initial probabilistic array into a filled  $3 \times 3$  array. Defect-free arrays are achieved by filling vacancies with nearby reservoir atoms.

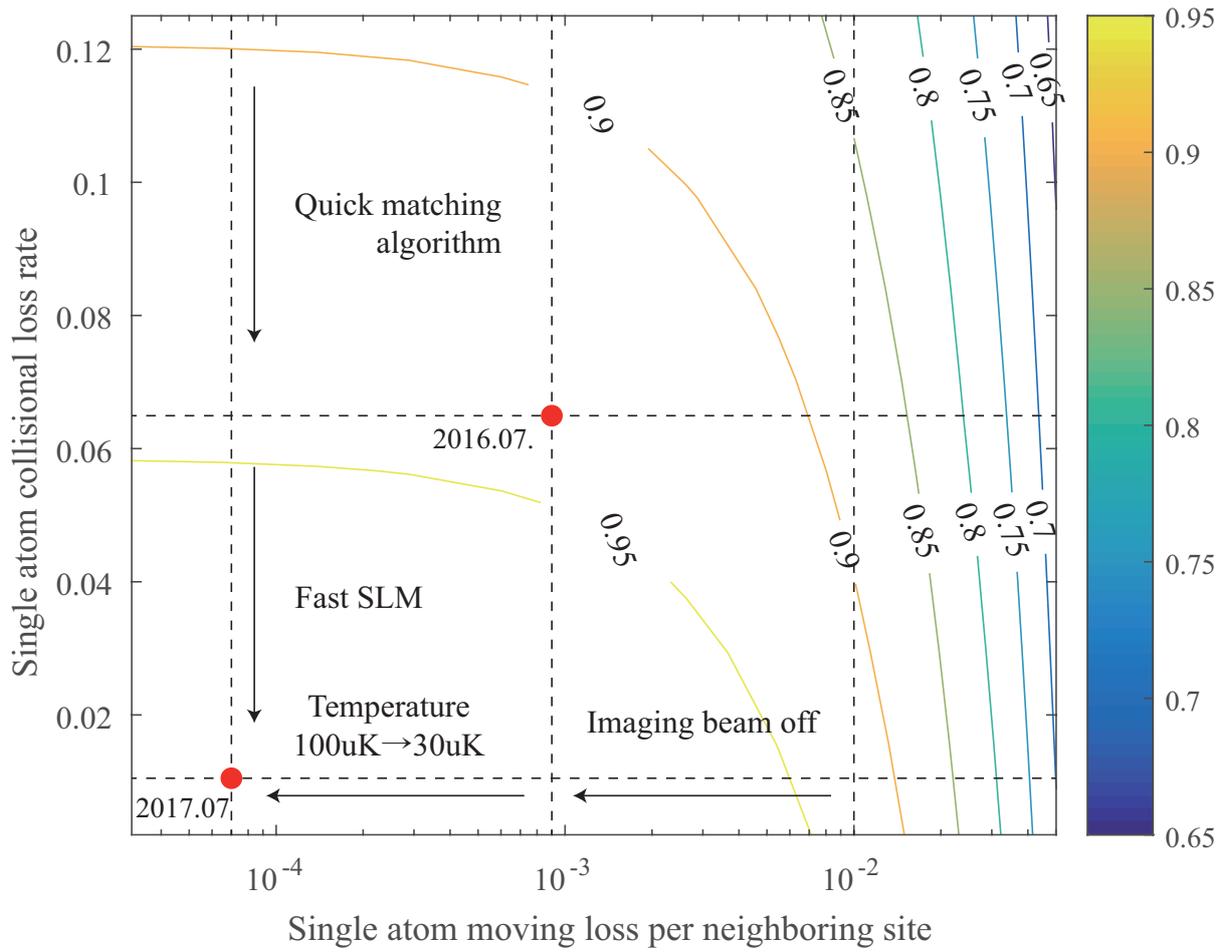


Figure 4.2: A road map of calculated transport efficiency for  $5 \mu\text{m}$  travel by single atom collisional loss rate and single atom moving loss per neighboring site. The transport efficiency has been improved by a quick matching algorithm, faster SLM, turning off the imaging beam, and reducing the atom temperature.

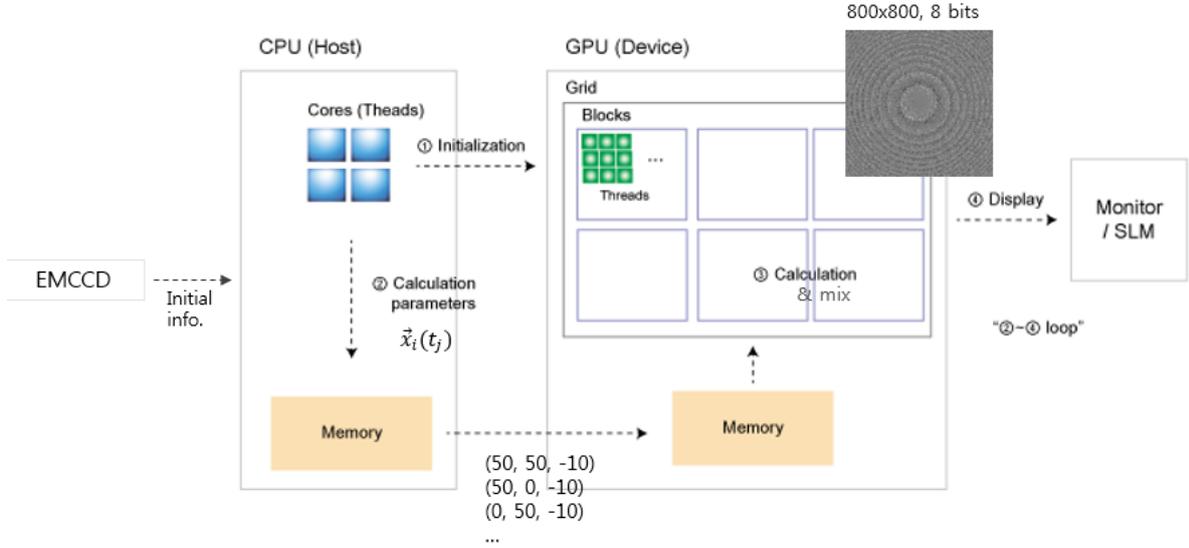


Figure 4.3: Structure and procedure of GPU-accelerated phase calculation.

## 4.1 Closed-loop feedback system

### 4.1.1 Optical and computational layout

The optical and computational setup mostly follows Ref. [36], or Chapter 2 of this thesis. A standard MOT was initially prepared and after a few seconds, dipole trap array beam was turned on. An initial probabilistic array of atoms was trapped in 0.5 seconds. Then the first readout of the initial array by an EMCCD was carried out. The image of the array was used for identification of the trap status. Based on the trap status, a path plan for each atom for vacancy filling was calculated by *Hungarian algorithm*, a matching algorithm which will be discussed in Sec. 4.3. Once the path plan was decided, individual atom transport began, while the SLM phase mask pattern was calculated in real-time during the operation. A different LCOS-SLM device (Meadowlarks XY Spatial light modulator  $512 \times 512$ ,  $15 \mu m$ , 200Hz), from the one in Chapter 2, is being used in ongoing continued experiment [68]. The real-time calculation of phase patterns was accelerated by a GPU (Graphic Processing Unit; the model used here is Nvidia GeForce GTX 1080), and will be discussed in the following subsection. After the first reconfiguration of the atom array, the second readout by the EMCCD was done. Like in the first operation, the atoms were controlled to fabricate a final defect-free array. If such an array was obtained, the iteration procedure was terminated.

### 4.1.2 GPU-accelerated phase calculation

For real-time operation of trap geometry reconfiguration, calculation of the phase pattern to be transferred to the SLM was accelerated by a graphics processing unit (GPU). The calculation of such phase calculation is mostly done in pixel-by-pixel independently, so it can be

much accelerated by exploiting a GPU’s parallel computing architecture, compared to CPU computing.

The structure and procedure of the GPU-accelerated calculation system is depicted in Fig. 4.3. First, the CPU initializes the GPU. When the EMCCD transfers the image of the atom array to the CPU, it performs path planning and stores position information  $\vec{x}_i(t_j)$  of each atom  $i$  in time sequence  $t_j$  in its memory, which runs in real-time. At each time step, the position information  $\vec{x}_i(t_j)$  is passed to GPU’s memory and the GPU carries out calculation of the corresponding phase mask pattern  $\phi(X, Y, t_j)$ , which is in 8-bit phase level and in size that matches the beam size shining on the SLM. While the created phase pattern  $\phi(X, Y, t_j)$  is transferred to SLM and applied, the next phase pattern  $\phi(X, Y, t_{j+1})$  is in turn calculated in the GPU.

## C++ code for real-time phase calculation and SLM control

In our method of single atom transport with an SLM, atoms move step-by-step, by a fixed order of step size, therefore atom loss by trap lifetime can be reduced by faster refresh of frames, or faster fps (frame-per-second). For this, we used C++ language, instead of MATLAB which is more accessible but calculates rather slowly, for our phase-calculating and SLM controlling computer program, along with GPU-assisted acceleration.

The program accepts a path plan, or a set of positions of each trap  $i$  at each time step  $t_j$ ,  $\vec{x}_i(t_j)$ , as inputs. The unit of the position is arbitrary for now. It also takes the information whether each site is designated to move to another site (if the site is trapping an atom and has assigned destination) or stays (if the site is not trapping any atom or has no assigned destination).

A prototype of the code using GPU-acceleration had been written in C++ and MATLAB and it was optimized by a software company “ArrayFire”, so the code is on ArrayFire platform.

The code is as following.

```

1 #include <arrayfire.h>
2 #include <cstdio>
3 #include <cstdlib>
4 #include <fstream>
5 #include <vector>
6 #include <time.h>
7
8
9 #include "stdafx.h" // Does nothing but #include targetver.h.
10
11 #include <vector>
12 #include <cstdio>
13 #include <conio.h>
14 #include "Blink.SDK.H" // Relative path to SDK header.
15

```

```

16 #include <string.h>
17
18 const int num_init = 25;
19
20 using namespace af;
21
22 // Typedef for the container for our phase targets.
23 typedef std::vector<unsigned char> uchar_vec;
24
25 void loadArraysFromFiles(array &posXpath, array &posYpath, array &assigned)
26 {
27     // posXpath and posYpath
28     {
29         std::vector<float> posXpathVec(num_init * 20); // Current size
30             from files
31
32         std::vector<float> posYpathVec(num_init * 20); // Current size from
33             files
34
35         std::ifstream posXFile("posXpaths.txt");
36         if(!posXFile.is_open())
37             std::cerr << "Error: posXpaths.txt failed to open" << std::endl;
38
39         int i = 0;
40         while(!posXFile.eof()) {
41             posXFile >> posXpathVec[i++];
42         }
43
44         std::ifstream posYFile("posYpaths.txt");
45         if(!posYFile.is_open())
46             std::cerr << "Error: posYpaths.txt failed to open" << std::endl;
47
48         i = 0;
49         while(!posYFile.eof()) {
50             posYFile >> posYpathVec[i++];
51         }
52
53         posXpath = array(20, num_init, posXpathVec.data()).T();
54         posYpath = array(20, num_init, posYpathVec.data()).T();
55     }
56
57 // assigned
58 {
59     std::vector<float> assignVec(num_init); // Current size from files
60
61     std::ifstream assignFile("assign.txt");
62     if(!assignFile.is_open())

```

```

62         std::cerr << "Error: assign.txt failed to open" << std::endl;
63
64         int i = 0;
65         while(!assignFile.eof()) {
66             assignFile >> assignVec[i++];
67         }
68         assigned = array(num_init, assignVec.data()).as(u32);
69     }
70 }
71
72 void loadTrig(array &loading, array &trig){
73     // loading and trig
74     {
75         std::vector<float> loadingVec(num_init+1); // Current size from
76             files
77
78         std::ifstream loadingFile("load.txt");
79         if (!loadingFile.is_open())
80             std::cerr << "Error: load.txt failed to open" << std::
81                 endl;
82
83         int i = 0;
84         while (!loadingFile.eof()) {
85             loadingFile >> loadingVec[i++];
86         }
87
88         array loadingA(num_init+1, loadingVec.data());
89         loadingA = loadingA.as(u32);
90         loading = loadingA(seq(num_init));
91         trig = loadingA(end);
92
93         //af_print(loadingA(end));
94     }
95 }
96 int main(int argc, char *argv[])
97 {
98     try {
99         af::info();
100
101         // SLM initialize
102         const int board_number = 1;
103
104         // Construct a Blink_SDK instance with Overdrive capability.
105
106         const unsigned int bits_per_pixel = 8U;
107         const unsigned int pixel_dimension = 512U;
108         const bool is_nematic_type = true;
109         const bool RAM_write_enable = true;

```

```

108     const bool          use_GPU_if_available = true;
109     const char* const   regional_lut_file = "slm4205.820.txt";
110     const char* const   loadlutfile = "slm4205.820nm.P8.lut";
111
112     unsigned int n_boards_found = 0U;
113     bool         constructed_okay = true;
114
115     Blink_SDK sdk(bits_per_pixel, pixel_dimension, &n_boards_found,
116                 &constructed_okay, is_nematic_type, RAM_write_enable,
117                 use_GPU_if_available, 2U, regional_lut_file);
118
119     // load_LUT for non-overdrive
120     //sdk.Load_LUT_file(board_number, loadlutfile);
121
122     // Check that everything started up successfully.
123     bool okay = constructed_okay && sdk.Is_slm_transient_constructed
124         ();
125
126     if (okay)
127     {
128         enum { e_n_true_frames = 3 };
129         sdk.Set_true_frames(e_n_true_frames);
130         sdk.SLM_power(true);
131         okay = sdk.Load_linear_LUT(board_number);
132     }
133     else
134     {
135         printf("gg");
136     }
137
138     //
139     // Load Arrays From Files
140     array posXpath, posYpath, loading, trig, assigned;
141     loadArraysFromFiles(posXpath, posYpath, assigned);
142     loadTrig(loading, trig);
143     // These variables are not used here. Define them later.
144     //array weights = constant(1, loading.elements(), f32);
145     //array peakValues = constant(0, loading.elements(), c32);
146     //
147     //int frameNumber = posXpath.dims(1);
148
149     const array X = iota(dim4(1, 512), dim4(512, 1), f32) - 256.f;
150     const array Y = iota(dim4(512, 1), dim4(1, 512), f32) - 256.f;
151
152     array suminit = constant(0, 512, 512, c32);
153     array suminit1 = constant(0, 512, 512, c32);
154

```

```

155     {
156         // prep work
157         static const cfloat factor(0, 0.0012); // same as 1i *
           0.0012
158         // Generate loading.eelements random numbers along 3rd
           dimension and then tile
159         array randVals = tile(2 * Pi * randu(1, 1, loading.
           elements(), f32), X.dims(0), X.dims(1), 1);
160
161         // This tiles each values into a matrix and the 3rd
           dimension is for k
162         // New dimensions are 512 x 512 x loading.elements()
163         array posXpathVals = tile(moddims(posXpath(span, 0), 1,
           1, loading.elements()),
           X.dims(0), X.dims(1), 1);
164         array posYpathVals = tile(moddims(posYpath(span, 0), 1,
           1, loading.elements()),
           Y.dims(0), Y.dims(1), 1);
165
166         suminit1 += sum(exp(factor *
167             ((posXpathVals * tile(X, 1, 1, loading.elements()
168                 ))
169             + (posYpathVals * tile(Y, 1, 1, loading.elements
170                 ()))
171             + randVals))
172             , 2);
173
174         // Clear the temporary arrays so that memory is not
           locked
175         randVals = array();
176         posXpathVals = array();
177         posYpathVals = array();
178     }
179
180     array phaseGSW1 = arg(suminit1);
181     // Do round instead of u32 cast. Important to do %256 after
182     array phaseGSW11 = round(((phaseGSW1 + 2 * Pi) * (255 / (2 * Pi))
           )) % 256; // Do Round
183     array phaseGSW2 = constant(0, 512, 512, c32);
184     array phaseAmp = constant(1, 512, 512, f32);
185     array phaseU = constant(128, 512 * 512, u8);
186     static const int numOver = 4;
187     static const int numOver512 = 512 * numOver;
188
189     // These require +1 because meshgrid does 1 index
190     array XX = iota(dim4(1, numOver512), dim4(numOver512, 1), f32) +
           1;
191     array YY = iota(dim4(numOver512, 1), dim4(1, numOver512), f32) +

```

```

192         1;
193     const array gaussianForm = exp(-(pow(XX - (numOver512 / 2), 2) +
194         pow(YY - (numOver512 / 2), 2))
195         / (2.f * 118 * 118));
196     const array gaussianForm2 = exp(-(pow(X, 2) + pow(Y, 2))
197         / (2.f * 118 * 118));
198     array circle = 1 - floor((pow(XX - (numOver512 / 2), 2) + pow(YY
199         - (numOver512 / 2), 2))
200         / ((512 / 2) * (512 / 2)));
201     array circle2 = 1 - floor((pow(X, 2) + pow(Y, 2))
202         / ((512 / 2) * (512 / 2)));
203     circle = circle > 0;
204     circle2 = circle2 > 0;
205
206     const array maske = gaussianForm * circle;
207     const array mask = gaussianForm2 * circle2;
208
209     array phaseover = constant(0, numOver512, numOver512, c32);
210     array phaseoverfft = constant(1, numOver512, numOver512, c32);
211     array targetAmp = constant(0, numOver512, numOver512, f32);
212     array phaseR = constant(0, numOver512, numOver512, c32);
213
214     const int GSWiter = 5;
215     const float rf = 0.7;
216     const float sf = 1;
217
218     const array psf = exp(-pow(((X * X) / rf) + ((Y * Y) / rf), sf)).
219         as(f32);
220     array psf2 = fft2(psf);
221     psf2 = abs(psf2) / tile(max(abs(Flat(psf2))), psf2.dims());
222
223     array weights = constant(1, loading.elements(), f32);
224     array peakValues = constant(0, loading.elements(), c32);
225     int frameNumber = posXpath.dims(1);
226
227     static const cfloat oneI(0, 1);
228     array loading0Idx = where(loading == 0); //loading.elements() is the
229         number of elements. seq(loading.elements()) is 0:1:#-1
230
231     array loading1Idx = where(loading == 1);
232     array weighter = constant(0.1, frameNumber, 1, f32);
233     weighter(0) = 0.7; // ((float)0.8, (float)0.6, (float)0.4, (float)
234         0.2);
235     weighter(1) = 0.4;
236     weighter(2) = 0.1;

```

```

234     weighter(frameNumber-5) = 0.1; // ((float)0.4, (float)0.6, (float
        )0.8, (float)1, (float)1);
235     weighter(frameNumber - 4) = 0.1;
236     weighter(frameNumber - 3) = 0.4;
237     weighter(frameNumber - 2) = 0.7;
238     weighter(frameNumber - 1) = 1;
239     af_print(weighter);
240     array qq = assigned(seq(loading0Idx.elements())) - 1;
241     array qq1;
242     if (loading0Idx.elements() == loading.elements()){
243         loading1Idx = loading0Idx;
244         qq1 = qq;
245     }
246     else{
247         seq qass(loading0Idx.elements(), loading.elements() - 1)
            ;// seq qass(a,a)=a
248         qq1 = assigned(qass) - 1;
249     }
250
251
252     // The offsets here are 0 indexed so thats why the increment/
        decrement
253     const seq S(numOver512 / 2 - 256, numOver512 / 2 + 255);
254
255     while (1){
256
257         FILE *f;
258
259         while (1){
260             Sleep(1);
261             loadTrig(loading, trig);
262             //printf("%d", trig.nonzeros());
263             if (trig.nonzeros()){ //from here to qqq-for-loop : 13ms.
264
265                 break;
266             }
267
268         }
269         trig = 0;
270
271         loadArraysFromFiles(posXpath, posYpath, assigned);
272
273         loading0Idx = where(loading == 0);
274         loading1Idx = where(loading == 1);
275
276         qq = assigned(seq(loading0Idx.elements())) - 1;
277
278         if (loading0Idx.elements() == loading.elements()){

```

```

279         loading1Idx = loading0Idx;
280         qq1 = qq;
281     }
282     else{
283         seq qass(loading0Idx.elements(), loading.elements() - 1);// seq
                qass(a,a)=a
284         qq1 = assigned(qass) - 1;
285     }
286
287
288     timer::start();
289     for (int qq = 0; qq < frameNumber; qq++) {
290
291         for (int gswiter = 0; gswiter < GSWiter; gswiter++) {
292             phaseGSW1 = abs(iff2(fft2(phaseGSW11.as(f32)) *
                psf2)).as(f32); //+2Hz (5 iter)
293             phaseGSW2 = mask * exp(oneI * (phaseGSW1 *
                0.0246));// *phaseAmp;
294             phaseover(S, S) = phaseGSW2;
295             phaseoverfft = shift(fft2(phaseover), phaseover.
                dims(0) / 2, phaseover.dims(1) / 2);
296
297             std::string fname_str = "phaseoverfft" + std::
                to_string(qq) + ".bin";
298             //af::saveImage(fname_str.c_str(), phaseoverfft);
299             af::saveArray("phase", phaseoverfft, fname_str.
                c_str(), false);
300
301             //peakValues = diag(phaseoverfft(
302             //round(posYpath(span, qq) / (1.271 * 8 /
                numOver) + (numOver512 / 2 - 1)),
303             //round(posXpath(span, qq) / (1.271 * 8 /
                numOver) + (numOver512 / 2 - 1))));
304             peakValues = diag(phaseoverfft(round(posYpath(
                span, qq) / (1.271 * 8 / numOver) + (
                numOver512 / 2 - 1)), round(posXpath(span, qq
                ) / (1.271 * 8 / numOver) + (numOver512 / 2 -
                1))));
305
306             if (loading0Idx.elements() == loading.elements()
                || loading1Idx.elements() == loading.elements
                ()){
307                 weights(span) *= tile(mean(abs(peakValues
                (span))), loading.elements()) / abs(
                peakValues(span)) / tile(mean(weights(
                span)), loading.elements());
308
309                 targetAmp(round(posYpath(span, qq) /

```

```

310         (1.271 * 8 / numOver) + (numOver512 /
311         2 - 1)) + round(posXpath(span, qq) /
312         (1.271 * 8 / numOver) + (numOver512 /
313         2 - 1)) * targetAmp.dims(0))
314         = weights(span);
315     }
316     else{
317         array pv1a = abs(peakValues(loading1Idx))
318         ;
319         array pv0a = abs(peakValues(loading0Idx))
320         ;
321         if (qqq > 2 && qqq < frameNumber - 3){
322             weights(loading0Idx) = tile(
323                 weighter(qqq), loading0Idx.
324                 elements());
325         }
326         else{
327             weights(loading1Idx) *= tile(mean
328                 (pv1a), loading1Idx.elements()
329                 ) / pv1a / tile(mean(weights(
330                 loading1Idx)), loading1Idx.
331                 elements());
332             weights(loading0Idx) *= (tile(
333                 mean(pv0a), loading0Idx.
334                 elements()) / pv0a
335                 / tile(mean(weights(
336                 loading0Idx)),
337                 loading0Idx.elements()
338                 ))
339                 * tile(weighter(qqq),
340                 loading0Idx.elements()
341                 );
342         }
343     }
344     /* (1 / ((float)qqq + 1) + 1 / ((float)
345     frameNumber - ((float)qqq + 1) + 1));
346
347     // Use linear indexing = y * d0 + x
348     // Simply doing targetAmp(x, y) will
349     result in M*N number of elements.
350     // This worked in the for loop as both x
351     and y were 1.
352     // But when using batched mode, x and y
353     are 121 each result in 14641 values
354     // while weights(qq) is 121 only.
355     // So use linear indexing.

```

```

335         targetAmp(round(posYpath(qq, qq) /
                      (1.271 * 8 / numOver) + (numOver512 /
                      2 - 1)) +
336         round(posXpath(qq, qq) / (1.271
                      * 8 / numOver) + (numOver512 /
                      2 - 1)) * targetAmp.dims(0))
337         = weights(qq);
338
339         targetAmp(round(posYpath(qq1, qq) /
                      (1.271 * 8 / numOver) + (numOver512 /
                      2 - 1)) +
340         round(posXpath(qq1, qq) / (1.271
                      * 8 / numOver) + (numOver512
                      / 2 - 1)) * targetAmp.dims(0))
341         = weights(qq1);
342     }
343     /*for (int q = 0; q < loading.elements(); q++) {
344         array qq = assigned(q)-1;
345         targetAmp(round(posYpath(qq, qq) /
                      (1.271 * 8 / numOver) + (numOver512 /
                      2-1)),
346         round(posXpath(qq, qq) / (1.271 * 8 /
                      numOver) + (numOver512 / 2-1)))
347         = weights(qq);
348     }*/
349
350
351     phaseR = ifft2(shift(targetAmp * exp(oneI * arg(
                      phaseoverfft)), numOver512 / 2, numOver512 /
                      2));
352     //phaseR = ifft2(shift(targetAmp , numOver512 /
                      2, numOver512 / 2));
353     phaseGSW1 = arg(phaseR(S, S));
354     phaseGSW11 = round(((phaseGSW1 + (6.2832)) *
                      (40.5845))) % 256; // Do Round
355 }
356
357
358 unsigned char *puc = phaseGSW11.as(u8).host<unsigned char
                      >(); // 7ms
359 //printf("elapsed seconds: %g\n", timer::stop());
360 okay = sdk.Write_overdrive_image(board_number, puc);
361 //okay = sdk.Write_image(board_number, puc,
                      pixel_dimension);
362 targetAmp = constant(0, numOver512, numOver512, f32);
363 af::freeHost(puc);
364
365

```

```

366         }
367         sync(); printf(" elapsed seconds: %g\n", frameNumber / timer::stop
           ());
368
369         f = fopen("load.txt", "w");
370         for (int i = 0; i < loading.elements()+1 ; i++){
371             fprintf(f, "%d\n", 0);
372         }
373         fclose(f); // 1ms
374         //printf(" elapsed seconds: %g\n", timer::stop());
375         //system("PAUSE");
376
377     }
378
379     system("PAUSE");
380
381 } catch (af::exception& e) {
382     fprintf(stderr, "%s\n", e.what());
383     throw;
384 }
385
386 return 0;
387 }

```

The summary of the code is as follows. The code starts at line 94 which is the starting point of the main function. The code before this line is pre-defined functions and variables. The SLM is initialized and setting is done in line 99-137. Next, it calls function “loadArraysFromFiles” in line 141. It loads the information about time trajectory of the positions (i.e., paths) of individual atoms and which site an atom comes from to each site, from the text file “posXpaths.txt”, “posYpaths.txt”, and “assign.txt”. The function input arguments are the addresses of the array variables. Function “loadTrig” in line 142 takes information about whether each site is loaded and the reconfiguration trigger is on (a loop run continuously in operation which will appear below and if the trigger is on, reconfiguration sequence begins) from text file “load.txt”. When the first frame of phase pattern movie is generated by GSW algorithm, the initial pattern is generated by superposition algorithm which is a decent guessing of the phase before iteration and this phase pattern is generated in line 155-178. Phase pattern generation starts at line 255. The while loop in line 255 ensures the operation runs every time the trigger is turned on. The while loop in line 259 runs until a trigger comes in and proceeds to next operation when it’s on. “loading0Idx” / “loading1Idx” contains the indices of sites that doesn’t have / have atoms, respectively. qq / qq1 contains the sites that are assigned to be atom source sites or not, respectively. Actual part for phase generations starts at line 289; loop variable qqq runs for time series of frame and gswiter runs for GSW iteration for each frame. Since the phase generating part of the code is complicated with plenty of variables, the core structure of the code, omitting any coefficients, is provided below.

**Field at SLM**  $mask * \exp[i \text{phaseGSW1}] \rightarrow \text{phaseGSW2}$

**Zero padding**  $\text{phaseGSW2}$  ( $512 \times 512$ )  $\rightarrow$  to the center of blank space  $\text{phaseover}$  ( $2048 \times 2048$ )

**Fourier transform**  $\text{phaseover} \rightarrow \text{phaseoverfft}$  (SLM to atom trap)

**Weighting**  $\text{weights}$  (obtained from  $\text{phaseoverfft}$ )  $\rightarrow \text{targetAmp}$

**Field at atom trap**  $\text{targetAmp} * \exp[i \text{arg}(\text{phaseoverfft})]$

**Inverse Fourier transform**  $\text{targetAmp} * \exp[i \text{arg}(\text{phaseoverfft})] \rightarrow \text{phaseR}$  (atom trap to SLM)

**Phase without zero padding**  $\text{phaseR}$  ( $2048 \times 2048$ )  $\rightarrow \text{phaseGSW1}$  ( $512 \times 512$ )

Above is one cycle in GSW algorithm and the SLM phase is acquired by,

**SLM phase acquisition**  $\text{phaseGSW1}$  ( $[0, 2\pi]$ )  $\rightarrow \text{phaseGSW11}$  ( $[0, 255]$ )

### 4.1.3 MATLAB code for analysing the array status and assigning path

This code is for entering position information of the atoms, making decision about which atom goes to which site (matching), and recording the path information, assignment information and trigger information on a text file.

```
1 addpath('C:\Program Files\ArrayFire\v3\examples\ModifiedGSWAlgorithm 180405
  backup\')
2 posX = [-260 -130 0 130 260 -300 -233.3333 -166.6667 -100 -33.3333 33.3333 100
  166.6667 233.3333 300 -300 -233.3333 -166.6667 -100 -33.3333 33.3333 100
  166.6667 233.3333 300];
3 posY = [300 300 300 300 300 410 410 410 410 410 410 410 410 410 190 190 190
  190 190 190 190 190 190 190]+25;
4 sorter=1:length(posX);
5
6
7 xx = [96 134 173 212 251 84 104 123 144 164 183 204 223 243 263 84 104 123 144
  164 183 204 223 243 263];
8 yy = [66 66 66 66 66 34 34 34 34 34 35 35 35 35 35 97 97 97 97 97 98 98 98 98
  98]+18;
9 for k=1:length(xx);
10 imagesum(k)=sum(sum(AA(yy(k)-3:yy(k)+3,xx(k)-3:xx(k)+3),1),2)/49;
11 end;
12
13
14 threscompatible502=[131 126.8 122.6 128.2 129.6 131 133.8 133.8 133.8 124 125.4
  125.4 131 125.4 124 125.4 126.8 125.4 129.6 125.4 128.2 125.4 125.4 128.2
  126.8];
15
16 digital=(sign(imagesum-threscompatible502)+1)/2;
17 loading=digital';
18 trig=quo;
19 sumloading=sum(loading);
20
```

```

21
22
23 targeter = [1:5];
24
25 % posXt=posX( sorter (1:sumloading) );
26 % posYt=posY( sorter (1:sumloading) );
27 posXt=posX( targeter );
28 posYt=posY( targeter );
29
30 costmat=zeros( length( posXt ), length( posX ) );
31 for kq=1:length( posXt );
32 costmat( kq, : )=((posX-posXt( kq )).^2+(posY-posYt( kq )).^2) ./ loading ;% distance
    square
33 end;
34
35 targetsorted = [1:length( posXt )];
36
37 costmat( isnan( costmat ) )=Inf;
38 assigned=(munkres( costmat ) ');
39 % assigned( assigned==0)=find( assigned==0);
40
41
42 loading2=loading;
43 loading2( assigned==0)=1;
44 assigned( assigned==0)=targetsorted( ( assigned==0) );
45 unassigned=setxor( 1:length( loading ), assigned );
46
47
48 framenummer=20;
49 posXpath=(zeros( length( posX ), framenummer) );
50 posYpath=(zeros( length( posY ), framenummer) );
51 for kqq=1:framenummer;%%path generation colliding path
52 posXpath( :, kqq)=posX;
53 posXpath( assigned, kqq)=(posX( assigned )-(posX( assigned )-posXt)/(framenummer-1)*(
    kqq-1));
54 posYpath( :, kqq)=posY;
55 posYpath( assigned, kqq)=(posY( assigned )-(posY( assigned )-posYt)/(framenummer-1)*(
    kqq-1));
56 if kqq>framenummer/2;
57 posXpath( loading2( targetsorted )==0,kqq)=posX( assigned( assigned>length( posYt ) ) ); %
    deterministic loading
58 posYpath( loading2( targetsorted )==0,kqq)=posY( assigned( assigned>length( posXt ) ) );
59
60 % posXpath( loading==0,kqq)=posX( sorter( sumloading+1:end) );
61 % posYpath( loading==0,kqq)=posY( sorter( sumloading+1:end) );
62
63 end;
64 end;

```

```

65
66 % fixing the position
67 %posXpath=posX' * ones(1,framenumber);
68 %posYpath=posY' * ones(1,framenumber);
69 %trig=0;
70
71
72 if trig==1
73 save('C:\Program Files\ArrayFire\v3\examples\ModifiedGSWAlgorithm 180405 backup\
      posXpaths.txt','posYpath','-ascii')
74 save('C:\Program Files\ArrayFire\v3\examples\ModifiedGSWAlgorithm 180405 backup\
      posYpaths.txt','posXpath','-ascii')
75 save('C:\Program Files\ArrayFire\v3\examples\ModifiedGSWAlgorithm 180405 backup\
      assign.txt','unassigned','assigned','-ascii')
76 save('C:\Program Files\ArrayFire\v3\examples\ModifiedGSWAlgorithm 180405 backup\
      load.txt','loading','trig','-ascii')
77 save('C:\Program Files\ArrayFire\v3\examples\ModifiedGSWAlgorithm 180405 backup\
      loadbefore.txt','loading','trig','-ascii')
78
79 else
80 save('C:\Program Files\ArrayFire\v3\examples\ModifiedGSWAlgorithm 180405 backup\
      loadafter.txt','loading','trig','-ascii')
81 end

```

The code sets the atom positions in lines 2-8. It catches which-atom-trapped information in lines 9-19 with pre-measured threshold intensity information in line 14. Target sites are decided in lines 23-28. The cost matrix is generated in lines 30-37 and the code for Hungarian algorithm is called in line 38. Loading information and paths information are generated in lines 42-69 accordingly. The information is written in text files in lines 72-81.

#### 4.1.4 MATLAB code of the Hungarian algorithm

This is an open source implementing the Hungarian algorithm with an input cost matrix “costMat”. The algorithm consists of processing operations on matrices and the reference of the procedure can be found in [69].

```

1 function [assignment,cost] = munkres(costMat)
2 % MUNKRES Munkres (Hungarian) Algorithm for Linear Assignment Problem.
3 %
4 % [ASSIGN,COST] = munkres(COSTMAT) returns the optimal column indices ,
5 % ASSIGN assigned to each row and the minimum COST based on the assignment
6 % problem represented by the COSTMAT, where the (i,j)th element represents the
      cost to assign the jth
7 % job to the ith worker.
8 %
9 % Partial assignment: This code can identify a partial assignment is a full
10 % assignment is not feasible. For a partial assignment, there are some

```

```

11 % zero elements in the returning assignment vector, which indicate
12 % un-assigned tasks. The cost returned only contains the cost of partially
13 % assigned tasks.
14
15 % This is vectorized implementation of the algorithm. It is the fastest
16 % among all Matlab implementations of the algorithm.
17
18 % Examples
19 % Example 1: a 5 x 5 example
20 %{
21 [assignment, cost] = munkres(magic(5));
22 disp(assignment); % 3 2 1 5 4
23 disp(cost); %15
24 %}
25 % Example 2: 400 x 400 random data
26 %{
27 n=400;
28 A=rand(n);
29 tic
30 [a,b]=munkres(A);
31 toc           % about 2 seconds
32 %}
33 % Example 3: rectangular assignment with inf costs
34 %{
35 A=rand(10,7);
36 A(A>0.7)=Inf;
37 [a,b]=munkres(A);
38 %}
39 % Example 4: an example of partial assignment
40 %{
41 A = [1 3 Inf; Inf Inf 5; Inf Inf 0.5];
42 [a,b]=munkres(A)
43 %}
44 % a = [1 0 3]
45 % b = 1.5
46 % Reference:
47 % "Munkres' Assignment Algorithm, Modified for Rectangular Matrices",
48 % http://csclab.murraystate.edu/bob.pilgrim/445/munkres.html
49
50 % version 2.3 by Yi Cao at Cranfield University on 11th September 2011
51
52 assignment = zeros(1, size(costMat,1));
53 cost = 0;
54
55 validMat = costMat == costMat & costMat < Inf;
56 bigM = 10^(ceil(log10(sum(costMat(validMat))))+1);
57 costMat(~validMat) = bigM;
58

```

```

59 % costMat(costMat~=costMat)=Inf;
60 % validMat = costMat<Inf;
61 validCol = any(validMat,1);
62 validRow = any(validMat,2);
63
64 nRows = sum(validRow);
65 nCols = sum(validCol);
66 n = max(nRows,nCols);
67 if ~n
68     return
69 end
70
71 maxv=10*max(costMat(validMat));
72
73 dMat = zeros(n) + maxv;
74 dMat(1:nRows,1:nCols) = costMat(validRow,validCol);
75
76 %*****
77 % Munkres' Assignment Algorithm starts here
78 %*****
79
80 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
81 % STEP 1: Subtract the row minimum from each row.
82 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
83 minR = min(dMat,[],2);
84 minC = min(bsxfun(@minus, dMat, minR));
85
86 %*****
87 % STEP 2: Find a zero of dMat. If there are no starred zeros in its
88 %         column or row start the zero. Repeat for each zero
89 %*****
90 zP = dMat == bsxfun(@plus, minC, minR);
91
92 starZ = zeros(n,1);
93 while any(zP(:))
94     [r,c]=find(zP,1);
95     starZ(r)=c;
96     zP(r,:)=false;
97     zP(:,c)=false;
98 end
99
100 while 1
101 %*****
102 % STEP 3: Cover each column with a starred zero. If all the columns are
103 %         covered then the matching is maximum
104 %*****
105     if all(starZ>0)
106         break

```

```

107     end
108     coverColumn = false(1,n);
109     coverColumn(starZ(starZ>0))=true;
110     coverRow = false(n,1);
111     primeZ = zeros(n,1);
112     [rIdx, cIdx] = find(dMat(~coverRow,~coverColumn)==bsxfun(@plus,minR(~coverRow
        ),minC(~coverColumn)));
113     while 1
114         %*****
115         % STEP 4: Find a noncovered zero and prime it. If there is no starred
116         % zero in the row containing this primed zero, Go to Step 5.
117         % Otherwise, cover this row and uncover the column containing
118         % the starred zero. Continue in this manner until there are no
119         % uncovered zeros left. Save the smallest uncovered value and
120         % Go to Step 6.
121         %*****
122         cR = find(~coverRow);
123         cC = find(~coverColumn);
124         rIdx = cR(rIdx);
125         cIdx = cC(cIdx);
126         Step = 6;
127         while ~isempty(cIdx)
128             uZr = rIdx(1);
129             uZc = cIdx(1);
130             primeZ(uZr) = uZc;
131             stz = starZ(uZr);
132             if ~stz
133                 Step = 5;
134                 break;
135             end
136             coverRow(uZr) = true;
137             coverColumn(stz) = false;
138             z = rIdx==uZr;
139             rIdx(z) = [];
140             cIdx(z) = [];
141             cR = find(~coverRow);
142             z = dMat(~coverRow, stz) == minR(~coverRow) + minC(stz);
143             rIdx = [rIdx(:);cR(z)];
144             cIdx = [cIdx(:);stz(ones(sum(z),1))];
145         end
146         if Step == 6
147             %
148             %*****
149             % STEP 6: Add the minimum uncovered value to every element of each
150             % covered

```

```

149         %           row, and subtract it from every element of each uncovered
150         %           column.
151         %           Return to Step 4 without altering any stars, primes, or
152         %           covered lines.
153         %*****
154         [minval, rIdx, cIdx]=outerplus(dMat(~coverRow,~coverColumn),minR(~
155         %           coverRow),minC(~coverColumn));
156         minC(~coverColumn) = minC(~coverColumn) + minval;
157         minR(coverRow) = minR(coverRow) - minval;
158     else
159         break
160     end
161 end
162 %*****
163 % STEP 5:
164 % Construct a series of alternating primed and starred zeros as
165 % follows:
166 % Let Z0 represent the uncovered primed zero found in Step 4.
167 % Let Z1 denote the starred zero in the column of Z0 (if any).
168 % Let Z2 denote the primed zero in the row of Z1 (there will always
169 % be one). Continue until the series terminates at a primed zero
170 % that has no starred zero in its column. Unstar each starred
171 % zero of the series, star each primed zero of the series, erase
172 % all primes and uncover every line in the matrix. Return to Step 3.
173 %*****
174 rowZ1 = find(starZ==uZc);
175 starZ(uZr)=uZc;
176 while rowZ1>0
177     starZ(rowZ1)=0;
178     uZc = primeZ(rowZ1);
179     uZr = rowZ1;
180     rowZ1 = find(starZ==uZc);
181     starZ(uZr)=uZc;
182 end
183 end
184
185 % Cost of assignment
186 rowIdx = find(validRow);
187 colIdx = find(validCol);
188 starZ = starZ(1:nRows);
189 vIdx = starZ <= nCols;
190 assignment(rowIdx(vIdx)) = colIdx(starZ(vIdx));
191 pass = assignment(assignment>0);
192 pass(~diag(validMat(assignment>0,pass))) = 0;
193 assignment(assignment>0) = pass;
194 cost = trace(costMat(assignment>0,assignment(assignment>0)));
195

```

```

193 function [minval, rIdx, cIdx]=outerplus(M,x,y)
194 ny=size(M,2);
195 minval=inf;
196 for c=1:ny
197     M(:,c)=M(:,c)-(x+y(c));
198     minval = min(minval, min(M(:,c)));
199 end
200 [rIdx, cIdx]=find(M==minval);

```

## 4.2 Flicker-free phase solution for dynamic holographic traps

A trap array is defined by the profile of the phase mask pattern applied to the LCOS-SLM. Since only phase modulation is allowed, not amplitude modulation, generally there is no exact solution for phase calculation, and iterative methods are widely used instead. GS algorithm is one of typical iterative methods for such phase retrieving in many fields including optical tweezers and x-ray diffraction analysis, because of its high fidelity and efficiency compared to other algorithms [70].

However, when it comes to individual atom transport by sequential phase pattern operation, frame-to-frame intensity flickering matters. This intensity flickering comes from phase jumps in a large number of pixels (details in Chapter 2). Basic GS algorithm does not give frame-to-frame phase proximity so it does not guarantee generation of flicker-free sequential phase patterns. There exist flicker-free solutions at some cost of power efficiency. They also have various computational times required. A comparison of these methods are presented in table 4.1. Among them, *superposition algorithm* was used in this experiment because of its flicker-free characteristic, its better efficiency than *random distribution* and limitation of calculation time.

### Gerchberg-Saxton algorithm

Gerchberg-Saxton (GS) algorithm is a typical and widely used algorithm for phase retrieving. It tries to find a Fourier transform of a complex function, in 2D or other dimensions, when control of the phase part in the Fourier transform is only allowed, and the amplitude part is

Table 4.1: Comparison of phase generating algorithms.

Type	Power per trap	F-to-f proximity	Frame rate (Nvidia GeForce Titan X)
GS	$1/N$	X	$\leq 100Hz$
Random dist.	$1/N^2$	O	$\gg 100Hz$
Superposition	$1/N$	O	$\approx 100Hz$
Inductive GSW	$1/N$	O	$\leq 100Hz$

given fixed. The phase retrieval is achieved, in terms of our purpose, by repeatedly performing forward and inverse Fourier transform and replacing the amplitude, between the fields on the trap domain  $U(x, y) = A(x, y)e^{i\theta(x, y)}$  and the SLM domain  $V(X, Y) = B(X, Y)e^{i\phi(X, Y)}$  as following:

---

```

set  $\theta(x, y) \leftarrow 0$ .
repeat for  $N_{iter}$  times :
     $U(x, y) \leftarrow A_0 e^{i\theta}$ 
     $V(X, Y) = \mathcal{FT}[U(x, y)]$ 
     $V(X, Y) = B_0 e^{i\phi}$ 
     $U(x, y) = \mathcal{IFT}[V(X, Y)]$ 
Retrieve  $\phi(X, Y)$ 

```

---

where  $|A_0|^2$  and  $|B_0|^2$  are respectively the target array intensity and the beam profile on the SLM. The equality mark (“=”) here means substitution of the l.h.s. by the r.h.s., as in computational fields. Usually the resultant trap array profile formed from the retrieved  $\phi(X, Y)$  converges to the target profile as the iteration number increases.

GS algorithm shows good power efficiency of the result so it is good for general purpose of phase retrieval. However it requires quite much amount of computation which comes from the repeated forward and inverse Fourier transforms for real-time operation compared to other non-iterative methods. Moreover, it occurs intensity flickering when used for sequential phase mask operation since frame-to-frame phase proximity is not guaranteed. For transport of atoms, flicker-free phase retrieval algorithms are considered as below.

### Random distribution

Random distribution is a simple method for phase retrieving which is also intuitive. (See Chapter 2, phase section, for details) The resultant phase pattern is given by a mixture of phase patterns for each traps. The individual  $N$  phase patterns  $\phi_n(i)$  (We use an index to represent coordinates here. The index runs along the whole SLM plane and is not restricted to 1D.) are mixed into the resultant phase pattern  $\phi_{mixed}(i)$  as,

$$\phi_{mixed}(i) = \phi_{M(i)}(i) \quad (4.1)$$

where  $M(i)$  is a random matrix which has integers between 1 and  $N$  as its elements. In other words, each value of the mixed phase pattern  $\phi_{mixed}(i)$  is drawn from one of the individual  $\phi_n(i)$ ’s, with which one to draw from randomly. Individual phase patterns can be as Fourier transform of single trap profiles and usually in simple form of linear (in the 2D plane) or additional quadratic (perpendicular shift) phase profiles.

It is a flicker-free solution for atom transport because increment in trap position shift always corresponds to increment of phase difference. Also, it does not demand much calculation time

since it is mostly a set of simple additions. On the other hand, the method has low efficiency which scales as  $1/N^2$  per trap, because in this way the pixels on the SLM are independently decided, failing to exploiting any constructive interference between components from different pixels unlike other efficient methods which have efficiency of  $1/N$ .

### Superposition algorithm

Superposition algorithm is another algorithm that mixes individual phase patterns into a synthesized phase pattern. In this method, the phase value at each pixel in the mixed pattern is obtained as the argument of the complex sum of all the fields from the individual patterns at the same index, i.e.,

$$\phi_{mixed}(i) = \arg \left[ \sum_{n=1}^N e^{i\phi_n(i)} \right] \quad (4.2)$$

Unlike *Random Distribution*, this method allows fields to constructively interfere, so efficiency of this method is better, scaling as  $1/N$ . The increment condition is satisfied, so is flicker-free condition. Computational demand of this method is larger than that of *Random Distribution*, but smaller than that of iterative methods.

However, superposition algorithm fails to generate images with highly periodic geometries. The reason is that a simple Fourier transform of a periodic geometry shows amplitude peaks (as in typical Fourier transform of monotonic signal), and modulation of the SLM finds it hard to meet such a profile requirement only with phase modulation. GS algorithm is made to iteratively converge to the target profile, so the periodicity problem is relieved in this method.

### Inductive GSW

Inductive weighted GS (Inductive GSW) algorithm has two additional manipulation in its procedure, starting from GS algorithm. First is the “weighted” part; to increase the uniformity of the resultant array profile, it performs feedback control to each of the traps every iteration.

$$A_0^{k+1}(x, y)|_{(x,y) \in P} = G^k(x, y)A_0^k(x, y)|_{(x,y) \in P} \quad (4.3)$$

where  $A_0^k$ ,  $P$ , and  $G^k(x, y)$  represent the target amplitude of the array in the  $k$ -th frame, the set of trap points, and the gain value for each point in the  $k$ -th frame. The gain value can be determined as various ways; it can be just the inverse of the previous target amplitudes, times the mean of them, or given by a servo loop output.

Second, the “inductive” part means, in the iteration for calculating the  $(k + 1)$ -th frame  $\phi^{k+1}(X, Y)$ , the initial condition of the iteration is replaced by the resultant phase of the  $k$ -th frame  $\phi^k(X, Y)$ , instead of just setting  $\theta(x, y) = 0$ . By applying this, frame-to-frame phase proximity occurs, giving the sequential phase patterns flicker-free characteristic.

### 4.3 Path planning

Neutral atom arrays in two or three dimensional space may play an important role in quantum information processing (QIP), because of their scalability to a massive number of qubits [29, 72, 75, 76, 77, 78]. Currently, arrays of several hundred atoms have been implemented with optically-addressable spacings of a few  $\mu\text{m}$  [8, 79], and this number is expected to increase to a few thousand as laser power permits. These atoms are confined by an array of optical-dipole traps made through various methods including holographic devices [44], diffractive optical elements [23, 80], micro-lens arrays [81], and optical lattices [82]. Ultimately, neutral-atom platforms for QIP may require (i) a significant number of atoms, (ii) a high-dimensional architecture, preferably with an arbitrary lattice geometry, (iii) single-atom loading per site, and (iv) the ability to be individually addressable. However, no existing method satisfies all these requirements. For example, optical lattices can provide a large number of atoms singly loaded per site through the Mott insulator transition [82], but they have rather limited geometries and often lack individual addressability; other methods have advantages of arbitrary configurations and site addressability but fail the single-atom loading condition due to the collisional blockade effect [52].

In optical-dipole traps, the probability of single-atom trapping per site is about 50 percent. Both the filling factor and the configuration of the entire array are, in consequence, probabilistic. The probability of filling an entire array with  $N$  atoms scales as  $0.5^N$ , which is extremely small for a large  $N$ . Significant efforts are being devoted to achieve a deterministic or near-deterministic single-atom loading; one approach uses an array of bottle-shaped blue-detuned optical well potentials [71], and the others include light-assisted, controlled inelastic collision [73, 74, 65]. The loading probability of defect-free arrays however still remains distant from one, especially when we consider a large number of atoms.

Recently, methods have been devised to achieve defect-free atom arrays at a high probability by filling vacancies with nearby reservoir atoms [66, 67, 36, 61, 83, 84], along with the development of atom transport techniques [48, 50, 85, 45, 46, 86]. In this vacancy-filling scheme, as illustrated in Fig. 4.1, the probability of achieving  $N$  completely filled lattice points is a product of the probability of initially trapping more than or equal to  $N$  atoms and the probability of successful transport of  $N$  atoms to target sites. Since the former is a conditional probability that approaches one as the number of initial traps exceeds  $2N$ , the vacancy-filling of the target sites is mainly governed by the latter, or how  $N$ -atom transport is performed. The shorter the overall travel path of all atoms, the smaller the loss that is given as a function of travel time and distance. Thus, successful transport depends on a “good” atom-guiding plan that minimizes the travel time and distance as well as any lossy transport paths. This is a combinatorial optimization problem, and can be specifically categorized as bipartite matching, for which the solutions can be efficiently found with graph theories such as the Hungarian method, or Hungarian matching algorithm [87].

Designing an optimized path plan for compactifying an imperfect initial array into the target deterministic filled array, there are things are to be considered. One is the amount of required calculation time for computing path plan which is a critical factor regarding the finite trap time. Calculation time should be sufficiently short compared to the trap decay time and it depends much on which algorithm we use to obtain the path plan. Another is the amount of required memory for the calculation. It is a practical limitation which involves the physical memory of the system. Another thing to consider, which is fundamental, is whether the algorithm is complete. In other words, “If there is a solution for the current array situation, does this algorithm always give a solution?”

Regarding the things, we can come up with some kinds of path planning algorithms; heuristic move, backpropagation, atom-target matching (brute force), and rigorous matching algorithm such as the Hungarian algorithm. Comparisons of them in terms of the discussed factors is presented in table 4.3. The path plan algorithm is needed to obtain the optimized efficiency considering passive loss, moving loss depending on the step displacement and travel distance, number of sequence steps, total spent time, etc. Also, algorithms should not be complicated for speedy overall operation. We simulated several methods stated here and decided to employ Hungarian matching algorithm for complete matching with reasonably short calculation time. The algorithms will be described in following in detail.

In the following descriptions, symbols stands;  $N_A$ : number of trapped atoms,  $N_T$ : number of target sites,  $N_U$ : number of total sites,  $d$ : spatial dimension in which traps and atoms exists,  $s$ : allowed number of trial stages.

In this section, we consider the Hungarian matching algorithm as an efficient means to achieve defect-free atomic lattice formation through vacancy-filling. In following subsections, we first compare atom-site matching methods, namely the brute-force and heuristic approaches as well as the Hungarian, to discuss their pros and cons, and then explain how to obtain collision-free paths using the Hungarian algorithm in Sec. 4.4. The experimental procedure of capturing atoms with optical-dipole traps, identifying the vacancies, calculating the optimal path plans accordingly, and finally verifying the filling is described in Sec. 4.5. In Sec. 4.5.2, we present the results of experiments utilizing the optimal path planning before concluding in Sec. 4.6.

Table 4.2: Comparison of path planning algorithms.

Algorithm	Completeness	Calculation time
Heuristic move	Not comp.	Very short
Brute force	Comp.	Very long
Hungarian matching	Comp.	Short

### 4.3.1 Heuristic move

The simplest way of vacancy filling path plan one can think of is to make some heuristic rules and let the atoms move based on it. Since this is not a rigorous way of planning, there could be a range of various strategies. For example, if the target sites are all centered, a trivial and macroscopic, or global rule should be *move all atoms towards the center, if there is any vacancy*, like gravitation does. Once a global rule is set, additional microscopic, or local rules also have to be decided, which could be quite different case by case according to the array geometry. One issue for this kind of rule would be how to select one direction between two possible directions which are equally close to the central direction, i.e., when they are symmetric. Another can be how to distribute spatial atom population if it is localized in some regions and too sparse in another.

This kind of strategy is simple and does not require much computation power and memory. Instead, it is not complete; in some special situations, it might not find any solution for our objective, even if there is. In order to maximize the success rate, we considered some further strategies to overcome the issue.

### 4.3.2 Backpropagation method

One simple way to achieve rigorous solving of vacancy filling is to address all the cases of array status and provide a pre-found solution for it in experiment. Then, how can we figure out the predetermined solutions for all the cases? It can be found out by starting from the final filled array we want to obtain (virtually, in simulation), and addressing all the configurations after moving individual atoms along every possible path set; i.e., we record the path sets for all possible cases of backpropagation of the atoms starting from the final objective configuration and recall the corresponding one in experiment.

This method is rigorous because every single possible configurations is investigated and addressed on the precalculation stage. It guarantees if the queried configuration is not on the address list, there is no solution for it. On the other hand, it clearly requires much of memory space since there are plenty number of cases of possible backpropagation, especially with a large number of traps. It also requires long time when investigating all the possible path sets. Moreover, the time required to query the address on the lookup table during experiment is not short, either.

### 4.3.3 Atom-target site matching (source indexing)

Instead of investigating all possible configurations of atom locations on an array, there exists a less time-demanding, memory-demanding and still rigorous way. The idea is to set our objective to be assigning one atom to every target site, or finding out an one-to-one matching between a set of atoms and all the target sites, with the constraint that the sum of the distances for atoms to move by is minimized.

This can be computationally realized as following. Identify the array status, and for each target site  $g(x_j)$ , record distances from every trapped atom  $a(x_i)$ . This would constitute a *distance matrix*  $d_{ij}$ . Now try to choose one atom from each target site, exclusively. No two target sites could pick the same atom. Calculate the total distance. Repeating this way, find a matching which gives the minimal total distance, investigatively.

The amount of investigation could be much reduced by limiting the allowed atom-site distance, and by sorting the candidate atoms in the order of distance so that the investigation starts from the minimal distances. This *source indexing method* requires much shorter time compared to the *backpropagation method*, but this way of finding the optimal matching still requires somewhat long investigation time and might not be sufficiently efficient for certain real-time experiments.

When we consider the relocation of atoms to transform a partially-filled atomic lattice to a completely-filled one, finding a set of relocation paths can be viewed as the problem to find a match between every target site and a corresponding atom. Although there are a plethora of algorithms to assign matching between the target sites and the same number of atoms, we must consider their operational efficiency in actual experiments. Not only do atoms in optical-dipole traps have a finite trapping time, but they also escape from the traps during transport with a certain probability given as a function of both time and distance. In choosing a specific algorithm, therefore, we need to consider the time and travel distance. The time is the sum of computational time for the matching algorithm, and execution time for the subsequent guiding operation (transport), with the latter closely related to the travel distance. In our case of about half-filled lattices, the travel distance (or the execution time) does not change much for various initial configurations and algorithms; however, the computation time changes significantly depending on the choice of algorithm.

Figure 4.4 compares the computational times of various atom-site matching algorithms. The brute-force algorithm requires a factorial increasing computational time as the size of the target site  $N$  increases, and the Hungarian algorithm scales as  $N^3$  [88]. While the result of the heuristic method (the shortest move method [83]) provides a shorter computational time, the resulting matching is not only sub-optimal but also often involves path collisions (see Sec. 4.4). The pros and cons of these algorithms are summarized in Table 4.3, with the details of each method discussed in the following subsections.

Table 4.3: Comparison of atom-site matching algorithms

Algorithm	Calculation complexity	Rigorosity
Brute-force method	$O(N!)$	yes
Heuristic shortest-move	$O(N^3)$	no
Hungarian matching	$O(N^3)$	yes

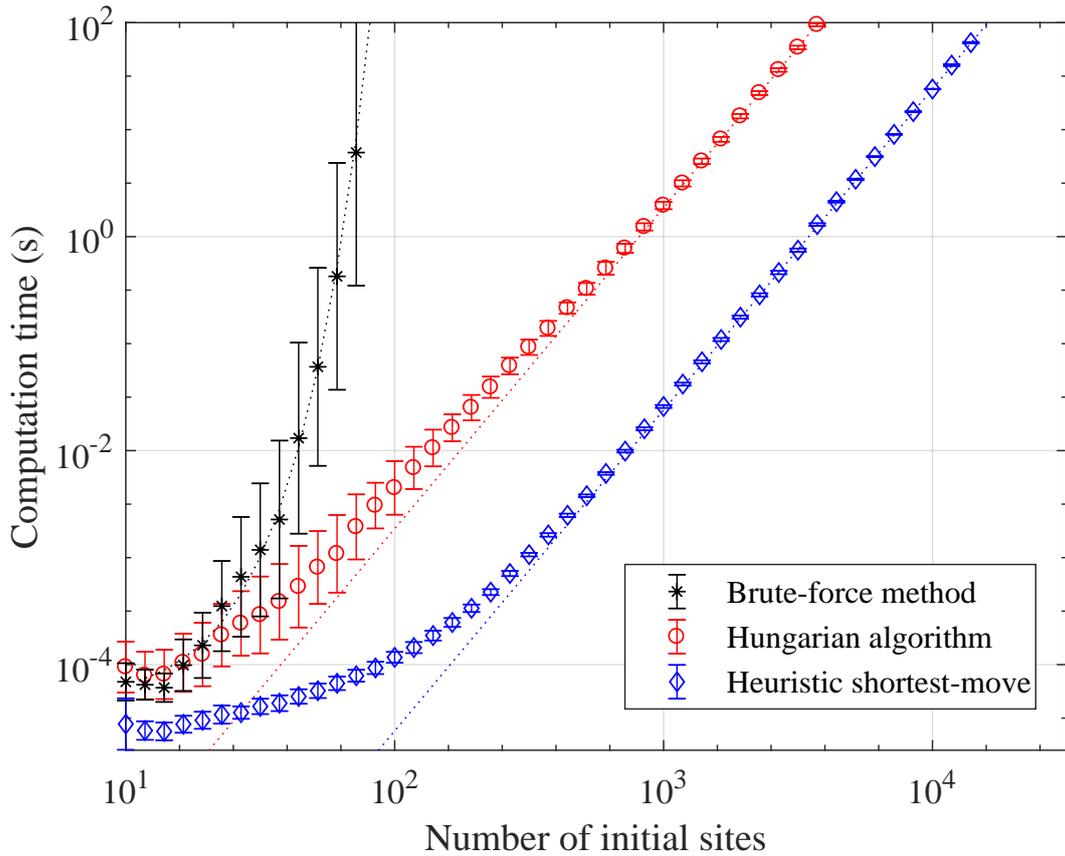


Figure 4.4: Computational time vs. the number of initial sites. The computational time of atom-matching to target sites using the brute-force, heuristic shortest move, and Hungarian algorithms, when the numbers of target sites, atoms and initial sites are given by  $N : N_A : N_i = 1 : 2 : 4$ , respectively. Each errorbar represents the standard deviation [105].

## Brute-force atom-site matching

The brute-force method extensively searches all possible matching solutions; thus, it finds the optimal solution without failure, but in an extremely time-inefficient way. In this method, after identifying the initial configuration of atoms in the lattice, we calculate the distance matrix  $D$ , of which the element  $d_{i,j}$  is the distance between each target site  $t_i$  and the initial position of each trapped atom  $a_j$ . When all the target sites are indexed with  $T = \{t_i | 1 \leq i \leq N\}$  and the positions of the trapped atoms with  $A = \{a_j | 1 \leq j \leq N_A\}$ , the objective is to find a one-to-one matching  $f : T \rightarrow A$  which minimizes the total distance between atoms and target sites, where  $d_{\text{total}} = \sum_i d_{i,f(i)}$  and  $d_{i,f(i)} = |t_i - f(t_i)|$ . All possible subsets of  $A$  of size  $N$  are sequentially selected with all possible permutations inspected. This method ensures the optimal solution (*i.e.*, the one-to-one function with the minimum total distance); however, it requires a tremendous amount of calculation time. As shown in Fig. 4.4, the brute-force calculation time scales factorially as a function of the total number of initial sites,  $N_i$ , and as a result it takes more than an hour for  $N_i = 100$ , which is not practical in our experiments. (The comparative benchmarking of matching computation time in Fig. 4.4 was performed with a MATLAB code and an Intel CPU i5-4670.)

## Heuristic shortest-move matching

Heuristic algorithms can find a solution in a time-efficient manner. One example used in Ref. [83], which may be referred to as heuristic shortest-move matching, finds a solution in such a way that  $N$  smallest elements are sequentially selected from the distance matrix  $D$  with the condition of choosing only one element from each row and column. So, in the distance matrix, this algorithm finds the smallest element  $d_{l,m}$  and assigns  $a_m$  to  $t_l$ , *i.e.*,  $a_m = f(t_l)$ . Then, the  $l$ 'th row and  $m$ 'th column are eliminated from the matrix  $D$  and the process repeats  $N_T$  times until all target sites are assigned to atoms. As shown in Fig. 4.4, the heuristic approach allows fast calculation, with a computational time an order smaller than the Hungarian algorithm for typical cases (30 times faster for  $N_i = 100$ ). This algorithm is fast but not rigorous; this sub-optimality can be improved by using additional restriction rules. Compared with the Hungarian algorithm, which will be introduced in the following subsection, our computer simulation of the heuristic shortest-move method, without additional rules, finds an optimal solution with a 50% chance; the sub-optimal solutions involve either a longer travel distance (14%), atom-atom collision en route (93%), or both (7%).

## Matching with Hungarian algorithm

Finding a path set to fill the vacancies of the target array can be viewed as finding a matching between every target site and a corresponding atom. We considered some matching theories such as Hall's marriage theorem, the Hopcroft-Karp algorithm, and the Hungarian algorithm to achieve complete matching between target sites and atoms.

First, Hall’s marriage theorem, or Hall’s theorem, gives necessary and sufficient condition for the existence of a matching that covers at least one side of a bipartite graph. In our experiment, there is the set of target sites  $T = \{t_i \mid 1 \leq i \leq N_T\}$  and each target site has its candidate atoms  $A_i = \{a_j \mid d(t_i, a_j) \leq \text{range}\}$  which are adjacent to the site itself. Interpreted for our experiment, the theorem tells whether there exists any possible exclusive matching  $M(t_i, a(t_i))$  between each target site and a corresponding atom among its candidates.

The Hopcroft-Karp algorithm finds an actual maximum matching from a bipartite graph. It assigns as many as exclusive atoms for target sites, accomplishing complete filling if the number of matched atoms equals the number of target sites.

The Hungarian algorithm finds a maximum matching in the constraint of minimizing the cost. In our problem, the algorithm finds a matching between atoms and target sites where the cost means the distance from an atom to a target site in this case. A formal description of the mathematical formalism of the algorithm is as following.

- Index all atoms  $\{a(x_i)\}$  and construct a cost matrix, or in this case, distance matrix,  $d_{ij}$  from the atoms to each goal site  $\{g(x_j)\}$ .
- The objective is to find one-to-one matching between a set of goal sites and atoms  $G(U, V, E)$  in which the total distance is minimized.  $U = \{g(x_j)\}$ ,  $V \subset \{a(x_i)\}$ .
- This is an assignment problem which assigns atoms exclusively to each of goal sites, the distances corresponding to the costs. Hungarian algorithm is known to be an efficient algorithm to solve with time complexity of  $n^3$ , where dimension of the cost matrix is  $n \times n$ .

The detailed procedure of the algorithm to solve the problem is beyond the topic of the thesis, so is not to be discussed here. The calculation of path planning using this algorithm was done by an open source computer program.

#### 4.3.4 Hungarian matching algorithm

Graph theories, such as Hall’s marriage theorem, the Hopcroft-Karp algorithm, and the Hungarian algorithm, provide useful theoretical backgrounds to achieve a fast and rigorous matching between target sites and atoms. Hall’s marriage theorem [89], or Hall’s theorem, provides the necessary and sufficient condition for the existence of a matching  $M$  that covers at least one side of a bipartite graph  $G(U, V; E)$ , where  $U$  and  $V$  are two finite sets, and  $E$  is the set of edges that connect  $U$  and  $V$ . In the current work, we consider  $U = T$  and  $V = A$ , and this theorem tells whether there exists in  $G$  any possible exclusive matching between each target site and a corresponding atom among all trapped atoms. The Hopcroft-Karp algorithm [90] finds an actual matching  $M$  that allows the maximal one-to-one connection between  $U$  and  $V$ , from a given bipartite graph  $G$ . When all elements in  $U = T$  are one-to-one connected to  $V = A$ , in

other words maximal matching, the complete filling of the target sites in our case is possible. This theorem however only finds possible matching, without considering distance minimization.

As total distance minimization is necessary, we focus on the Hungarian matching algorithm, which can use cost functions when finding a maximal matching  $M$  in  $G$  [87]. The Hungarian method efficiently finds the maximal matching with a time complexity of  $N^3$  for an  $N \times N$  cost matrix, when the constraint is given to minimize the cost function. Our Monte Carlo simulation using the total travel distance as the cost function shows the same scaling behavior of computational time as in Fig. 4.4. Furthermore, some modifications to the original Hungarian algorithm can significantly reduce the calculation time, either by employing a sparse-matrix Hungarian algorithm or by using the sub-domains of trapped atom sites to apply the algorithm to each domain (a divide-and-conquer approach).

#### 4.4 Collision-free property of matching with Hungarian algorithm

Examples of actual atom-guiding plans obtained with the heuristic shortest-move and Hungarian algorithms are shown in Fig. 4.5. The initial configuration is a 7-by-7 square lattice ( $N_i = 49$ ) randomly occupied by  $N_A = 21$  atoms, as in Fig. 4.5(a), where filled circles represent the initial atoms and unfilled circles the vacancies in the 3-by-3 target lattice ( $N = 9$ ). The result of the heuristic shortest-move method without any additional rule is shown in Fig. 4.5(b). However, some guiding paths cross each other or trespass on existing atoms (orange dotted circles), which leads to possible atom loss or improper guiding due to the merging of optical-dipole traps en route.

The Hungarian matching algorithm in Fig. 4.5(c), on the other hand, intrinsically shows no path crossing. This is because the matching with path crossing gives a bigger travel distance than the corresponding collision-free matching that swaps the targets, and the Hungarian algorithm minimizes the total distance. However, trespassing still remains, as shown with the dotted circle in Fig. 4.5(c). In order to avoid such trespassing, we can employ an alternative cost matrix  $D$ , for example, with a modified distance metric  $d_{i,j}^\alpha$ . With the modified distance metric, trespassing is avoided when  $\alpha > 1$ . If, for example  $\alpha = 2$ , since the matching  $A \rightarrow B$ ,  $B \rightarrow C$  (“relaying path”) in Fig. 4.5(d) gives lower cost ( $1^2 + 1^2 = 2$ ) than  $A \rightarrow C$ ,  $B \rightarrow B$  (trespass) in Fig. 4.5(c) ( $2^2 + 0^2 = 4$ ).

A similar principle can also apply to “nearly trespassing paths” where, for instance, atom  $B$  is near the  $A \rightarrow C$  path. Since atom traps have finite sizes in space, by avoiding the atoms which are too close, atom loss could be reduced. In a similar manner to the trespassing case, a relaying path is chosen when  $\alpha > \alpha_c$ , in which the minimum interatomic distance is increased. Sufficient  $\alpha_c$  can vary according to the array configuration. For the square lattice in our case, it is found that  $\alpha > 1.12$  ensures the minimum interatomic distance of  $1/\sqrt{2}$ , as follows. We consider nearly trespassing configurations that involve the minimum interatomic distance, in

which  $(0, 0) \rightarrow (1, l)$  and  $(0, 1) \rightarrow (0, 1)$  is the nearly trespassing path (see Fig. 4.6(a)). The condition for choosing the relaying path (Fig. 4.6(b)) is  $1^\alpha + (\sqrt{1 + (l - 1)^2})^\alpha < (\sqrt{1 + l^2})^\alpha$ . For  $l = 1$ , a nearly trespassing path is allowable because the minimum distance in this case is  $1/\sqrt{2}$ , which is sufficiently larger than the trap size. For  $l = 2$ ,  $\alpha_c \approx 1.12$ , and as  $\alpha_c$  has smaller values for larger  $l$ 's,  $\alpha > \alpha_c$  ensures the minimum distance not to be smaller than  $1/\sqrt{2}$ , which is the condition for collision-free matching.

## 4.5 Experiment

### 4.5.1 Procedure

The experimental setup, similar to what is described in our earlier work [36, 61], includes a magneto-optical trap (MOT) for cold rubidium atoms ( $^{87}\text{Rb}$ ), a dipole-trapping laser beam programmable with a 2D spatial light modulator (SLM, Meadowlarks XY spatial light modulator,  $512 \times 512$  pixels, 200 Hz frame rate) in the Fourier domain, a single-atom imaging system with an electron multiplying charge-coupled device (EMCCD) and a high numerical aperture lens ( $\text{NA} = 0.5$ ), and a computing system that calculates possible atom-relocation paths. Atoms were first cooled and trapped in the MOT which took 0.5 seconds. Simultaneously, the dipole-trapping beams were turned on to prepare an initial array of atoms that were probabilistically loaded in the collisional blockade regime [52], with a filling factor of about 50 percent. Then, the imaging system read out the filling and vacancy configuration of the initial atom array, and the computing system calculated an atom-transport path plan to a completely-filled smaller-size lattice. The matching algorithm, such as the Hungarian algorithm, was used at this stage. Once the atom guide plan was finalized, all the atoms to be relocated were simultaneously transported, while the mask pattern for the SLM was calculated in real time, which was accelerated with a graphic processing unit (GPU, Nvidia Titan X). For hologram generation, we used a modified GS (Gerchberg-Saxton) algorithm [91]. When the first trial of atom reconfiguration was completed, the actual array configuration was confirmed through a second readout. If the configuration was incomplete due to moving or collision loss during the operation, the whole process was repeated until a defect-free array was achieved. The whole experiment was performed in a closed feedback loop with up to nine iterations within the trap lifetime of  $\tau = 18$  s.

### 4.5.2 Result

#### Circular target with phase generation comparison

The experimental sequence was mainly iterations of vacancy filling operation. From an initial MOT, atoms were loaded to an initial array of  $N_{res}$  traps. Trapped atoms were identified and array reconfiguration for vacancy filling was performed. This operation was iterated for sequential stages  $N_f$  at certain time period of  $T$ .

Fig. 4.7 shows several trial stages of vacancy filling with experimental parameter sequence

on the top, and selected captured atom images, each followed by atom number histogram in the final target array. After the initial trapping, the experimental parameters were repeated for times. The leftmost column of the atom images shows the initial status of the array and the other columns show the images after vacancy filling operations at corresponding trial stages  $N_f$ . The upper demonstration collects the final array of  $N_T = 10$  with a reservoir of  $N_{res} = 20$ . The lower one collects the final array of  $N_T = 15$  with  $N_{res} = 30$ . In both demonstrations, the highest probabilities, 63% and 43% respectively, of defect-free array were acquired after two trials, which are over  $\times 600$  and  $\times 14000$  times of the natural trapping scheme of 0.5 per site. The result shows possibility of preparation of filled single atom arrays for experiments on single atom arrays in a reasonable number of trials.

Fig. 4.8 shows atom loss rate per a single transport step for various types of phase generation method. The superposition algorithm shows better performance than GS algorithm because of GS algorithm's undermined frame-to-frame continuity. The superposition algorithm also outperforms random distribution algorithm

### Square target with matching algorithm comparison

Experimental demonstration of our defect-free atom-lattice formation using the Hungarian algorithm is shown in Fig. 4.9. Representative atom lattice images at various stages are shown in Fig. 4.9(a-d). The initial configuration was a partially filled 7-by-7 square lattice, having three vacancies in the central 3-by-3 target zone, as shown in Fig. 4.9(a). As indicated by the five arrows, the neighboring atoms were simultaneously moved to construct a completely-filled central lattice. However, as the images in Fig. 4.9(b) and (c) show, some atoms in the target lattice disappeared during transport due to time-dependent atom loss. To fill the vacancies, neighboring atoms were additionally moved along the paths indicated with arrows, until a completely filled 3-by-3 lattice was achieved, as shown in Fig. 4.9(d). The atom-site matching in each stage and the corresponding guiding paths were obtained using the Hungarian algorithm with  $\alpha = 1.5$ . The success probability  $P_s$ , defined as the number of successful events (achieving the defect-free 3-by-3 target lattice) divided by the total number of events (250), increased from about  $0.5^9 \simeq 0.2\%$  in the initial configuration, to 24% after the first relocation, then 50% after the second relocation, and ultimately 61% after the ninth relocation.

Figure 4.10 compares the experimental success probabilities of the Hungarian algorithm with  $\alpha = 0.5, 1.5$  and 3 with that of the heuristic shortest-move matching. The experimental data (circles) shows that the success probability to achieve a defect-free array is notably bigger when the Hungarian matching algorithm with either  $\alpha = 1.5$  or  $\alpha = 3$  is employed rather than the heuristic shortest-move algorithm or Hungarian with  $\alpha = 0.5$ . This result is in good agreement with the analysis in Sec. 4.4, where it was predicted that the former cases are collision-free but the latter cases are not. Note that the success probability  $P_s$  first increases as a function of the stage number, but decreases in the end, which is attributed to the fact that the longer travel distance required for the later stages brings about bigger losses. In the experiment, each

atom move between sites was divided into  $N_{\text{frame}} = 15$  segmented moves, and each segmented move was driven by the SLM frame evolution between two stationary frames. The atom survival probability in each segmented move can be modeled as  $P = P_{\text{time}}P_{\text{moving}}P_{\text{cross}}$ , where  $P_{\text{time}} = e^{-t/\tau}$  is the survival probability against the time-dependent loss due to background gas collision, with  $\tau$  the trap lifetime,  $P_{\text{moving}} = e^{-\beta N_{\text{frame}}d^2}$  is the survival probability against the moving loss due to intensity flickering of the optical dipole traps [61], with  $\beta$  the moving loss coefficient and  $d$  the travel distance, and  $P_{\text{cross}} = 1 - e^{-\gamma d_{\text{min}}^2}$  is the survival probability against the loss due to path collisions. In Fig. 4.10, the numerical simulation using the above models (dotted lines) for each relocation stage are shown. The fitted parameters obtained through curve fitting are given by  $\tau = 18$  s,  $\beta = 0.0079/a^2$ , and  $\gamma = 84/a^2$ , where  $a$  is the lattice constant. Each data point is statistically averaged over 250 events, where the errorbar represents the standard error.

Finally, Fig. 4.11 shows a few examples of atom arrays formed by the Hungarian matching algorithm. In each of the demonstrations, the upper images show examples of the random initial configurations with  $N \sim 100$  initial sites, and the lower images show the final configurations following atom relocation. Each image was from a single shot. Success rates were 30.6%, 32.6%, and 19.1%, respectively. For array-formation of various geometries of target sites, the Hungarian algorithm performed best.

## 4.6 Conclusion

Three methods of vacancy site filling have been compared. The advantage of Hungarian matching over the brute-force method is clear because the calculation time of the latter greatly exceeds the former as the number of atoms increases. The heuristic shortest-move method seemingly has an advantage in short calculation time, but the issue of path collisions becomes serious, in particular when the vacancy occurs in the central region of the target lattice. It is concluded that the Hungarian matching method has at least three advantages over the heuristic shortest-move method: it provides rigorous solutions, high success probabilities, and advantages in atom vacancy healing cases, where the second and third advantages are attributed to the collision-free path planning of the Hungarian algorithm.

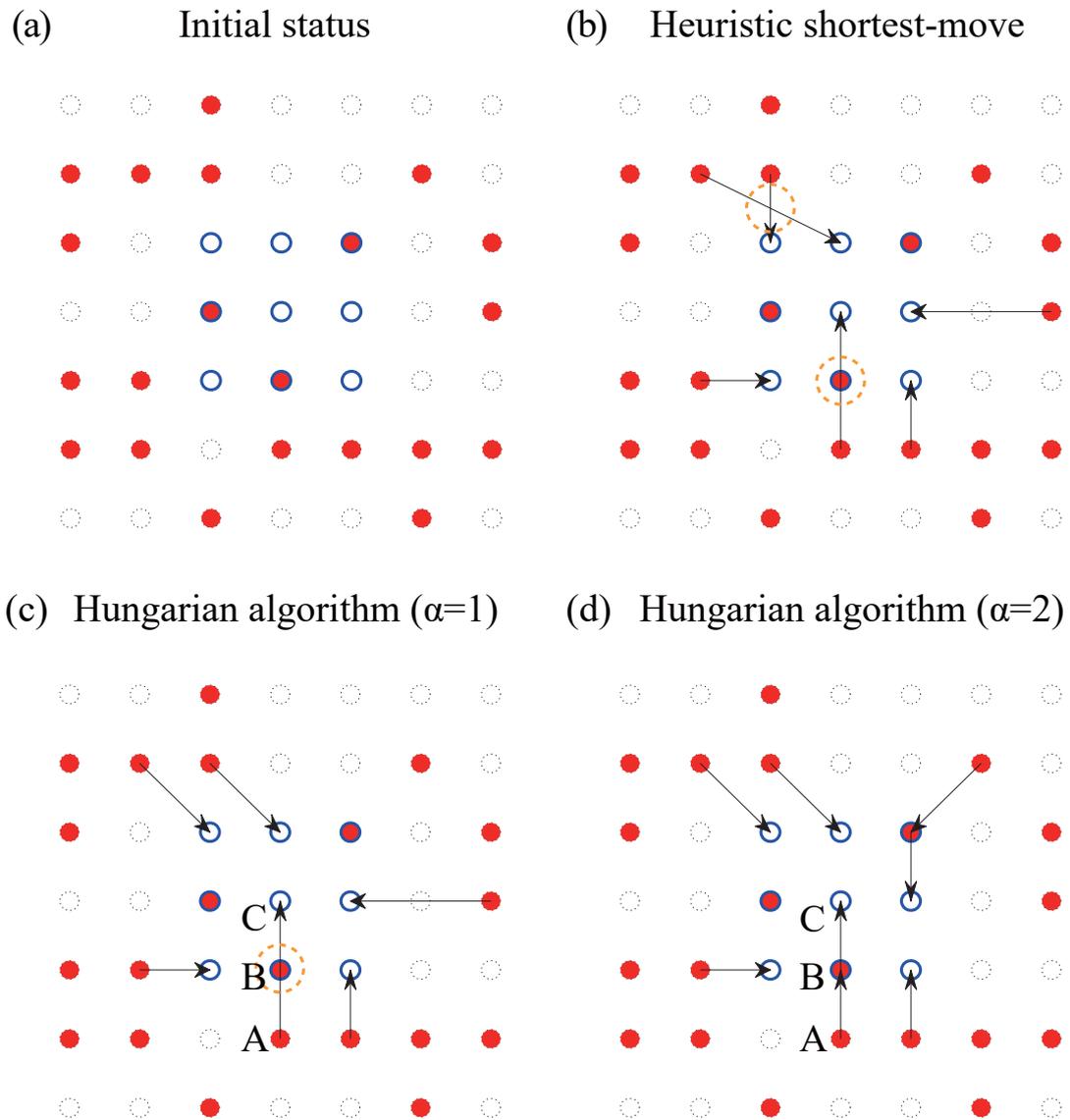


Figure 4.5: Visualization of move solutions from (a) the 7-by-7 initial array, by (b) shortest-move matching algorithm, and (c-d) Hungarian algorithm matching with  $\alpha = 1$  and 2, respectively, to the central 3-by-3 target array. The orange dotted circles show overlapping of the paths and trespassing of atom sites.

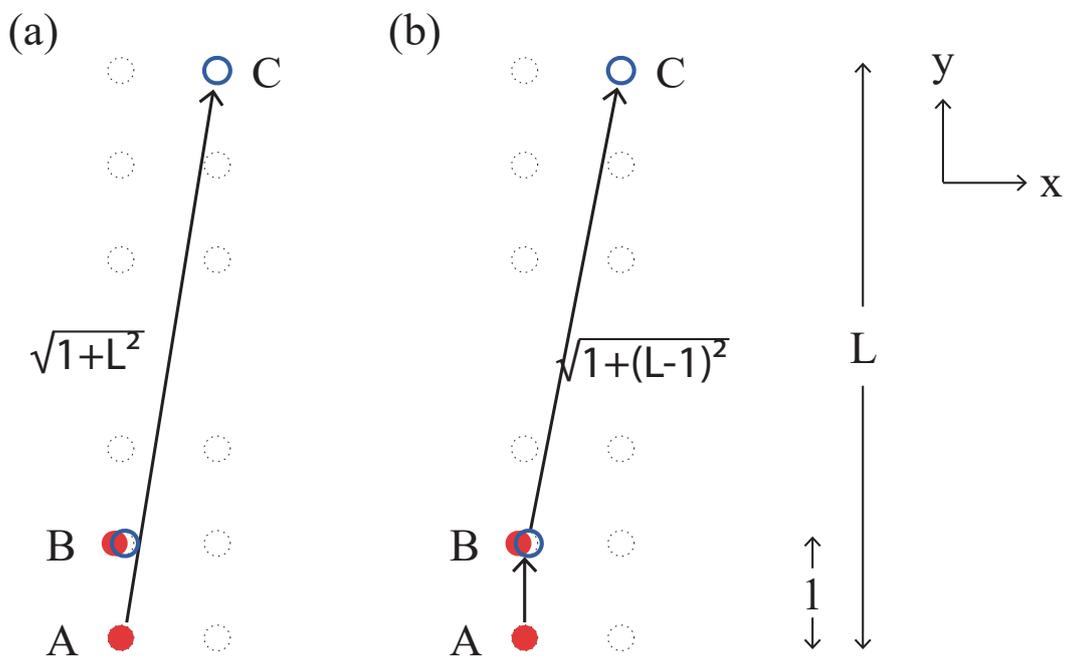


Figure 4.6: (a) "Nearly trespassing path" and (b) "Relaying path" for a configuration with two atoms and two targets in a 1-by- $L$  lattice.

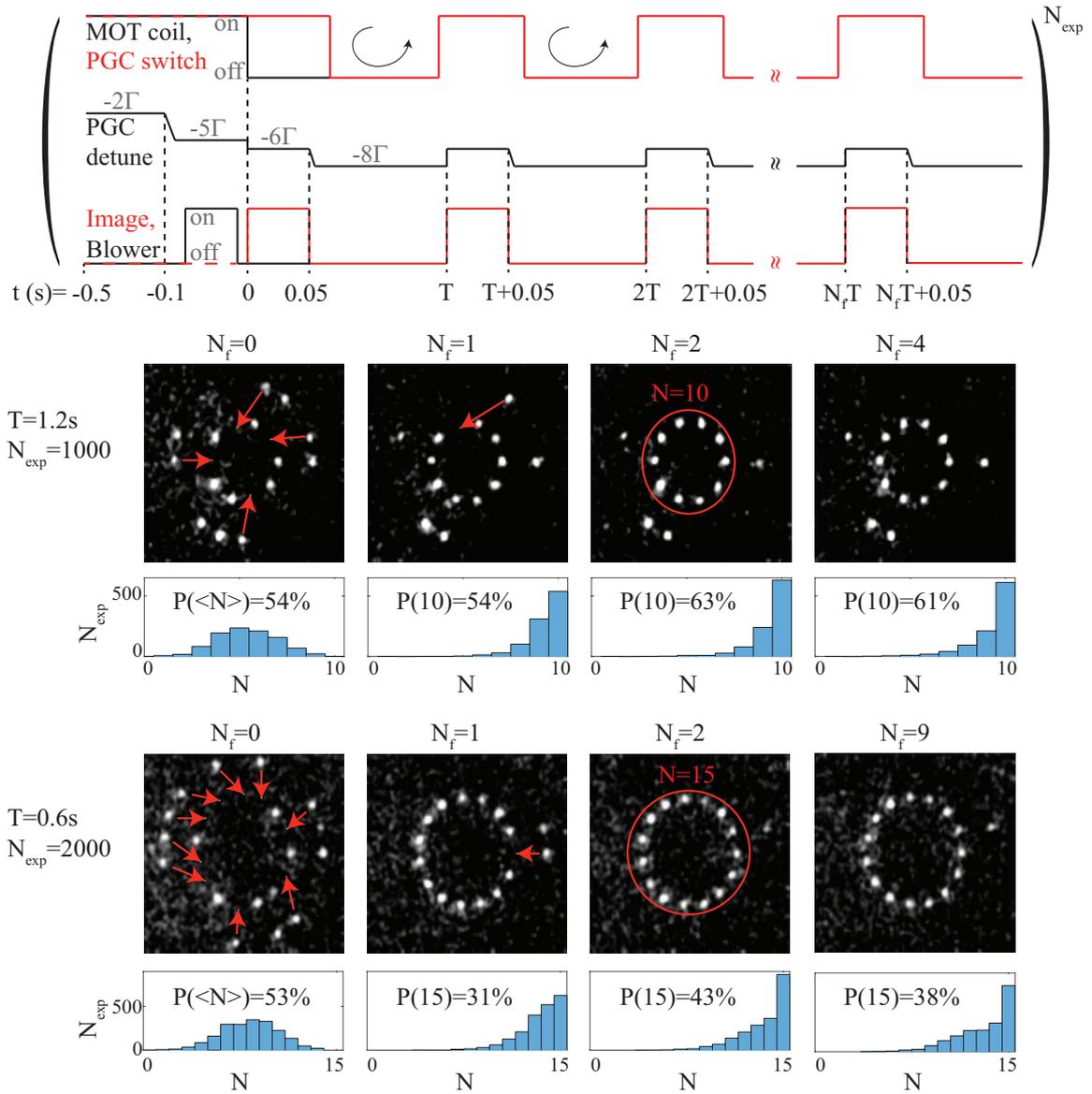


Figure 4.7: Deterministic circular chain arrays. Several trial stages of vacancy filling with experimental parameter sequence and selected captured atom images are shown, each followed by atom number histogram in the final target array.

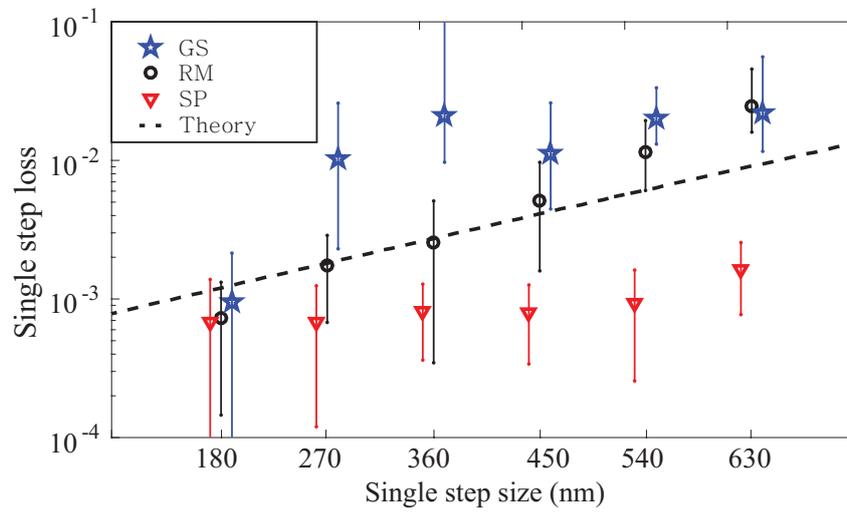


Figure 4.8: Single step loss for GS, Random Distribution, Superposition (SP), and theory.

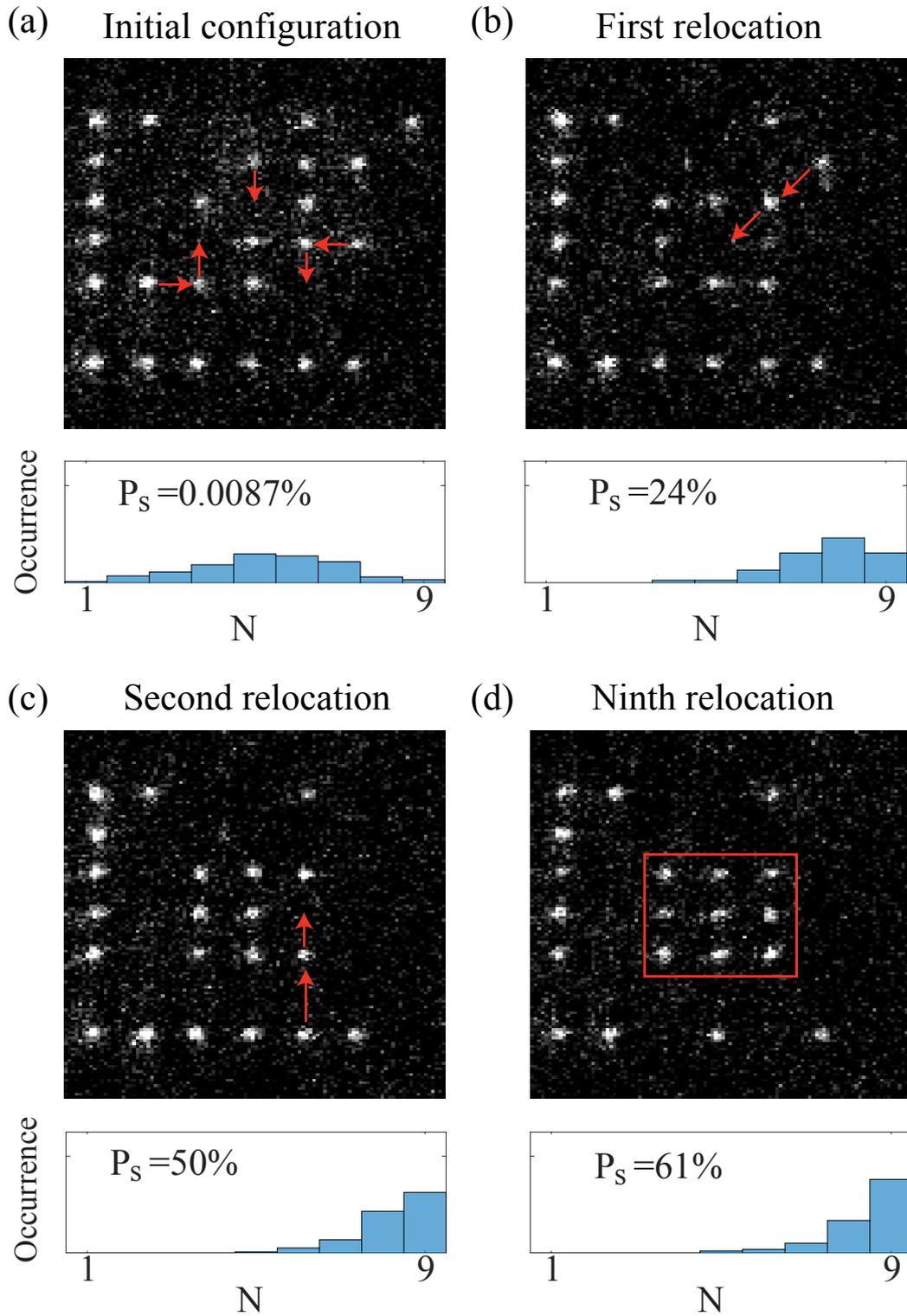


Figure 4.9: Experimental examples of the formation of a 3-by-3 atom array from a partially filled 7-by-7 array using the Hungarian matching algorithm, where  $P_s$  is the success probability of achieving a filled target lattice. Histograms of the atom number in the target lattice are shown below from a total of 250 events [105].

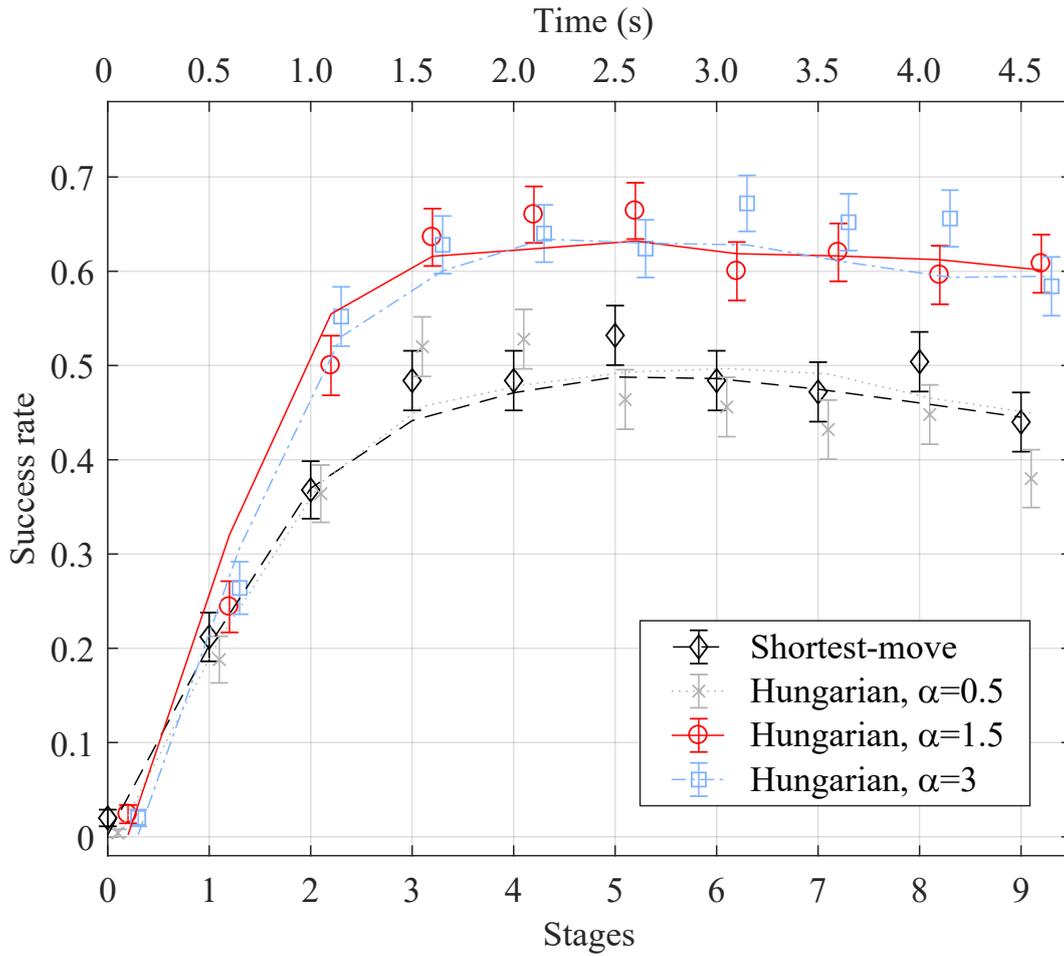


Figure 4.10: Success rate comparison for shortest-move and Hungarian algorithm matching with various  $\alpha$  values in the 7-by-7 lattice for the target 3-by-3 lattice in the central region. The circles and errorbars correspond to the experimental data and the dotted lines show the simulation results [105].

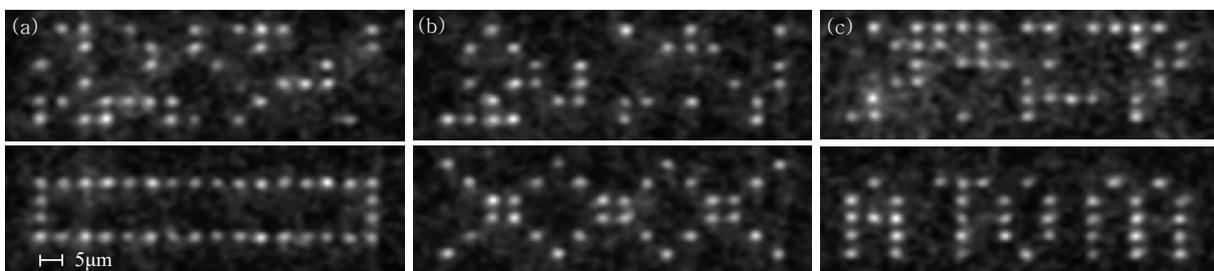


Figure 4.11: Examples of defect-free atomic array formation: (a) a rectangular ring, (b) a triple X, and (c) the capital letters of the word “atom” with upper and lower images showing the initial and final configurations, respectively [105].

## Chapter 5. N-atom interacting system with Rydberg blockade

### 5.1 Introduction

Since evidence of quantum entanglement was discovered and advocating theories was made, there has been demonstrations on quantum entanglement in many kinds, in forms of photon-photon, photon-atom, etc. One of well-known interactions between atoms is dipole interaction, which casts us possibility of creating and controlling atom-atom entanglement via a phenomenon called dipole blockade, or in Rydberg state case, Rydberg blockade [9, 25, 93, 94, 95, 96, 97].

Let us consider dipole-dipole interaction between adjacent two atoms [30, 92]. The leading term in dipole-dipole interaction hamiltonian is expressed as,

$$V(\vec{r}_1, \vec{r}_2) = (1 - 3\cos^2\theta_{12})\frac{d_1d_2}{R^3} \quad (5.1)$$

where  $\theta_{ij}$  is the angle between the interatomic axis and the quantization axis of the atoms,  $d_i$  and  $d_j$  are the electric dipole operators of the atoms, and  $R$  is the distance between them. For a two-atom Rydberg state  $|rr\rangle$ , where  $|r\rangle = |nd\rangle$ , the largest contribution to the energy shift caused by the dipole interaction comes from the state  $|r'r''\rangle$ , where  $|r'\rangle = |(n+1)p\rangle$  and  $|r''\rangle = |(n-1)f\rangle$ . With these two atom-atom states as basis, the interaction Hamiltonian becomes in the form of,

$$H = \begin{pmatrix} 0 & \frac{C}{R^3} \\ \frac{C}{R^3} & \delta_F \end{pmatrix} \quad (5.2)$$

where  $C$  is a constant proportionanl to the dipole moments of the atoms and  $\delta_F$  is the energy difference between two states, called the Förster defect [30].  $R$  is the distance between the two atoms. As shown in Sec. 1.4, the energy shift is,

$$\Delta V_{int}(R) = \pm \frac{C^2}{4\delta_F R^6}, \quad (5.3)$$

which will be denoted simply as  $C_6/R^6$ .

Let us now consider two levels of an atom, the ground state  $|g\rangle$  and the Rydberg state  $|r\rangle$  with Rabi frequency  $\Omega$ . Now two atoms are apart by a distance at which van der Waals interaction occurs with basis of  $|gg\rangle$ ,  $|gr\rangle$ ,  $|rg\rangle$ ,  $|rr\rangle$ . In this case, the Hamiltonian is,

$$\begin{aligned} H &= \frac{\Omega}{2}(|g\rangle\langle r| \otimes \mathbf{1} + \mathbf{1} \otimes |r\rangle\langle g| + H.c.) - \frac{C_6}{R^6} |rr\rangle\langle rr| \\ &= \frac{\Omega}{2}(|gg\rangle\langle gr| + |gg\rangle\langle rg| + |gr\rangle\langle rr| + |rg\rangle\langle rr| + H.c.) - \frac{C_6}{R^6} |rr\rangle\langle rr|. \end{aligned} \quad (5.4)$$

Note that the state  $|-\rangle = (|gr\rangle - |rg\rangle)/\sqrt{2}$  is decoupled from the system. Then we can use new basis  $|gg\rangle$ ,  $|+\rangle = (|gr\rangle + |rg\rangle)/\sqrt{2}$ , and  $|rr\rangle$ . Then the Hamiltonian becomes,

$$H = \frac{\sqrt{2}\Omega}{2}(|gg\rangle\langle +| + |+\rangle\langle rr| + H.c.) - \frac{C_6}{R^6}|rr\rangle\langle rr| \quad (5.5)$$

Compared to single atom Rabi oscillation, the Rabi oscillation frequency between  $|gg\rangle$  state and an entangled state  $|+\rangle$  state turns out to be enhanced by  $\sqrt{2}$ . In strong interaction regime where  $\frac{|C_6|}{R^6} \ll \Omega$ ,  $|rr\rangle$  state decouples from the dynamics, which is called Rydberg blockade. Two 67S Rydberg atoms apart from each other by  $6 \mu\text{m}$ , for example, have interaction energy of about  $2\pi \times 10 \text{ MHz}$ , which is 10 times of single atom Rabi frequency if the frequency is  $2\pi \times 1 \text{ MHz}$ . Due to its strong interaction and versatility of single atom dipole trapping, Rydberg blockade is studied as a promising way of interaction in neutral atom quantum computing [98, ?, 100] and quantum simulation [101, 102, 103, 104].

## 5.2 Experimental setup

The experimental setup mostly follows previous experiments, except for Rydberg excitation lasers and a set of Helmholtz coils for assignment of quantization axis. The interatomic distance was around  $5 \mu\text{m}$ , considering the magnitude of the interaction energy between them. The schematic diagram is as in Fig. 5.1 and the details will be described below.

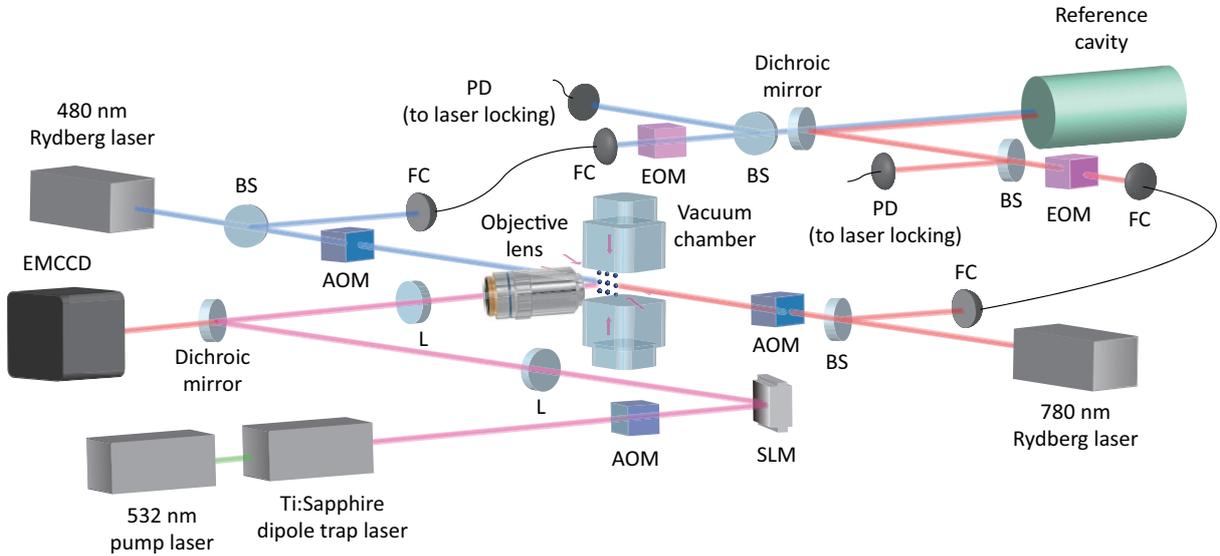


Figure 5.1: Schematic diagram of Rydberg atom experiment setup.

### 5.2.1 780 nm laser and 480 nm laser

Since two-photon excitation from the ground state to the Rydberg state requires the linewidths of the lasers less than MHz, frequency locking of the 780 nm and 480 nm lasers was achieved by locking to an ultra-low expansion (ULE) reference cavity.

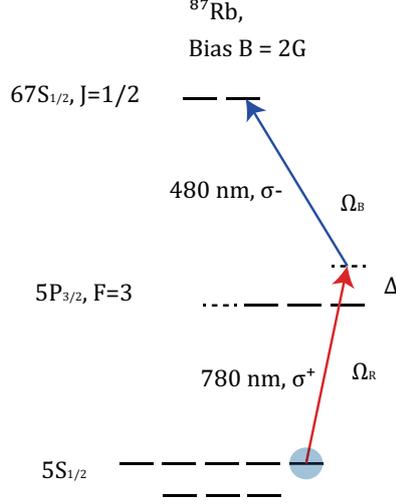


Figure 5.2: Energy level diagram for excitation of the atoms. The atoms initially in  $|g\rangle = |5S_{1/2}\rangle$  state are excited to  $|r\rangle = |67S_{1/2}\rangle$  Rydberg state by non-resonant two-photon transition through  $|5P_{3/2}\rangle$  state.

The 780 nm and 480 nm Rydberg excitation lasers (Toptica diode laser and Toptica DL pro, respectively) were locked to an ULE reference cavity. The detuning of the two lasers from the intermediate state was about  $\Delta \approx 660$  MHz. If one of the transmission peaks the laser through the ULE cavity approximately coincides the desired offset frequency, the laser frequency can be locked with the transmission signal along with typical locking devices. If any of transmission peaks are not in adjacency of the desired frequency, additional parts, like an EOM, for sideband generation so that one of the sideband matches the frequency of the cavity transmission peak.

### 5.2.2 Quantization Axis

Since we expect the atoms to be excited from a well-defined hyperfine state, the  $m_F$  states needs to be well-defined with a certain quantization axis. This quantization axis for magnetic states can be defined by applying some bias magnetic field along the direction of excitation beam axis. The magnitude of the bias field was 3.2 G.

## 5.3 Experiment procedure

### 5.3.1 Atom Trap and Optical Pumping

The method to trap single atoms was as in the previous Chapters. Atoms were trapped in two holographically generated microtraps and their separation was  $5 \mu\text{m}$ . There could be four kinds of trap status after the initial trapping: a) both traps are occupied, b) trap A is occupied, c) trap B is occupied, d) none is occupied. The results for a), b), and c) are recorded and analyzed for comparison of single atom Rydberg rabi oscillation and two-atom collective

oscillation.

Two-photon Rydberg excitation goes from the ground state  $|5S_{1/2}, F = 2, m_F = 2\rangle$ , through the intermediate state  $|5P_{3/2}, F = 3, m_F = 3\rangle$  to the Rydberg state  $|67S_{1/2}, m_J = 1/2\rangle$ . The initial state  $|5S_{1/2}, F = 2, m_F = 2\rangle$  therefore needs to be prepared after the first readout of the atoms. This is realized by the MOT cooling beam  $|5S_{1/2}, F = 2\rangle \rightarrow |5P_{3/2}, F = 3\rangle$  with angular momentum  $m_F = +1$  which makes  $|5S_{1/2}, F = 2\rangle$  a dark state. Then our desired initial state becomes dark state and so the initial population is prepared.

### 5.3.2 Rydberg Excitation

The Rydberg transition from  $|5S_{1/2}, F = 2, m_F = 2\rangle$  through  $|5P_{3/2}, F = 3, m_F = 3\rangle$  to  $|67S_{1/2}, m_J = 1/2\rangle$  states is excited by two laser beams in 780 nm and 480 nm. After the first readout and optical pumping, simultaneous shining of two excitation beams occurs Rydberg rabi oscillation of trapped atoms. For accurate control of excitation timing, 480 nm laser shines for a longer time window than 780 nm laser and control of excitation time is adjusted by 780 nm laser time window.

### 5.3.3 State measurement by recapture method

After excitation beams are turned off, quantum population is measured by recapture method. When the trap beam is turned on again after the excitation finishes, it only traps atoms in the ground state and the atoms in the excited state are not captured again. So when we image the atom array, only the atoms in ground state appear. For example, by measuring a single atom in a resonance light, one can see Rabi oscillation between  $|c_g(t)|^2$  and  $|c_e(t)|^2$  of the atom system  $|\phi(t)\rangle = c_g(t) |5S_{1/2}\rangle + c_e(t) |67S_{1/2}\rangle$  by measuring the recapture rate in pulse duration time  $t$ .

## 5.4 Rydberg blockade and collective excitation

Two holographic traps trapped atoms, the first readout verified which of the traps were filled, then the initial state was prepared by optical pumping, and 780 nm and 480 nm excitation pulses were applied along two trap sites for various pulse lengths. After applying the excitation pulses, the second readout was carried out. The final population of two atoms was measured, in terms of the excitation pulse length, through the recapture method. Figure 5.3 shows the evidence of atom-atom Rydberg entanglement in our experiment. One, two, and three atoms were excited respectively, with all atoms in the blockade radius in each set.  $|\uparrow\rangle = |r\rangle = |67S_{1/2}\rangle$  and  $|\downarrow\rangle = |g\rangle = |5S_{1/2}\rangle$ .

Due to Rydberg blockade, any double-excitation state population is near zero as expected (not shown). Comparing the three plots where one, two and three atoms are involved in each dynamics, Rabi oscillation frequency is enhanced by  $\sqrt{N}$ , by theory, when  $N$  qubits are max-

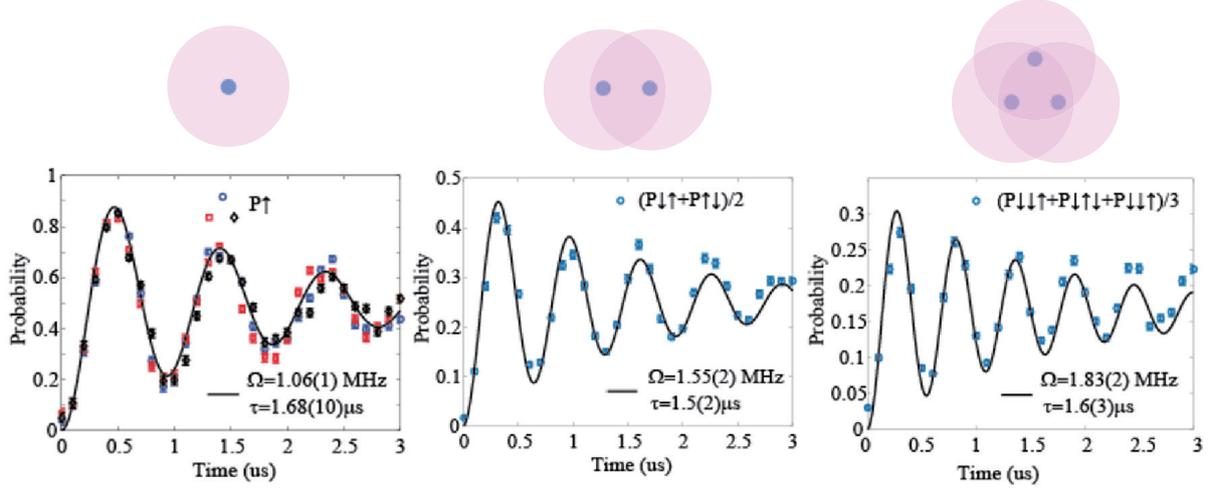


Figure 5.3: Evidence of atom-atom Rydberg entanglement in our experiment. One, two, and three atoms were excited respectively, with all atoms in the blockade radius in each set.  $|\uparrow\rangle = |r\rangle = |67S_{1/2}\rangle$  and  $|\downarrow\rangle = |g\rangle = |5S_{1/2}\rangle$ .

imally entangled. The result shows the enhancement of 1.46, which is close to  $\sqrt{2}$ , and 1.73, close to  $\sqrt{3}$  that we expect from two atom entanglement, suggesting that entanglement actually occurred in this experiment.

## 5.5 Dynamics of $N = 3 \sim 5$ atoms with environmental errors

Dipole-dipole interaction between neighboring Rydberg atoms shifts the energy of the double excitation state, preventing excitation of an atom during the other is excited, which, for example, can be implemented as a C-NOT gate. However, quantum evolution of neutral atoms is still vulnerable to the environment which makes quantum computation using Rydberg atoms practically hard.

Here, we investigate and analyze the effect of the environment in a dynamics of  $N=3\sim 5$  Rydberg atoms by fitting simulation with the environmental effect to the data, including stochastic atom loss, spontaneous emission from the intermediate state and the Rydberg state, and the phase noise of the excitation lasers.

### 5.5.1 Main results

#### Hamiltonian of the dynamics

We regard the atomic system, under atom-light interaction, as an effective two-level system. The Rabi-oscillating single atom Hamiltonian is,

$$H_o = \frac{\hbar\Omega}{2}\sigma_x + \frac{\hbar\Delta_r}{2}\sigma_z, \quad (5.6)$$

with the Rabi frequency  $\Omega$  and the detuning  $\Delta_r$ .

We examine atom-atom interaction by Rydberg blockade in atom chains, which embeds 1d Ising-like model,

$$H = \sum_{1 < j < N} \left\{ \frac{\hbar\Omega}{2} \sigma_x^j + \frac{\hbar\Delta_r}{2} \sigma_z^j \right\} + \sum_{k < l} V_{kl} n_k n_l \quad (5.7)$$

for  $N$  atoms with interaction energy of  $V_{kj}$  between each pair of atoms.  $n_k \in \{0, 1\}$  means if the atom is excited.

## Experimental setup and procedure

Here we briefly introduce the experimental setup and procedure (the schematic setup diagram is as Fig. 5.1; detailed explanation is in Sec. 5.5.2). From an  $^{87}\text{Rb}$  atom cloud trapped in a MOT (magneto-optical trap), single atoms are trapped in far-off resonance dipole traps in 820 nm wavelength. To make a vacancy-free single atom array, trapped atoms are gathered into the target array site by single atom reconfiguration technique [105, 106]. The atoms were located in either linear or zig-zag configurations, for  $N = 3 \sim 5$  atoms, and the fluorescence of the atoms was captured by an EMCCD (electron multiplying charge-coupled device). After optical pumping, the atoms initially in  $|g\rangle = |5S_{1/2}\rangle$  state were excited to  $|r\rangle = |67S_{1/2}\rangle$  Rydberg state by non-resonant two-photon transition through  $|5P_{3/2}\rangle$  state with various pulse areas to track the quantum evolution of the system (Fig. 5.2). The detuning from the intermediate state was  $\Delta \approx 660$  MHz. Projection measurement was conducted to distinguish whether each atom was in  $|g\rangle$  or  $|r\rangle$  state, by checking whether it was recaptured or not after applying a beam nearly resonant to  $|5P_{3/2}\rangle$  and EMCCD capture, since the trap wavelength was designed to trap atoms in  $|g\rangle$  state while not to trap the atoms in  $|r\rangle$ . The dynamics data was obtained by varying the evolution time from 0 to 3  $\mu\text{s}$ , by 0.1  $\mu\text{s}$ , total 31 time steps.

## Environmental error consideration

Among many factors that causes decoherence on the dynamics, we focus on main four factors that affect the most; stochastic atom loss due to background collision and atom temperature (depopulation), spontaneous emission from the Rydberg state to the ground state (depopulation), fast spontaneous emission from the leakage to the intermediate 5P state to the ground state (dephasing,  $\sim 20$  kHz), and finally the phase noise of the 780 nm and 480 nm excitation lasers.

### 1. Stochastic atom loss

During quantum evolution of the atoms, stochastic loss of atoms occurs due to collision with a background hot atom, excessive portion of its kinetic energy distribution, or any environmental condition that results in failure in recapturing. The probability that an atom that is trapped in the measurement before evolution is not trapped in the afterward measurement  $p(r|g)$  can be derived from the experimental data at time-zero (false negative for ground state population). For a three-atom experiment, we can obtain  $P(r|g) =$

$\{P_{ggr}(t=0) + P_{grg}(t=0) + P_{rgg}(t=0)\}/3$  (the two-atom-loss case was neglected). The probability is applied to the simulation by replacing diagonal density matrix elements at each temporal step. For example,  $P'_{ggg} = \{1 - p(r|g)\}^3 P_{ggg}$ ,  $P'_{ggr} = \{1 - p(r|g)\}^2 P_{ggr} + p(r|g)P_{ggg}$ ,  $P'_{grg} = \{1 - p(r|g)\}^2 P_{grg} + p(r|g)P_{ggg}$ , and so on.

2. Spontaneous emission from the Rydberg state to the ground state

Rydberg state atoms have lifetime of  $\sim 100 \mu\text{s}$  for  $n \sim 60$  states and the decay can cause measurement error  $p(g|r)$  which is opposite of the atom loss error. The probability is given as  $p(g|r) = 1 - e^{-t_{recap}/\tau}$  for recapture time  $t_{recap}$  and the atom lifetime  $\tau$ . In our experiment, the trap beam is turned off for  $t_{recap} \approx 3.5 \mu\text{s}$  and the lifetime of the Rydberg 67S state is  $\tau \approx 140 \mu\text{s}$ . Similarly to atom loss error,  $p(g|r)$  is applied to the simulation in the way that  $P'_{rrr} = \{1 - p(g|r)\}^3 P_{rrr}$ ,  $P'_{rrg} = \{1 - p(g|r)\}^2 P_{rrg} + p(g|r)P_{rrr}$ , and so on.

3. Fast spontaneous emission from the leakage to the intermediate 5P state to the ground state

Despite the large detuning of the two-photon transition to the intermediate  $|5P_{3/2}\rangle$  state, a small portion of leakage to this state exists and it causes dephasing in evolution, although depopulation is negligible. We use Lindblad equation to count the non-unitary characteristic of the experimental system [107, 108, 109].

$$\frac{d}{dt}\rho = -\frac{i}{\hbar}[H, \rho] + L\rho L^\dagger - \frac{1}{2}\{L^\dagger L, \rho\} \quad (5.8)$$

In our case of quantum evolution, leakage to the intermediate  $|5P_{3/2}\rangle$  state, from Rydberg excitation, followed by rapid decay ( $\sim 30 \text{ ns}$ ) to the ground state, causes unintended non-unitary evolution. We suppose the following form of the Lindblad operator,

$$L = \begin{pmatrix} \sqrt{\gamma/2} & 0 \\ 0 & -\sqrt{\gamma/2} \end{pmatrix} \quad (5.9)$$

with a dephasing rate  $\gamma$ . The rate  $\gamma$  is obtained when reducing the two-photon excitation in a single atom, which is a three-level system of Hamiltonian, into an effective two-level system per atom, and numerically comparing the rates from the two types of expression. The three-level system evolves with Hamiltonian

$$H_{three-level} = \begin{pmatrix} 0 & \Omega_B & 0 \\ \Omega_B & \Delta & \Omega_R \\ 0 & \Omega_R & 0 \end{pmatrix} \quad (5.10)$$

and Lindblad operator for the spontaneous emission

$$L_{three-level} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \sqrt{\Gamma} & 0 \end{pmatrix} \quad (5.11)$$

where  $\Gamma \approx 6 \text{ MHz}$  is the spontaneous emission rate from  $|5P_{3/2}\rangle$  to  $|5S_{1/2}\rangle$  in Rb atoms. Numerical simulation of a single atom dynamics for  $3 \mu\text{s}$  with the two-level system Hamiltonian 5.6 with  $\delta\Delta_r = 0$  and two-level Lindblad operator 5.9 fitted with the three-level system Hamiltonian in Eq. (5.10) and three-level Lindblad operator in Eq. (5.11) gives  $\gamma \approx 2\pi \times 20 \text{ kHz}$ .

#### 4. Phase noise of the excitation lasers

A laser has its intrinsic phase noise and the phase noise affects quantum evolution of the dynamics. Since direct measurement of the laser phase noise is not possible in our setup, a power spectral density of frequency noise  $S_\nu(f)$  from PDH locking electronics is obtained to convert to phase noise  $\phi(t)$  [110]. The measured  $S_\nu(f)$  is shown in Fig. 5.4. The power spectral density of phase noise  $S_\phi(f)$  is obtained as

$$S_\phi(f) = S_\nu(f)/f^2. \quad (5.12)$$

The phase noise is given, from  $S_\phi(f)$ ,

$$\phi(t) = \sum_f 2\sqrt{S_\phi(f)}\cos(2\pi ft + \phi_f)\sqrt{\Delta f} \quad (5.13)$$

with some randomly assigned phase  $\phi_f$  of which the actual value is unknown. The phase noise of 780 nm and 480 nm laser, respectively,  $\phi_{780}(t)$  and  $\phi_{480}(t)$ , is applied to the simulated quantum evolution replacing  $\Omega$  by  $\Omega e^{i\{\phi_{780}(t)+\phi_{480}(t)\}}$  in Eq. 5.7. Since it contains random phases, the evolution is averaged over 100 times to give a simulation result. As in the result in later sections, the phase noise of the excitation lasers turns out to be the largest dephasing source in our system.

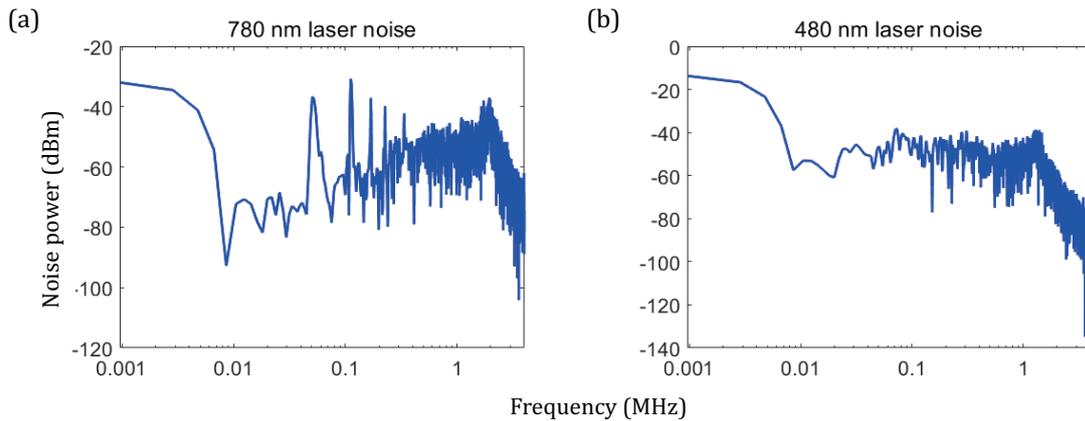


Figure 5.4: The Power spectral density of frequency noise  $S_\nu(f)$  from PDH error signal of the (a) 780 nm and (b) 480 nm laser obtained by a spectrum analyzer from PDH locking electronics.

## Fitting parameters by scanning

To fit simulation to the experimental data, we use the R-squared value between simulation and experimental data as the goodness measure of fitting. We scan two parameters,  $\Omega$ , and  $\Delta S$ , where  $\Delta S$  is a free parameter which is an offset to  $S_\nu$  (i.e.,  $S'_\nu = S_\nu + \Delta S$ ), since a measured  $S_\nu$  is a relative value of which the offset value is unknown. Other parameters are fixed to the estimated values.  $\gamma$  is chosen as the value estimated in the above subsection.  $\delta\Omega$ , the fluctuation of  $\Omega$  is measured to be  $\sim 1\%$  which is negligible in the evolution.  $\Delta_r$  and  $\delta\Delta_r$ , the fluctuation  $\Delta_r$ , is less than 200 kHz, which can be neglected, in our spectroscopy. The interaction strength comes from the known values [111] and  $P(r|g)$  and  $P(g|r)$  comes from the measured data, as stated in the previous subsection. The scanning and fixed values are summarized in Table. 5.1, where the approximated values of  $\Omega$  and  $\Delta S$  appears in the following results.

Total six configurations are used;  $N = 3 \sim 5$  atom in both linear and zig-zag geometry. The Rydberg blockade radius is about  $8.8 \mu\text{m}$  and the lattice constant is  $6.1 \mu\text{m}$ . Figure. 5.5 shows the fitting result for three atoms in the linear configuration. The schematic geometry and the image of the atoms are shown in fig. 5.5(a) and the scan result is presented in fig. 5.5(b). The scan range is  $2\pi \times 0.87 \sim 1.17$  MHz, by  $2\pi \times 50$  kHz step, total 7 values, for  $\Omega$ , and  $2\pi \times 1.405 \sim 1.705$  MHz, by  $2\pi \times 50$  kHz step, total 7 values, for  $\Delta S$ , which makes total 49 sets of paramters. The range is chosen heuristically by pre-scanning within the possible range. There exists a local maximum at  $(\Omega, \Delta S) = 2\pi \times (0.97, 1.07)$  MHz in the scanning range in which the  $R^2$  value exceeds 0.85 (red circle). The fitting curves at this point are drawn along with the experimental data in fig. 5.5(c), for symmetric basis states (“0” in the basis indicates  $|g\rangle$  state and “1”,  $|r\rangle$  state). The curves with different values of  $\Delta S$ , apart from the maximal  $R^2$  by 2 steps (gray circles in fig. 5.5(b)), are also drawn for comparison. The effect of difference in  $\Omega$  is thought to be rather trivial so curves of different  $\Omega$  are not presented. The simulation curve of the maximal  $R^2$  appears highly fitted to the experimental data, in comparison to the curves with other paramters that do not fit well at all.

The same procedure is applied to the data of rest configurations, for the symmetric basis state for each configuration. Likewise, the simulation curves with the parameters of the maximum  $R^2$  fit well to the experimental data, while other curves do not.

### 5.5.2 Detailed implementation

#### Experimental layout

1. Single atom configuration generation

A cold atom cloud of  $^{87}\text{Rb}$  was formed in magneto-optical trap (MOT). The temperature was estimated  $< 70 \mu\text{K}$ . Stray B-field was compensated by hyperfine states spectroscopy in the ground state. Stray E-field was shielded by grounded electrodes placed around the chamber. 780 nm and 480 nm excitation lasers were frequency stabilized to  $< 30$  kHz by an ultralow expansion cavity. After MOT was made, a far-off-resonance trap beam

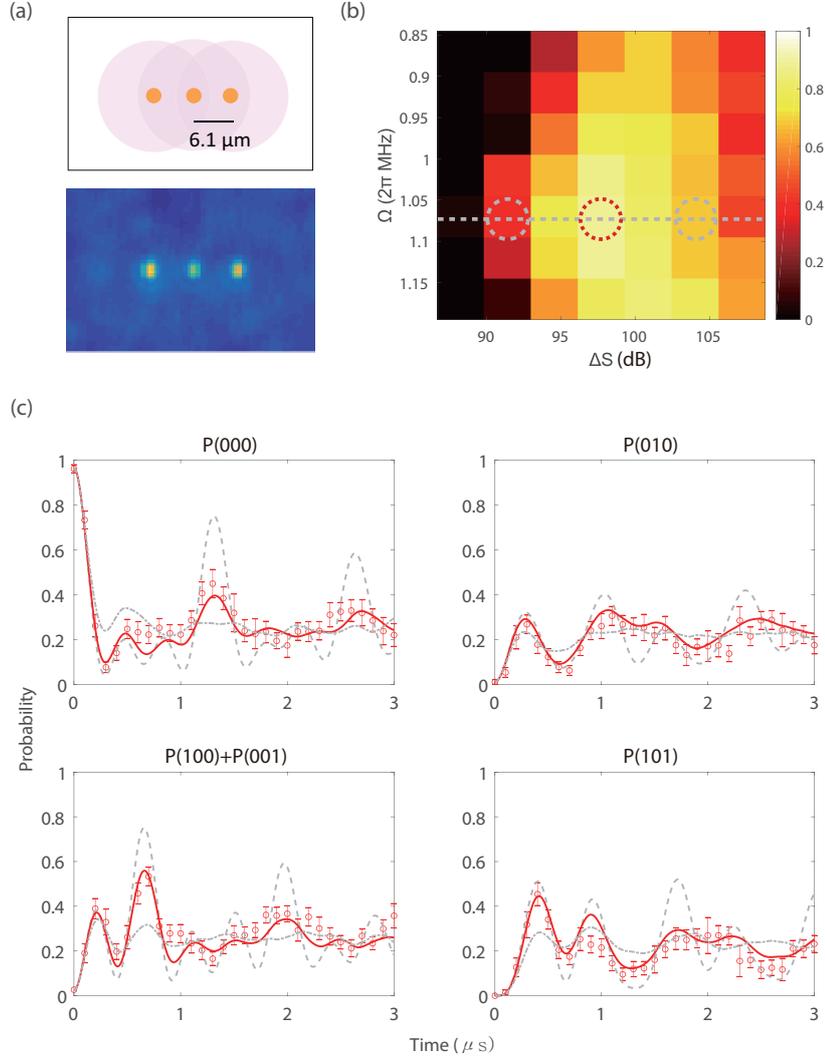


Figure 5.5: (a) Geometry of  $N=3$  atoms in linear configuration. (b) Scan of  $R^2$  between simulation and experimental data of time-sampled projection measurement with two free parameters  $\Delta S$  and  $\Omega$  in simulation. The local maximum existed (red dashed circle). The curves in the points of other  $\Delta S$  values (gray dashed circles) were also examined in (c) for comparison. (c) Experimental data (red circles) and simulation (red curves) with the best fitting parameters, for symmetric basis states, along with non-best fit simulation with difference in  $\Delta S$  by  $\pm 3.1$  dB. “0” and “1” indicate  $|g\rangle$  and  $|r\rangle$  states, respectively.

Table 5.1: Fitting and fixed parameters in simulation

Scanning parameters		
Param.	Value	Description
$\Omega$	$\approx 1$ MHz	Rabi frequency of single atom
$\Delta S$	$\approx 100$ dB	Frequency noise level offset
Fixed parameters		
Param.	Value	Description
$\gamma$	$2\pi \times 20$ kHz	Dephasing due to spontaneous emission from 5P state
$\delta\Omega$	0 kHz	Shot-to-shot fluctuation width of Rabi frequency
$\Delta_r$	0 kHz	Detuning in two-photon transition
$\delta\Delta_r$	0 kHz	(Shot-to-shot) fluctuation width of detuning
$V = C_6/R^6$	$(520 \text{ GHz}/\mu\text{m}^6)/R^6$	Interaction strength (67S state) [111]
$P(r g)$	$\approx 0.01$	False negative for $ g\rangle$ (e.g. stochastic atom loss)
$P(g r)$	$\approx 0.03$	False positive for $ g\rangle$ (e.g. spontaneous emission from 67S)

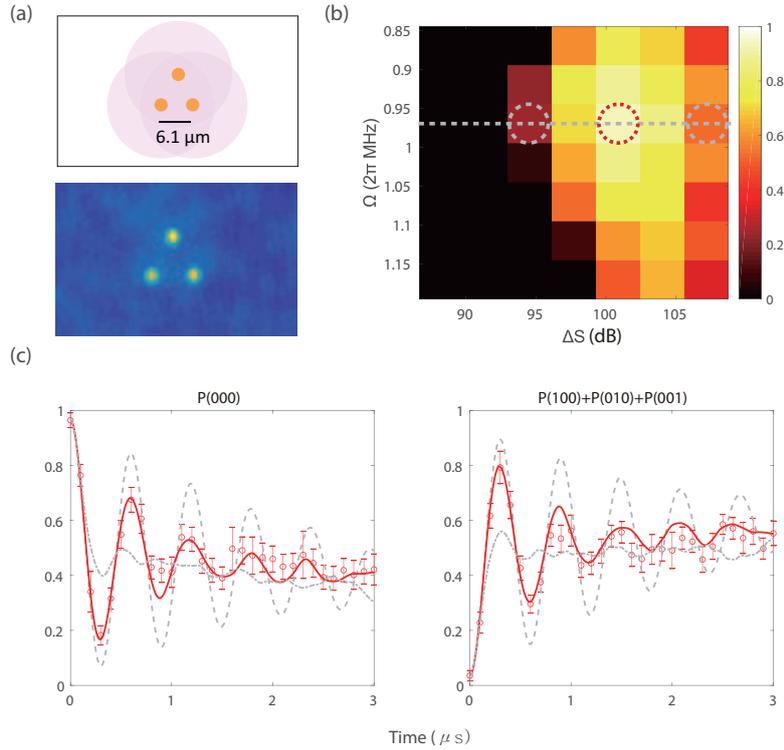


Figure 5.6: (a) Geometry of the atoms, (b) Scan of  $R^2$  between simulation and experimental data, and (c) Experimental data (red circles) and simulation (red curves) with the best fitting parameters, for  $N = 3$  atoms in triangle configuration

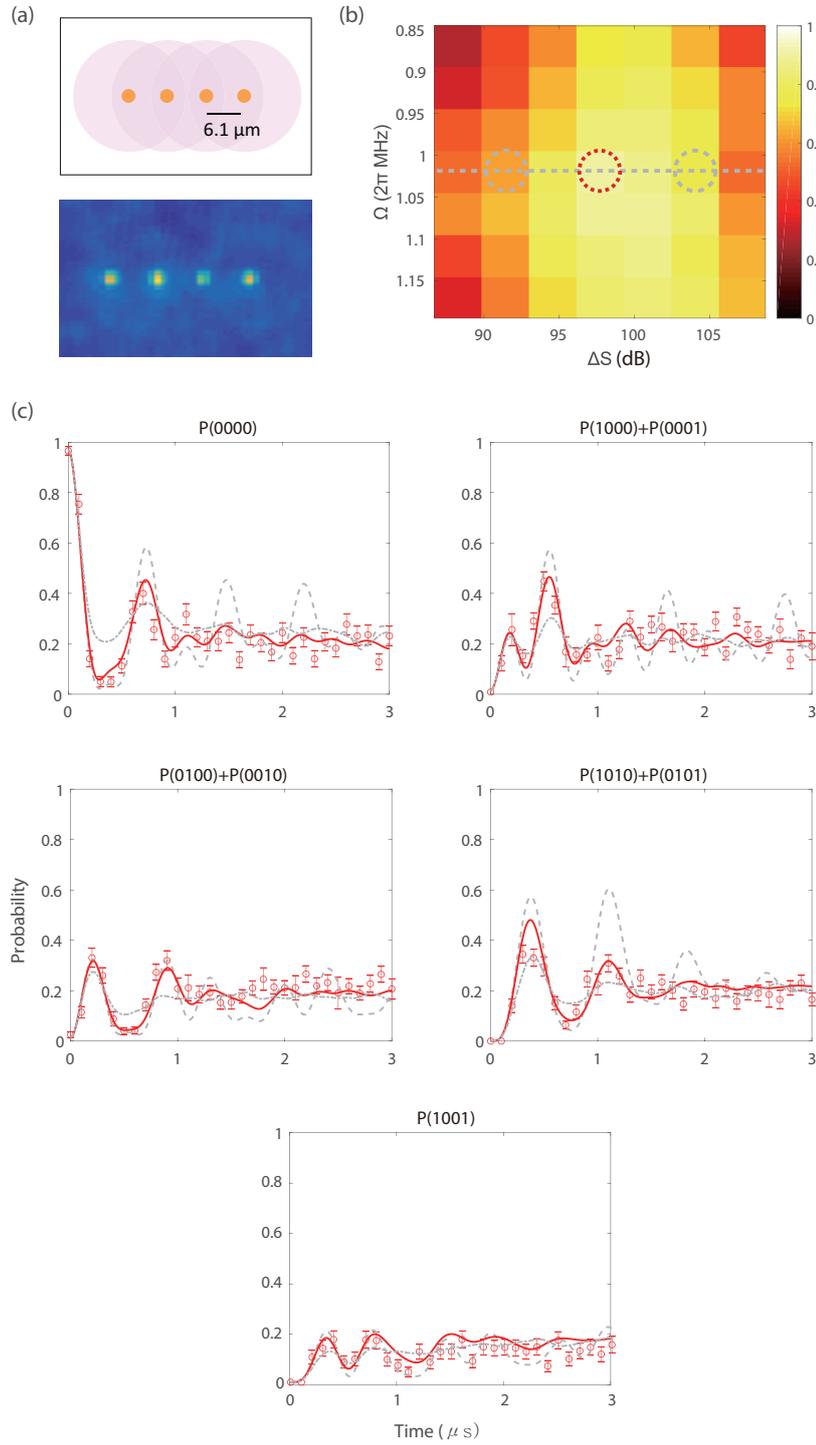


Figure 5.7: (a) Geometry of the atoms, (b) Scan of  $R^2$  between simulation and experimental data, and (c) Experimental data (red circles) and simulation (red curves) with the best fitting parameters, for  $N = 4$  atoms in linear configuration.

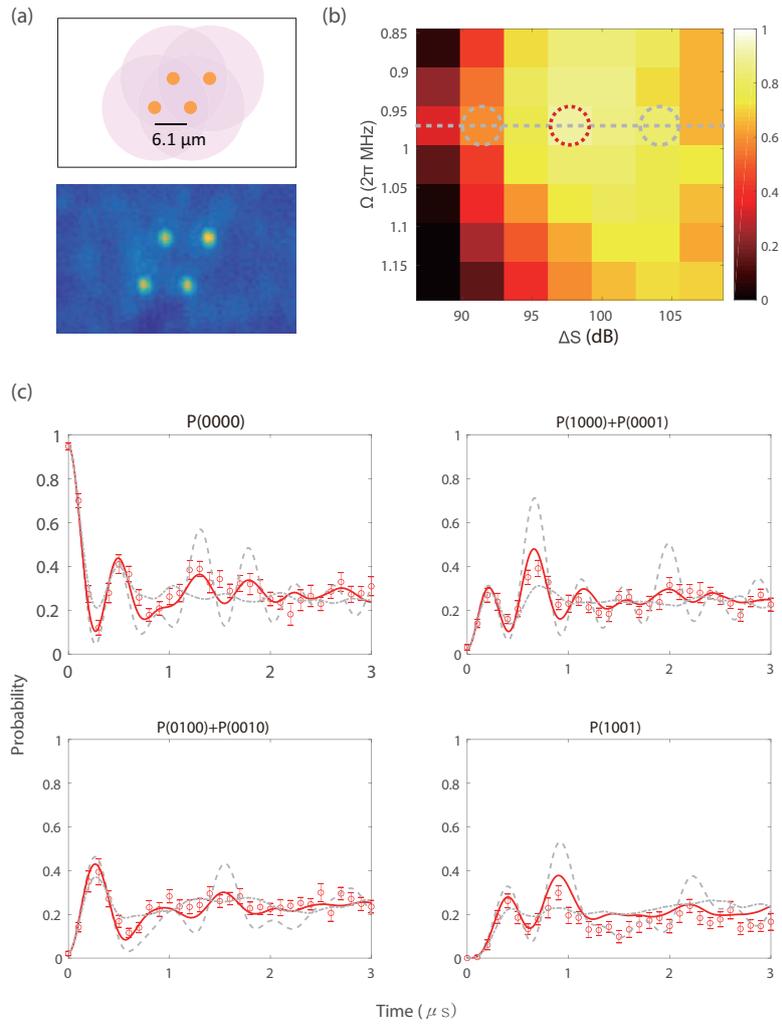


Figure 5.8: (a) Geometry of the atoms, (b) Scan of  $R^2$  between simulation and experimental data, and (c) Experimental data (red circles) and simulation (red curves) with the best fitting parameters, for  $N = 4$  atoms in zig-zag configuration.

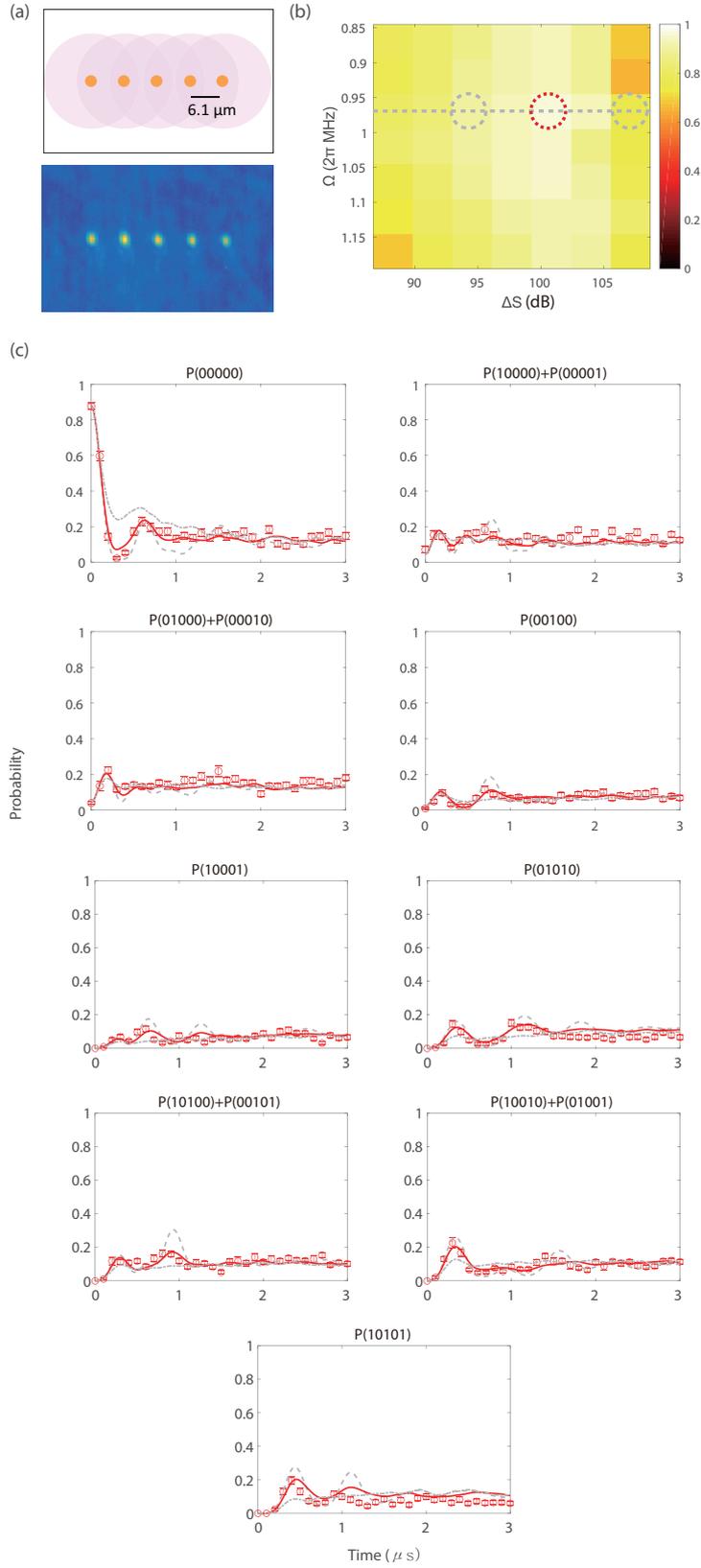


Figure 5.9: (a) Geometry of the atoms, (b) Scan of  $R^2$  between simulation and experimental data, and (c) Experimental data (red circles) and simulation (red curves) with the best fitting parameters, for  $N = 5$  atoms in linear configuration.

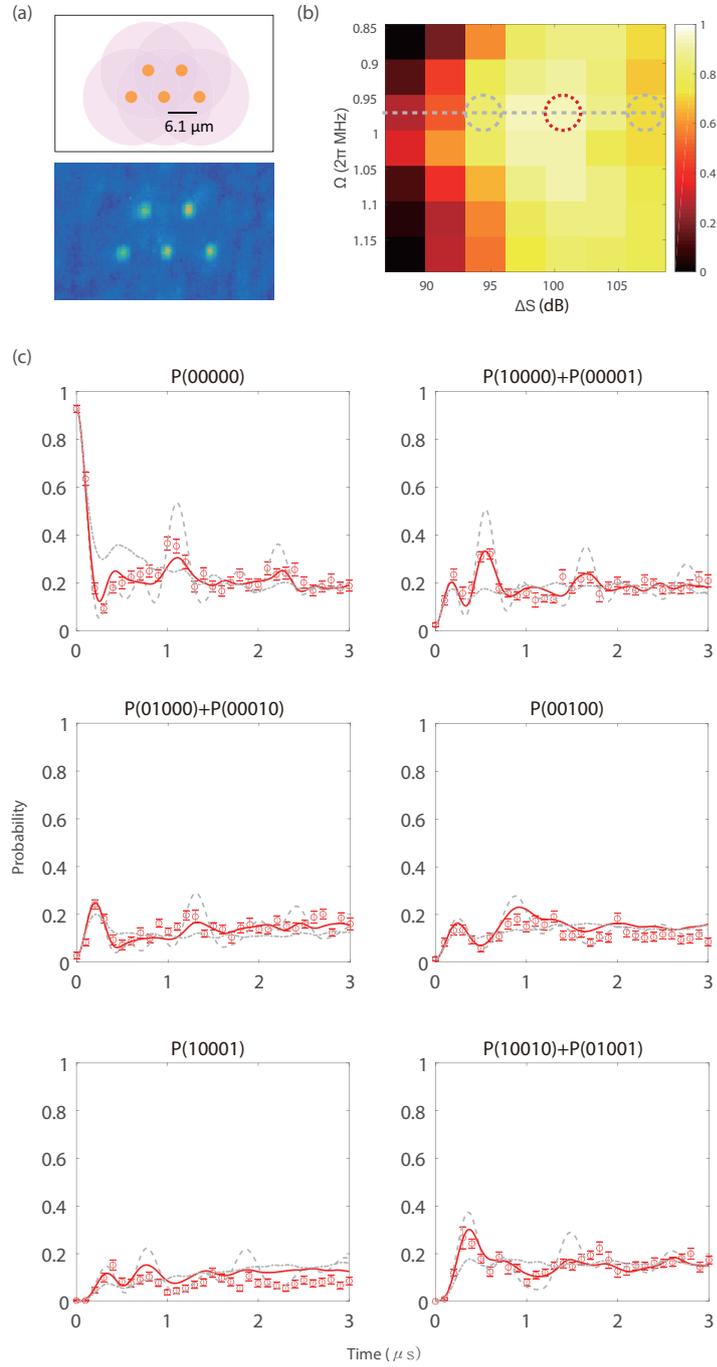


Figure 5.10: (a) Geometry of the atoms, (b) Scan of  $R^2$  between simulation and experimental data, and (c) Experimental data (red circles) and simulation (red curves) with the best fitting parameters, for  $N = 5$  atoms in zig-zag configuration.

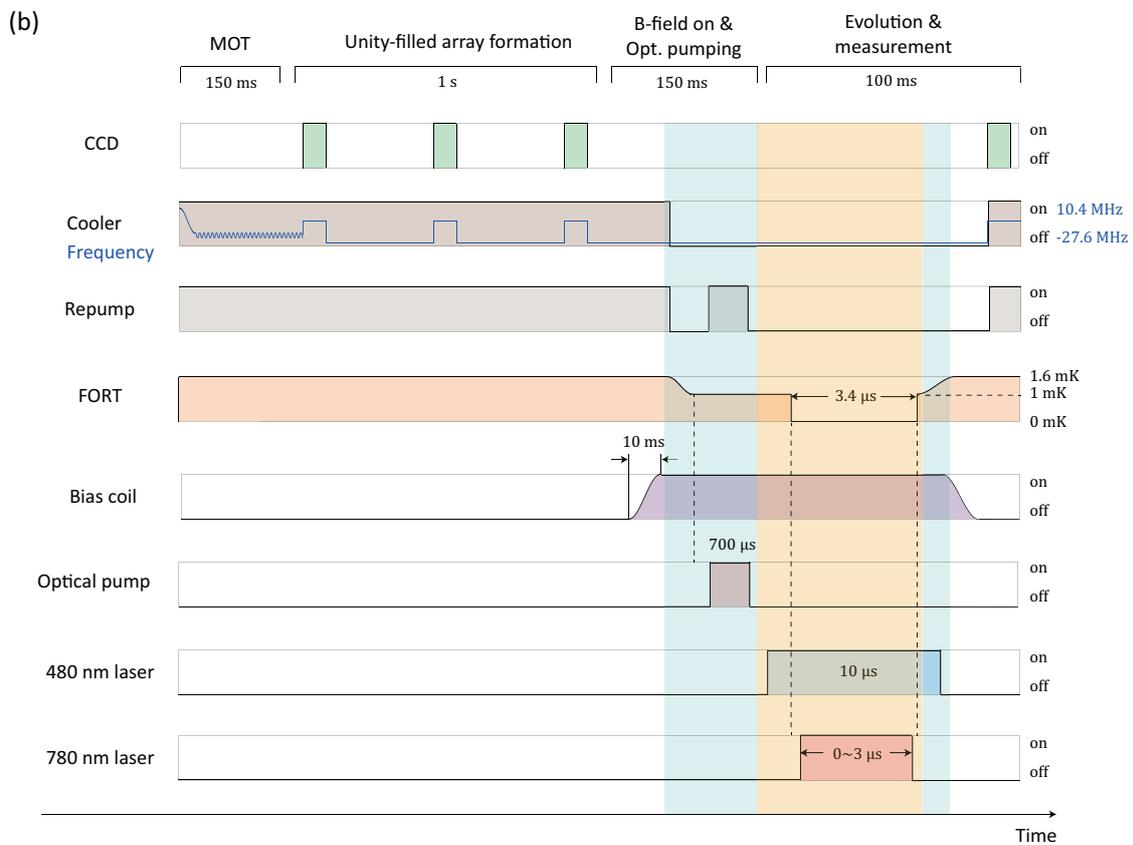
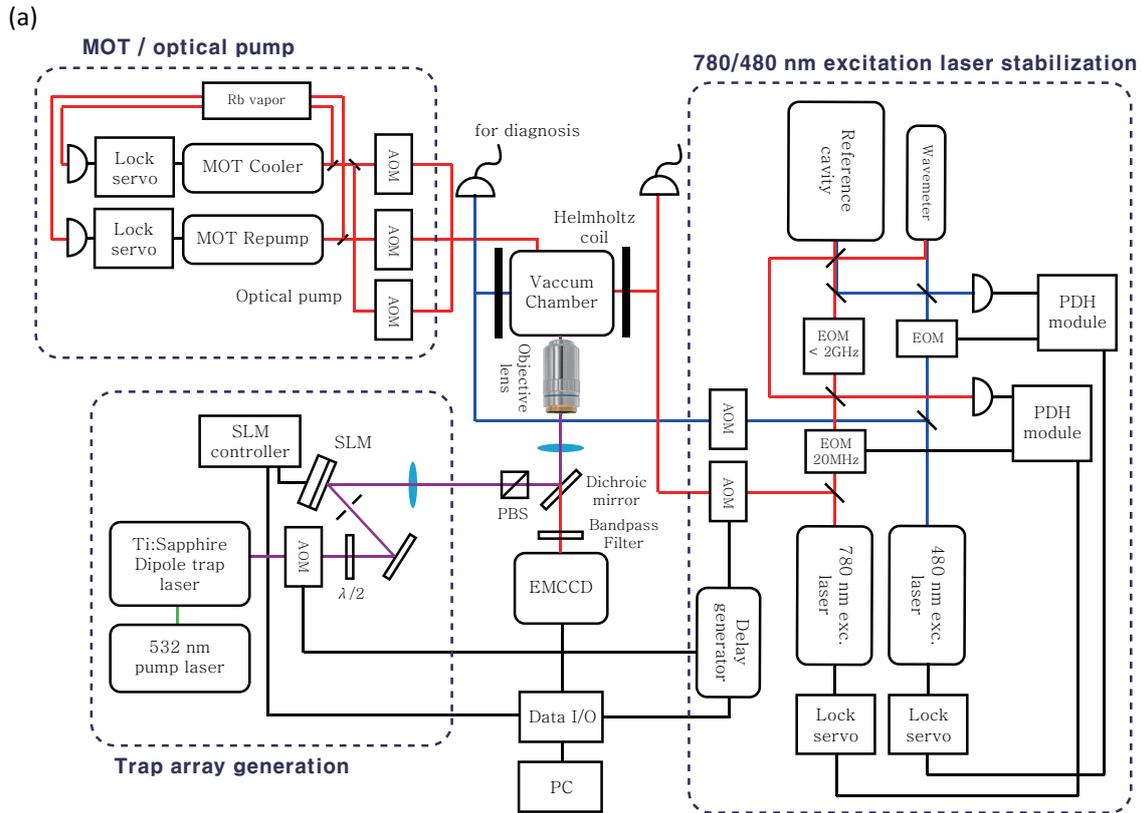


Figure 5.11: (a) Schematic illustration of the experimental setup. (b) Timeline of the experimental procedure.

array of 820 nm wavelength, each trap in 1.4  $\mu\text{m}$  diameter, was turned on to create single atom chain in the MOT volume. The trapped atoms were further cooled by polarization gradient cooling. Typical arrays were half-filled single atom arrays by collisional blockade [52], imaged by the EMCCD (Electron-multiplying CCD). To obtain unity-filled array, reconfiguration of existing atoms was processed [105, 106] by two time trials of imaging-reconfiguration-checking sequence.

## 2. Frequency stabilization of excitation lasers

The excitation lasers, 780 nm and 480 nm wavelength diode lasers (Homemade with Toptica laser diode & DL pro, a Toptica commercial diode laser system), respectively, were frequency stabilized to an ultralow expansion (ULE) reference cavity (Stable Laser Systems) for getting extra narrow linewidth of the lasers. The ULE had finesse of 15,000 and AR coated at dual wavelengths 780 nm and 480 nm. PDH locking (Pound-Drever-Hall locking) technique was used to lock the laser frequencies to the Fabry-Perot signal from the reference cavity reflection, along with fast lock servos. (Stable Laser Systems PDH module, Toptica PDD 110, Toptica FALC110) The wavelength of the lasers were roughly monitored by a wavemeter (HighFinesse WS7-60) within 60 MHz accuracy. The linewidth of the lasers, tens of kHz, were estimated by the width of the error signals, compared to the PDH lock signal peaks, considering peak widths. See Fig. 5.11(a).

## 3. Quantum evolution

After reconfiguration phase, excitation lasers shed on the atom chain. The excitation was two-photon transition, one in 780 nm wavelength and 10  $\mu\text{W}$  and the other in 480 nm, 10 mW, which were resonant to the 5S-6S level and non-resonant to the intermediate 5P state by  $\Delta_{inter}$ . In the excitation phase, first Helmholtz magnetic coils were turned on to make bias B-field for quantization axis. Before excitation, optical pump ( $F=2 \rightarrow F'=2$ ) along with repump beam was applied for 2 ms. Then, the atom tweezers were turned off for 3.4  $\mu\text{s}$  to avoid ac stark shift by them when the atoms were under excitation, after 480 nm excitation laser was turned on. 780 nm laser was turned on for a certain length of pulse time  $t_{exp}$ , to be measured, where the actual excitation happened in this period of time. Right after the 780 nm laser ended, the trapping laser was turned on to recapture the ground state populated atoms and the bias coil was turned off. The final state of the atom array were obtained by the EMCCD. See Fig. 5.11(b).

## 4. Power stabilization of excitation lasers

The stabilization of the laser power was once conducted for examination, to verify whether the power fluctuation affects the evolution. It turned out that the effect was negligible. We present the procedure here anyway for recording. For stabilization of the 780 nm excitation laser power, a photodiode was placed after the beam passed the experiment chamber. The 780 nm laser was turned on for 3  $\mu\text{s}$  a few milliseconds before the excitation for measurement of the laser power. Based on the power measurement, the laser power

was feedbacked through amplitude modulation of the AOM. The power fluctuation of 3% was improved to 0.7% after stabilization.

## Enumeration of possible errors

Major and minor possible errors that may occur in the experiment are listed and discussed.

### 1. Individual atom dephasing sources

- Leakage to and spontaneous decay from intermediate level

During two-photon excitation to the Rydberg state, there occurs transition leakage to 5P state by the 780 nm excitation laser despite of the large detuning  $\sim 600$  MHz from the intermediate state. The leakage can be estimated from the amount of the detuned Rabi oscillation  $\Omega^2/(\Delta^2 + \Omega^2) \sim 0.2\%$ . The leaked population goes through decay to the ground state  $\sim 30$  ns, causing decoherence in the quantum evolution. This Markov process can be considered by Lindblad Master equation as in Sec. 5.5.1.

- Spontaneous decay from the Rydberg state

The excitation to the Rydberg state is two-photon transition via the intermediate state  $5P_{3/2}$  with detuning  $\Delta \approx 600$  MHz. The decay time of 67S Rydberg state is in order of  $100 \mu s$ . The experiment time scale is  $< 10 \mu s$  which is small compared to the decay time, but the decay probability  $\sim e^{t/100}$  is still finite. The effect is applied to the simulation by adjusting the population (see Sec. 5.5.1).

- Atomic thermal motion

Although the positions of the atoms are assumed to be at the center of corresponding traps, they thermally move in a volume determined by the size of the trap and the atom temperature. This might cause uncertainty in the interaction strength. Also, due to finite velocity of the trapped atoms, the frequencies of the excitation lasers are doppler-shifted. Assuming  $30 \mu K$  of the atom temperature, one-dimensional velocity is  $0.058$  m/s, which causes shift by (0.074, 0.12) MHz for 780 nm and 480 nm lasers respectively. This would make a certain, but not much error.

Table 5.2: Individual dephasing sources.

Error sources	Effective Rabi decay $\tau$	Treatment
$5P_{3/2}$ decay, 26 ns (causing dephasing)	$\simeq 15.5 \mu s$	Two-level approximation and Lindblad operator $L_i = \sqrt{\gamma/2}\sigma_z^i$ , where $\gamma = 2\pi \times 20$ kHz
$67S_{1/2}$ decay, 100 $\mu s$ , Ref. [112]	-	Population adjustment
Thermal motion (80 $\mu K$ ), Ref. [113]	$\delta V < 0.5V$ $\delta\Delta_r \simeq 100$ kHz	-

### 2. Collective dephasing sources

- Laser phase fluctuation / laser linewidth

The finite linewidth of the laser from the diode itself causes a certain amount of de-coherence in the quantum evolution. The laser itself has noise in phase, which causes degradation of the coherence of the quantum evolution. Our 780 nm / 480 nm excitation lasers were frequency stabilized to an ULE reference cavity to  $< 30$  kHz. The phase noise can be estimated from the measurement of power spectral density [110] which causes considerable dephasing effect. A detailed consideration is discussed in Sec. 5.5.1.

- Laser power fluctuation

The laser power can fluctuate, which is due to the fluctuation of the power from the laser diode itself, from error in AOM modulation, or from beam pointing error in relaying fiber inputs. This results in fluctuation of  $\Omega$  and  $\delta$ , by ac stark shift. From the measurement in our setup, the fluctuation is about 3% without feedback (0.7% with feedback), and the change in stark shift due to intensity change occurs as  $\delta\Delta_r \sim 2\pi \times 40$  kHz.

- Stray E-field fluctuation

Stray E-field can exist by electronics devices around the experiment chamber, although most amount has been suppressed by grounded electrodes set up around the chamber. Our spectroscopy measurement tells that  $\delta\Delta_r$  is less than  $2\pi \times 100$  kHz.

Table 5.3: Collective dephasing sources

Error sources	Effective Rabi decay $\tau$	Treatment
Rydberg lasers linewidth, $\leq 30$ kHz	$\leq 20$ $\mu s$	-
Rydberg lasers phase noise	$\simeq 1$ $\mu s$	Replace $\Omega$ by $\Omega e^{i\phi(t)}$ , $\phi(t)$ converted from $S_\nu(f)$
Intensity fluctuation 4% (480), 2% (780)	$\delta\Omega \sim 0.03\Omega$ , $\delta\Delta_r \sim 2\pi \times 40$ kHz	$\simeq 20$ $\mu s$
Static electric field drift	$\delta\Delta_r \sim 2\pi \times 50$ kHz	-

### 3. Projection measurement error sources

- Atom loss at the second imaging (false negative)

- Background collision

After excitation, an atom is determined to collapse into the ground state by being recaptured. However, atoms are lost by finite probability mainly by collision with hot background gas atoms. This leads to wrong measurement that a certain amount of ground state population is transferred to Rydberg state population. Trap-off time  $t_{recap}$  out of trap lifetime  $\tau$  causes loss of  $p(r|g) = 1 - e^{-t_{recap}/\tau}$  and it is estimated 0.01 (see Sec. 5.5.1).

- Leakage to other Rydberg states
 

A small portion of the excitation to the Rydberg states can evolve to other Rydberg state than the intended state due to finite polarization extinction ratio.
- Atom incoming (false positive)
  - Background incoming
 

Although with a low probability, background atoms sometimes income into vacancy traps at the post-excitation capture, occuring false positive measurement, although the amount is negligible in our setup.
  - Leakage to other ground states
 

The de-excitation to the ground state can evolve to other ground state than the intended state due to finite polarization extinction ratio (additional to the spontaneous emission from the intermediate state).
  - Spontaneous emission from the Rydberg state after recapture time
 

The atoms that are in the Rydberg state can decay to the ground state by spontaneous emission during the time between the recapture and measurement. For the time length  $t_{rm}$ , the false-positive probability is  $p(g|r) = 1 - e^{-t_{rm}/\tau} \sim 0.01$ .

Table 5.4: Projection measurement error sources

Error sources	$\langle \hat{n} \rangle$ error	Treatment
Atom image projection	$\ll 0.001$	-
Background gas collision	0.015	Population adjustment
Leakage to the other Rydberg levels [37]	$< 0.001$	-
Leakage to another ground level	$\simeq 0.003$	-
Spontaneous emission from Rydberg state	$\sim 0.01$	-

### 5.5.3 Conclusion

We have analyzed the dynamics of  $N = 3 \sim 5$  dipole-trapped atoms in linear and zig-zag configurations under interaction via Rydberg blockade. Based on the rather simple Hamiltonian with self-energy terms and interaction terms, environmental effects such as spontaneous emission, stochastic atom loss, and phase noise of the excitation lasers have been considered as dephasing sources.

For a specific set of fitting parameters, the simulated evolution agreed well with the experimental data. It turned out in the result that the phase noise of the excitation laser is the major factor of the dephasing in the dynamics. We expect the detailed analysis for the Rydberg atom dynamics to be an useful background for further studies.

## 5.6 Machine learning of N-atom dynamics

### 5.6.1 Curve fitting vs. machine learning: computational complexity

Understanding the Rydberg atom dynamics requires either performing a full set of quantum tomography measurements or determining the governing equation, e.g., Lindblad equation. However, both are daunting tasks due to the huge size of density matrix. Here we present our machine learning approach, detouring the difficulty of classical computational complexity, to the measurement-based analysis of the dynamics of our Rydberg-atom quantum simulators.

Curve fitting of the measured data  $\rho_{ii}^{exp}(t_j)$  by the diagonal elements  $\rho_{ii}^{sim}(t_j)$  of simulation data, with a Hamiltonian  $H(\Omega; N)$  and a master equation  $\dot{\rho} = f(\rho; \gamma_k)$ , where  $\gamma_k$  are some set of dephasing rates, the amount of time required for computation (simulation) of the N-atom dynamics time-series increases exponentially in N. If our objective is to obtain fitting parameters for some data set,  $N_{iter}$  iterations of single N-atom evolution calculation which takes  $x^N t_1$  is needed where  $t_1$  indicates the required time for N=1 system ( $x$  might differ by the calculation algorithm). The benchmark result of curve fitting (CF) calculation time for  $N = 5, 6, 7, 8$  atoms with a typical personal computer (CPU model Intel i5-4670) is shown in Fig. 5.12. The results exactly falls on an exponential line in N. Fitting time of the dynamics of  $N = 9$  atoms already exceeds one hour in time, which makes numerical fitting impractical for more atoms.

In contrast, by machine learning the system using artificial neural network (ANN) requires no heavy calculation, even in real-time in many cases, when we are to retrieve the system parameters from some data. The heavy calculation part can be transferred to ‘training phase’ which is done prior to the actual retrieval of the outputs. In our benchmark, the retrieval time using a neural network was less than or around 1 s. The ANN-feasible number of qubits is only limited by long-term training time which can be reduced by a high-performance workstation. So, by using ANN, analysis of experimental data can be done efficiently, at the cost of pre-training time and resources. In this section, a toy model of quantum system machine learning with  $N = 2, 3$  atoms and the result fitted with experimental data are presented.

### 5.6.2 N-atom dynamics

The physical system we consider is an experimental Rydberg quantum simulation system with  $N$  atoms trapped with an array of optical tweezers and entangled through excitation to a Rydberg energy level. When the atoms are modeled with an effective two-level pseudo-spin system of ground and Rydberg states [91], the Hamiltonian is the transverse-field quantum Ising-like model,

$$H = \sum_j \frac{\hbar\Omega}{2} \sigma_x^i + \sum_{j < k} V_{jk} n_j n_k. \quad (5.14)$$

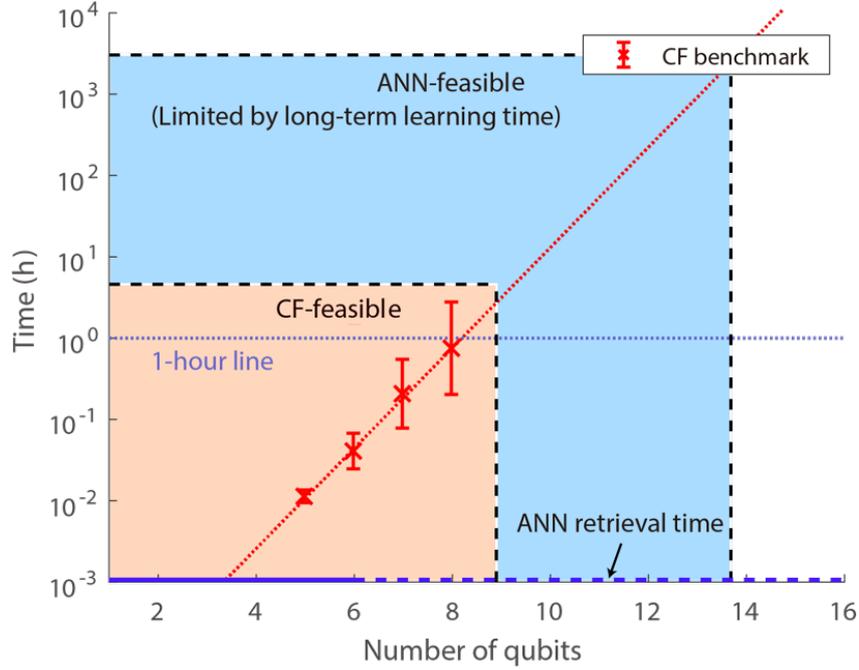


Figure 5.12: The computational complexity of computing N-atom dynamics.

The system dynamics is assumed to be under Markovian systematic decoherence, which follows the Lindblad master equation:

$$\frac{d}{dt}\rho = -\frac{i}{\hbar}[H, \rho] + \sum_i \left[ L_i \rho L_i^\dagger - \frac{1}{2}\{L_i^\dagger L_i, \rho\} \right]. \quad (5.15)$$

Based on this dephasing N-qubit Ising like model, we use a dephasing model which has two dephasing rates, which approximately fits for  $N \leq 3$  atoms. In the model, one parameter is  $\gamma_1$  which accounts for the dephasing of individual atoms, and the other is  $\gamma_2$  which is for the global (collective) dephasing. The Lindblad operators are given, for the individual dephasing, in forms of

$$L_1 = \begin{pmatrix} \sqrt{\gamma_1/2} & 0 \\ 0 & -\sqrt{\gamma_1/2} \end{pmatrix} \otimes I^{N-1}, \quad \dots, \quad L_N = I^{N-1} \otimes \begin{pmatrix} \sqrt{\gamma_1/2} & 0 \\ 0 & -\sqrt{\gamma_1/2} \end{pmatrix}, \quad (5.16)$$

where  $L_1, \dots, L_N$  are for individual dephasing of each atom. For collective dephasing, a Lindblad operator of the form

$$L_0 = \begin{pmatrix} \sqrt{\gamma_2/2} & 0 \\ 0 & -\sqrt{\gamma_2/2} \end{pmatrix} \otimes I^{N-1} + \dots + I^{N-1} \otimes \begin{pmatrix} \sqrt{\gamma_2/2} & 0 \\ 0 & -\sqrt{\gamma_2/2} \end{pmatrix} \quad (5.17)$$

was used.

### 5.6.3 Artificial neural network

Machine learning enables solving a variety of complex problems, by discovering visible and/or hidden patterns of data. Here, we use an artificial neural network to determine system

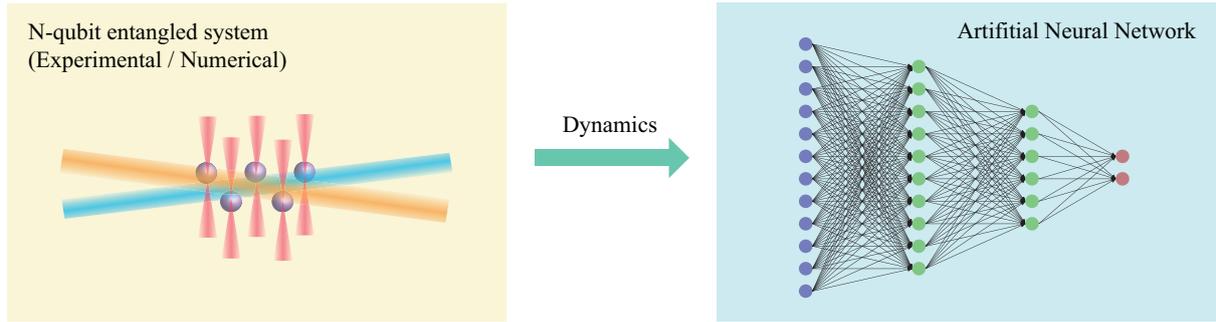


Figure 5.13: The concept of dynamics mapping on ANN.

parameters in the Markovian dynamics for our experimental Rydberg-atom quantum simulator. For determination of the system parameters from the experimental data, a four-layer feed-forward artificial neural-network was used. The neural network we designed consisted of an input layer which takes the time series of projection measurement data of dynamics,  $\rho_{aa}(t_b)$  ( $a = 1, 2, \dots, 2^N$ ,  $b = 1, 2, \dots, N_t$ ), by nodes  $x_i$  ( $i = 1, 2, \dots, N_t 2^N$ ). The output layer had nodes  $y_l$  ( $l = 1, 2, \dots, N_p$ ) where each node was for each parameter to be retrieved. Additionally, the artificial neural network had two hidden layers in-between,  $h_j$  for Hidden layer 1 and  $g_k$  for Hidden layer 2. From each  $x_i$  of input layer to each  $h_j$  of the Hidden layer 1, there is a weight  $w_{ji}$  from the first layer to Hidden layer 2,  $v_{kj}$ , and from Hidden layer 2 to the output layer,  $u_{lk}$ .

To obtain a set of output values from a dataset  $X_i$  through our neural network, “forward” propagation had occurred. The weights were set random initially. The value of a node  $h_j$  in Hidden layer 1 was determined by  $h_j = \sigma(\sum_i w_{ji} X_i / \beta_w)$ . Similarly, the values of Hidden layer 2 were calculated as  $g_k = \sigma(\sum_j v_{kj} h_j / \beta_v)$ , and the output layer,  $y_l = \sigma(\sum_k u_{lk} g_k / \beta_u)$ .

First, the neural network was trained through supervised learning with numerically generated data of projection measurements which are simulations of the actual experimental data and the corresponding system parameters used to create the data. After training, the neural network was used to determine the system parameters of the experimental data. A schematic figure is shown in Fig. 1(a). The number of input layer nodes was 62, which was from 31, the time sample length, times two, the number of projection states we used. The single hidden layer had six nodes, being decided heuristically, and the output layer had two nodes to give  $\gamma_1$  (individual dephasing) and  $\gamma_2$  (global dephasing).

#### 5.6.4 Program codes

- initialize.m

This code initializes the calculation; i.e., clears existing memories, sets the number of qubits, time step and length, generates the basis, loads the experimental data, and generates the original Hamiltonian.

```
1 clear all
2
```

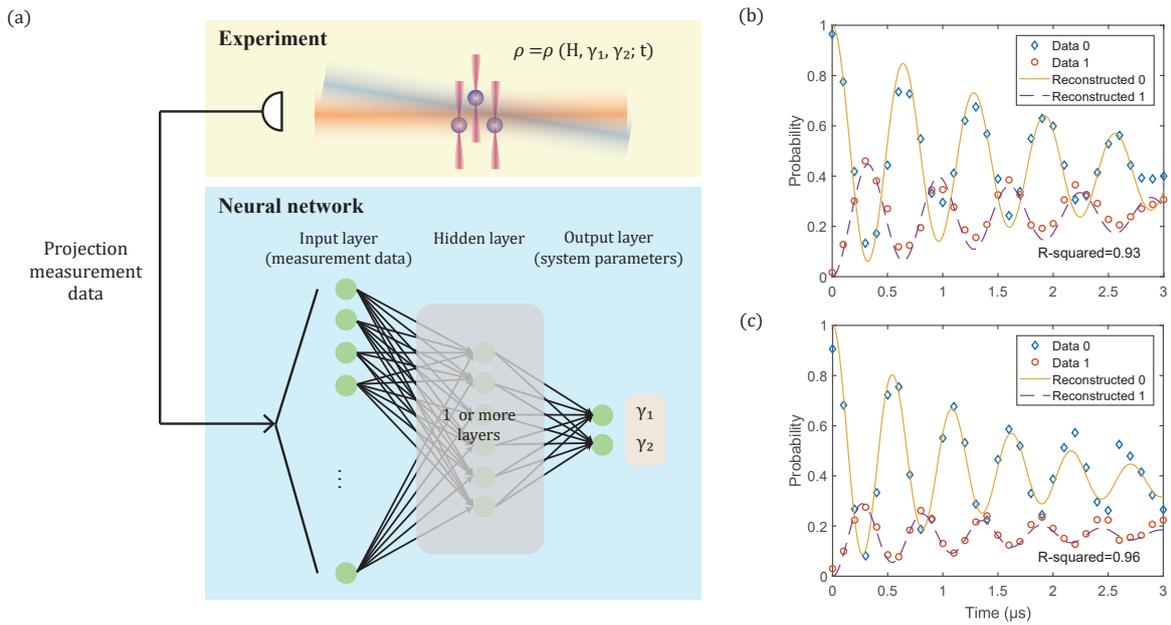


Figure 5.14: (a) The scheme of machine learning of multi-qubit dynamics. In the experimental setup, a physical multi-qubit system evolves and is measured in time series. The measured data is sent into the input layer of the neural network to determine the system parameters. The neural network is pre-trained with numerically generated data. (b) Result of learning in  $N = 2$  qubit system. A set of data and the corresponding reconstructed dynamics from the output of the neural network. “0” indicates the probability in the state with zero excited atoms, and “1” indicates the average of the probabilities of the one-atom excited states. (c) Result for  $N = 3$  qubit triangular system.

```

3 % addpath('C:\Users\woojun\Documents\MATLAB\qit ')
4 % init
5
6 nOfTrainingSets=1;
7 nOfQubits=4;
8 length_sample=31*2;
9 training_set=zeros(nOfTrainingSets,length_sample);
10 answer_set=zeros(nOfTrainingSets,2);
11
12 hbar=1.05e-34;
13
14 dt=1e-9;
15
16 length_t=3000;
17 time_axis=(1:length_t)*dt;
18 Omega=7e6;
19
20 basis=decimalToBinaryVector(0:2^nOfQubits-1);
21 binDigitSum=sum(basis,2);
22 % [binDigitSum, I]=sort(binDigitSum,'descend');
23 % basis=basis(I,:);
24
25
26 mult_Omega_coeff=0.5;
27 mult_Omega=1+random('normal',0,mult_Omega_coeff);
28 mult_Delta_coeff=0.5;
29 mult_Delta=random('normal',0,mult_Delta_coeff);
30
31 clear H_0
32 sigma_x=[0 1; 1 0];
33 sigma_z=[1 0; 0 -1];
34 for i=1:nOfQubits
35     H_0_temp=1;
36     for j=1:nOfQubits
37         if j==i
38             H_0_temp=kron(H_0_temp,sigma_x);
39 %             H_0_temp=kron(H_0_temp,mult_Omega*sigma_x+mult_Delta*0.5*
sigma_z);
40         else
41             H_0_temp=kron(H_0_temp,eye(2));
42         end
43     end
44     H_0(:, :, i)=H_0_temp;
45 end
46 H_0=hbar*Omega/2*sum(H_0,3);
47
48 clear tau_v
49 clear purity_m

```

```

50 clear V
51
52 mode=1;
53 interaction_4qubit;
54
55 H=H_0+V;
56
57 % figure;
58 % hold on;
59 for iter=1:nOfTrainingSets
60     iter
61
62     rand_param=rand(2,1);
63 %     rand_param=[-99999 0.5];
64 %     rand_param=[-99999 -99999];
65     evolution;
66     training_set(iter,:)=rhodiag_sample(:).*(1+random('normal',0,0.1,1,numel
        (rhodiag_sample)))';
67     answer_set(iter,:)=rand_param;
68
69     plot(rhodiag)
70
71 end

```

- evolution.m

This code conducts the actual evolution of the system, with dephasing factors generated.

```

1 gamma=2*pi*10.^(2*(rand_param)+3.5);
2
3 L=zeros(2^nOfQubits);
4 l1=[sqrt(gamma(1)/2) 0; 0 -sqrt(gamma(1)/2)];
5 l2=[sqrt(gamma(2)/2) 0; 0 -sqrt(gamma(2)/2)];
6
7 % L1_1=kron(kron(l1,eye(2)),eye(2));
8 % L1_2=kron(kron(eye(2),l1),eye(2));
9 % L1_3=kron(kron(eye(2),eye(2)),l1);
10 % L1(:, :, 1)=L1_1;
11 % L1(:, :, 2)=L1_2;
12 % L1(:, :, 3)=L1_3;
13 L1=zeros(2^nOfQubits,2^nOfQubits,nOfQubits);
14 for i=1:nOfQubits
15     L_temp=1;
16     for j=1:nOfQubits
17         if j==i
18             L_temp=kron(L_temp,l1);
19         else
20             L_temp=kron(L_temp,eye(2));
21         end

```

```

22     end
23     L1(:, :, i)=L_temp;
24 end
25
26 % L2=kron(kron(l2, eye(2)), eye(2))+kron(kron(eye(2), l2), eye(2))+kron(kron(eye
    (2), eye(2)), l2);
27 L2=zeros(2^nOfQubits, 2^nOfQubits, nOfQubits);
28 for i=1:nOfQubits
29     L_temp=1;
30     for j=1:nOfQubits
31         if j==i
32             L_temp=kron(L_temp, l2);
33         else
34             L_temp=kron(L_temp, eye(2));
35         end
36     end
37     L2(:, :, i)=L_temp;
38 end
39 L2=sum(L2, 3);
40
41 rho=zeros(2^nOfQubits);
42 rho(1)=1;
43
44 clear y ryFrac purity
45
46 rho_record=evolution_core(dt, length_t, H, L1);
47 % toc
48 rho_sample=time_sample(rho_record, 151, 20);

```

- evolutionCore.m The core part of the code where the Lindblad master equation is computed with the Runge-Kutta method.

```

1 function [rho_record] = evolution_core(dt, length_t, H, L1)
2 hbar=1.05e-34;
3
4 rho=zeros(size(H,1));
5 rho(end)=1;
6
7 rho_record=zeros(length_t+1,length(rho(:)));
8
9 rho_ones=ones(size(H,1));
10 L_decom=zeros(size(H,1));
11 for j=1:size(L1,3)
12     L_temp=L1(:, :, j);
13     L_decom=L_decom+( L_temp*rho_ones*conj(L_temp) -0.5*(conj(L_temp)*L_temp*
        rho_ones+rho_ones*conj(L_temp)*L_temp) );
14 end
15

```

```

16 for i=1:length_t+1
17     rho_record(i,:)=rho(:);
18
19     k1=1/i/hbar*(H*rho-rho*H) + rho.*L_decom;
20     k2=1/i/hbar*(H*(rho+dt/2*k1)-(rho+dt/2*k1)*H) + (rho+dt/2*k1).*L_decom;
21     k3=1/i/hbar*(H*(rho+dt/2*k2)-(rho+dt/2*k2)*H) + (rho+dt/2*k2).*L_decom;
22     k4=1/i/hbar*(H*(rho+dt*k3)-(rho+dt*k3)*H) + (rho+dt*k3).*L_decom;
23
24     drho=dt/6*(k1+2*k2+2*k3+k4);
25     rho=rho+drho;
26
27 end
28
29 end

```

- neuralNetwork.m

The code that conducts learning of an ANN. It came from an open source ANN code and modified.

```

1 %% Very simple and intuitive neural network implementation
2 %
3 % Carl L?ndahl, 2008
4 % email: carl(dot)londahl(at)gmail(dot)com
5 % Feel free to redistribute and/or to modify in any way
6
7 % DATA SETS; demo file
8 % [training_set , answer_set] = mendez;
9
10 clear answer y_v
11
12 n = 1.2;
13 nbrOfNodes1 = 6;
14 % nbrOfNodes2 = 5;
15 % nbrOfNodes3 = 5;
16 nbrOfEpochs = 150000;
17
18 % Initialize matrices with random weights 0-1
19 w1 = rand(nbrOfNodes1 , length(training_set(1,:)));
20 % w2 = rand(nbrOfNodes2 , nbrOfNodes1);
21 % w3 = rand(nbrOfNodes3 , nbrOfNodes2);
22 w4 = rand(length(answer_set(1,:)) , nbrOfNodes1);
23
24 m = 0;
25 e = size(training_set);
26 y_v=zeros(nbrOfEpochs,1);
27 RMS_best=100;
28
29 while m < nbrOfEpochs

```

```

30
31 % Increment loop counter
32 m = m + 1;
33
34 % Iterate through all examples
35 for i=1:e(1)
36     % Input data from current example set
37     input = training_set(i,:).';
38     answer = answer_set(i,:).';
39
40     % Propagate the signals through network
41     forward;
42
43     % Output layer error
44     delta_i = output.*(1-output).*(answer-output);
45
46     % Calculate error for each node in layer_(n-1)
47 %     delta_j = hidden3.*(1-hidden3).*(w4'*delta_i);
48 %     delta_k = hidden2.*(1-hidden2).*(w3'*delta_j);
49     delta_l = hidden1.*(1-hidden1).*(w4'*delta_i);
50
51     % Adjust weights in matrices sequentially
52     w4 = w4 + n.*delta_i*(hidden1');
53 %     w3 = w3 + n.*delta_j*(hidden2');
54 %     w2 = w2 + n.*delta_k*(hidden1');
55     w1 = w1 + n.*delta_l*(input');
56 end
57
58 RMS_Err = 0;
59
60 % Calculate RMS error
61 for i=1:e(1)
62     answer = answer_set(i,:).';
63     input = training_set(i,:).';
64     forward;
65     RMS_Err = RMS_Err + norm(answer-output,2);
66 end
67
68 y = RMS_Err/e(1);
69 y_v(m)=y;
70 [m/10000 log10(y)]
71
72 if RMS_Err<RMS_best
73     epoch_best=m;
74     w1_best=w1;
75     w4_best=w4;
76     RMS_best=RMS_Err;
77 end

```

```

78
79     if log10(y)<-1.5
80         n=0.9;
81     end
82
83 %     plot(1:m,log10(y-v(1:m)), '* '); drawnow;
84
85 %     input=training_set(10,:);
86 %     forward;
87 %     [answer_set(10,1) output(1)]
88     if log10(y)<-2
89         break;
90     end
91 end
92
93 input = training_set(8,:).';
94 forward;
95 rand_param=output;
96 evolution;
97 RMS_Err = norm(input-sample);
98 y = RMS_Err/e(1);
99 log10(y)
100 figure; plot(reshape(input,31,2)); hold on; plot(rhodiag_sample);
101
102 rand_param=rand(2,1);
103 param_init=rand_param;
104 evolution;
105 input=real(rhodiag_sample(:));
106 forward;
107 rand_param=output;
108 evolution;
109 RMS_Err = norm(input-sample);
110 y = RMS_Err/e(1);
111 log10(y)
112 figure; plot(reshape(input,31,2)); hold on; plot(rhodiag_sample);
113
114 data=load('data.mat');
115 data=data.data;
116 input = data(:);
117 forward;
118 rand_param=output;
119 evolution;
120 RMS_Err = norm(input-rhodiag_sample(:));
121 figure; plot(reshape(input,31,2)); hold on; plot(rhodiag_sample);
122 r_squared=1-sum((input-rhodiag_sample(:)).^2) / sum((input-mean(input)).^2)
123 figure; plot(log10(y-v))

```

### 5.6.5 Results and discussion

The test results for a two-qubit system and a three-qubit system in triangular configuration (all atoms in the Rydberg blockade) are shown in Figs. 1(b, c). We used time series data from experiment, which were prepared in projection measurement basis and had a length of 31 for  $3 \mu\text{s}$ . For dephasing model, we used the Lindblad master equation, with two parameters,  $\gamma_1$ , the individual dephasing parameter, and  $\gamma_2$ , the global dephasing parameter. Among  $2^N$  projection states in  $N$ -qubit system, only non-doubly excited states were used for learning because otherwise the learning performance was poor.

The probabilities for ideally degenerate one-atom excited states were averaged before use due to the same reason. The reconstructed numerical dynamics with the system parameters determined by the trained neural networks well agreed with the test experiment data. The  $R$ -squared values for two- and three-qubit system tests were 0.93 and 0.96, respectively. The spent time for obtaining the system parameters was less than 10 ms in a PC, while the training time was around 15 minutes with 50 training sets, taking up to 200,000 epochs. The pre-trained neural network analyzed the systems in much less time, promising in-situ monitoring of the experimental setup.

## Chapter 6. Conclusion and outlook

Experimental progress and their theoretical backgrounds in Rydberg atom array creation and control for quantum information processing have been reported and discussed. Especially, a novel method for transport and reconfiguration of single atom arrays, constructing defect-free arrays using this method and analysis of dynamics of N atom quantum system in Rydberg blockade regime are suggested.

Construction and setting of a magneto-optical trap, the basic of experimental setup required for single atom array creation and control, was discussed in Chapter 2. Designing and operating of the parts of the magneto-optical trap, vacuum chamber, cooling and repump lasers, anti-Helmholtz coil, Earth's magnetic field compensation coil were introduced. Our MOT condition had temperature of  $< 100 \mu K$  ( $30 \sim 70 \mu K$  typical), density of  $10^{10} / cm^3$  in vacuum pressure of  $3 \times 10^{-10}$  Torr.

In chapter 3, the novel method of single atom transport and array reconfiguration was discussed. Since the GS algorithm does not guarantee frame-to-frame phase continuity, huge frame-to-frame flicker caused by phase jump in large portion of the SLM, makes single atom transport impossible with this method. Instead, we devised and applied flicker-free phase algorithms, random mask, superposition, and GSW algorithms, enabling single atom array transport and reconfiguration.

In chapter 4, methods for constructing a defect-free array, out of naturally half-filled arrays limited by collisional blockade effect were discussed. A novel method of atom-site matching for path planning was introduced and applied to show it causes less atom loss than the existing heuristic methods, in reasonably short time compared to brute-force methods. Also, GPU-acceleration and feedback system in experimental setup to make the phase calculation and the vacancy-filling operation in real-time scale was introduced.

Applying atom-atom interaction via Rydberg blockade is introduced in chapter 5. An experimental evidence of Rydberg blockade is demonstrated, and N-atom Rydberg dynamics data is analyzed considering the environmental condition such as laser intensity fluctuation, frequency fluctuation (or phase noise), error in absolute frequency of the laser, Doppler shift, spontaneous decay, population leakage, beam pointing fluctuation, stray fields, thermal motion, atom loss, etc. Also, analysis of data with artificial neural network is suggested, with application of real-time diagnosis of the experiment condition.

Further studies can be done in single atom array reconfiguration and Rydberg dynamics analysis. Although reconfiguration of single atom arrays has been done even in three dimensions, simultaneous imaging of multiple image planes in atom array region is not yet done. This can be accomplished by setting up a tunable lens in front of the EMCCD camera and synchronizing it with the imaging and reconfiguration sequence. In Rydberg dynamics experiment, full

consideration of the environmental condition can be done by setting up devices that control relevant experimental parameters. With this, more precise modeling and error analysis will be possible for Rydberg atom experiment.

## Bibliography

- [1] T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J. L. O'Brien, "Quantum computers," *Nature* **464**, 45 (2010).
- [2] J. Benhelm, G. Kirchmair, C. F. Roos, and R. Blatt, "Towards fault-tolerant quantum computing with trapped ions," *Nat. Phys.* **4**, 463 (2008).
- [3] T. P. Harty, D. T. C. Allcock, C. J. Ballance, L. Guidoni, H. A. Janacek, N. M. Linke, D. N. Stacey, and D. M. Lucas, "High-fidelity preparation, gates, memory, and readout of a trapped-ion quantum bit," *Phys. Rev. Lett.* **113**, 220501 (2014).
- [4] C. J. Ballance, T. P. Harty, N. M. Linke, M. A. Sepiol, and D. M. Lucas, "High-fidelity quantum logic gates using trapped-ion hyperfine qubits," *Phys. Rev. Lett.* **117**, 060504 (2015).
- [5] J. M. Chow, J. M. Gambetta, A. D. Córcoles, S. T. Merkel, J. A. Smolin, C. Rigetti, S. Poletto et al., "Universal quantum gate set approaching fault-tolerant thresholds with superconducting qubits," *Phys. Rev. Lett.* **109**, 060501 (2012)
- [6] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White et al., "Superconducting quantum circuits at the surface code threshold for fault tolerance," *Nature* **508**, 500 (2014).
- [7] T. Xia, M. Lichtman, K. Maller, A. W. Carr, M. J. Piotrowicz, L. Isenhower, and M. Saffman, "Randomized benchmarking of single-qubit gates in a 2D array of neutral-atom qubits," *Phys. Rev. Lett.* **114**, 100503 (2015).
- [8] F. Nogrette, H. Labuhn, S. Ravets, D. Barredo, L. Béguin, A. Vernier, T. Lahaye, and A. Browaeys, "Single-atom trapping in holographic 2D arrays of microtraps with arbitrary geometries," *Phys. Rev. X* **4**, 021034 (2014).
- [9] T. Wilk, A. Gaëtan, C. Evellin, J. Wolters, Y. Miroshnychenko, P. Grangier, and A. Browaeys, "Entanglement of two individual neutral atoms using Rydberg blockade," *Phys. Rev. Lett.* **104**, 010502 (2010).
- [10] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press (2000).
- [11] M. Saffman and T. G. Walker, "Analysis of a quantum logic device based on dipole-dipole interactions of optically trapped Rydberg atoms," *Phys. Rev. A*, **72**, 022347, (2005).

- [12] I. Bloch, “Quantum coherence and entanglement with ultracold atoms in optical lattices,” *Nature* **453**, 1016 (2008).
- [13] J. I. Cirac and P. Zoller, “Goals and opportunities in quantum simulation,” *Nat. Phys.* **8**, 264 (2012).
- [14] I. Bloch, J. Dalibard, and S. Nascimbene, “Quantum simulations with ultracold quantum gases,” *Nat. Phys.* **8**, 267 (2012).
- [15] P. Hauke, F. M. Cucchietti, L. Tagliacozzo, I. Deutsch, and M. Lewenstein, “Can one trust quantum simulators?” *Rep. Progr. Phys.* **75**, 082401 (2012).
- [16] A. Abragam, “Principles of Nuclear Magnetism,” Oxford University Press, (2002).
- [17] S. R. Hartmann and E. L. Hahn, “Nuclear double resonance in the rotating frame,” *Phys. Rev.* **128**, 2042 (1962).
- [18] M. H. Levitt and R. Freeman, “NMR population inversion using a composite pulse,” *J. of Mag. Resonance* **33**, 473 (1978).
- [19] J. Lim, H. G. Lee, S. Lee, C. Y. Park, and J. Ahn, “Ultrafast Ramsey interferometry to implement cold atomic qubit gates,” *Sci. Rep.* **4**, 05867 (2014).
- [20] D. Schrader, I. Dotsenko, M. Khudaverdyan, Y. Miroshnychenko, A. Rauschenbeutel, and D. Meschede, “Neutral atom quantum register,” *Phys. Rev. Lett.* **93**, 150501 (2004).
- [21] S. Olmschenk, R. Chicireanu, K. D. Nelson, and J. V. Porto, “Randomized benchmarking of atomic qubits in an optical lattice,” *New J. Phys.* **12**, 113007 (2010).
- [22] D. D. Yavuz, P. B. Kulatunga, E. Urban, T. A. Johnson, N. Proite, T. Henage, T. G. Walker, and M. Saffman, “Fast ground state manipulation of neutral atoms in microscopic optical traps,” *Phys. Rev. Lett.* **96**, 063001 (2006).
- [23] C. Knoernschild, X. L. Zhang, L. Isenhower, A. T. Gill, F. P. Lu, M. Saffman, and J. Kim, “Independent individual addressing of multiple neutral atom qubits with a MEMS beam steering system,” *Appl. Phys. Lett.* **97**, 134101 (2010).
- [24] Y. Wang, A. Kumar, T.-Y. Wu, and D. S. Weiss, “Single qubit gates based on targeted phase shifts in a 3D neutral atom array,” *Science* **352**, 1562 (2016).
- [25] L. Isenhower, E. Urban, X. L. Zhang, A. T. Gill, T. Henage, T. A. Johnson, T. G. Walker, and M. Saffman, “Demonstration of a neutral atom controlled-NOT quantum gate,” *Phys. Rev. Lett.* **104**, 010503 (2010).
- [26] R. Grimm, M. Weidemüller, and Y. B. Ovchinnikov, “Optical dipole traps for neutral atoms,” *Adv. At. Mol. Opt. Phy.* **42**, 95 (2000).

- [27] S. Bergamini, B. Darquié, M. Jones, L. Jacubowicz, A. Browaeys, and P. Grangier, “Holographic generation of microtrap arrays for single atoms by use of a programmable phase modulator,” *J. Opt. Soc. Am. B* **21**, 1889-1894 (2004).
- [28] K. D. Nelson, X. Li, and D. S. Weiss, “Imaging single atoms in a three-dimensional array,” *Nat. Phys.* **3**, 556 (2007).
- [29] M. Saffman, “Quantum computing with atomic qubits and Rydberg interactions: progress and challenges,” *J. Phys. B: At., Mol. Opt. Phys.* **49**, 202001 (2016).
- [30] T. G. Walker and M. Saffman, “Consequences of Zeeman degeneracy for the van der Waals blockade between Rydberg atoms,” *Phys. Rev. A* **77**, 032723 (2008).
- [31] J. Estève, “Trapped by nanostructures,” *Nat. Nanotechnol.*, **8**, 317-318 (2013).
- [32] A. L. Migdall, J. V. Prodan, W. D. Phillips, T. H. Bergeman, and H. J. Metcalf, “First observation of magnetically trapped neutral atoms,” *Phys. Rev. Lett.* **54**, 2596 (1985).
- [33] S. Chu, J. E. Bjorkholm, A. Ashkin, and A. Cable, “Experimental observation of optically trapped atoms,” *Phys. Rev. Lett.* **57**, 314 (1986).
- [34] V. S. Bagnato, G. P. Lafyatis, A. G. Martin, E. L. Raab, R. N. Ahmad-Bitar, and D. E. Pritchard, “Continuous stopping and trapping of neutral atoms,” *Phys. Rev. Lett.* **58**, 2194 (1987).
- [35] E. L. Raab, M. Prentiss, A. Cable, S. Chu, and D. E. Pritchard, “Trapping of neutral sodium atoms with radiation pressure,” *Phys. Rev. Lett.* **59**, 2631 (1987).
- [36] W. Lee, H. Kim, and J. Ahn, “Three-dimensional rearrangement of single atoms using actively controlled optical microtraps,” *Opt. Express*, **24**, 9816 (2016).
- [37] J. E. Bjorkholm, R. R. Freeman, A. Ashkin, and D. B. Pearson, “Observation of focusing of neutral atoms by the dipole forces of resonance-radiation pressure,” *Phys. Rev. Lett.* **41**, 1361 (1978).
- [38] S. Chu, J. E. Bjorkholm, A. Ashkin, and A. Cable, “Experimental observation of optically trapped atoms,” *Phys. Rev. Lett.* **57**, 314 (1986).
- [39] V. Boyer, R. M. Godun, G. Smirne, D. Cassetari, C. M. Chandrashekar, A. B. Deb, Z. J. Laczik, and C. J. Foot, “Dynamic manipulation of Bose-Einstein condensates with a spatial light modulator,” *Phys. Rev. A* **73**, 031402 (2006).
- [40] M. P. MacDonald, L. Paterson, K. Volke-Sepulveda, J. Arlt, W. Sibbett, and K. Dholakia, “Creation and manipulation of three-dimensional optically trapped structures,” *Science* **296**, 1101 (2002).

- [41] Y. Tanaka, H. Kawada, S. Tsutsui, M. Ishikawa, and H. Kitajima, “Dynamic micro-bead arrays using optical tweezers combined with intelligent control techniques,” *Opt. Express* **17**, 24102 (2009).
- [42] A. G. Banerjee, S. Chowdhury, W. Losert, and S. K. Gupta, “Real-time path planning for coordinated transport of multiple particles using optical tweezers,” *IEEE Trans. Autom. Sci. Eng.* **9**, 669 (2012).
- [43] J. I. Cirac and P. Zoller, “A scalable quantum computer with ions in an array of micro-traps,” *Nature* **404**, 579 (2000).
- [44] S. Bergamini, B. Darquie, M. Jones, L. Jacubowicz, A. Browaeys, and P. Grangier, “Holographic generation of microtrap arrays for single atoms by use of a programmable phase modulator,” *J. Opt. Soc. Am. B* **21**, 1889 (2004).
- [45] F. K. Fatemi, M. Bashkansky, and Z. Dutton, “Dynamic high-speed spatial manipulation of cold atoms using acousto-optic and spatial light modulation,” *Opt. Express* **15**, 3589 (2007).
- [46] X. He, P. Xu, J. Wang, and M. Zhan, “Rotating single atoms in a ring lattice generated by a spatial light modulator,” *Opt. Express* **17**, 21014 (2009)
- [47] J. Beugnon, C. Tuchendler, H. Marion, A. Gaëtan, Y. Miroshnychenko, Y. R. P. Sortais, A. M. Lance, M. P. A. Jones, G. Messin, A. Browaeys, and P. Grangier, “Two-dimensional transport and transfer of a single atomic qubit in optical tweezers,” *Nat. Phys.* **3**, 696 (2007).
- [48] S. Kuhr, W. Alt, D. Schrader, M. Müller, V. Gomer, and D. Meschede, “Deterministic delivery of a single atom,” *Science* **293**, 278 (2001).
- [49] M. P. MacDonald, L. Paterson, K. Volke-Sepulveda, J. Arlt, W. Sibbett, and K. Dholakia, “Creation and manipulation of three-dimensional optically trapped structures,” *Science* **296**, 1101 (2002).
- [50] Y. Miroshnychenko, W. Alt, I. Dotsenko, L. Förster, M. Khudaverdyan, D. Meschede, D. Schrader, and A. Rauschenbeutel, “An atom-sorting machine,” *Nature* **442**, 151 (2006).
- [51] M. J. Piotrowicz, M. Lichtman, K. Maller, G. Li, S. Zhang, L. Isenhower, and M. Saffman, “Two-dimensional lattice of blue-detuned atom traps using a projected Gaussian beam array,” *Phys. Rev. A* **88**, 013420 (2013).
- [52] N. Schlosser, G. Reymond, and P. Grangier, “Collisional blockade in microscopic optical dipole traps,” *Phys. Rev. Lett.* **89**, 023005 (2002).

- [53] T. Xia, M. Lichtman, K. Maller, A. W. Carr, M. J. Piotrowicz, L. Isenhower, and M. Saffman, “Randomized benchmarking of single-qubit gates in a 2D array of neutral-atom qubits,” *Phys. Rev. Lett.* **114**, 100503 (2015).
- [54] I. Buluta, S. Ashhab, and F. Nori, “Natural and artificial atoms for quantum computation,” *Rep. Prog. Phys.* **74**, 104401 (2000).
- [55] O. Mandel, M. Greiner, A. Widera, T. Rom, T. W. Hänsch, and I. Bloch, “Controlled collisions for multi-particle entanglement of optically trapped atoms,” *Nature* **425**, 937 (2003).
- [56] I. Bloch, “Ultracold quantum gases in optical lattices,” *Nat. Phys.* **1**, 23 (2005).
- [57] M. Schlosser, S. Tichelmann, J. Kruse, and G. Birkl, “Scalable architecture for quantum information processing with atoms in optical micro-structures,” *Quant. Inf. Proc.* **10**, 907 (2011).
- [58] H. Dammann and K. Görtler, “High-efficiency in-line multiple imaging by means of multiple phase holograms,” *Opt. Commun.* **3**, 312 (1971).
- [59] T. Grünzweig, A. Hilliard, M. McGovern, and M. F. Andersen, “Near-deterministic preparation of a single atom in an optical microtrap,” *Nat. Phys.* **6**, 951 (2010).
- [60] D. McGloin, G. C. Spalding, H. Melville, W. Sibbett, and K. Dholakia, “Applications of spatial light modulators in atom optics,” *Opt. Express* **11**, 158-166 (2003).
- [61] H. Kim, W. Lee, H.-g. Lee, H. Jo, Y. Song, and J. Ahn, “In situ single-atom array synthesis by dynamic holographic optical tweezers,” *Nat. Commun.* **7**, 13317 (2016).
- [62] A. Jechow, E. W. Streed, B. G. Norton, M. J. Petrusiunas, and D. Kielpinski, “Wavelength-scale imaging of trapped ions using a phase Fresnel lens,” *Opt. Lett.* **36**, 1371 (2011).
- [63] R. W. Bowman, G. M. Gibson, A. Linnenberger, D. B. Phillips, J. A. Grieve, D. M. Carberry, S. Serati, M. J. Miles, and M. J. Padgett, ““Red Tweezers”: Fast, customisable hologram generation for optical tweezers,” *Comput. Phys. Commun.* **185**, 268 (2014).
- [64] F. O. Fahrbach, F. F. Voigt, B. Schmid, F. Helmchen, and J. Huisken, “Rapid 3D light-sheet microscopy with a tunable lens,” *Opt. Express* **21**, 21010 (2013).
- [65] Y. H. Fung and M. F. Andersen, “Efficient collisional blockade loading of a single atom into a tight microtrap,” *New J. Phys.* **17**, 073011 (2015).
- [66] D. S. Weiss, J. Vala, A. V. Thapliyal, S. Myrgren, U. Vazirani, and K. B. Whaley, “Another way to approach zero entropy for a finite system of atoms,” *Phys. Rev. A* **70**, 040302 (2004).

- [67] J. Vala, A. V. Thapliyal, S. Myrgren, U. Vazirani, D. S. Weiss, and K. B. Whaley, “Perfect pattern formation of neutral atoms in an addressable optical lattice,” *Phys. Rev. A* **71**, 032324 (2005).
- [68] G. Thalhammer, R. W. Bowman, G. D. Love, M. J. Padgett, and M. Ritsch-Marte, “Speeding up liquid crystal SLMs using overdrive with phase change reduction,” *Opt. Express* **21**, 1779 (2013).
- [69] “Munkres’ Assignment Algorithm”, Murray State University (accessed October 27, 2018), <http://http://csclab.murraystate.edu/~bob.pilgrim/445/munkres.html>.
- [70] R. Di Leonardo, F. Ianni, and G. Ruocco, “Computer generation of optimal holograms for optical trap arrays,” *Opt. Express* **15**, 1913 (2007).
- [71] X. He, S. Yu, P. Xu, J. Wang, and M. Zhan, “Combining red and blue-detuned optical potentials to form a Lamb-Dicke trap for a single neutral atom,” *Optics express* **20**, 3711-3724 (2012).
- [72] R. Raussendorf and H. J. Briegel, “A one-way quantum computer,” *Phys. Rev. Lett.* **86**, 5188 (2001).
- [73] T. Grünzweig, A. Hilliard, M. McGovern, and M. F. Andersen, “Near-deterministic preparation of a single atom in an optical microtrap,” *Nat. Phys.* **6**, 951 (2010).
- [74] B. J. Lester, N. Luick, A. M. Kaufman, C. M. Reynolds, and C. A. Regal, “Rapid Production of Uniformly Filled Arrays of Neutral Atoms,” *Phys. Rev. Lett.* **115**, 073003 (2015).
- [75] L. Isenhower, E. Urban, X. L. Zhang, A. T. Gill, T. Henage, T. A. Johnson, T. G. Walker, and M. Saffman, “Demonstration of a neutral atom controlled-NOT quantum gate,” *Phys. Rev. Lett.* **104**, 010503 (2010).
- [76] T. Wilk, A. Gaëtan, C. Evellin, J. Wolters, Y. Miroshnychenko, P. Grangier, and A. Browaeys, “Entanglement of two individual neutral atoms using Rydberg blockade,” *Phys. Rev. Lett.* **104**, 010502 (2010).
- [77] A. M. Kaufman, B. J. Lester, M. Foss-Feig, M. L. Wall, A. M. Rey, and C. A. Regal, “Entangling two transportable neutral atoms via local spin exchange,” *Nature* **527**, 208 (2015).
- [78] Y. Y. Jau, A. M. Hankin, T. Keating, I. H. Deutsch, and G. W. Biedermann, “Entangling atomic spins with a Rydberg-dressed spin-flip blockade,” *Nat. Phys.* **12**, 71 (2016).
- [79] M. J. Piotrowicz, M. Lichtman, K. Maller, G. Li, S. Zhang, L. Isenhower, and M. Saffman, “Two-dimensional lattice of blue-detuned atom traps using a projected Gaussian beam array,” *Phys. Rev. A* **88**, 013420 (2013).

- [80] K. Gillen-Christandl, and B. D. Cosey, “Polarization-dependent atomic dipole traps behind a circular aperture for neutral-atom quantum computing,” *Phys. Rev. A* **83**, 023408 (2011).
- [81] R. Dumke, M. Volk, T. Mütter, F. B. J. Buchkremer, G. Birkl, and W. Ertmer, “Micro-optical realization of arrays of selectively addressable dipole traps: a scalable configuration for quantum computation with atomic qubits,” *Phys. Rev. Lett.* **89**, 097903 (2002).
- [82] M. Greiner, O. Mandel, T. Esslinger, T. W. Hänsch, and I. Bloch, “Quantum phase transition from a superfluid to a Mott insulator in a gas of ultracold atoms,” *Nature* **415**, 39 (2002).
- [83] D. Barredo, S. De Léséleuc, V. Lienhard, T. Lahaye, and A. Browaeys, “An atom-by-atom assembler of defect-free arbitrary 2d atomic arrays,” *Science*, aah3778, (2016).
- [84] M. Endres, H. Bernien, A. Keesling, H. Levine, E. R. Anschuetz, A. Krajenbrink, and M. D. ... Lukin, “Atom-by-atom assembly of defect-free one-dimensional cold atom arrays,” *Science*, aah3752, (2016).
- [85] J. Beugnon, C. Tuchendler, H. Marion, A. Gaëtan, Y. Miroshnychenko, Y. R. Sortais, A. M. Lance, M. P. Jones, G. Messin, A. Browaeys, and P. Grangier, “Two-dimensional transport and transfer of a single atomic qubit in optical tweezers,” *Nat. Phys.* **3**, 696 (2007).
- [86] A. Lengwenus, J. Kruse, M. Schlosser, S. Tichelmann, and G. Birkl, “Coherent transport of atomic quantum states in a scalable shift register,” *Opt. Express* **105**, 170502 (2010).
- [87] H. W. Kuhn, “The Hungarian method for the assignment problem,” *Nav. Res. Logist.* **2**, 83-97 (1955).
- [88] J. Edmonds and R. M. Karp, “Theoretical improvements in algorithmic efficiency for network flow problems,” *J. ACM* **19**, 248-264 (1972).
- [89] P. Hall, “On representatives of subsets,” *J. London Math. Soc.* **1**, 26-30 (1935).
- [90] J. E. Hopcroft and R. M. Karp, “An  $n^{5/2}$  algorithm for maximum matchings in bipartite graphs,” *SIAM J. Comput.* **2**, 225-231 (1973).
- [91] H. Kim, Y. Park, K. Kim, H.-S. Sim, J. Ahn, “Detailed Balance of Thermalization Dynamics in Rydberg-Atom Quantum Simulators,” *Phys. Rev. Lett.* **120**, 180502 (2018).
- [92] K. Singer, J. Stanojevic, M. Weidemüller, and R. Côté, “Long-range interactions between alkali Rydberg atom pairs correlated to the ns-ns, np-np and nd-nd asymptotes,” *J. Phys. B* **38**, S295 (2005).
- [93] E. Urban, T. A. Johnson, T. Henage, L. Isenhower, D. D. Yavuz, T. G. Walker, and M. Saffman, “Observation of Rydberg blockade between two atoms,” *Nat. Phys.* **5**, 110 (2009).

- [94] Y. Miroshnychenko, T. Wilk, A. Chotia, M. Viteau, D. Comparat, P. Pillet, and P. Grangier, “Observation of collective excitation of two individual atoms in the Rydberg blockade regime,” *Nat. Phys.* **5**, 115 (2009).
- [95] T. Wilk, A. Gaëtan, C. Evellin, J. Wolters, Y. Miroshnychenko, P. Grangier, and A. Browaeys, “Entanglement of two individual neutral atoms using Rydberg blockade,” *Phys. Rev. Lett.* **104**, 010502 (2010).
- [96] X. L. Zhang, L. Isenhower, A. T. Gill, T. G. Walker, and M. Saffman, “Deterministic entanglement of two neutral atoms via Rydberg blockade,” *Phys. Rev. A* **82**, 030306 (2010).
- [97] Y. Y. Jau, A. M. Hankin, T. Keating, I. H. Deutsch, and G. W. Biedermann, “Entangling atomic spins with a Rydberg-dressed spin-flip blockade,” *Nat. Phys.* **12**, 71-74 (2016).
- [98] I. Bloch, “Quantum coherence and entanglement with ultracold atoms in optical lattices,” *Nature* **453**, 1016 (2008).
- [99] J. A. Blackmore, L. Caldwell, P. D. Gregory, E. M. Bridge, R. Sawant, J. Aldegunde, J. Mur-Petit, D. Jaksch, J. M. Hutson, B. E. Sauer, M. R. Tarbutt, and S. L. Cornish, “Ultracold molecules for quantum simulation: rotational coherences in CaF and RbCs,” *Quantum Sci. Technol.* **4**, 014010 (2018).
- [100] A. Browaeys, D. Barredo, and T. Lahaye, “Experimental investigations of dipole-dipole interactions between a few Rydberg atoms,” *J. Phys. B: At., Mol. Opt. Phys.* **49**, 152001 (2016).
- [101] H. Kim, Y. Park, K. Kim, H. S. Sim, and J. Ahn, “Detailed Balance of Thermalization Dynamics in Rydberg-Atom Quantum Simulators,” *Phys. Rev. Lett.* **120**, 180502 (2018).
- [102] H. Labuhn, D. Barredo, S. Ravets, S. De Léséleuc, T. Macrì, T. Lahaye, and A. Browaeys, “Tunable two-dimensional arrays of single Rydberg atoms for realizing quantum Ising models,” *Nature* **534**, 667 (2016).
- [103] H. Bernien, S. Schwartz, A. Keesling, H. Levine, A. Omran, H. Pichler, S. Choi, A. S. Zibrov, M. Endres, M. Greiner, V. Vuletic, and M. D. Lukin, “Probing many-body dynamics on a 51-atom quantum simulator,” *Nature* **551**, 579 (2017).
- [104] H. Weimer, M. Müller, I. Lesanovsky, P. Zoller, and H. P. Büchler, “A Rydberg quantum simulator,” *Nat. Phys.* **6**, 382 (2010).
- [105] W. Lee, H. Kim, H., and J. Ahn, “Defect-free atomic array formation using the Hungarian matching algorithm,” *Phys. Rev. A* **95**, 053424 (2017).
- [106] D. Barredo, V. Lienhard, S. De Léséleuc, T. Lahaye, and A. Browaeys, “Synthetic three-dimensional atomic structures assembled atom by atom,” *Nature* **561**, 79 (2018).

- [107] E. Braaten, H. W. Hammer, and G. P. Lepage, “Lindblad equation for the inelastic loss of ultracold atoms,” *Phys. Rev. A* **95**, 012708 (2017).
- [108] G. Lindblad, “On the generators of quantum dynamical semigroups,” *Commun. Math. Phys.* **48**, 119-130 (1976).
- [109] V. Gorini, A. Kossakowski, and E. C. G. Sudarshan, “Completely positive dynamical semigroups of N-level systems,” *J. Math. Phys.* **17**, 821-825 (1976).
- [110] S. De Léséleuc, D. Barredo, V. Lienhard, A. Browaeys, and T. Lahaye, “Analysis of imperfections in the coherent optical excitation of single atoms to Rydberg states,” *Physical Review A* **97**, 053803 (2018).
- [111] L. Béguin, A. Vernier, R. Chicireanu, T. Lahaye, and A. Browaeys, “Direct measurement of the van der Waals interaction between two Rydberg atoms,” *Phys. Rev. Lett.* **110**, 263201 (2013).
- [112] K. M. Maller, M. T. Lichtman, T. Xia, Y. Sun, M. J. Piotrowicz, A. W. Carr, and M. Saffman, “Rydberg-blockade controlled-not gate and entanglement in a two-dimensional array of neutral-atom qubits,” *Phys. Rev. A* **92**, 022336 (2015).
- [113] I. I. Beterov, I. I. Ryabtsev, D. B. Tretyakov, and V. M. Entin, “Quasiclassical calculations of blackbody-radiation-induced depopulation rates and effective lifetimes of Rydberg nS, nP, and nD alkali-metal atoms with  $n \leq 80$ ,” *Phys. Rev. A* **79**, 052504 (2009).

## Acknowledgments in Korean

논문이 완성되기까지 많은 분들의 도움이 있었습니다. 미흡하지만 학위 논문을 마치면서 그 분들께 감사의 말씀을 전합니다. 먼저 이 논문이 있기까지 부족한 저를 하나하나 세심하고 꼼꼼하게 지도해주신 안재욱 교수님께 진심으로 감사드립니다. 또한 바쁘신 와중에 시간을 내서 논문 심사를 해주신 조용훈 교수님, 서민교 교수님, 최형순 교수님, 최재운 교수님께 깊은 감사를 드립니다.

연구실 동료로서 연구에 대한 값진 토론을 나눴던 사람들이 많습니다. 김효섭, 이한결, 한대훈, 임종석, 정희정 박사님, 그리고 한래 형, 윤희, 민혁, 우정, 하은, 경태, 희재에게 감사 인사를 전합니다. 또한 연구에 대한 토론과 더불어 대학원 생활에 힘이 되어준 분들에게도 감사합니다. 과 동기인 기현이 형, 기환이 형, 상진, 성환, 인우 형, 진엽, 오랫동안 관계의 끈을 유지하고 있는 학부 동기들인 철희, 경옥, 지수 형, 미르 형, 주혁이, 고교 동창으로서 만나고 있는 민석, 성환, 영민, 그리고 과 친구이자 동아리 친구들인 한빛, 동하, 지현, 단호에게도 고맙다는 말을 전하고 싶습니다. 마지막으로 학위 기간동안 전적인 지원과 신뢰로 대학원 과정을 버틸 수 있게 해 준 가족에게도 깊은 감사를 전합니다.

## Curriculum Vitae

Name : Woojun Lee  
E-mail : ujun0909@kaist.ac.kr

### Education

2013. 2. – **Integrated Master - Ph.D. Program in Physics**  
*Korea Advanced institute of Science and Technology (KAIST), Daejeon, Korea*  
Research topic: Single atom tweezers and Rydberg interaction for quantum computation.
2007. 2. – 2013. 2. **B.S. in Physics**  
*Korea Advanced institute of Science and Technology (KAIST), Daejeon, Korea*

### Teaching and Tutoring

2015. 9. – 2015. 12. Teaching Assistant, Physics Experiment II, Dept. of Physics, KAIST
2015. 3. – 2015. 12. Teaching Assistant, Applied Physics Experiment, Dept. of Physics, KAIST
2015. 3. – 2015. 6. Teaching Assistant, Classical Mechanics, Dept. of Physics, KAIST
2014. 3. – 2014. 6. Teaching Assistant, General Physics I, Dept. of Physics, KAIST
2013. 3. – 2013. 6. Teaching Assistant, General Physics II, Dept. of Physics, KAIST
2012. 1. – 2013. 5. Tutoring, High school Mathematics, KAIST Voluntary Tutoring Club
2011. 9. – 2011. 12. Tutoring, General Physics II, Dept. of Physics, KAIST

### Research experience

2013. 7. – **Research Assistant**  
*Ultrafast Quantum Optics Lab., Dept of Physics, KAIST, Daejeon, Korea*  
Ultrafast coherent control of alkali atoms, single atom trapping, and Rydberg blockade for quantum computation.
2011. 6. – 2011. 8. **Internship**  
*Korea Astronomy and Space Science Institute (KASI), Daejeon, Korea*  
Participated in Giant Magellan Telescope secondary mirror design.

2011. 9. – 2013. 5. **Research Assistant**

*Convergence Optoelectronic Device Engineering Lab., Dept. of Electrical Engineering, KAIST, Daejeon, Korea*  
Development of Optical ID technology.

## **Publications**

1. Hyosub Kim, Minhyuk Kim, **Woojun Lee**, Jaewook Ahn, “Gerchberg-Saxton algorithm for tweezer-trap atom arrangements,” preprint (2018).
2. Changjun Kim, June-Koo Kevin Rhee, **Woojun Lee**, and Jaewook Ahn, “Mixed Quantum State Dynamics Estimation with Artificial Neural Network,” ICTC 2018, IEEE, 740-747 (2018).
3. **Woojun Lee**, Hyosub Kim, and Jaewook Ahn, “Defect-free atomic array formation using the Hungarian matching algorithm,” *Physical Review A* **95**, 053424 (2017).
4. Hyosub Kim, **Woojun Lee**, Han-gyeol Lee, Hanlae Jo, Yunheung Song, and Jaewook Ahn, “In situ single-atom array synthesis using dynamic holographic optical tweezers,” *Nature Communications* **7**, 13317 (2016).
5. **Woojun Lee**, Hyosub Kim, and Jaewook Ahn, “Three-dimensional rearrangement of single atoms using actively controlled optical microtraps,” *Optics Express* **24**, 9816 (2016).
6. **Woojun Lee**, Hyosub Kim, Kyungtae Kim, and Jaewook Ahn, “Coherent control of resonant two-photon transitions by counter-propagating ultrashort pulse pairs,” *Physical Review A* **92**, 033415 (2015).
7. Hyosub Kim, **Woojun Lee**, Han-gyeol Lee, and Jaewook Ahn, “Nonparaxial aberrations in the optical Talbot effect probed by quantum-dot fluorescence tomography,” *Physical Review A* **91**, 033817 (2015).

## **Conference Presentations**

1. “Density matrix reconstruction through Hamiltonian learning of entangled N-particle systems,” Woojun Lee, Hyosub Kim, and Jaewook Ahn, H11.02, 2018 KPS Spring Meeting, 2018.04.25-2018.04.27.
2. “Storage and retrieval of entanglement states in Rydberg atom chains,” Woojun Lee, Geol Moon, Hyosub Kim, Jaewook Ahn, P1-at.007, 2017 KPS Fall Meeting, 2017.10.25-2017.10.27.
3. “Controlling the Collective Rabi Oscillation of N Rydberg Atoms with Ancillary Atoms,” Woojun Lee, Hyosub Kim, Kyungtae Kim, and Jaewook Ahn, H6.00007, 5-9 June, DAMOP 2017, Sacramento, CA, USA.

4. “Defect-free preparation of two-dimensional single atom arrays by feedback control,” Woojun Lee, Hyosub Kim, Jaewook Ahn, KPS 2016 Fall, Kimdaejoong Convention Center, Gwangju, A2.07, 10/19-21 (10/19), 2016.
5. “Reformation of holographic single-atom arrays for 3D quantum system architectures,” Woojun Lee, Hyosub Kim, Jaewook Ahn, ICAP 2016, Mon-077, Seoul, South Korea, July 24-29, 2016.
6. “Three-Dimensional Dynamic Reconfiguration of Single-Atom Arrays Using Liquid-Crystal Spatial Light Modulator,” Woojun Lee, Hyosub Kim, Jaewook Ahn, CLEO 2016, FM2C.4, San Jose, CA, USA, 6/5-10 (6/6), 2016.
7. “Spectro-Spatial Coherent Control of Ultrafast Laser Interaction with Atomic Vapor,” Woojun Lee, Hyosub Kim, Kyungtae Kim, and Jaewook Ahn, CLEO-PR 2015, BEXCO, Korea, 27G1-4, 8/24-28 (8/27), 2015.

### Honors and Awards

1. Best Oral Presentation, *KPS 2018 Spring Meeting*.
2. Predoctoral Fellowship at *Dept. of Physics, KAIST*, 2018.

### Patent

1. Jaewook Ahn, Hyosub Kim, **Woojun Lee**, “Dynamic holographic single atom tweezers and tweezing method using thereof,” Korea Patent 1017834040000 (2017-09-25).

