

Chapman University

Chapman University Digital Commons

Computational and Data Sciences (PhD)
Dissertations

Dissertations and Theses

Fall 12-2021

Quantum State Estimation and Tracking for Superconducting Processors Using Machine Learning

Shiva Lotfallahzadeh Barzili

Chapman University, lotfa100@mail.chapman.edu

Follow this and additional works at: https://digitalcommons.chapman.edu/cads_dissertations



Part of the [Computational Engineering Commons](#), and the [Quantum Physics Commons](#)

Recommended Citation

S. Barzili, "Quantum state estimation and tracking for superconducting processors using machine learning," Ph.D. dissertation, Chapman University, Orange, CA, 2021. <https://doi.org/10.36837/chapman.000317>

This Dissertation is brought to you for free and open access by the Dissertations and Theses at Chapman University Digital Commons. It has been accepted for inclusion in Computational and Data Sciences (PhD) Dissertations by an authorized administrator of Chapman University Digital Commons. For more information, please contact laughtin@chapman.edu.

Quantum State Estimation and Tracking for Superconducting Processors Using Machine Learning

A Dissertation by

Shiva Lotfallahzadeh Barzili

Chapman University

Orange, CA

Schmid College of Science and Technology

Submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computational and Data Sciences

December 2021

Committee in charge:

Justin Dressel, Ph.D. (Co-Chair)

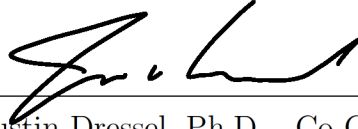
Jeff Tollaksen, Ph.D. (Co-Chair)

Matthew Leifer, Ph.D.

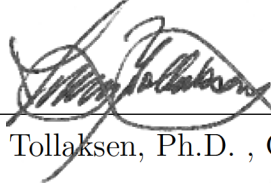
Andrew Jordan, Ph.D.



The dissertation of Shiva Lotfallahzadeh Barzili is approved.



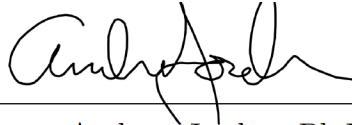
Justin Dressel, Ph.D. , Co-Chair



Jeff Tollaksen, Ph.D. , Co-Chair



Matthew Leifer, Ph.D.



Andrew Jordan, Ph.D.

September 2021

Quantum State Estimation and Tracking
for Superconducting Processors Using Machine Learning

Copyright © 2021

by Shiva Lotfallahzadeh Barzili

ACKNOWLEDGMENTS

At the end of this wonderful journey, I would like to thank everyone who has supported me and helped me through this process. First, I would like to acknowledge and thank Dr. Justin Dressel for all the opportunities he has given me throughout my years at Chapman University, and for patiently guiding me toward the right direction whenever I needed. This dissertation would not have been possible without his supervision and support.

I thank my dissertation committee members, Dr. Matthew Leifer, Dr. Andrew Jordan, and Dr. Jeff Tollaksen, for their comments and feedback that have helped to improve the quality of this dissertation. I also want to thank them for being extremely flexible with respect to the scheduling of the defense. I am deeply grateful to Dr. Hesham El-Askary for providing helpful advice and support many times during my graduate studies.

I am thankful to the collaborators whose expertise and intuition helped me throughout my research: Dr. Razieh Mohseninia, Luke Burns, Sacha Greenfield, at University of California, Berkeley: Dr. Irfan Siddiqi, Dr. Gerwin Koolstra, and Noah Stevenson, and the collaborators at Google: Dr. Alexander Korotkov, and Dr. Mostafa Khezri.

I was fortunate to be surrounded by many good friends during my graduate studies. I would like to thank Ismael for being a supportive friend from the beginning and confusing me sometimes by explaining things in his own unique way. Cai, for all his support and positive energy. Nancy, for her friendship, and delicious foods. Mina, for being such a kind and caring friend during my studies and hopefully for the rest of my life.

I would like to thank my family, Baba, Maman, Bitu, Asma, and Alireza for the continuous support and unconditional love they have given me throughout my time in graduate school; I could not have done it without them. Finally, there are not enough words to thank my husband, Salar, for his unwavering support and constant encouragement.

ABSTRACT

Quantum State Estimation and Tracking
for Superconducting Processors Using Machine Learning

by Shiva Lotfallahzadeh Barzili

Quantum technology has been rapidly growing; in particular, the experiments that have been performed with superconducting qubits and circuit QED have allowed us to explore the light-matter interaction at its most fundamental level. The study of coherent dynamics between two-level systems and resonator modes can provide insight into fundamental aspects of quantum physics, such as how the state of a system evolves while being continuously observed. To study such an evolving quantum system, experimenters need to verify the accuracy of state preparation and control since quantum systems are very fragile and sensitive to environmental disturbance. In this thesis, I look at these continuous monitoring and state estimation problems from a modern point of view. With the help of machine learning techniques, it has become possible to explore regimes that are not accessible with traditional methods: for example, tracking the state of a superconducting transmon qubit continuously with dynamics fast compared with the detector bandwidth. These results open up a new area of quantum state tracking, enabling us to potentially diagnose errors that occur during quantum gates. In addition, I investigate the use of supervised machine learning, in the form of a modified denoising autoencoder, to simultaneously remove experimental noise while encoding one and two-qubit quantum state estimates into a minimum number of nodes within the latent layer of a neural network. I automate the decoding of these latent representations into positive density matrices and compare them to similar estimates obtained via linear inversion and maximum likelihood estimation. Using a superconducting multiqubit chip, I experimentally verify that the neural network estimates the quantum state with greater fidelity than either traditional method. Furthermore, the network can be trained using only product states and still achieve high fidelity for entangled states. This simplification of the

training overhead permits the network to aid experimental calibration, such as the diagnosis of multi-qubit crosstalk. As quantum processors increase in size and complexity, I expect automated methods such as those presented in this thesis to become increasingly attractive.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	IV
ABSTRACT	V
LIST OF TABLES	X
LIST OF FIGURES	XI
1 INTRODUCTION	1
2 BACKGROUND TOPICS IN QUANTUM MECHANICS	7
2.1 A Brief Introduction to Linear Algebra for Quantum Computation	7
2.1.1 State vectors	7
2.1.2 Dirac Notation	8
2.1.3 Operators and Observables in Quantum Mechanics	9
2.1.4 Uncertainty Principle	11
2.1.5 Time Evolution of a Quantum System	12
2.1.6 Heisenberg Picture	14
2.1.7 Rotating Frames and the Interaction Picture	15
2.1.8 Tensor Product and Partial Trace	15
2.2 Qubit and Bloch Sphere	16
2.2.1 Pauli Matrices	18
2.2.2 Density Matrix	18
2.2.3 Harmonic Oscillators and Coherent States	19
2.3 Rabi Oscillations and Jaynes-Cummings Model	25
2.3.1 Rabi Oscillations	25
2.3.2 The Jaynes-Cummings Model	27

3	SUPERCONDUCTING QUBITS AND CIRCUIT QED	31
3.1	Quantum LC Oscillator	34
3.2	Josephson Junctions	36
3.3	Superconducting Qubits	39
3.3.1	Transmon Qubit	39
3.4	Resonator Coupling	42
4	MONITORING QUBIT EVOLUTION	47
4.1	Projective Measurement	48
4.2	Generalized Measurement	50
4.3	Repeated Measurement and Monitoring	52
4.3.1	Discrete Monitoring: Bayesian Trajectories	53
4.3.2	Continuous Monitoring: Stochastic Master Equations	54
5	MACHINE LEARNING FOR DATA PROCESSING	58
5.1	A Brief History of Neural Networks	59
5.2	The Building Blocks of Neural Networks	60
5.3	Training Neural Networks	64
5.4	Backpropagation	67
5.5	Cross Validation	68
5.6	Overfitting	69
5.7	Long Short Term Memory Recurrent Neural Networks	70
5.8	Autoencoders	74
6	MONITORING FAST SUPERCONDUCTING QUBIT DYNAMICS USING A NEU- RAL NETWORK	77
6.1	Motivation	77
6.2	Circuit QED Model	78
6.3	Analytic Quantum Trajectory Reconstruction	81

6.4	Training of LSTM RNN	85
6.5	Qubit Trajectory Reconstruction with a Neural Network vs. a Bayesian Update	87
6.6	Resonator Memory Corrections To Qubit Trajectories	91
6.7	Time-Dependent Rabi Drive	93
7	STATE TOMOGRAPHY WITH A DENOISING AUTOENCODER	97
7.1	Motivation	97
7.2	Traditional State Tomography Strategies	98
7.2.1	Preliminary Definitions	99
7.2.2	Linear inversion (LI)	100
7.2.3	Maximum Likelihood Estimation (MLE)	101
7.3	Denoising Autoencoder Model and Methods	102
7.4	Data Preparation for Denoising Autoencoder	105
7.5	Measurement Strategies	108
7.6	Results	110
7.6.1	Numerical Simulation of One-Qubit System	110
7.6.2	Experimental Data of a Planar Multi-Transmon Qubit	116
7.6.3	Two-Qubit System	117
8	CONCLUSIONS AND OUTLOOKS	122
	REFERENCES	124

LIST OF TABLES

7.1	The average fidelity vs. number of nodes in latent layer.	113
7.2	Fidelity of five different measurement strategies of simulated data.	115
7.3	Fidelity of five different measurement strategies of experimental data.	117
7.4	Fidelity vs. Number of Nodes in Latent Layer for Deep Network	120

LIST OF FIGURES

2.1	Qubit state representation on the Bloch sphere	17
2.2	Wavefunctions and energies of the first few harmonic oscillator eigenstates . . .	22
2.3	Representation of the state space occupied by the coherent state $ \alpha\rangle$	24
2.4	Rabi oscillations in the lab frame vs. the rotating frame	28
2.5	Dressed states vs bare states	30
3.1	LC Oscillator	34
3.2	A Josephson Junction	36
3.3	LJ Oscillator	38
3.4	Cosine potential well of the transmon qubit.	41
3.5	Experimental Setup of cQED	43
4.1	Stern-Gerlach experiment	49
5.1	The architecture of a feedforward neural network with one hidden layer.	61
5.2	Activation functions	63
5.3	An example of a gradient descent search for a minimum	66
5.4	The architecture of recurrent neural network.	71
5.5	Long Short Term Memory (LSTM) cell	73
5.6	Denoising autoencoder architecture	76
6.1	Schematic setup of a driven superconducting qubit subject to continuous measurement.	79
6.2	Cross-validation cost vs epochs	86
6.3	Breakdown of Markovian trajectory reconstruction	88
6.4	Phase difference in the voltage records	89
6.5	QuTiP master equation simulation of a driven qubit dispersively coupled to the readout resonator	90
6.6	Unraveling Lindblad trajectory dynamics from LSTM trajectories	91
6.7	Measurement back-action effects due to a narrow linewidth cavity	94

6.8	Time-dependent Rabi drive	95
7.1	Training DAE vs. testing DAE	104
7.2	Tomographic denoising autoencoder	105
7.3	Three different measurement bases	107
7.4	Comparing five different strategies	108
7.5	Fidelity heatmap	111
7.6	Fidelity vs. training ensemble size	113
7.7	State representation inside latent layer	114
7.8	Fidelity of one qubit experimental data	116
7.9	Two qubits shallow denoising autoencoder performance	119

1 Introduction

The idea of quantum computing first originated with Paul Benioff in 1980, who proposed a quantum mechanical model for the Turing machine [1], a minimal yet universal computer designed originally with classical physics. Then after that in 1982, Richard Feynman realized that with a classical computer simulating quantum systems would require exponentially large computational resources, but quantum computers could in theory overcome this challenge since they also would have similarly complex structure [2]. Nobody knew what such a quantum computer might look like until 1985, when David Deutsch proposed the mathematical framework of a quantum Turing machine to model quantum computation [3]. A theoretical foundation for quantum computing was laid for the first time by this work.

These developments raised the question of whether quantum computers could actually outperform classical computers, which was quickly answered by a sequence of ground-breaking papers. In 1992, David Deutsch and Richard Jozsa give the first example of quantifiable advantage by proposing a quantum algorithm to efficiently distinguish balanced functions from constant functions [4]. In 1996, Grover designed a quantum algorithm for faster search in unsorted databases [5]. In 1997, Ethan Bernstein and Umesh Vazirani demonstrated that for the problem of learning a secret bit-string encoded in an unknown function, quantum computers could be significantly faster than known classical algorithms, even if classical computers were allowed a small probability of error [6].

These developments were provocative, but did not settle the question whether a quantum computer could demonstrate any advantage when solving a practical problem of interest. In 1999, Peter Shor settled this debate by showing that quantum computers can factor large integers efficiently, a task for which there is no known efficient classical algorithm [7]. Another development of algorithmic importance came in 1996 from Seth Lloyd, who developed a quantum algorithm for simulating quantum mechanical systems [8], which was a first step

toward realizing the quantum simulation ideas of Feynman. Since that time, simulation of quantum mechanics has become one of the most promising uses of a quantum computer, with many potential applications, such as drug design [9, 10], materials science [11], and high-energy physics [12].

These potential applications started a surge of interest in quantum computing and in developing hardware that could implement the proposed quantum algorithms. Developing the hardware rapidly converged to identifying suitable physical implementations of quantum bits (qubits) that are the basic units of quantum information, which could be realized in principle by any two-level quantum system that could be both protected and controlled. A qubit is the quantum version of the classic binary bit; however, whereas the state of a bit can only be either 0 or 1, the general state of a qubit can be a superposition of both [13] and thus encode a much richer set of possibilities.

Although quantum computing was a promising technology with potentially powerful computational capabilities, actually building a large-scale quantum system would involve several imposing challenges in terms of fabrication, verification, and architecture [14–16]. These difficulties arise due to the fragility of quantum systems (e.g. qubits) and the rapid degradation of quantum effects when the hardware inevitably interacts with a noisy environment. Individual quantum systems like a qubit can be built in two ways: naturally and artificially. Photons, single atoms, and single ions are examples of natural quantum systems with distinguishable and controllable quantum states that can serve as qubits. Although they can make a system which is very strongly quantum mechanical however in the end, they are small and have limited design flexibility. In contrast, artificial atoms could be made much larger and be engineered to have customized properties, but were harder to fabricate as precisely [17–20].

Despite these challenges, quantum hardware development has been remarkably fast over the last two decades. In 1999, Yasunobu Nakamura and Jaw-Shen Tsai demonstrated that

a superconducting circuit could be used as a quantum bit [21]. In 2003, Stephan Gulde, Mark Riebe, Gavin P T Lancaster et al implemented the Deutsch-Jozsa algorithm using the hyperfine ground states of ions that were trapped and controlled with lasers [22]. An important milestone was passed in 2004 when Schoelkopf’s group at Yale reproduced with superconducting circuits [23, 24] the experiment performed in cavity quantum electrodynamics [25, 26], a field of atomic physics in which the individual atoms interact with photons stored in a resonant cavity.

The behavior of these new superconducting circuits had many similarities to the atom-laser interactions investigated in the 1960’s, which greatly accelerated further hardware development. In particular, the optical cavity quantum electrodynamics formalism developed for atom-laser interactions was readily adapted into electrical circuit quantum electrodynamics (cQED) suitable for these new superconducting processors. The atom in cavity QED is replaced by a an artificial atom created from electrical circuit components, such as a Cooper pair box or charge qubit [21]. Similarly, the optical cavity is replaced by a resonant tank circuit, such as a lumped element LC oscillator or a quarter-wavelength resonant waveguide. In 2007 the persistent issue of environmental charge noise was addressed by the introduction of a modified charge qubit design, named the transmon qubit, which was developed by Koch et al. The transmon design was very simple but significantly improved the coherence, so it and its variations have now become the most widely used superconducting qubits in practice¹ [27, 28].

Progress in superconducting hardware design rapidly accelerated after the introduction of the transmon, scaling up to multi-qubit processors with individual control wires and readout with resonant microwave signals. Cutting edge designs by companies like Google, IBM, and Rigetti Computing are based on these superconducting transmon architectures, and have

1. Commonly used as a quality metric, the coherence time is the time constant for a qubit to exponentially decay from a quantum superposition state to a classical probabilistic bit state.

dozens of fully controllable qubits fabricated onto layered 2D planar chips.

A close interaction between engineering and science in superconducting circuit technology has prompted active research from a variety of different perspectives, ranging from new developments in fundamental physics to novel practical applications for actively controllable quantum effects. In particular, significant progress has been made in understanding the physics of open quantum systems and the subtleties of the measurement process [29–32]. All physical systems are not perfectly isolated, so are open to interactions with a complex and usually uncontrolled environment.

Information stored in the system can be transferred to the environment and lost, which manifests on average as decoherence and dissipation affecting the quantum system. However, from the point of view of quantum measurement, if we are able to monitor the information that was transferred to the environment, then we can reconstruct a more accurate picture of how the quantum system coherently evolves [33, 34].

In fact, measurement on a quantum system can be used as a resource for feedback control, enabling conditional customization of the dynamics of the system [35–37]. Specifically, monitoring slow information loss from the system produces a weaker measurement effect, so can allow near continuous monitoring of the dynamics of a quantum system while largely preserving its quantum coherence [38–40]. The resulting estimates for the quantum state dynamics of these continuously monitored systems are called quantum trajectories. Due to their applications in feedback control and quantum system characterization, there is considerable interest in improving both the accuracy and efficiency of such quantum trajectory estimations.

Machine learning is starting to attract attention in the quantum science community as a useful tool for managing the complexity of high-dimensional quantum systems [41–46]. Machine learning aims to identify complex patterns in data, match those patterns to a suitable

model describing the data, and predict future behaviors. In many ways, this data analysis and modeling process is the same as that used in physics, but often with fewer assumptions of principles or other structure. It is natural to ask whether machine learning methods can offer potential advantages for characterizing and controlling quantum systems, compared to the standard methods used in the laboratory. The high degree of complexity of realistic quantum processors makes it difficult to predict how to optimally compress the relevant information to a tractable form, but this sort of task is precisely what machine learning is well suited for.

This thesis explores several applications of machine learning to superconducting quantum computation. Specifically, I apply machine learning models based on neural network architectures [47, 48] to estimate and track the states of a two-dimensional transmon qubit. The work presented here was carried out in collaboration with the experimental group of Dr. Irfan Siddiqi at the University of California, Berkeley.

This work is structured as follows,

Chapter 2 introduces quantum mechanics and its key concepts related to this work, including a brief introduction to the linear algebra of quantum computing, the qubit and Bloch representation of it, the qubit density matrices, qubit as a harmonic oscillator, and qubit dynamics in the presence of drive fields.

Chapter 3 is about superconductivity and circuit quantum electrodynamics. The focus in this section is to connect electrical engineering intuition of classical circuits with a quantization based quantum optics approach. After introducing superconducting qubits, specifically the transmon architecture, the role of open quantum systems in the qubit-resonator coupling required to measure information about the system is discussed.

Chapter 4 provides a basic theoretical discussion about quantum measurement of open systems, then extends this to simple models for continuously monitoring the qubit states.

Chapter 5 can be used as an independent introduction to machine learning, tailored to the

topics relevant to this work.

Chapter 6 revisits monitoring the superconducting qubit dynamics and quantum state tomography in an automated way using neural networks. Notably these new tools allow going beyond the traditional adiabatic coupling regime to track rapidly driven superconducting qubit trajectories more precisely in the presence of rapid dynamics.

Chapter 7 discusses the standard methods to estimate the state of a quantum system, then introduces machine learning tools to automate the estimation of a quantum state.

Chapter 8 concludes and discusses possible future directions.

2 Background Topics in Quantum Mechanics

2.1 A Brief Introduction to Linear Algebra for Quantum Computation

2.1.1 State vectors

In quantum mechanics the state of a physical system is a vector in a complex vector space V . In a vector space, also called a linear space, vectors can be added together and multiplied by numbers, called scalars. A vector space can be infinite in dimension, but in most application in quantum computation, finite dimensional vector spaces are used and are denoted by $\mathbb{C}^n$, which is the space of vectors of positive integer dimension n having n complex entries. A basis for $\mathbb{C}^n$ consists of exactly n linearly independent vectors. If $V = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ is a basis for $\mathbb{C}^n$, then a generic vector can be written as,

$$\mathbf{v} = \sum_{i=1}^n a_i \mathbf{v}_i, \quad (2.1)$$

where the coefficients $a_i \in \mathbb{C}$ are complex numbers.

The inner product is a operation, denoted with $\langle \cdot, \cdot \rangle$. There are different ways to define an inner product on a vector space. If,

$$\mathbf{v} = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}, \quad (2.2)$$

then, $\langle \mathbf{v}, \mathbf{u} \rangle = \sum_{i=1}^n a_i^* b_i$.

In a vector space one can define orthogonality in the following way: two vectors are orthog-

onal if their inner product is zero. The length of a vector or norm of a vector can be defined also using the inner product as $\|\mathbf{v}\| = \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle}$. A normalized vector is a vector with norm 1. A basis is called orthonormal if all vectors are normalized and mutually orthogonal.

2.1.2 Dirac Notation

Dirac invented a useful alternative notation which is introduced in the context of quantum mechanics to help simplify some manipulations of linear algebra. For a vector, Dirac notation uses $\mathbf{v} = |v\rangle$ and the terminology ‘ket’ with its components arranged as a column vector. Its associated linear functional, or dual vector is denoted by $\langle v|$ and is called a ‘bra’ with components conjugated and transposed to a row vector. The ket and bra vectors have the explicit matrix forms

$$|v\rangle = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}, \quad \langle v| = \begin{bmatrix} a_1^* \dots a_n^* \end{bmatrix}. \quad (2.3)$$

The application of the functional $\langle v|$ to the vector $|v\rangle$ is denoted by $\langle v|v\rangle = \sum_{i=1}^n a_i^* a_i$ and is equivalent to the matrix product between the row and column matrix representations. The contraction ‘bra-ket’ is used for such an inner product. If $\{|v_1\rangle, \dots, |v_n\rangle\}$ is an orthonormal basis, then,

$$\langle v_i|v_j\rangle = \delta_{ij} \quad (2.4)$$

where δ_{ij} is the Kronecker delta. The Kronecker delta function is 1 if the variables are equal, and 0 otherwise,

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \quad (2.5)$$

It is also very common to use matrix product of a column vector with a row vector $|v\rangle\langle v|$,

which is known as the outer product and its result is an $n \times n$ matrix.

2.1.3 Operators and Observables in Quantum Mechanics

In quantum mechanics an observable is any property of a system that can be measured, such as position x , momentum p , energy E , and so on. An operator on the other hand is a rule for changing vectors into other vectors. Associated with each observable there is a quantum mechanical Hermitian operator. Operators are distinguished by hats on top of symbols. Some common operators that we will encounter throughout this thesis are,

- **Unitary operator**, an operator $\hat{U}$ is unitary if $\hat{U}^\dagger \hat{U} = \hat{U} \hat{U}^\dagger = \hat{\mathbb{I}}$.
- **Hermitian operator**, an operator $\hat{A}$ is Hermitian if $\hat{A}^\dagger = \hat{A}$.
- **Orthogonal projector**, an operator $\hat{P}$ is an projector if $\hat{P}^2 = \hat{P}$ and $\hat{P}^\dagger = \hat{P}$.
- **Positive operator**, an operator $\hat{A}$ is positive if $\langle v | \hat{A} | v \rangle \geq 0$ for any $|v\rangle \in \mathcal{H}$.

Any Hermitian operator $\hat{\mathcal{O}}$ on a vector space V can be written as a diagonal operator in a particular orthonormal eigenbasis $\{|v_i\rangle\}$ called its spectral basis,

$$\hat{\mathcal{O}} = \sum_{i=1}^n \lambda_i |v_i\rangle \langle v_i|, \quad (2.6)$$

where the complex numbers λ_i are the eigenvalues of $\hat{\mathcal{O}}$ and $|v_i\rangle$ are its corresponding eigenvectors. For a continuous basis Eq. (2.6) becomes,

$$\hat{\mathcal{O}} = \int \lambda(v) |v\rangle \langle v| dv. \quad (2.7)$$

According to the Born rule, if an observable corresponding to an operator $\mathcal{O}$ is measured

in a system with normalized vector $|\psi\rangle$ in the Hilbert space¹, the probability of finding the system in the state $|v_i\rangle$ is given by,

$$P(v_i|\psi) = |\langle v_i|\psi\rangle|^2. \quad (2.8)$$

Note that the probability in Eq. (2.8) does not change if we replace $|\psi\rangle$ with $e^{i\phi}|\psi\rangle$, which implies that the vectors that differ only by such a global phase are physically indistinguishable.

The change of the state from $|\psi\rangle = \sum_i \psi_i |v_i\rangle$ to $|v_i\rangle$ is what is often referred to as the collapse of the state caused by measuring $\hat{\mathcal{O}}$. The average over repeated measurements of the observable $\hat{\mathcal{O}}$ over many systems independently prepared in the state $|\psi\rangle$ gives us the expectation value of observable $\hat{\mathcal{O}}$,

$$\langle \mathcal{O} \rangle = \langle \psi | \hat{\mathcal{O}} | \psi \rangle = \sum_{i,j} \lambda_j \psi_i^* \psi_j \langle v_i | v_j \rangle = \sum_j \lambda_j \psi_j^* \psi_j = \sum_j \lambda_j P(v_i|\psi). \quad (2.9)$$

For a continuous basis, Eq. (2.9) changes to,

$$\langle \mathcal{O} \rangle = \langle \psi | \hat{\mathcal{O}} | \psi \rangle = \int \mathcal{O}(x) \psi^*(x) \psi(x) dx = \int \mathcal{O}(x) P(x|\psi) dx \quad (2.10)$$

If we have more than one operator acting on a state, the order of operations matters. The *commutator* of two operators $\hat{A}$ and $\hat{B}$ is,

$$[\hat{A}, \hat{B}] \equiv \hat{A} \hat{B} - \hat{B} \hat{A}. \quad (2.11)$$

1. A Hilbert space is a Banach space with an inner product. A Banach space is a normed vector space that is complete, meaning that it contains the limit points of all convergent sequences of vectors.

The position $\hat{x}$ and momentum $\hat{p}$ operators satisfy the canonical commutation relation [49],

$$[\hat{x}, \hat{p}] \equiv \hat{x}\hat{p} - \hat{p}\hat{x} = i\hbar\hat{\mathbb{I}} \quad (2.12)$$

where $\hat{\mathbb{I}}$ is the identity operator and $\hbar$ is the Planck constant.

2.1.4 Uncertainty Principle

The uncertainty principle, also known as Heisenberg's uncertainty principle, states that no preparation of a quantum system can determine precise predictions for both a measurement of its position and for a measurement of its momentum simultaneously [50]. We saw previously that the position operator $\hat{x}$ and the momentum operator $\hat{p}$ do not commute $[\hat{x}, \hat{p}] = i\hbar\hat{\mathbb{I}}$. We also saw that given a quantum state, we can compute expectation values for both $\hat{x}$ and $\hat{p}$. The standard deviation of an observable is

$$\Delta x = \sqrt{\langle x^2 \rangle - \langle x \rangle^2}, \quad \text{and} \quad \Delta p = \sqrt{\langle p^2 \rangle - \langle p \rangle^2}. \quad (2.13)$$

The Heisenberg uncertainty principle states that,

$$\Delta x \Delta p \geq \frac{\hbar}{2}. \quad (2.14)$$

Inequality (2.14) generalizes to any pair of Hermitian (i.e. $\hat{A}^\dagger = \hat{A}$) operators $\hat{A}$ and $\hat{B}$. The two operators' commutator provides the lower bound on the product of standard deviations

$$\Delta A \Delta B \geq \frac{|\langle [\hat{A}, \hat{B}] \rangle|}{2}. \quad (2.15)$$

2.1.5 Time Evolution of a Quantum System

In quantum mechanics the evolution of a closed system is unitary and governed by the time-dependent Schrödinger equation

$$i\hbar \frac{\partial |\psi(t)\rangle}{\partial t} = \hat{H} |\psi(t)\rangle, \quad (2.16)$$

where $\hat{H}$ is the Hamiltonian operator of the system, and $\hbar$ is the reduced Planck's constant. The statement that the evolution of a closed system is unitary means that the state of the system at a later time t is given by

$$|\psi(t+t_0)\rangle = \hat{U}(t) |\psi(t_0)\rangle, \quad (2.17)$$

where $\hat{U}(t)$ is a unitary operator. Substituting Eq. (2.17) into Eq. (2.16) leads to,

$$\hat{U}(t) = e^{-i\hat{H}t/\hbar}. \quad (2.18)$$

The Hamiltonian is,

$$\hat{H} = \frac{1}{2m} \hat{p}^2 + \hat{V}(\hat{\mathbf{r}}), \quad (2.19)$$

where $\hat{p}$ is the canonical momentum operator and $\hat{V}$ is the potential energy operator. The time-dependent Schrödinger equation for a state $|\psi(t)\rangle$ in the position basis is,

$$i\hbar \frac{\partial \psi(\mathbf{r}, t)}{\partial t} = \left[-\frac{\hbar^2}{2m} \nabla^2 + \hat{V}(\mathbf{r}) \right] \psi(\mathbf{r}, t). \quad (2.20)$$

Here $\hat{p} = -i\hbar \nabla$ (∇ is the gradient vector that stores all the partial derivative information for every variable.).

Note that in a closed system $|\psi\rangle$ is referred to as a pure state. However, $|\psi\rangle$ is not the

most general way defining a quantum state as we may not have always complete knowledge of what state the system is in but only know the probabilities of it being in various states (mixed state). Therefore our description of the state and predictions for the results of any measurements on it must take account of this lack of knowledge. The way to do this is through the density operator or density matrix. The density operator is defined by,

$$\hat{\rho} \equiv \sum_i P_i |\psi_i\rangle\langle\psi_i|, \quad (2.21)$$

where pure state $|\psi_i\rangle$ is prepared with probability P_i , and the probabilities must sum to one, $\sum_i P_i = 1$. $\hat{\rho}$ is a Hermitian matrix, with trace 1, and non-negative eigenvalues. Any matrix that satisfies these three conditions is a valid density matrix.

In the case of a mixed state, the expectation value of an observable is given by weighting the expectation value for each state by the probability of that state,

$$\langle\mathcal{O}\rangle = \sum_i P_i \langle\psi_i|\hat{\mathcal{O}}|\psi_i\rangle. \quad (2.22)$$

In terms of the density operator of Eq. (2.21), Eq. (2.22) can be written as

$$\begin{aligned} \langle\mathcal{O}\rangle &= \sum_i \sum_{mn} P_i \langle\psi_i|n\rangle \langle n|\hat{\mathcal{O}}|m\rangle \langle m|\psi_i\rangle \\ &= \sum_{mn} \rho_{mn} \mathcal{O}_{mn} = \text{Tr}(\hat{\rho}\hat{\mathcal{O}}). \end{aligned} \quad (2.23)$$

The time evolution of the density matrix follows naturally from the definition of $\hat{\rho}$ and the

time-dependent Schrödinger equation, Eq. (2.16),

$$\begin{aligned}
\frac{\partial \hat{\rho}}{\partial t} &= \sum_j P_j \frac{\partial}{\partial t} [|\psi_j\rangle\langle\psi_j|] \\
&= \sum_j P_j \left(\left[\frac{\partial}{\partial t} |\psi_j\rangle \right] \langle\psi_j| + |\psi_j\rangle \frac{\partial}{\partial t} \langle\psi_j| \right) \\
&= \sum_j P_j \left(-\frac{i}{\hbar} \hat{H} |\psi_j\rangle\langle\psi_j| + \frac{i}{\hbar} |\psi_j\rangle\langle\psi_j| \hat{H} \right) \\
&= -\frac{i}{\hbar} [\hat{H}, \hat{\rho}].
\end{aligned} \tag{2.24}$$

and is known as the Liouville - Von Neumann equation.

2.1.6 Heisenberg Picture

So far we have discussed the time evolution in the Schrödinger picture, where state $|\psi(t)\rangle$ evolves according to the Schrödinger equation while operators are time-independent. There is another way of thinking about the dynamics in quantum mechanics besides the Schrödinger picture, and that is the Heisenberg picture. In the Heisenberg picture, the state is independent of time, but the operators are time dependent. We define the Heisenberg operator by,

$$\hat{\tilde{\mathcal{O}}}(t) = \hat{U}^\dagger(t) \hat{\mathcal{O}} \hat{U}(t), \tag{2.25}$$

where tilde is used to contrast the time-independent operators (Schrödinger representation), $\hat{\mathcal{O}}$, with the time-dependent operators (Heisenberg representation), $\hat{\tilde{\mathcal{O}}}$. Note that the two representations are completely equivalent, and it is a matter of convenience which one is used in a given problem.

The time-dependent operators are governed by a differential equation known as Heisenberg equation of motion [51],

$$\frac{d\hat{\tilde{\mathcal{O}}}(t)}{dt} = \frac{i}{\hbar} [\hat{H}, \hat{\tilde{\mathcal{O}}}(t)] + \frac{\partial \hat{\tilde{\mathcal{O}}}(t)}{\partial t}. \tag{2.26}$$

2.1.7 Rotating Frames and the Interaction Picture

A useful application of a rotating frame is to eliminate the time-dependence of certain Hamiltonians. This is what we do when we switch to an accelerating reference frame, like a rotating planet in classical mechanics.

If we consider a system with density matrix $\hat{\rho}$ which is evolving according to von Neumann's Equation (2.24) but with a time-dependent Hamiltonian $\hat{H}(t)$, for simplicity we can always move to a *rotating frame* by defining a new density matrix,

$$\hat{\tilde{\rho}}(t) = \hat{U}(t) \hat{\rho} \hat{U}^\dagger(t), \quad (2.27)$$

where $\hat{U}(t)$ is an arbitrary unitary. $\hat{\tilde{\rho}}(t)$ will also obey a von Neumann equation

$$\frac{\partial \hat{\tilde{\rho}}(t)}{\partial t} = -\frac{i}{\hbar} [\hat{\tilde{H}}(t), \hat{\tilde{\rho}}(t)], \quad (2.28)$$

but with an effective Hamiltonian

$$\hat{\tilde{H}}(t) = i \hbar \frac{d\hat{U}(t)}{dt} \hat{U}^\dagger(t) + \hat{U}(t) \hat{H} \hat{U}^\dagger(t). \quad (2.29)$$

2.1.8 Tensor Product and Partial Trace

Composite quantum systems are described by the tensor product of their individual descriptions as Hilbert spaces. If a state $|\psi\rangle_A$ belongs to Hilbert space $\mathcal{H}_A$ and a state $|\psi\rangle_B$ belongs to $\mathcal{H}_B$, then their joint state is,

$$|\psi\rangle_{AB} = |\psi\rangle_A \otimes |\psi\rangle_B, \quad (2.30)$$

where $|\psi\rangle_{AB} \in \mathcal{H}_A \otimes \mathcal{H}_B$. A state written in this form is called a product or a separable state. If a state is not a product state, it is called an entangled or non-separable state. In this case, the state associated with each subsystem can be written with the definition of a partial trace. The partial trace over the subsystem B , denoted by Tr_B , is defined as,

$$\hat{\rho}_A \equiv \text{Tr}_B[\hat{\rho}_{AB}] \equiv \sum_i (\hat{\mathbb{I}}_A \otimes \langle i|_B) \hat{\rho}_{AB} (\hat{\mathbb{I}}_A \otimes |i\rangle_B), \quad (2.31)$$

where $\{|i\rangle_B\}$ is an orthonormal basis for the Hilbert space $\mathcal{H}_B$ of subsystem B .

2.2 Qubit and Bloch Sphere

Quantum bits are the smallest and most fundamental units of quantum information. A qubit is any system that may be described by a two-dimensional complex vector space (Hilbert space) $\mathbb{C}^2$. We call the preferred vector basis of logical states the computational basis, and write the spanning vectors in Dirac notation as $|0\rangle$ and $|1\rangle$ (or $|g\rangle$ and $|e\rangle$, that denote the qubit ground and excited states, respectively).

If the qubit state is perfectly known, it is in a pure state represented by a normalized complex vector

$$|\psi_q\rangle = \alpha |g\rangle + \beta |e\rangle \quad (2.32)$$

where α and β are complex coefficients which satisfy the condition $|\alpha|^2 + |\beta|^2 = 1$ and $|g\rangle$ and $|e\rangle$ are

$$|g\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |e\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (2.33)$$

To describe any qubit state one needs four real numbers, two for α and two for β , since they are complex numbers. The constraint $|\alpha|^2 + |\beta|^2 = 1$ reduces it to three real numbers. Moreover two qubits that differ only in global phase are indistinguishable. By eliminating

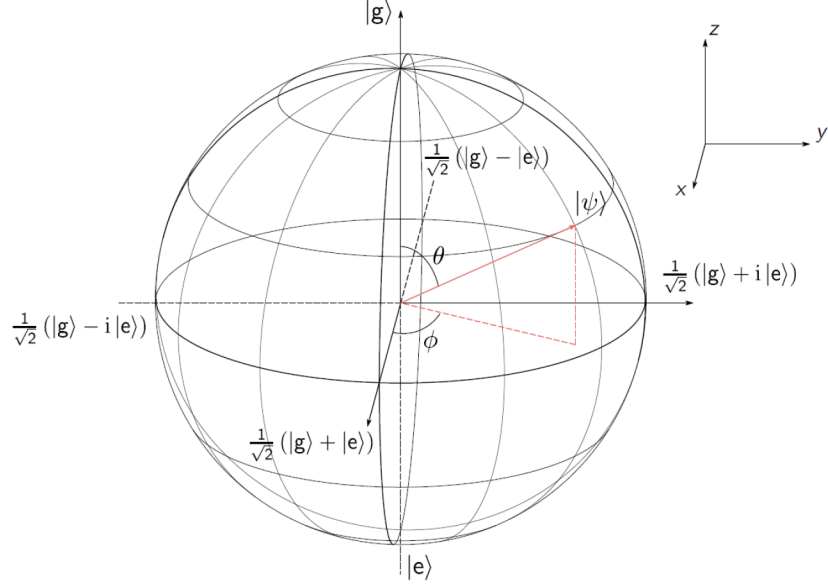


Figure 2.1: **Qubit state representation on the Bloch sphere**, which provides a useful visualization of a single qubit and its operations. The poles of the Bloch sphere corresponds to the computational basis states, $|g\rangle$ and $|e\rangle$.

this factor, a qubit can be described by two real angles θ and ϕ as follows [52],

$$|\psi_q\rangle = \cos \frac{\theta}{2} |g\rangle + \sin \frac{\theta}{2} e^{i\phi} |e\rangle. \quad (2.34)$$

where $0 \leq \theta \leq \pi$ and $0 \leq \phi < 2\pi$. This parameterization makes it clear that each pure qubit state $|\psi_q\rangle$ can be visualized as a point on the surface of a sphere with unit radius, called the Bloch sphere, see Fig. (2.1).

While any pure state $|\psi_q\rangle$ can be represented by a point on the surface of the Bloch sphere, the interior of the Bloch sphere is used to describe the mixed states of a qubit, which are needed in the presence of dissipation or decoherence. As discussed in section 2.1.5, in order to describe the qubit state even in the presence of imperfect knowledge, we use the density matrix formalism.

2.2.1 Pauli Matrices

The Pauli matrices are defined as

$$\hat{\sigma}_1 = \hat{\sigma}_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \hat{\sigma}_2 = \hat{\sigma}_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \hat{\sigma}_3 = \hat{\sigma}_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (2.35)$$

Pauli matrices are unitary and Hermitian, and their eigenvalues are equal to ± 1 . Any 2×2 Hermitian matrix can be written in a unique way as a linear combination of Pauli matrices (considering $\hat{\sigma}_0$ as the identity operator) with real coefficients.

2.2.2 Density Matrix

For a pure state $|\psi_q\rangle$, the density matrix is $\hat{\rho}_q = |\psi_q\rangle\langle\psi_q|$. Using Eq. (2.32), we can write the density matrix as

$$\hat{\rho}_q = \begin{pmatrix} |\alpha|^2 & \alpha\beta^* \\ \alpha^*\beta & |\beta|^2 \end{pmatrix}. \quad (2.36)$$

The diagonal elements describe the population of each energy eigenstate, and the off-diagonal elements describe the coherence between different energy eigenstates. When considering a statistical mixture of states $|\psi_{q_i}\rangle$, the density matrix becomes $\hat{\rho}_q = \sum_i P_i |\psi_{q_i}\rangle\langle\psi_{q_i}|$. Since $\hat{\rho}_q$ is Hermitian, positive and of trace 1, it can be decomposed into the Pauli matrix basis

$$\hat{\rho}_q = \frac{1}{2} (\hat{\mathbb{I}} + x \hat{\sigma}_x + y \hat{\sigma}_y + z \hat{\sigma}_z) = \frac{1}{2} (\hat{\mathbb{I}} + \mathbf{r} \cdot \hat{\sigma}), \quad (2.37)$$

where $\mathbf{r} = (x, y, z) \in \mathbb{R}^3$ is the Bloch vector, $\hat{\mathbb{I}}$ is the identity operator, and $\hat{\sigma} = (\hat{\sigma}_x, \hat{\sigma}_y, \hat{\sigma}_z)$. Any qubit state can be represented uniquely by specifying the Bloch coordinates x , y , and

z . Using the relation $\text{Tr}[\hat{\sigma}_i \hat{\sigma}_j] = 2\delta_{i,j}$, we can show² that $x = \langle \hat{\sigma}_x \rangle \equiv \text{Tr}[\hat{\sigma}_x \hat{\rho}]$, and similar for y and z . Also, since Eq. (2.37) is constrained to have positive eigenvalues, the Bloch vector $\mathbf{r}$ only represents a physical state if $\|\mathbf{r}\| = \sqrt{x^2 + y^2 + z^2} \leq 1$. $\hat{\rho}_q$ is pure if and only if $\|\mathbf{r}\| = 1$.

2.2.3 Harmonic Oscillators and Coherent States

So far we have considered only systems with a finite number of energy levels (e.g. a qubit). In this section we will discuss a system with an infinite number of energy levels; the quantum harmonic oscillator. In quantum mechanics, the harmonic oscillator is an important paradigmatic example because it provides a model for a variety of systems, such as the modes of the electromagnetic field (photons) as we will see frequently in the next chapters, or the vibrations of molecules and solids (phonons).

An object follows harmonic motion in a restoring potential of the form $V(x) = \frac{1}{2} k x^2$, where $k = m\omega^2$ is the spring constant defining the parabolic potential, ω is the angular frequency of oscillation and m is the mass of the object. The Hamiltonian that describes the system is,

$$\hat{H} = \frac{\hat{p}^2}{2m} + \frac{1}{2} m \omega^2 \hat{x}^2, \quad (2.38)$$

here, $\hat{x}$ and $\hat{p}$ are the position and momentum operators, where, in the position representation, $\hat{p} = -i\hbar d/dx$. The time-independent Schrödinger equation of the Hamiltonian (2.38) will be,

$$\hat{H}\psi = \left[-\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + \frac{m\omega^2}{2} \hat{x}^2 \right] \psi = E\psi. \quad (2.39)$$

We can solve Eq. (2.39) analytically in terms of modified Hermite polynomials [53], however, the *ladder operator* method, developed by Dirac, gives us a much simpler algebraic method

2. The expectation value of the measurement can be calculated by extending from the case of pure states, $\langle \hat{A} \rangle = \sum_i P_i \langle \psi_i | \hat{A} | \psi_i \rangle = \sum_i P_i \text{Tr}(|\psi_i\rangle\langle\psi_i| \hat{A}) = \text{Tr}\left(\sum_i P_i |\psi_i\rangle\langle\psi_i| \hat{A}\right) = \text{Tr}(\hat{\rho} \hat{A})$.

to find the energy spectrum without directly solving the differential equation. Defining the ladder operators as

$$\begin{aligned}\hat{a} &= \sqrt{\frac{m\omega}{2\hbar}} \left(\hat{x} + \frac{i\hat{p}}{m\omega} \right) && \text{annihilation (lowering) operator,} \\ \hat{a}^\dagger &= \sqrt{\frac{m\omega}{2\hbar}} \left(\hat{x} - \frac{i\hat{p}}{m\omega} \right) && \text{creation (raising) operator.}\end{aligned}\tag{2.40}$$

These operators [2.40](#) each annihilate/create a quantum of energy $E = \hbar\omega$, a property that gives them their respective names. Since we know $[\hat{p}, \hat{x}] = -i\hbar$, we also have

$$[\hat{a}, \hat{a}^\dagger] = 1.\tag{2.41}$$

The inverse relations of Eq. [\(2.40\)](#) are,

$$\begin{aligned}\hat{x} &= \sqrt{\frac{\hbar}{2m\omega}} (\hat{a}^\dagger + \hat{a}), \\ \hat{p} &= i\sqrt{\frac{\hbar m\omega}{2}} (\hat{a}^\dagger - \hat{a}).\end{aligned}\tag{2.42}$$

Considering again the Hamiltonian from Eq. [\(2.38\)](#), we can rewrite it in terms of $\hat{a}^\dagger \hat{a}$,

$$\hat{H} = \frac{1}{2m} (\hat{p}^2 + (m\omega \hat{x})^2) = \frac{\hbar\omega}{2} (\hat{a}^\dagger \hat{a} + \hat{a} \hat{a}^\dagger).\tag{2.43}$$

Using the commutator [2.41](#) we can further simplify the Hamiltonian [2.43](#),

$$\hat{H} = \hbar\omega \left(\hat{a}^\dagger \hat{a} + \frac{1}{2} \right).\tag{2.44}$$

The energy eigenstates that are denoted by $|n\rangle$, when operated on by the annihila-

tion/creation operators, give

$$\begin{aligned}\hat{a}|n\rangle &= \sqrt{n}|n-1\rangle, \\ \hat{a}^\dagger|n\rangle &= \sqrt{n+1}|n+1\rangle.\end{aligned}\tag{2.45}$$

Here, we can see that $\hat{a}^\dagger$, creates a single quantum of energy in the oscillator, while $\hat{a}$ removes a quantum. From the relations in Eq. (2.45), we can also define a number operator as, $\hat{n} = \hat{a}^\dagger \hat{a}$, which has the following property, $\hat{n}|n\rangle = n|n\rangle$, meaning $|n\rangle$ is the eigenstate of the number operator with non-negative ³ integer eigenvalue n . The reason we call $\hat{a}^\dagger \hat{a}$ the number operator is that it counts the number of quanta in a given state; specifically, given a state $|n\rangle$, we first apply $\hat{a}$ to annihilate a quantum and then $\hat{a}^\dagger$ to create it back again. the proportionality factor is the eigenvalue n .

The ground state with $n = 0$ is called the vacuum state, $|0\rangle$ and is defined by $\hat{a}|0\rangle = 0$. We can build all number states by starting from the vacuum and applying $\hat{a}^\dagger$ in an iterative way [51],

$$|n\rangle = \frac{(\hat{a}^\dagger)^n}{\sqrt{n!}}|0\rangle.\tag{2.46}$$

Using the Eq. (2.46) and the algebra of $\hat{a}$ and $\hat{a}^\dagger$, it then follows that the states $|n\rangle$ form an orthonormal basis, $\langle n|m\rangle = \delta_{n,m}$, which means, experimentally, one should be able to distinguish $|n\rangle$ from $|m\rangle$ without any ambiguity. The states $|n\rangle$ are also called *Fock states*.

Now that we have introduced the number operator, rewriting the Hamiltonian (2.43) in the form, $\hat{H} = \hbar\omega(\hat{n} + 1/2)$ provides the energy eigenvalues,

$$E_n = \hbar\omega\left(n + \frac{1}{2}\right).\tag{2.47}$$

The corresponding energy eigenstates, $\psi_n(x)$ can be determined by finding the solutions to

3. $\hat{n}$ is a positive semi-definite operator, so n can not be negative, $n = \langle n|\hat{a}^\dagger \hat{a}|n\rangle \geq 0$.

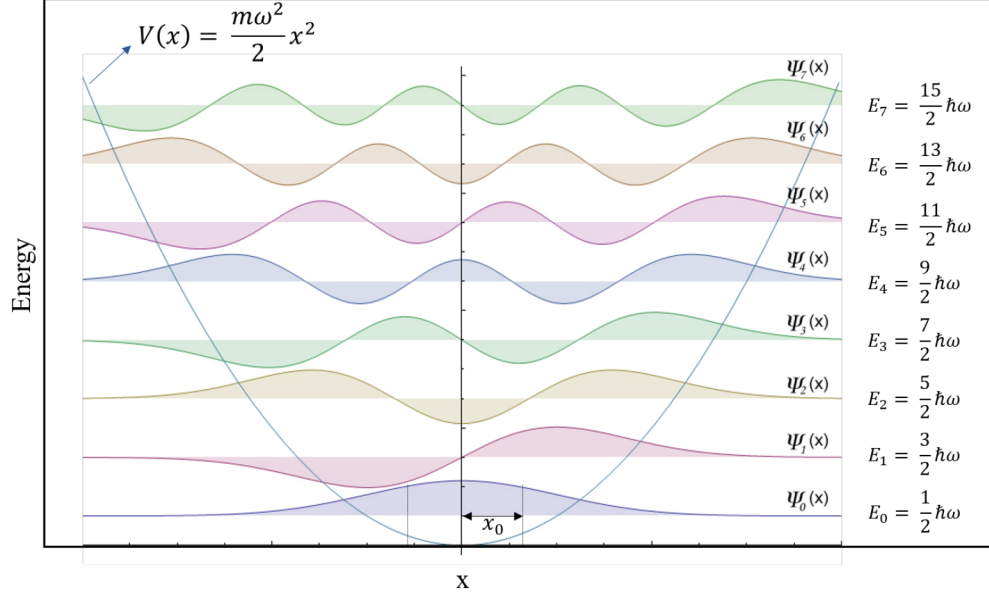


Figure 2.2: Wavefunctions and energies of the first few harmonic oscillator eigenstates, the wavefunctions form a ladder of alternating even and odd energy states, which are each separated by a quantum of energy $\hbar\omega$, i.e. equally spaced. The ground state is a Gaussian distribution with width $x_0 = \sqrt{\hbar/m\omega}$.

the Schrödinger equation in position space, $\hat{H}|\psi_n(x)\rangle = E_n|\psi_n(x)\rangle$, the solutions are [51]

$$\psi_n(x) = \frac{1}{\sqrt{n!}}(\hat{a}^\dagger)^n\psi_0(x) = \frac{1}{\sqrt{n!}}\left(\frac{m\omega}{\pi\hbar}\right)^{\frac{1}{4}}(\hat{a}^\dagger)^n \exp\left(-\frac{m\omega}{2\hbar}x^2\right), \quad (2.48)$$

where $\psi_0(x)$ is the ground state ($n = 0$) which is a Gaussian with variance $\sqrt{\hbar/m\omega}$.

The wavefunctions $\psi_n(x)$ for the first few energy eigenstates are shown in Fig. (2.2). From the figure, we can see that as n increases, the wavefunction becomes more dispersed in position x and has n nodes in the amplitude.

The coherent states of the harmonic oscillator are special nondispersive wave packets that describe states in which the ground state wave packet is displaced from the origin of the system. These states interest us because they follow the classical motion of a harmonic oscillator with the minimum amount of uncertainty allowed by the Heisenberg uncertainty

principle. They also play a key role in the quantum theory of light, because the output of a laser or a signal generator is coherent light. The coherent states are expressed in the $|n\rangle$ basis as follows [54, 55],

$$|\alpha\rangle = \sum_{n=0}^{\infty} c_n |n\rangle, \quad c_n = e^{-\frac{|\alpha|^2}{2}} \frac{\alpha^n}{\sqrt{n!}}, \quad (2.49)$$

here c_n indicates the contribution of each photon number state in the coherent state. $\alpha \in \mathbb{C}$ (complex displacement) is a constant with magnitude related to the average number of photons, $\langle \hat{n} \rangle = |\alpha|^2$, of the coherent state $|\alpha\rangle$. We can show that a coherent state is an eigenstate of annihilation operator, $\hat{a}|\alpha\rangle = \alpha|\alpha\rangle$. However, since $\hat{a}$ is a non-Hermitian operator, the eigenstates $|\alpha\rangle$ do not form an orthogonal basis and are overcomplete.

We can write the Eq. (2.49) in the form

$$|\alpha\rangle = e^{-\frac{|\alpha|^2}{2}} \sum_{n=0}^{\infty} \frac{(\alpha \hat{a}^\dagger)^n}{n!} |0\rangle = e^{-\frac{|\alpha|^2}{2}} e^{\alpha \hat{a}^\dagger} |0\rangle = e^{\alpha \hat{a}^\dagger - \alpha^* \hat{a}} |0\rangle, \quad (2.50)$$

where the last step is completed by applying the Baker–Campbell–Hausdorff Lemma [51]. The operator applied to the ground state is called the displacement operator $\hat{D}(\alpha) = e^{\alpha \hat{a}^\dagger - \alpha^* \hat{a}}$ because it displaces the ground state from the origin by an amount α in phase space, see Fig. (2.3).

To look at the dynamics of a coherent state, we first need to find the expectation values $\langle \hat{x} \rangle$ and $\langle \hat{p} \rangle$ (defined in Eq. (2.42)), calculated in the interaction picture (moving to a frame that is rotating with frequency $\omega = \hat{H}/\hbar$ in phase space⁴) where $\hat{a}^\dagger \rightarrow \hat{a}^\dagger e^{i\omega t}$ and $\hat{a} \rightarrow \hat{a} e^{-i\omega t}$,

4. It is sometimes convenient to express the behavior of a mechanical system in phase-space, where the position is plotted on the x-axis and the momentum is plotted on the y-axis. One of the key features of the phase-space formulation of quantum mechanics is that the quantum state is described by a quasiprobability distribution [56] instead of wave function, state vector, or density matrix. The formalism for quantum mechanics in phase-space is explained in more details in this paper [57].

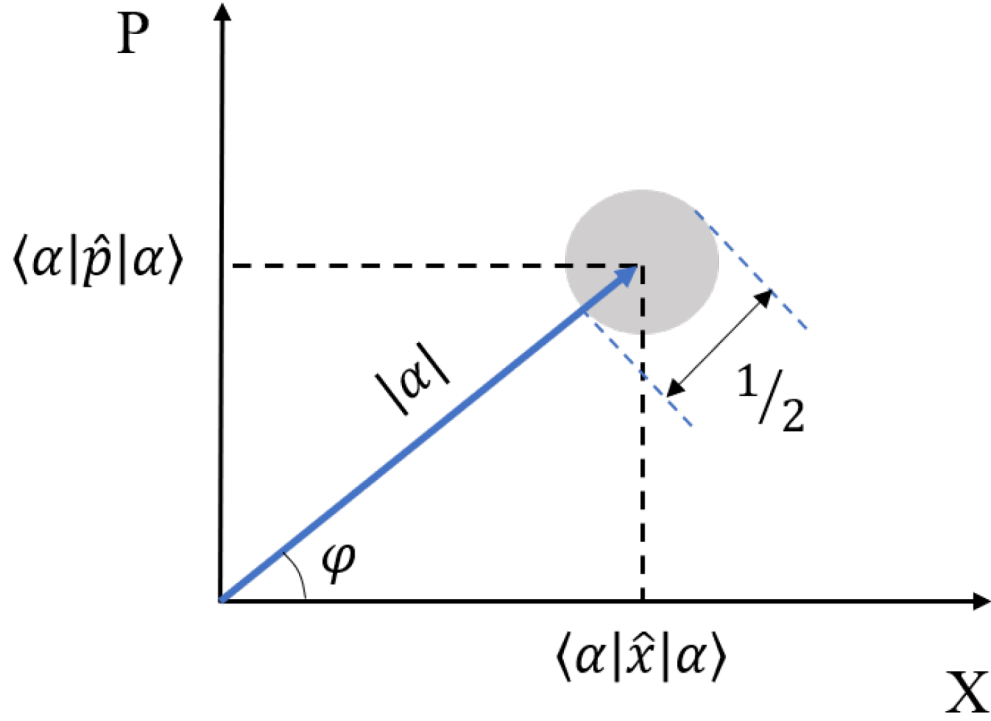


Figure 2.3: Representation of the state space occupied by the coherent state $|\alpha\rangle$, where $\hat{X}$ and $\hat{P}$ are the field quadratures, φ is the phase angle of the coherent state, $|\alpha|$ is the amplitude of the coherent state, and the value of $1/2$ is the uncertainty.

we have

$$\langle\alpha|\hat{x}|\alpha\rangle = \sqrt{\frac{\hbar}{2m\omega}} \langle\alpha|(\hat{a}e^{-i\omega t} + \hat{a}^\dagger e^{i\omega t})|\alpha\rangle = \sqrt{\frac{2\hbar}{m\omega}} |\alpha| \cos(\omega t + \varphi), \quad (2.51)$$

$$\langle\alpha|\hat{p}|\alpha\rangle = -i\sqrt{\frac{\hbar m\omega}{2}} \langle\alpha|(\hat{a}e^{i\omega t} - \hat{a}^\dagger e^{-i\omega t})|\alpha\rangle = \sqrt{\frac{\hbar m\omega}{2}} |\alpha| \sin(\omega t + \varphi), \quad (2.52)$$

where $\varphi = \arg[\alpha]$. In scaled units the relation between α and the expectation values of position and momentum in Eqs. (2.51) and (2.52) are

$$\hat{X} \propto \langle\alpha|\hat{x}|\alpha\rangle = \text{Re}[\alpha] = \frac{\langle\hat{a} + \hat{a}^\dagger\rangle}{2}, \quad (2.53)$$

$$\hat{P} \propto \langle\alpha|\hat{p}|\alpha\rangle = \text{Im}[\alpha] = \frac{\langle\hat{a} - \hat{a}^\dagger\rangle}{2i}. \quad (2.54)$$

$\hat{X}$ and $\hat{P}$ are dimensionless field quadratures (see Fig. (2.3)). In Chapter 3, we will see these quadrature operators represent continuous variable observables that are able to be measured via homodyne detection [58] (in Chapter 3, X is denoted by I , and P is denoted by Q).

Having a minimum uncertainty, the dynamics of a coherent state can be described by a Gaussian Wigner function [59] travelling in phase space. As a representation, the Wigner function is usually replaced by a circle with a diameter corresponding to the width of the Gaussian (minimum uncertainty here), see Fig. (2.3).

2.3 Rabi Oscillations and Jaynes-Cummings Model

Quantum information has always been related to quantum optics, so light-matter interaction is an essential topic in the field. The two foundational models in this sense are the Rabi and Jayne-Cummings models, which both describe the interaction of a light field (radiation mode) with a single atom, approximated as a two-level systems (i.e., a qubit). Although these effects could take place in free space, we are usually interested in controlled experiments performed when the qubit is coupled to a circuit quantum electrodynamics (cQED) resonator⁵ (resonators).

2.3.1 Rabi Oscillations

For an isolated qubit, the only term in the Hamiltonian is $\hat{H}_0 = \hbar\omega_q \hat{\sigma}_z/2$ that comes from energy splitting between the ground and excited states $\Delta E_q = E_{|e\rangle} - E_{|g\rangle} = \hbar\omega_q$, where ω_q is the frequency which the Bloch vector rotates around the $\hat{\sigma}_z$ axis.

5. The resonator field is usually represented by a harmonic oscillator

Solving the free-evolution Schrödinger equation yields

$$|\psi_q(t)\rangle = \alpha e^{i\omega_q t/2}|g\rangle + \beta e^{-i\omega_q t/2}|e\rangle. \quad (2.55)$$

If we write Eq. (2.55) in a density matrix form, $\hat{\rho}_q = |\psi_q(t)\rangle\langle\psi_q(t)|$, we will see that in the off-diagonal elements, the relative phase between $|g\rangle$ and $|e\rangle$ varies, but the populations in each state (diagonal elements) is constant.

To change the populations in $|g\rangle$ and $|e\rangle$, i.e. to control the state of a qubit, we need to rotate it about an axis other than $\hat{\sigma}_z$, i.e. by applying an oscillatory driving field. The resulting Hamiltonian for this system is called the Rabi model, and describes the state of the qubit interacting with a driving field. The dynamics are called Rabi oscillations and are associated with oscillations of the qubit's energy level populations. The frequency of the periodic exchange of energy and of the oscillation of the population probabilities is the Rabi frequency.

The Hamiltonian of the driven qubit, which is explained above is

$$\frac{\hat{H}}{\hbar} = -\frac{1}{2}\omega_q \hat{\sigma}_z - \Omega_R \cos(\omega_d t) \hat{\sigma}_x, \quad (2.56)$$

where Ω_R is the Rabi frequency and ω_d is the drive frequency. For now we assume that $\omega_d = \omega_q = \omega$ (also called resonant Rabi drive). It is convenient to express the operator $\hat{\sigma}_x$ in terms of the ladder operator $\hat{\sigma} = |1\rangle\langle 0|$ that acts on qubit energy eigenstates as follows, $\hat{\sigma}|0\rangle = |1\rangle$, and $\hat{\sigma}^\dagger|1\rangle = |0\rangle$. Using the ladder operator and expanding the cosine, the Hamiltonian (2.56) takes the form

$$\frac{\hat{H}}{\hbar} = -\frac{1}{2}\omega \hat{\sigma}_z - \frac{1}{2}\Omega_R (e^{i\omega t} + e^{-i\omega t})(\hat{\sigma} + \hat{\sigma}^\dagger). \quad (2.57)$$

By going to the rotating frame of the drive, the Hamiltonian will change to the effective

Hamiltonian shown in Eq. (2.29). Choosing the unitary operator as $\hat{U}(t) = e^{-i\omega \hat{\sigma}_z t/2}$, simply removes the first part in Eq. (2.57) and only keeps the Rabi drive dynamics. The Hamiltonian in the rotating frame takes the form

$$\begin{aligned}\frac{\hat{H}}{\hbar} &= -\frac{1}{2}\Omega_R(e^{i\omega t} + e^{-i\omega t})(\hat{\sigma}e^{i\omega t} + \hat{\sigma}^\dagger e^{-i\omega t}) \\ &= -\frac{1}{2}\Omega_R \hat{\sigma}_x - \frac{1}{2}\Omega_R(\hat{\sigma}e^{2i\omega t} + \hat{\sigma}^\dagger e^{-2i\omega t}).\end{aligned}\tag{2.58}$$

The second part of Eq. (2.58) can be neglected using the *rotating wave approximation* (RWA). The idea is motivated by the fact that the last terms in Eq. (2.58) oscillate rapidly (with the frequency of $\pm 2\omega$) around a zero average and therefore will have a small effect on the average dynamics, leaving the RWA Rabi drive Hamiltonian ⁶

$$\frac{\hat{H}}{\hbar} = -\frac{1}{2}\Omega_R \hat{\sigma}_x.\tag{2.59}$$

Most of the time in experiment a resonant drive is used to control the qubit state. Fig. (2.4) shows driven qubit state evolution in Bloch coordinates for both the lab and rotating frame.

2.3.2 The Jaynes-Cummings Model

The Jaynes-Cummings model describes the dynamics of a two-level system (qubit) interacting with a single mode of an electromagnetic cavity (resonator). Initially proposed by Edwin Jaynes and Fred Cummings in 1963 [60, 61], it has been widely used in circuit quantum electrodynamics. The total Hamiltonian to describe a system consisting of a resonator and a

6. In the case that the oscillating field is detuned from the qubit frequency, $\omega_d \neq \omega_q$, for making the $\hat{H}$ time-independent, we should choose $\hat{U}(t) = e^{-i\omega_d \hat{\sigma}_z t/2}$. In this case we will see that the Rabi frequency will be $\Omega_R = \sqrt{\Omega_R^2 + \Delta^2}$, where $\Delta = \omega_q - \omega_d$ [51].

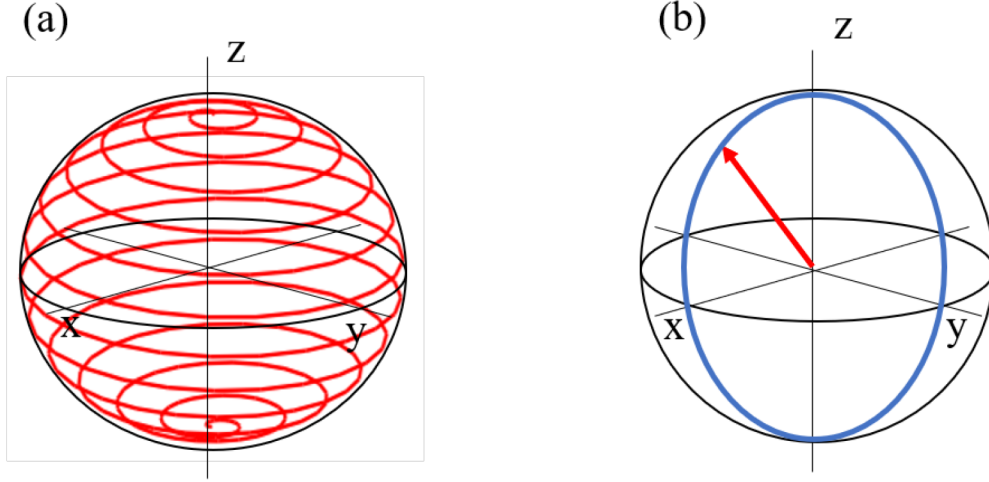


Figure 2.4: Rabi oscillations in the (a) lab frame vs. (b) the rotating frame.

qubit has three parts,

$$\hat{H}/\hbar = \frac{\omega_q}{2}\hat{\sigma}_z + \omega_r\left(\hat{a}^\dagger\hat{a} + \frac{1}{2}\right) + g(\hat{a} + \hat{a}^\dagger)(\hat{\sigma} + \hat{\sigma}^\dagger) \quad (2.60)$$

where ω_r is the frequency of the resonator (as a harmonic oscillator), ω_q is the frequency of the qubit, g is the strength of the capacitive coupling between the qubit and the resonator, $\hat{a}(\hat{a}^\dagger)$ is the annihilation (creation) operator for the resonator mode, and $\hat{\sigma}(\hat{\sigma}^\dagger)$ is the lowering (raising) operator of the qubit.

In the case that there is no interaction between the resonator and qubit ($g = 0$), the eigenstates of the qubit-resonator system become the tensor product of the resonator and qubit eigenstates, which are called bare states. However, for $g \neq 0$, to calculate the eigenstates we will need to diagonalize the total Hamiltonian, which is not easy [62]. We can perform this calculation most easily if we first simplify the interaction Hamiltonian using the rotating wave approximation [63]. The interaction Hamiltonian is composed of four terms, $\hat{H}_{int} \Rightarrow \hat{a}^\dagger\hat{\sigma} + \hat{a}\hat{\sigma}^\dagger + \hat{a}^\dagger\hat{\sigma}^\dagger + \hat{a}\hat{\sigma}$. The last two terms (which are exciting or deexciting the oscillator and the qubit simultaneously) will oscillate with frequency $\pm\omega_r + \omega_q$ in the inter-

action picture. Knowing that the rotating wave approximation is valid when the coupling strength is much less than the frequency of both the qubit and resonator, $g \ll \omega_q, \omega_r$, we can see that $\pm(\omega_r + \omega_q)$ would be much larger than g , meaning that the dynamics of the last two processes are so fast, they roughly average to zero. Thus, we can neglect them in our calculations.⁷ With this rotating wave approximation we obtain the Jaynes-Cummings Hamiltonian,

$$\hat{H}/\hbar = \omega_r(\hat{a}^\dagger \hat{a} + \frac{1}{2}) - \frac{1}{2}\omega_q \hat{\sigma}_z - g(\hat{a}^\dagger \hat{\sigma} + \hat{a} \hat{\sigma}^\dagger) \quad (2.61)$$

We can obtain the approximate eigenvalues and eigenstates of the system by diagonalizing this Hamiltonian [63]. The resulting eigenstates are no longer bare states, but are instead dressed states [64],

$$\begin{aligned} \frac{E_{\pm,n}}{\hbar} &= (n+1)\omega_r \pm \frac{1}{2}\sqrt{\Delta^2 + 4ng^2}, \\ |n, +\rangle &= \cos(\theta_n/2)|e, n-1\rangle + \sin(\theta_n/2)|e, n\rangle, \\ |n, -\rangle &= \cos(\theta_n/2)|g, n\rangle - \sin(\theta_n/2)|g, n-1\rangle, \\ \tan(\theta_n) &\equiv -\frac{2\sqrt{n}g}{\Delta}. \end{aligned} \quad (2.62)$$

where $\Delta = \omega_q - \omega_r$, and $\theta_n = \tan^{-1}(2g\sqrt{n}/\Delta)$, which quantifies the level of qubit-resonator hybridization. In Fig. (2.5) one can compare the dressed state energy levels to the corresponding uncoupled system energy levels, the bare states.

One of the interesting regimes to consider in the solution to Jaynes-Cummings is when the detuning, Δ , is much larger than the coupling strength, g , which is called the dispersive regime. In this case, we can apply the unitary transformation, $\hat{U}_{\text{disp}} = \exp[(g/\Delta)(\hat{a}\hat{\sigma}^\dagger -$

7. Those non-RWA terms become relevant at higher energy, i.e., when we pump the resonator to larger photon numbers, but for weaker drives they are safely neglected

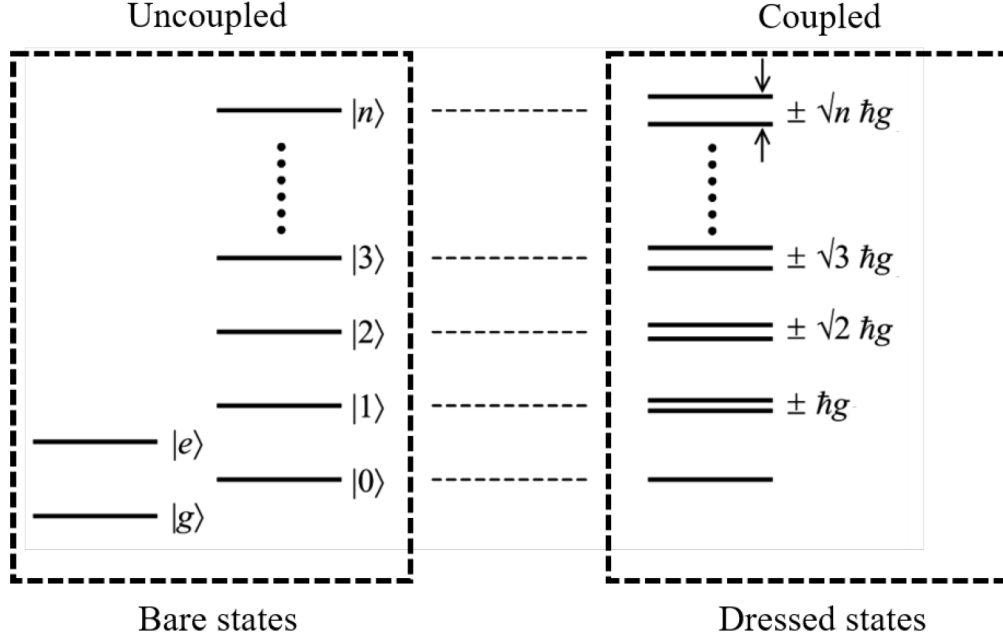


Figure 2.5: Dressed states vs bare states. The right panel illustrates the dressed states of the qubit-cavity system in comparison with the bare states, left panel.

$\hat{a}^\dagger \hat{\sigma})]$, and using second-order perturbation theory [65], we can rewrite the Hamiltonian,

$$\begin{aligned} \hat{H}_{\text{disp}} &= (\omega_r - \chi \hat{\sigma}_z) \hat{a}^\dagger \hat{a} - \frac{1}{2} \omega_q \hat{\sigma}_z \\ \chi &\equiv \frac{g^2}{\Delta} \end{aligned} \tag{2.63}$$

where χ is the dispersive shift⁸. We see that the Hamiltonian in the dispersive regime leads to a qubit state dependent shift of the resonator. This means that even though the qubit and the resonator do not directly exchange the energy; however, they still interact in such a way that we can use the resonator to probe the qubit state.

8. We will see in Chapter 3 that for a specific type of qubit called transmon qubit, when the transmon energy levels outside of the qubit subspace (the first two energy levels) are taken into account, the state-dependent dispersive shift χ will be different.

3 Superconducting Qubits and Circuit QED

Modern quantum technologies have opened the door to a new era of experimentation with actively controllable quantum physics. Prior to this “second quantum revolution”¹, most technology used quantum mechanics only in a passive way, like in transistors and material engineering. Modern quantum technology now allows us to directly manipulate and use more exotic quantum effects and test the foundations of quantum theory, which is important for both fundamental and applied research.

Even though everything in our world behaves according to quantum mechanics, it is usually difficult to see these quantum effects and control them because frequent environmental interactions like temperature fluctuations and ambient radiation effectively scramble quantum features so they appear classical. To build a controllable quantum system that preserves and manipulates these fragile quantum effects, that system must be completely isolated from the environment, except for a few specific knobs that we can control. Since any control is also a potential dissipation channel, these control knobs have to be designed carefully to prevent environmental decoherence while still permitting enough interaction with the system for the control to be useful and fast. The simpler the system, the better it can be controlled. Suitably simple systems can come from nature directly, or from clever engineering.

Simple examples of natural quantum systems are photons, single atoms, and single ions. Isolating individual atoms, or ions, for example with laser traps, yields a controllable system that has distinguishable quantum mechanical features. In modern ion traps, for example, the hyperfine ground state energy levels of the ion are nearly decoupled from the environment so can have long-lived coherence, but can still be manipulated by suitably tuned lasers. Choosing such a system, however, has the disadvantage that nature provides only a restricted set of possible parameters; for example, a particular choice of ion will always have the

1. This name is used by Jonathan Dowling for the first time [J. Dressel, private communication].

same energy level structure. Additionally, individual ions are very small, so controlling and measuring them can require incredibly precise tools that are challenging to engineer and operate with sufficient accuracy. A more subtle problem arises when controlling multiple trapped ions, since the speed of such control is limited by their mechanical motion, which is subject to inertia.

In contrast, engineered quantum systems like superconducting electronic circuits are man-made systems that give larger amount of flexibility in how well one can tune the operating parameters to desired values. They are also concrete examples where quantum mechanical effects persist in macroscopic systems, making them fruitful new experimental territory for testing fundamental physics. However, engineered systems have their own disadvantages. As larger and more complex systems, they interact more strongly with their environments, causing more noise and decoherence compared to microscopic ions. Getting them into a regime where the quantum effects dominate the environmental noise can be very challenging, and usually requires cooling to very low temperatures near absolute zero, which makes scaling to larger composite systems problematic.

Yasunobu Nakamura and Jaw-Shen Tsai studied quantum mechanical effects at the macroscopic level for the first time in 1991 in superconducting circuits. For an integrated circuit to behave quantum mechanically, the first requirement is to eliminate energy dissipation. To achieve this, all parts of the circuit need to be made of a material that has zero resistance at the operating temperature. Superconductivity has played a key role since superconducting material naturally has no resistance when cooled below a certain critical temperature e.g., about 1K for superconducting aluminum [66–68]. Below the critical temperature, the valence electrons weakly bind into bosonic Cooper pairs [69] that can all condense into the same macroscopic quantum state and thus travel through the metal in unison with no resistance [70].

Since 1991 these superconducting circuits have been increasingly used as qubits in quan-

tum processors. Like normal circuits, they can be controlled with external voltage drives without disrupting the superconductivity. Unlike normal circuits, the lack of resistance allows quantum effects to manifest, such as superpositions of different currents along wires, or superpositions of different accumulations of charge on capacitive plates. The design of controllable superconducting circuits became much easier after identifying a convenient analogy between resonant tank circuits (like LC oscillators) and optical laser cavities. Just like lasers in optical cavities could be used to manipulate nearby atoms, tank circuits in superconducting chips can be used to manipulate neighboring capacitively-coupled qubits. The formal descriptions of both these interactions involve photon absorption and emission by a two-level quantum system from resonant electromagnetic modes, so much of the extensive work done in atom-laser interactions since the 1960's could be adapted to describe superconducting circuits directly. The field of circuit quantum electrodynamics (cQED) is the result after adapting these interactions between light and matter to the setting of superconducting circuits [71, 72].

Creating an artificial atom using superconducting circuits is relatively simple, but requires a new nonlinear circuit component: a Josephson junction. As with any quantum mechanical system, a binding potential will quantize the allowed energy levels. Thus, any resonant circuit will have discrete levels that could be used in principle as logical states, with the lowest two energies being a natural choice for encoding a qubit. However, a linear resonant tank circuit has a quadratic binding potential and evenly spaced energy levels, which makes it impossible to address specific pairs of levels via absorption and emission of specific photon energies for the purposes of control. Introducing a nonlinear element like a Josephson junction into the tank circuit creates uneven energy spacing that can be uniquely addressed, and thus controlled. A Josephson Junction, which is a tunnel junction made from a thin insulating layer sandwiched between two superconducting metals, behaves as a non-linear and non-dissipating inductor so enables remarkably simple circuit designs for artificial atoms with

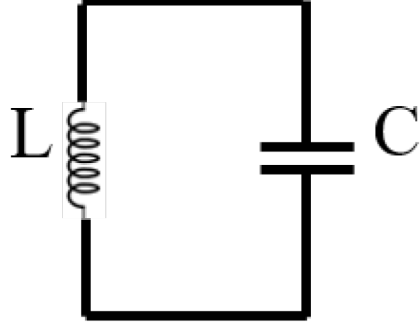


Figure 3.1: An LC oscillator is the simplest example of a resonant quantum circuit. It consists of a capacitor C connected to an inductor L .

addressable energy levels [73, 74, 74–76].

In this chapter I will more formally introduce the simplest resonant quantum circuit: the quantum tank circuit or LC oscillator, as well as the architecture of the Josephson junction, its use in the design of superconducting qubits like the widely used transmon [77], and how a coupled tank circuit can be used to measure the qubit energy using an external drive.

3.1 Quantum LC Oscillator

An LC oscillator is the simplest example of a resonant quantum circuit. It consists of a capacitor C connected to an inductor L , see Fig. (3.1) and is made up of superconducting materials to eliminate dissipative resistance. When the capacitor and the inductor connect electric charge, Q , can oscillate between the capacitive plates by traveling through the inductor at the circuit resonance frequency $\omega = \sqrt{1/LC}$. Though no energy is dissipated in this oscillation by resistance, the charge flow still has a net impedance $Z = \sqrt{L/C}$. The energy oscillates between charging energy in the capacitor $E_C = Q^2/2C$, and magnetic field energy in the inductor, $E_L = \Phi^2/2L$, where Φ is the magnetic flux and the current $I = dQ/dt$ flowing through the inductor, in accordance with Faraday’s induction law [78].

After identifying Φ as the Lagrangian coordinate, where $\mathcal{L} = E_C - E_L$, we can identify that the charge is its conjugate momentum since $d\mathcal{L}/d\dot{\Phi} = C\dot{\Phi} = Q$. To quantize this resonant circuit, we can thus follow the same procedure as for quantizing the usual quantum harmonic oscillator, where the charge on the capacitor Q and the flux in the inductor Φ are analogous to the momentum p and the position x respectively. The charge and flux variables are promoted to canonically conjugate quantum operators² satisfying the commutation relation $[\hat{\Phi}, \hat{Q}] = i\hbar$, where $\hbar = h/2\pi$ is Planck's reduced constant. The Hamiltonian of this circuit is

$$\hat{H} = \frac{\hat{\Phi}^2}{2L} + \frac{\hat{Q}^2}{2C}. \quad (3.1)$$

Following the treatment of the quantum harmonic oscillator in Chapter 2, we can immediately define the annihilation $\hat{a}$ and creation $\hat{a}^\dagger$ operators in terms of $\hat{Q}$ and $\hat{\Phi}$. Representing these coordinates in terms of ladder operators, we have

$$\begin{aligned} \hat{\Phi} &= \sqrt{\frac{\hbar Z}{2}}(\hat{a}^\dagger + \hat{a}), \\ \hat{Q} &= -i\sqrt{\frac{\hbar}{2Z}}(\hat{a}^\dagger - \hat{a}). \end{aligned} \quad (3.2)$$

With these definitions, the Hamiltonian (3.1) takes its usual form as the Hamiltonian of a quantum harmonic oscillator $\hat{H} = \hbar\omega(\hat{a}^\dagger\hat{a} + 1/2)$. As with the standard quantum harmonic oscillator, the ladder operator $\hat{a}^\dagger$ ($\hat{a}$) creates (annihilates) one quantum of oscillation energy $\hbar\omega$ stored in the circuit, where $\omega = 1/\sqrt{LC}$ is the resonance frequency of the circuit.

2. Using the Lagrangian framework to describe the system, $\mathcal{L}(\dot{\Phi}, \Phi) = \frac{C\dot{\Phi}^2}{2} - \frac{\Phi^2}{2L}$, one can calculate $\frac{\partial \mathcal{L}}{\partial \dot{\Phi}} = C\dot{\Phi} = Q$ (we used $\dot{\Phi} = v$ and $CV = Q$), where Q (i.e. momentum) is canonically conjugate to Φ (i.e. position).

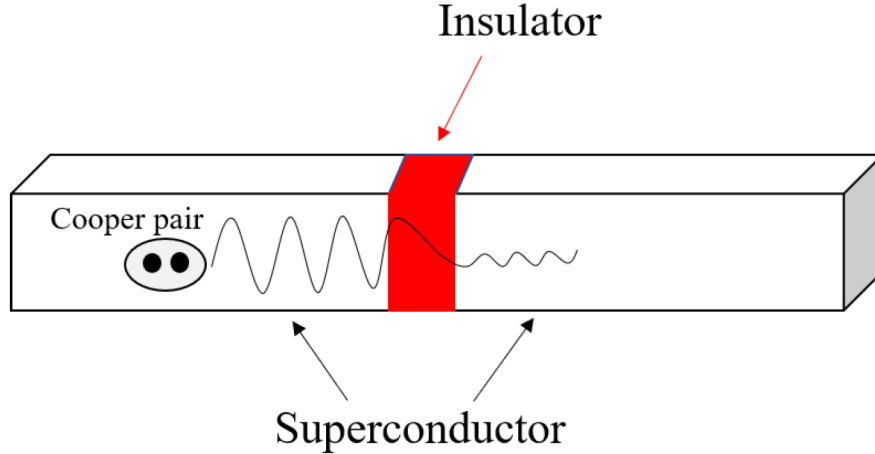


Figure 3.2: The Josephson junction consists of two superconductors separated by a thin layer of the insulator. The Cooper pairs on each side can tunnel through the insulator. The amplitude of wavefunction that represents the macroscopic boson condensate wavefunction of all the condensed Cooper pairs in the left electrodes decays in the insulating gap treated as a potential barrier. The wave functions on either side of the junction have different superconducting wavefunctions, and thus two different superconducting phases as order parameters. The difference between the two phases is the reduced φ flux appearing in the cosine potential for the junction.

3.2 Josephson Junctions

An LC circuit lacks an important ingredient for its discrete energy levels to be independently addressable as needed for a qubit: non-linearity. To address this problem, the linear inductor can be replaced with a non-linear inductor in the form of a Josephson junction. The Josephson junction consists of two superconducting electrodes (e.g., aluminum) separated by a thin insulating layer (e.g., aluminum oxide), see Fig. (3.2). If the insulating barrier is thin enough, there is a probability for the superconducting Cooper pairs to tunnel through the insulator. Due to this tunneling effect, a current can flow through the junction even when no voltage is applied if the wavefunctions describing the Cooper pair Bose-Einstein condensates on either side of the junction have mismatched kinetic phases. The resulting

tunneling current is given by the *first Josephson relation* [79, 80]

$$I = I_c \sin \varphi, \quad (3.3)$$

where I_c is a parameter of the junction and called the critical current that depends on the superconducting material and geometry and φ is the kinetic phase difference between the macroscopic Bose-Einstein condensate wavefunctions for the two superconducting electrodes on both sides of the junction. If a current is forced through the junction that exceeds the critical tunneling current I_c , then the superconductivity will be broken down (by breaking the Cooper pairs' bonds) and resistance will develop in the circuit [79], invalidating the Josephson relation (3.3).

The time variation of $\varphi(t)$ is related to the externally applied voltage across the junction V according to the *second Josephson relation* [79, 80]

$$\frac{d\varphi}{dt} = \frac{2\pi V}{\Phi_0}, \quad (3.4)$$

here $\Phi_0 = h/2e$ is the superconducting flux quantum, h is Planck's constant, and $(2e)$ is the charge of one Cooper pair of electrons. The dynamical behavior of Eqs. (3.3) and (3.4) can be understood by first differentiating Eq. (3.3) and substituting $d\varphi/dt$ into it from Eq. (3.4), which yields

$$\frac{dI}{dt} = \frac{2\pi I_c V \cos \varphi}{\Phi_0}. \quad (3.5)$$

From the relation $V = L dI/dt$, we can infer the effective inductance of the Josephson junction

$$L_J = V \left(\frac{dI}{dt} \right)^{-1} = \frac{\Phi_0}{2\pi I_c \cos \varphi}, \quad (3.6)$$

where the $1/\cos \varphi$ factor makes the inductor L_J nonlinear with respect to changes in the tunneling current, with the superconducting flux Φ_0 setting the scale of the nonlinearity [72].

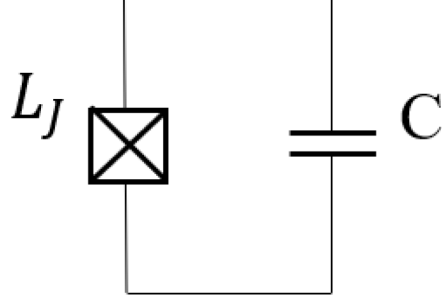


Figure 3.3: By replacing the inductor L of the simple LC oscillator with a Josephson junction with non-linear inductance L_J , the oscillator becomes weakly anharmonic with a discrete energy spectrum that is no longer evenly spaced.

Since the power across the junction is $P = IV$, the energy stored in the junction is

$$\begin{aligned}
 E &= \int_{-\infty}^t IV d\tau = \frac{\Phi_0}{2\pi} \int_{-\infty}^t I_c \sin \varphi \frac{d\varphi}{d\tau} d\tau \\
 &= \frac{I_c \Phi_0}{2\pi} \int_{-\infty}^t \sin \varphi d\varphi = E_J [1 - \cos \varphi],
 \end{aligned} \tag{3.7}$$

where $E_J = \Phi_0 I_c / 2\pi$ is the Josephson energy and is related to the linear Josephson inductance $E_J = L_J I_c^2$, when $\varphi = 0$.

By replacing the inductor L of the simple LC oscillator discussed in Section 3.1 with a Josephson junction with non-linear inductance L_J , see Fig. (3.3), the oscillator becomes weakly anharmonic with a discrete energy spectrum that is no longer evenly spaced. The uneven spacing is precisely what is needed to allow individual level pairs to be independently addressed as qubits.

3.3 Superconducting Qubits

3.3.1 Transmon Qubit

In the last two decades, there have been many circuit QED experiments demonstrating many different species of artificial qubits. There are three key types of superconducting qubit: charge qubits, also known as Cooper pair box [21, 81, 82], phase qubits [83] and flux qubits [84, 85]. In this section we will focus on an specific type of superconducting qubit: a transmon [77].

A transmon is a Josephson junction which is shunted with a large capacitor. Its dynamics can be modeled by adding a capacitive term to the potential energy stored in the junction in Eq. (3.7). The quantized Hamiltonian of the transmon is

$$\hat{H} = E_C [2(\hat{n} - n_g)]^2 + E_J [1 - \cos \hat{\varphi}], \quad (3.8)$$

where $E_C = e^2/2C$ is the charging energy of a single electron on the capacitor. Note that C in Eq. (3.8) is the total capacitance which includes the junction's capacitance and the additive shunt capacitance (dominated by the large shunt capacitor). The number operator that is introduced in the Eq. (3.8), $\hat{n} = \hat{Q}/2e = \sum_{n=-\infty}^{\infty} n|n\rangle\langle n|$ counts the difference of Cooper pairs (of charge $2e$) residing on either side of the junction, while offset charge parameter (also called the gate charge) n_g can be biased by an externally applied gate voltage.

Due to the large shunt capacitor, a transmon operates in the regime $E_J/E_C \sim 100 \gg 1$. The suppression of the charging energy compared to the Josephson energy makes the transmon relatively insensitive to environmental charge noise that makes the offset n_g fluctuate [86].

Expanding the cosine potential in Eq. (3.8) around $\varphi = 0$, the limit of $E_J/E_C \gg 1$ implies

that $\varphi \ll 1$. Therefore, expanding to the 4th order and neglecting ³ n_g , the Hamiltonian conveniently approximates an anharmonic Duffing oscillator,

$$\hat{H} = 4E_C \hat{n}^2 + \frac{E_J}{2} \hat{\varphi}^2 - \frac{E_J}{4!} \hat{\varphi}^4. \quad (3.9)$$

From Eq. (3.9), the Hamiltonian can be separated conceptually into two parts: the Hamiltonian of the harmonic oscillator (sum of the first two terms) plus a small anharmonic correction (the last term). Following the harmonic oscillator, it will be useful to define the creation and annihilation operators that diagonalize the first two terms in Eq. (3.9), using the canonical commutation relation $[\hat{\varphi}, \hat{n}] = i$ [77],

$$\hat{\varphi} = \left(\frac{8E_C}{E_J} \right)^{1/4} \frac{\hat{b} + \hat{b}^\dagger}{\sqrt{2}}, \quad (3.10)$$

$$\hat{n} = \left(\frac{E_J}{8E_C} \right)^{1/4} \frac{\hat{b} - \hat{b}^\dagger}{i\sqrt{2}}. \quad (3.11)$$

Rewriting the Hamiltonian (3.9) in terms of the lowering $\hat{b}$ and raising $\hat{b}^\dagger$ operators gives ⁴

$$\begin{aligned} \hat{H} &= \sqrt{8E_C E_J} \hat{b} \hat{b}^\dagger - \frac{E_C}{12} (\hat{b} + \hat{b}^\dagger)^4 \\ &\approx \sqrt{8E_C E_J} \hat{b} \hat{b}^\dagger - \frac{E_C}{2} (\hat{b}^\dagger \hat{b}^\dagger \hat{b} \hat{b} + 2\hat{b}^\dagger \hat{b}) \\ &= \underbrace{(\sqrt{8E_C E_J} - E_C)}_{\hbar\omega_q} \hat{b}^\dagger \hat{b} - \frac{E_C}{2} \hat{b}^\dagger \hat{b}^\dagger \hat{b} \hat{b}. \end{aligned} \quad (3.12)$$

For a harmonic oscillator the energy gap between the ground and first excited states would be $E_{10} = E_1 - E_0 = \hbar\omega_p$, where $\omega_p = \sqrt{1/LJC} = \sqrt{8E_C E_J}/\hbar$ is the plasma frequency. In contrast, the energy gap between the lowest two levels of the transmon is slightly shifted

3. The small charging energy makes the lower energy levels relatively sensitive to the offset charge n_g [77]

4. According to RWA which is discussed in Chapter 2, any combination of $\hat{b}$ and $\hat{b}^\dagger$ that cause fast oscillations average to zero.

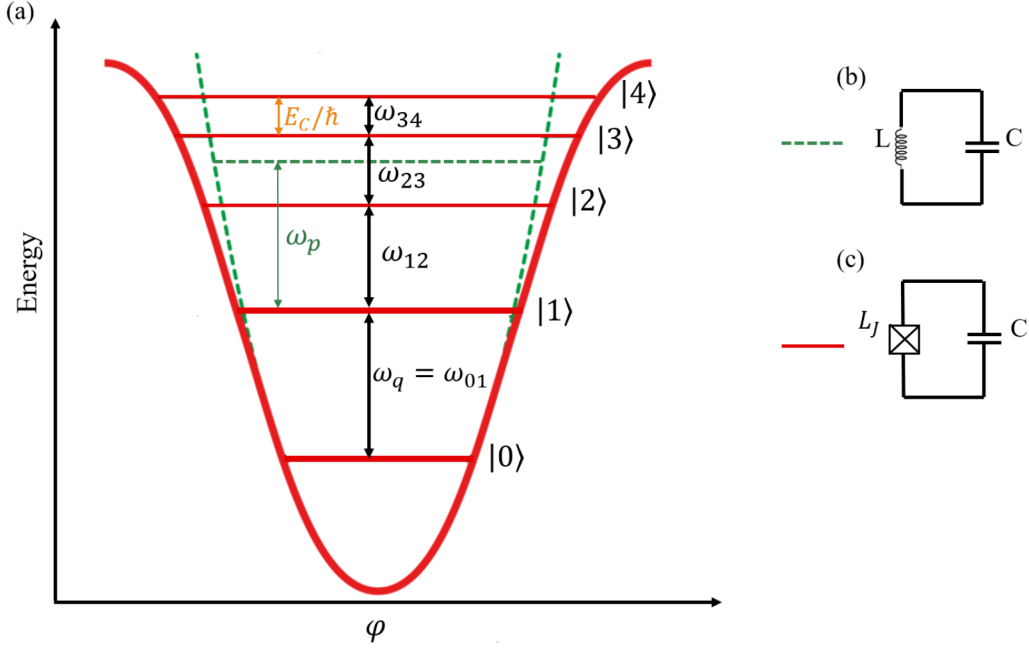


Figure 3.4: Cosine potential well of the transmon qubit (red line) compared to the quadratic potential of the LC oscillator (dashed green lines). The frequency between the ground and first excited state of the transmon is larger by $\delta_q = E_C/\hbar$ than the transition between the first excited and second excited state.

by the small charging energy, yielding the effective qubit frequency $\omega_q = \omega_p - E_C/\hbar$, see Fig. (3.4a). The frequency between the ground and first excited state of the transmon is larger by $\delta_q = E_C/\hbar$ than the transition between the first excited and second excited state. This anharmonic frequency shift prevents the higher transmon level pairs from absorbing or emitting energy quanta tuned to the qubit frequency $\hbar\omega_q$ of the lowest two levels, so we can treat the lowest two levels of a transmon as an effective two-level system, i.e., a qubit.

3.4 Resonator Coupling

In circuit QED, we measure the state of the transmon by dispersively coupling it to a strongly detuned readout resonator. This dispersive coupling shifts the frequency of the resonator by different amounts that depend on the occupied transmon energy without allowing energy to be disruptively transferred between the transmon and the resonator. Since the resonator has distinct frequencies, reflected signals accumulate distinct phases that depend on the transmon energy. It is thus possible to read out what the energy state of the transmon must have been by looking at the phase shift in a signal reflected off the resonator with a homodyne measurement.

To monitor the energy state of the transmon first we apply a weak probe tone to the resonator tuned to the mid-point between the distinct transmon-dependent resonance frequencies, This near-resonant probe tone transfers energy to the resonator, populating it with a discrete number of photons as quanta of the resonator mode $\hat{a}$. This injected resonator field amplitude interacts with the qubit, shifting its frequency accordingly, then leaks from the resonator at the rate $\kappa/2$. This leaked field is then amplified⁵ and measured in a homodyne setup that identifies the reflected phase-shift by comparing it with a copy of the original (unshifted) reference signal, see Fig. (3.5).

When a transmon qubit with bare frequency ω_q is dispersively coupled, with coupling energy $\hbar g$, to a microwave resonator with a bare (midpoint) frequency ω_r that is strongly detuned by $\Delta_{qr} \equiv \omega_q - \omega_r$ from the qubit, the coupling shifts the resonator frequency by $\pm\chi$ depending on the qubit state, yielding a total frequency splitting of 2χ between the two qubit-state-dependent resonances. In terms of the qubit-resonator coupling g , the dispersive shift of the

5. Amplification of the output field is because of the fact that superconducting qubits operate at GHz frequencies, therefore the energy of a single qubit excitation is much smaller than the thermal fluctuations associated with the room temperature electronics required to record a measurement result. In order to have a high measurement efficiency, the qubit state distinguishability must be boosted in the final collected signal so that it is much larger than this thermal noise floor.

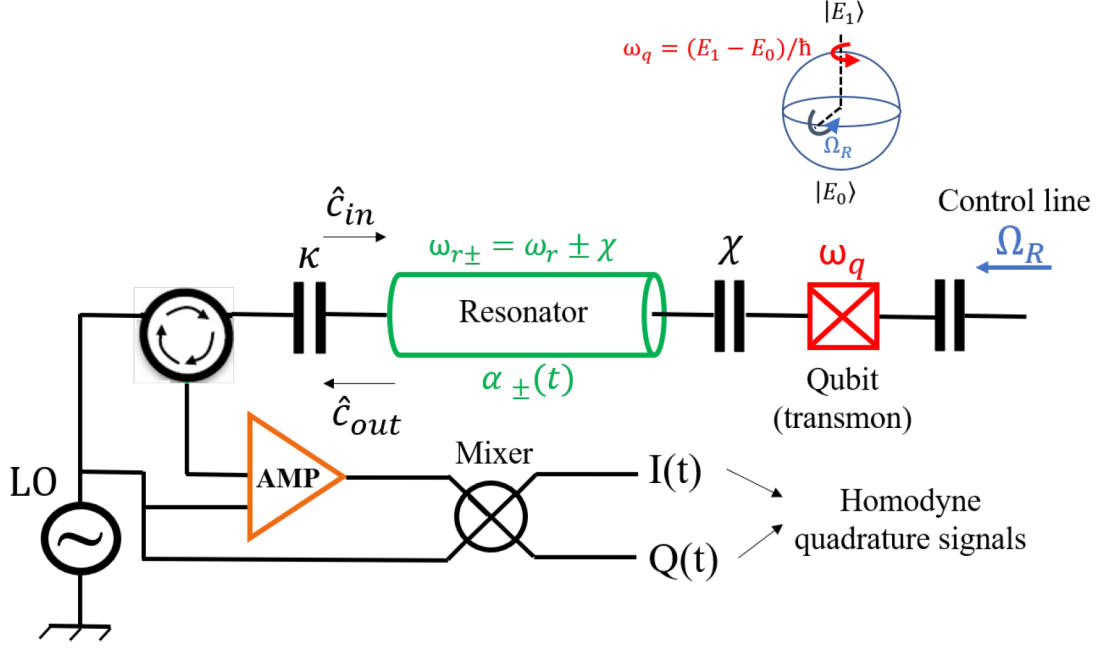


Figure 3.5: Schematic setup of a driven superconducting qubit, a transmon, capacitively coupled to a superconducting coplanar waveguide resonator. Given an input drive, the resonator becomes phase-entangled with the qubit energy eigenstates, then the field escapes from the resonator at the energy decay rate $\kappa/2$. The leaked resonator field mixes with the reflected part of the drive field to produce a propagating output signal in the transmission line. The signal field is amplified with a phase-preserving amplifier then evenly split to measure two orthogonal quadrature signals $I(t)$ and $Q(t)$ with balanced homodyne detectors.

resonator frequency is [87],

$$\chi \approx \frac{\omega_r}{\omega_q} \frac{g^2 \delta_q}{\Delta_{qr} (\Delta_{qr} - \delta_q)}, \quad (3.13)$$

where $\delta_q = \omega_q - (E_2 - E_1)/\hbar = E_C/\hbar$ is the qubit anharmonicity. The qubit-resonator Hamiltonian describing this coupling is [23],

$$\frac{\hat{H}_{qr}}{\hbar} = \frac{\omega_q}{2} \hat{\sigma}_z + (\omega_r + \chi \hat{\sigma}_z) \hat{a}^\dagger \hat{a}. \quad (3.14)$$

The resonator is also coupled to a transmission line that supports traveling input-output fields: The input $\hat{c}_{in}$ is a pump field entering the resonator while, the outgoing field $\hat{c}_{out}$ combines the escaping resonator field and the reflected pump field, and will be amplified

and detected later. The interaction Hamiltonian that describes the dynamics of the coupled resonator-transmission line is [88]

$$\frac{\hat{H}_{\text{rt}}}{\hbar} = i \left(\frac{\sqrt{\kappa}}{2} \right) (\hat{a} + \hat{a}^\dagger) (\hat{c}_{\text{in}} - \hat{c}_{\text{in}}^\dagger - \hat{c}_{\text{out}} + \hat{c}_{\text{out}}^\dagger), \quad (3.15)$$

which is subjected to the boundary input-output condition [89]

$$\sqrt{\kappa} \hat{a} = \hat{c}_{\text{in}} + \hat{c}_{\text{out}}. \quad (3.16)$$

The input-output fields have commutation $[\hat{c}_{\text{in}}, \hat{c}_{\text{in}}^\dagger] = [\hat{c}_{\text{out}}, \hat{c}_{\text{out}}^\dagger] = \Delta\omega/2\pi = \Delta f$ equal to the detection bandwidth, which we assume is $\Delta f \sim 1\text{GHz}$. This bandwidth leads to a temporal uncertainty of $\Delta t \sim 1/\Delta f \sim 1\text{ns}$ over which collected fields are averaged.

Following Heisenberg's equation of motion, introduced in Chapter 2, and the Hamiltonian in Eqs. (3.14) and (3.15) the resonator field evolves as

$$\begin{aligned} \frac{d\hat{a}}{dt} &= \dot{\hat{a}} = \frac{i}{\hbar} [\hat{H}_{\text{qr}} + \hat{H}_{\text{rt}}, \hat{a}] \\ &= -i(\omega_r + \chi \hat{\sigma}_z) \hat{a} + \left(\frac{\sqrt{\kappa}}{2} \right) (\hat{c}_{\text{in}} - \hat{c}_{\text{in}}^\dagger - \hat{c}_{\text{out}} + \hat{c}_{\text{out}}^\dagger). \end{aligned} \quad (3.17)$$

Using $[\hat{a}, \hat{a}^\dagger] = 1$, $[\hat{a}^\dagger \hat{a}, \hat{a}] = -\hat{a}$, and the boundary condition in Eq. (3.16) this becomes

$$\dot{\hat{a}} = -i(\omega_r + \chi \hat{\sigma}_z) \hat{a} + \sqrt{\kappa} (\hat{c}_{\text{in}} - \hat{c}_{\text{in}}^\dagger) - \frac{\kappa}{2} (\hat{a} - \hat{a}^\dagger). \quad (3.18)$$

Using the boundary condition of Eq. (3.16) and assuming that the resonator field remains in a coherent state such that $\langle \hat{c}_{\text{in}} \rangle = -i\varepsilon(t)e^{-i\omega_d t}/\kappa$ with $\omega_d \sim \omega_r$ and is conditioned on a particular qubit state, we can simplify Eq. (3.15) to an effective form [90–92]

$$\frac{\hat{H}_{\text{rt}}}{\hbar} = \hat{a}^\dagger \varepsilon(t) e^{-i\omega_d t} + \hat{a} \varepsilon^*(t) e^{i\omega_d t}, \quad (3.19)$$

assuming the drive envelope $\varepsilon(t)$ is varying slowly compared to the drive frequency ω_d , and applying the RWA to neglect fast-oscillating terms, Eq. (3.18) simplifies to the following pair of RWA coherent field evolution equations, as expressed in the rotating frame of the drive,

$$\dot{\alpha}_{\pm}(t) = -i(\omega_r - \omega_d \pm \chi) \alpha_{\pm}(t) - i\varepsilon(t) - \frac{\kappa}{2} \alpha_{\pm}(t). \quad (3.20)$$

Here $\alpha_{\pm}$ are the resonator coherent states conditioned on the qubit in the excited (+) and ground (−) states. Recall that $|\alpha_{\pm}|^2 = \bar{n}_{\pm}$ is the average number of photons in the resonator. In the special case when the drive ε is constant, we can find the two steady states ($\dot{\alpha}_{\pm} = 0$) in the resonator,

$$\alpha_{\pm}^{(\text{ss})} \equiv \frac{2\varepsilon}{\kappa} \frac{-i}{1 + i[2(\Delta_{rd} \pm \chi)/\kappa]}. \quad (3.21)$$

where $\Delta_{rd} \equiv \omega_r - \omega_d$ is the drive detuning from the midpoint resonator frequency.

Eq. (3.20) and the boundary condition in Eq. (3.16) imply that the outgoing field $\hat{c}_{\text{out}}$ can be described as

$$\alpha_{\pm}^{\text{out}}(t) \equiv \langle \hat{c}_{\text{out}} \rangle = \sqrt{\kappa} \alpha_{\pm}(t) + i \frac{\varepsilon(t)}{\sqrt{\kappa}}, \quad (3.22)$$

where $\varepsilon(t)/\sqrt{\kappa}$ is the drive envelope that is assumed that varies slowly compared to the drive frequency ω_d . The collected part of the output during Δt (the field amplitude are collected and sequentially integrated over bins of duration Δt) is

$$\alpha_{\pm}^{\text{col}}(t) \equiv i\sqrt{\eta \Delta t} \alpha_{\pm}^{\text{out}}(t), \quad (3.23)$$

where η is the measurement efficiency. And the lost part of the output during Δt is

$$\alpha_{\pm}^{\text{loss}}(t) \equiv i\sqrt{(1 - \eta) \Delta t} \alpha_{\pm}^{\text{out}}(t). \quad (3.24)$$

The leaked resonator field is amplified and then downconverted by mixing it with the orig-

inal microwave drive. The mixer produces two collected output homodyne signals, which have ensemble-averaged means that are proportional to the real and imaginary parts of the amplified and collected signal. In Section 2.2.3 we called the real part the in-phase homodyne signal $I(t) \propto \text{Re}[\langle \hat{a} \rangle] = \langle \hat{a} + \hat{a}^\dagger \rangle / 2$ and the imaginary part the quadrature homodyne signal $Q(t) \propto \text{Im}[\langle \hat{a} \rangle] = \langle \hat{a} - \hat{a}^\dagger \rangle / 2i$. For each signal realization in an ensemble, these mean values will be modified by additive Gaussian noise $\xi(t)$, with zero ensemble-averaged mean $\langle \xi(t) \rangle = 0$.

A phase-preserving amplifier, as will be used later in this thesis, will amplify both quadratures $I(t)$ and $Q(t)$ ⁶. Collecting both homodyne quadratures [95, 96], yields a net collection efficiency η that characterizes the fraction of signal photons contributing to the observed readout, with $\eta/2$ arriving at each homodyne quadrature.

Conditioned state trajectories have been reconstructed by filtering these noisy homodyne signals with different methods such as Bayesian updates, and stochastic master equations, as will be discussed in the next chapter. Prior to the work in this thesis, the efficient reconstruction of the state trajectories with these methods has been limited to known dynamics in Markovian regimes, where the qubit evolves slowly compared to the decay rate of the resonator. Chapter 6 describes extensions to these methods that are able to reconstruct the trajectories of a rapidly driven superconducting qubit even if the particular control drive is not known beforehand.

6. The result of this thesis is discussed with phase-preserving amplifiers. More details about the amplifiers are explained in these papers [93, 94].

4 Monitoring Qubit Evolution

Measurement of the quantum properties of a physical system is fundamental to connecting the theory of quantum mechanics and quantum information science to experiment and technology.[97–100]. However, sudden projective measurements are usually the only form of measurement discussed in the quantum mechanics postulates appearing in most undergraduate textbooks [101]. For such a projective measurement, precise information about a measured observable is immediately acquired and the system state is projected onto the eigenspace corresponding to the observed outcome.

In reality, however, this description is not sufficient to explain two important situations: first, measurements are not instantaneous, but occur over a nonzero duration of time that depends on the detailed response of the detector. In other words, a general treatment of quantum measurement must consider a gradual and continuous process of wavefunction collapse. Second, when some aspects of a system are continuously monitored. For example we saw in the previous chapter that the transmon has been monitored continuously using a weak probe tone. It is the subject of continuous measurement that describes such a measurement. This continuous measurement allows the observer to reconstruct the dynamics of a quantum system, and to track the evolution of its wavefunction before its collapse to an eigenstate.

In this chapter, we first discuss the basic notion of projective measurement in more detail then introduce generalized types of measurement, including weak and continuous measurements. Due to the growing interest in using real-time information for feedback control in quantum systems, these more general types of measurement have become increasingly important in the last decade [37, 102–108]. We place particular emphasis on monitoring a transmon qubit by continuously measuring the qubit-resonator system introduced in previous chapter by amplifying and collecting the microwave field that escapes the resonator.

4.1 Projective Measurement

The Stern-Gerlach experiment is a historically important example of quantum measurement that provided experimental evidence for the quantization of electron spin into only two discrete angular momentum values [51]. In this experiment, a beam of silver atoms was generated in an atomic beam furnace, then sent towards an inhomogeneous magnetic field with a large spatial gradient. According to classical physics one would expect the magnetic moments of the silver atoms to be randomly oriented and thus deflected by the inhomogeneous magnetic field by random amounts to produce a broad distribution of impact positions on a final scintillation plate. However, the researchers observed that the beam was split into only two well-separated positions having similar uncertainty to the initial beam, suggesting only two values, which were later named spin up and spin down.

To make this example more precise, consider atomic spins that have been polarized to all point along the x-axis. When these spins pass through such an inhomogeneous magnetic field with a large enough magnetic field gradient, dB/dz , in the z-direction, they will impact the screen to create two separated distributions along the z-axis as illustrated in Fig. (4.1). These two distinguishable lobes are defined to be the eigenstate representations of $\hat{\sigma}_z$, and associated with normalized eigenvalues of ± 1 . We can easily distinguish whether the spins are up (+1) or down (−1) after the measurement, so this is an example of a projective measurement.

However, the magnetic field gradient in this same experiment could also be chosen to be very weak, thus causing much smaller deflections of each atom. The distributions of the impacts on the screen will then overlap substantially, so we will not be able to distinguish whether the spins are up or down with certainty. However, we can still infer some information about which spin value would be more likely given an observed impact position. This complementary situation is an example of a weak measurement where only partial information about the

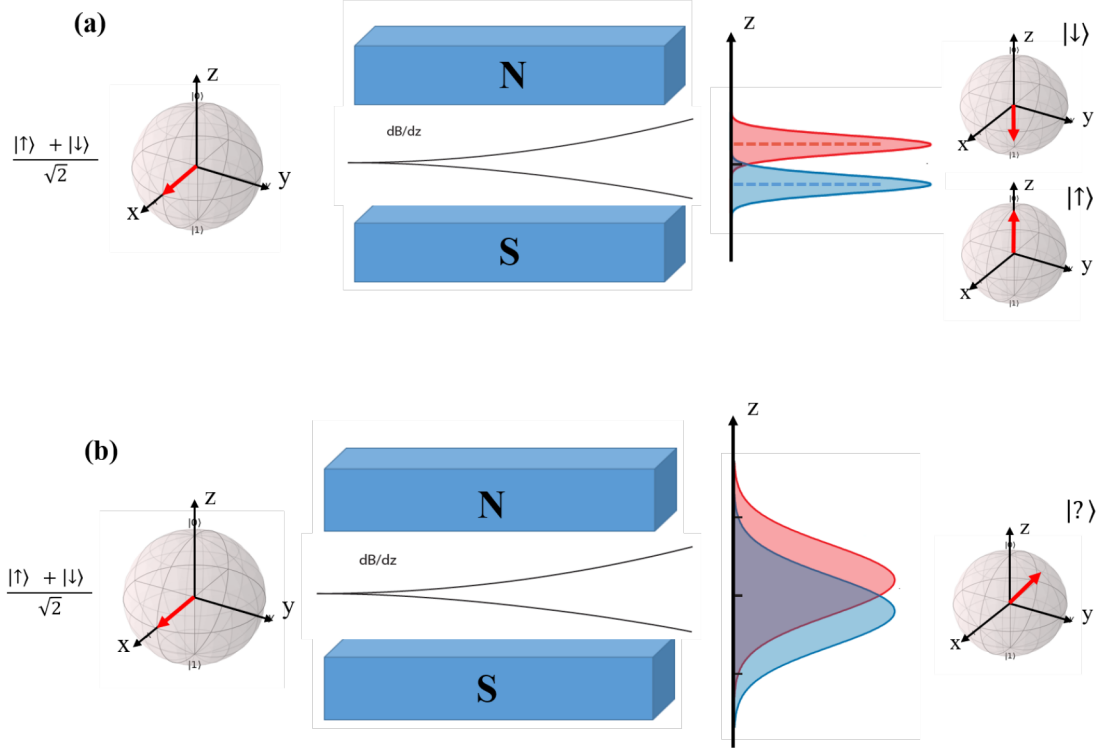


Figure 4.1: Stern-Gerlach experiment. The spin initially is aligned along x, depending on the strength of the magnetic field gradient (a) the spins will make two separated distribution (strong gradient) or (b) the distributions will overlap (weak gradient)

spin state is extracted [109]. In this case, the spin state is not fully projected into the eigenspace of a definite spin value.

To model the outcomes of this experiment mathematically, we need to review how projective measurements work in quantum mechanics. The essential rule that connects the vector space description of quantum mechanics to the probabilistic events observed in the laboratory is Born's rule. Born's rule defines the probability of getting each particular outcome of a measurement for a given observable. For simplicity, model the two spin states as the computational basis states of a qubit such that the spin of each atom is prepared in a particular superposition state $|\psi\rangle = \alpha|g\rangle + \beta|e\rangle$. If one passes this atom through a strong Stern-Gerlach apparatus, it will randomly impact in one of the lobes. According to Born's rule, the probability of finding the spin in the lobe corresponding to ground state $|g\rangle$ would

be

$$P(+z|\psi) = |\langle g|\psi\rangle|^2 = |\alpha|^2, \quad (4.1)$$

the probability of finding the qubit in the excited state $|e\rangle$ is $|\beta|^2$. Note that mathematically, this projective measurement is treated as instantaneous and non-unitary (irreversible), and causes the spin state $|\psi\rangle$ to collapse onto one of the two measurement eigenstates $|g\rangle$ or $|e\rangle$ afterwards.

As postulated, Born's rule applies only to precise projective measurements, which dramatically affect the dynamics of the system. Since the time-dependence of the detector dynamics are omitted, projective measurements can only reveal information available at temporal boundaries of an evolution interval, in this case about the spin state just before the atom enters the Stern-Gerlach device. This boundary information is generally insensitive to transient evolution during the device interaction. A generalization of Born's rule is required to model realistic experiments, and will be introduced in the next section. This generalization will account for imperfect measurements, arising from limited detection accuracy, and permit the inclusion of time-dependence in the measurement process.

4.2 Generalized Measurement

Recall that a quantum observable is modeled as a Hermitian operator, such as $\hat{\mathcal{O}}$, and can be expanded into its eigenvalues λ_i and eigenvectors $|v_i\rangle$ as $\hat{\mathcal{O}} = \sum_i \lambda_i |v_i\rangle\langle v_i|$. The eigenvalues are interpreted to be the distinct measurement outcome values for $\hat{\mathcal{O}}$, while their associated eigenvectors are the resulting states of certainty about the observable value. For each eigenstate $|v_i\rangle$, $\hat{P}_i = |v_i\rangle\langle v_i|$ is a measurement operator that transforms the quantum state $\hat{\rho}$ into that particular eigenstate. That is, after the measurement takes place, the density

matrix $\hat{\rho}$ updates according to the rule

$$\hat{\rho} \xrightarrow{i} \hat{\rho}' = \frac{\hat{P}_i \hat{\rho} \hat{P}_i}{\text{Tr}(\hat{P}_i \hat{\rho} \hat{P}_i)}. \quad (4.2)$$

The numerator projects $\hat{\rho}$ onto the $\hat{P}_i$ subspace while the denominator renormalizes the state by the probability to obtain the specific result i , $P(i|\hat{\rho}) = \text{Tr}(\hat{P}_i \hat{\rho} \hat{P}_i) = \text{Tr}(\hat{P}_i \hat{\rho})$, using the cyclic property of the trace and $\hat{P}_i^2 = \hat{P}_i$.

The update rule in Eq. (4.2) can be generalized to describe any kind of quantum measurement possible by replacing the projectors $\hat{P}_i$ with more general measurement operators $\hat{M}_k$ that only partially project the state towards more probable observable values. The density matrix $\hat{\rho}$ after the measurement updates according to the rule

$$\hat{\rho} \xrightarrow{k} \hat{\rho}' = \frac{\hat{M}_k \hat{\rho} \hat{M}_k^\dagger}{\text{Tr}(\hat{M}_k \hat{\rho} \hat{M}_k^\dagger)}. \quad (4.3)$$

Such a replacement would imply that the probabilities for the outcomes have the form $P(k|\hat{\rho}) = \text{Tr}(\hat{M}_k \hat{\rho} \hat{M}_k^\dagger) = \text{Tr}(\hat{M}_k^\dagger \hat{M}_k \hat{\rho}) = \text{Tr}(\hat{E}_k \hat{\rho})$. The Kraus operators $\hat{M}_k$ obey $\hat{E}_k = \hat{M}_k^\dagger \hat{M}_k$, where $\hat{E}_k$ are positive semidefinite and encode the conditional probabilities for the detector response. Since $\sum_k P(k|\hat{\rho}) = 1$, the positive operators $\hat{E}_k$ must also partition the unity operator $\sum_k \hat{E}_k = \hat{\mathbb{I}}$. Such a complete partition $\{\hat{E}_k\}$ is called a positive operator-valued measure (POVM) over the index k that labels distinguishable detector outcomes. Notably, the index k may not perfectly correspond to the eigenvalue labels i , and may have a different number of possible outcomes than there are observable eigenvalues. Projective measurements are the special case when $\hat{M}_k$ is the projector onto the eigenspace corresponding to λ_k .

The most straightforward way of implementing a POVM in practice is to couple a quantum system to an ancillary detector sub-system, then perform a projective measurement on only

the detector. The result will be a generalized quantum measurement of the original system, with the amount of projection dependent upon the correlation between detector outcomes and observable eigenvalues. In other words, generalized measurements can be realized by enlarging the Hilbert space and performing a standard projective measurement on only the detector sub-system of the enlarged space. We will employ this technique to implement specific generalized measurement in later chapters.

4.3 Repeated Measurement and Monitoring

The measurement process in real experiments takes place over a finite time scale, determined by the strength of interaction between the quantum system and the probe system. If the quantum system is weakly coupled to the probe then the uncertainty in a single measurement is very large compared with the separation between the eigenvalues. For example if the probability distributions for the ground and excited state of a qubit overlap, it would be more challenging to determine whether the qubit is in the ground or excited state after the measurement [40]. Because we only get partial information about the system, its state only partially collapses toward the most probable eigenstate [38].

A continuous measurement is one in which information is continually extracted from a system. In other words, during the continuous measurement, the amount of information that we obtain from measuring it goes to zero as the duration of the measurement time goes to zero. In order to construct such a measurement, we can divide time into a sequence of intervals of length Δt , and consider a weak measurement in each time interval.

4.3.1 Discrete Monitoring: Bayesian Trajectories

By using the Born rule, the modulus squared of quantum state amplitudes are probabilities. These probabilities satisfy the principles of probability theory, including Bayes' rule for updating probability distributions given new information. Quantum state collapse could be viewed as an informational update related to this correspondence. In other words, the Bayes' rule for classical probabilities forces the existence of a corresponding quantum state collapse in order for the Born rule to be consistent.

Given a pure qubit state $|\psi\rangle_q = \alpha|0\rangle + \beta|1\rangle$, and a measurement result r , the probabilities update as $P(k|r) = P(r|k)P(k)$ ($k \in \{0, 1\}$) with multiplicative likelihoods $P(r|k)$, one can renormalize it after updating the distribution. The total probability for the result r is $P(r) = \sum_k P(r|k) P(k)$, and generally depends upon the prior qubit occupation information. These results can be compactly encoded into a Kraus operator $\hat{M}_r = \sum_k \sqrt{P(r|k)} |k\rangle\langle k|$ that multiplies the prior qubit state with the correct likelihood (up to an arbitrary unitary factor that would introduce additional phase backaction). The result probability by construction is then

$$P(r) = \langle\psi_q|\hat{M}_r^\dagger \hat{M}_r|\psi_q\rangle. \quad (4.4)$$

The state update corresponding to Bayes rule is $\hat{M}_r|\psi_q\rangle$, followed by renormalization.

Because of the central limit theorem, many laboratory detectors have probability distributions that approximate Gaussian statistics. A Gaussian Kraus operator has the form,

$$\hat{M}_r = \sum_k (2\pi \sigma_k^2)^{-\frac{1}{4}} e^{-(r-\mu_k)^2/(4\sigma_k^2)} |k\rangle\langle k|, \quad (4.5)$$

where μ_k is the state-dependent mean and σ_k is the state-dependent variance. Note that the readout r can always be shifted and rescaled so the two possible qubit state values correspond to readout values of ± 1 . With this simplification the Kraus operator in Eq. (4.5) will have

the form

$$\hat{M}_r = (2\pi\sigma^2)^{-\frac{1}{4}} e^{\left(-\frac{(r^2+1)}{4\sigma^2}\right)} \left[e^{\left(-\frac{r}{2\sigma^2}\right)} |0\rangle\langle 0| + e^{\left(+\frac{r}{2\sigma^2}\right)} |1\rangle\langle 1| \right], \quad (4.6)$$

where all components in the exponent have been factored out as state-independent except the cross-term components. These constant terms cancel during state renormalization, so they can be removed when we are only concerned about state dynamics of informational collapse. By using a Pauli operator $\hat{\sigma}_z = |0\rangle\langle 0| - |1\rangle\langle 1|$ we can encode the standardized readout values. The simplified Kraus operator will have the form $\hat{M}_r = \exp(r\hat{\sigma}_z/2\sigma^2)$.

A noisy sequence of independent Gaussian measurements can be averaged to coarse-grain the readout and reduce the variance. To have this property the variance must have the form $\sigma^2 = \tau/\Delta t$ for characteristic timescale τ to achieve unit variance and averaging time bin Δt . Following the law of sums of variance, averaging two successive time bins will have the variance $\tau/2\Delta t$. Any continuous monitoring setup of a qubit with Markovian time bins will have this property, and therefore conform to the form of the standard Gaussian time-sequence Kraus operator $\hat{M}_r = \exp(r\Delta t\hat{\sigma}_z/2\tau)$.

The state update is then described by the composite map (including the unitary evolution)

$$\hat{\rho}(t_{i+1}) = \frac{\hat{M}_r \hat{\rho}(t_i) \hat{M}_r^\dagger}{\text{Tr}(\hat{M}_r \hat{\rho}(t_i) \hat{M}_r^\dagger)} \quad (4.7)$$

4.3.2 Continuous Monitoring: Stochastic Master Equations

For analytic convenience, the time continuum limit $\Delta t \rightarrow 0$ can be used to derive stochastic master equations (SME) from the update equation Eq. (4.7) while taking this limit of small Δt the Markovianity is formally preserved. In order to get a description for the evolution of a quantum system under continuous measurement we first start with unitary evolution of a closed (isolated) system. In terms of $\hat{\rho}$ this is described by the Schrödinger von - Neumann

equation, Eq. (2.24)

$$\frac{d}{dt}\hat{\rho}(t) = -\frac{i}{\hbar}[\hat{H}(t), \hat{\rho}(t)]. \quad (4.8)$$

When the system interacts with the environment, as is required in order to control the quantum system, these unitary dynamics must be generalized to non-unitary dynamics appropriate for an open system. The unique generalization of Eq. (4.8) without environmental memory (i.e. Markovian dynamics) is the Lindblad master equation [97, 110],

$$\frac{d}{dt}\hat{\rho}(t) = \mathcal{L}\hat{\rho} = -\frac{i}{\hbar}[\hat{H}, \hat{\rho}(t)] + \sum_{n=1} \mathcal{D}[\hat{c}_n] \hat{\rho}(t), \quad (4.9)$$

which describes the system dynamics on average while treating the environment as a dissipation bath where system information is transferred and lost. Each $\hat{c}_n$ is a Lindblad operator that describes the dissipative coupling to the environment, while $\mathcal{D}[\hat{c}_n]$ is the dissipation ‘superoperator’ defined by its linear action on $\hat{\rho}$

$$\mathcal{D}[\hat{c}_n] \hat{\rho} = \frac{1}{2}(2\hat{c}_n \hat{\rho} \hat{c}_n^\dagger - \hat{c}_n^\dagger \hat{c}_n \hat{\rho} - \hat{\rho} \hat{c}_n^\dagger \hat{c}_n). \quad (4.10)$$

For the relaxation and dephasing mechanisms discussed in chapter 2, the Lindblad operators are simple. For qubit relaxation $\hat{c}_{\text{relax}} = \sqrt{\Gamma_1} \hat{\sigma}$, where $\Gamma_1 = 1/T_1$ is the relaxation rate, T_1 is the exponential time constant for relaxation decay, and σ is the lowering operator. For qubit dephasing, the Lindblad operator $\hat{c}_{\text{dephasing}} = \sqrt{\Gamma_d} \hat{\sigma}_z$ where $\Gamma_d = 1/2T_2$, where Γ_d is the decay envelope rate for coherent oscillations, T_2 is the exponential timescale of the dephasing decay, and $\hat{\sigma}_z$ is the Pauli operator diagonal in the qubit basis.

The Lindblad master equation is an averaged evolution equation, so does not take into account any particular observational records for the system. Measurement modifies the state of the system due to the backaction of the informational update, so we would like to have a similar time-evolution description for the state when conditioned upon the observed

measurement results of a particular experimental run.

The different evolution of the density matrix $\hat{\rho}(t)$ conditioned on a measurement record with additive Gaussian white noise can be described by adding a stochastic term to the Lindblad master equation that corrects the dynamics by keeping track of the deviation of the observed record from the expected mean

$$\frac{d}{dt}\hat{\rho}(t) = -\frac{i}{\hbar}[\hat{H}, \hat{\rho}(t)] + \sum_{n=1} \mathcal{D}[\hat{c}_n] \hat{\rho}(t) + \sqrt{\eta} \mathcal{H}[\hat{c}_n] \hat{\rho} dW, \quad (4.11)$$

where the Weiner increment dW is a zero-mean, Gaussian-distributed random variable with variance dt that satisfies the Itô rule $dW^2 = dt$ [110]. The innovation superoperator $\mathcal{H}[c]$ updates the density matrix ρ as follows,

$$\mathcal{H}[\hat{c}]\hat{\rho} = \hat{c}\hat{\rho} + \hat{\rho}\hat{c}^\dagger - \langle \hat{c} + \hat{c}^\dagger \rangle \hat{\rho}, \quad (4.12)$$

using the same Lindblad collapse operators $\hat{c}_n$ appearing in the dissipative state update. The scaling factor $\eta \in [0, 1]$ indicates the measurement efficiency which quantifies the fraction of the measured information retained in the collected signal. Note that when $\eta = 0$, the stochastic Eq. (4.11) reduces to the ensemble-averaged Lindblad Eq. (4.9). By feeding the continuously measured signal $dr(t)$ into this stochastic master equation, we construct stochastic evolution of the density matrix, known as quantum trajectories [111, 111–118]. Since dW has zero mean, the ensemble average of computed trajectories reduces to the deterministic trajectory of the Lindblad equation by construction.

A simple way to compute solutions for the stochastic master equation is with Monte-Carlo simulation, in which one generates dW using a random number generators and then computes $\hat{\rho}(t + \Delta t) = \hat{\rho}(t) + d\hat{\rho}$ iteratively. There are also different modifications of this procedure with better numerical stability such as Rouchon method [119]. The experimental use of

the master equation is the computation of a predicted state $\hat{\rho}(t, r(t))$ from a sequence of measurement outcomes dr . In this case, we can compute dW from dr rather than from a random number generator.

5 Machine Learning for Data Processing

Machine learning based on neural networks (NN) [120–122] has revealed a remarkable potential for solving a wide variety of complex problems. As the field continues to grow rapidly, its applications have appeared in many areas of science. More recently, a number of first applications to quantum physics have emerged. These include quantum state tomography [44, 123–125], reconstructing a quantum system’s dynamics [43, 126], quantum error correction [127–129], identification of quantum phases of matter [130–133], quantum entanglement [134, 135], and tackling quantum many-body dynamics [45, 136, 137].

An artificial neural network model is based on the idea of reproducing how neurons connect in the human brain. Such an artificial neuron, or “perceptron” is a simplified mathematical model of a biological neuron. In actual neurons, the dendrites receive electrical signals from other neurons via the axons, while in perceptrons, these electrical signals are expressed as numerical values. Different amounts of electrical stimuli flow at synapses between the dendrites and axons. This variability is modeled in the perceptron by multiplying each input value by a value called a weight. The intensity of the electrical signal determines the ease with which a neuron fires (firing means that the threshold for an action potential has been passed). To model this phenomenon, a perceptron calculates the weighted sum of the inputs and applies a step function to the sum to model the conditional firing of the neuron. Similar to biological neural networks, this thresholded output is then fed to other perceptrons as a chained cascade.

Despite the fact that we have not yet been able to replicate the brain, the field of artificial intelligence offers remarkably effective solutions to a wide range of problems.

Neural networks learn or are trained by processing examples. These examples contain a known ‘input’ and ‘target’¹, which form probability-weighted associations between them,

1. This is known as supervised learning [138]. There are other types of learning such as unsupervised

that are stored within the data structure of the network. The training is usually performed by determining the difference (error) between the predicted output of the network and the target. Then based on the value of the error (difference), the network adjusts its weighted associations. By making successive adjustments, the neural network’s output will become more similar to the target output. Backpropagation [144–146] is the algorithm that determines how a single training example would adjust the weights and biases. More details about the neural networks will be elaborated in the sections to come.

This chapter introduces foundational and state-of-the-art techniques in machine learning and data processing for physicists but will mainly focus on the two different types of neural network used in this thesis, Long Short Term Memory Recurrent Neural Networks (LSTM RNN) [147, 148] which can be directly used for the purpose of reconstructing strongly driven superconducting qubit trajectories, and denoising autoencoders [149, 150] which can be used for state estimation of the superconducting qubit.

5.1 A Brief History of Neural Networks

Even though the study of the human brain is thousands of years old, the first steps towards constructing artificial neural networks took place with the mathematical modeling of a biological brain by Walter Pitts and Warren McCulloch in 1943. The neuron explained in their paper, “*A Logical Calculus of the Ideas Immanent in Nervous Activity*” [151] is very simple and has very limited capability of learning but yet is the foundation for artificial neural networks. In 1957, Frank Rosenblatt, a neuro-biologist from Cornell university, started working on the “perceptron”, which was a neural network unit that computes a weighted sum of the inputs, subtracts a threshold, and passes one of two possible values out as the result. This

learning [139–141] and reinforcement learning [142, 143] but in this thesis we will only see the supervised models.

perceptron had true learning capabilities and was able to do binary classification [152]. It was the first neural network built in hardware and has remained in use even to this day.

The first version of back-propagation started in 1960 when Henry J. Kelley, a professor of aerospace at the Virginia Polytechnic Institute, was working on control theory and flight optimization. In the context of control theory, he was able to derive the basics of backpropagation using dynamic programming[153]. In 1962, a computer programmer Stuart Dreyfus showed a simplified backpropagation model that used simple applications of the derivative chain rule, instead of dynamic programming [154]. It was a small step that strengthened the future of deep learning.

We say future because interestingly back in those days people still did not know how to connect the back-propagation technique to a neural network. 1965 was the year that multi-layer neural networks were created by the mathematicians Alexey Grigoryevich Ivakhnenko along with Valentin Grigorevich Lapa. Ivakhnenko is often considered as the father of deep learning. Then, later from 1982 to 1998, many neural network architectures were created including convolutional neural networks, recurrent neural networks, Boltzmann machines, and more importantly, they successfully implemented backpropagation[155].

5.2 The Building Blocks of Neural Networks

Fig. (5.1) shows a simple Artificial Neural Network (ANN) architecture with only one hidden layer in the middle, but some networks have multiple hidden layers. The leftmost layer in this network is called the input layer, and the neurons within the layer are called input neurons, and the rightmost or output layer contains the output neurons.

The goal of neural networks is to approximate a function f which maps an input x to an output y . An ANN defines this mapping $y = F(x; \theta)$, by learning the best values of the

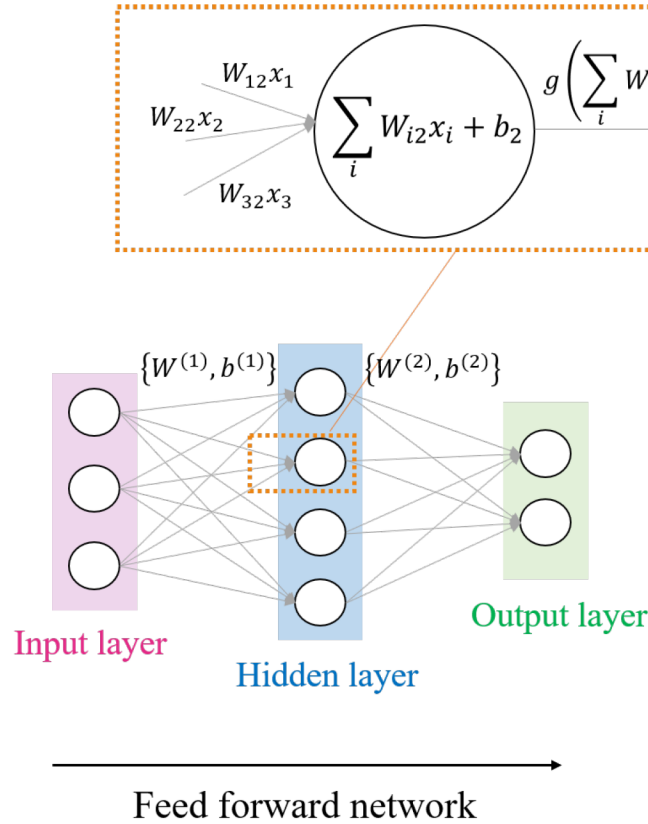


Figure 5.1: A simple feed forward neural network architecture with only one hidden layer in the middle. The leftmost layer in this network is called the input layer, and the neurons within the layer are called input neurons, and the rightmost or output layer contains the output neurons. Every neuron depends on the outputs of all the neurons in the previous layers (assuming that all neurons are connected together).

tunable parameters² θ that will lead to the best approximation of f .

ANN are often applied to problems with high-dimensional inputs and outputs, using many tunable parameters. One of the simplest ANN models is called a feedforward neural network because information flows through the function being evaluated from x , through the intermediate computations used to define F , and finally to the output y . The output y_j of a neuron n_j in this feedforward flow of information is computed by (this is just flow from one layer to the next layer. You should make it clear that this pattern will be composed with

2. For simplicity in the rest of this chapter we collectively call weights W and biases b , together θ , so θ would be a vector that contains all weights and biases.

each additional layer.)

$$y_j = g\left(\sum_{i=1}^N W_{ij}x_{ij} + b_j\right), \quad (5.1)$$

where W_{ij} and x_{ij} are the weights and inputs from the i^{th} neuron to j^{th} neuron, with a activation function g which I will talk about shortly. b_j refers to the associated bias with each neuron and N is the total number of incoming inputs. The goal of approximating the function f can be achieved by tuning the weights and biases during the training process. Neural networks are trained using the process called gradient descent which is an optimization algorithm to find a local minimum of a differentiable function (usually called a lost function) and consists of a backward propagation step (which is basically the chain rule) to change the weights in order to reduce the cost function. What follows in this chapter is a discussion of how we can accomplish this task.

In Eq. (5.1) the activation function g , transforms the input weighted sum to a thresholded output between a lower limit and upper limit. The purpose of having an activation function in the network is to introduce non-linearity into the output of a neuron in order to help the network learn complex patterns in the data. Some of the common activation functions are (see Fig. (5.2))

- **Sigmoid function**, also known as a logistic curve (or inverted Fermi-Dirac distribution), $g = \sigma(x) = 1/(1 + e^{-x})$. It limits the neuron's output to be between 0 and 1 which is ideal for binary classification problems. However, there are some disadvantages of using this function, e.g., the neurons' activation saturates at either tail of 0 or 1, because the gradient at these regions is almost zero. Therefore the small changes in the input in these regions will not be reflected in the output and gives rise to a problem called '*vanishing gradient*'.
- **Tanh**, it ranges between -1 and 1, $g(x) = \tanh(x) = 2/(1 + e^{-2x}) - 1$. Similar to the

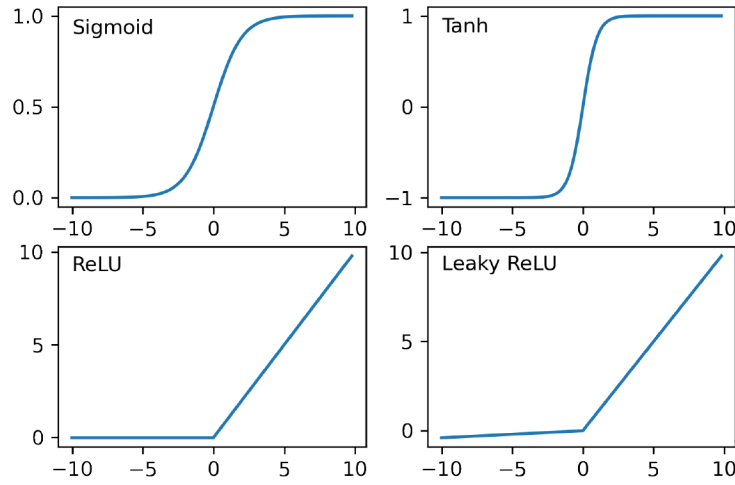


Figure 5.2: Commonly used activation functions.

Sigmoid, tanh also suffers from vanishing gradient problem.

- **ReLU** (Rectified Linear Unit), ranges from 0 to ∞ , $g(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$

In recent times, this is the most used activation function. There are a lot of reasons for this. One of the most important ones is that it is very easy to compute, so it is computationally efficient to work with for calculating the gradient. For positive inputs the linear ramp never has a vanishing gradient; however, the vanishing gradient still occurs for negative inputs.

- **Leaky ReLU**, is an improved version of the ReLU activation function. It addresses the remaining problem of a vanishing gradient for negative inputs in the ReLU activation function by defining a small slope in the negative part.

It is worth noting that adding more layers (or equivalently adding more neurons) to the network will increase its overall non-linearity. Because more layers will result in applying

more activation functions, thus adding more non-linearity into the system. However, more layers do not necessarily translate into better results, as we will see in the next section, because of the risk of overfitting.

5.3 Training Neural Networks

In the previous section we learned what each part of a neural network looks like; this section explains how the network learns. The network learns by adjusting its weights and biases and it does this by minimizing a function called a cost function, also called loss function. There are different types of cost functions such as mean squared error (MSE), mean absolute error (MAE), binary cross-entropy, and etc. A commonly used and well-behaved loss function adds up the squares of the differences between the targets (labels), y and the network's outputs³, $\hat{y}$ and is called mean squared error

$$\mathcal{L}(\theta) = \frac{1}{2M} \sum_{i=1}^M (y_i - \hat{y}_i)^2, \quad (5.2)$$

where θ denotes the collection of all weights and biases in the network. M is the total number of training inputs and makes the gradient become the average of M terms instead of a sum, therefore its scale does not change when we add more data points. The $1/2$ coefficient is merely for convenience; taking gradient of the cost function the 2 from the square term cancels out. There are some advantages of using the MSE as a loss function, e.g., in the form of quadratic equation, Eq. (5.2), the convexity guarantees a minimum [156]. In addition, By squaring errors, it penalizes the model for making larger errors. Note that the network output $\hat{y}$ depends on W , and b , but to keep the notation simple I have not indicated this dependence.

3. Here the “hat” symbol denotes estimates.

One important problem with MSE is that it treats $y = 0.01$ and $y = 0.00001$ as nearly equivalent, to fix this problem the logarithmic loss has been used. A common type of logarithmic loss function (that we also use in Chapter 6) is cross-entropy. In cross-entropy loss function, each predicted class probability is compared to the actual class output 0 or 1, and it penalizes the probabilities based on how far they are from the reference values. This penalty is logarithmic as I mentioned earlier, yielding a large score for large differences and small score for small differences. Cross-entropy is defined as (for n classes)

$$\mathcal{L}_{CE} = - \sum_i^n t_i \log(p_i), \quad (5.3)$$

where t_i is the actual label and p_i is the probability for the i^{th} class. For binary classification the Eq. (5.3) becomes,

$$\mathcal{L}_{CE} = -t \log(p) - (1 - t) \log(1 - p). \quad (5.4)$$

Binary cross-entropy is usually calculated as the average cross-entropy over all the training examples.

The aim of the training algorithm should be to minimize the cost function $\mathcal{L}(\theta)$, by finding a set of weights and biases to make the cost as small as possible using an algorithm known as gradient descent [157]. In order to minimize a convex function, we need to find a stationary point. There are different methods to do this. One possible approach is to start at an arbitrary point, and move along the gradient at the point towards the next point and repeat until converging to a stationary point, see Fig. (5.3). In the search for a stationary point we need to consider two things: the direction and the step size to move toward the stationary point. Therefore, the iterative search of gradient descent can be described as following the

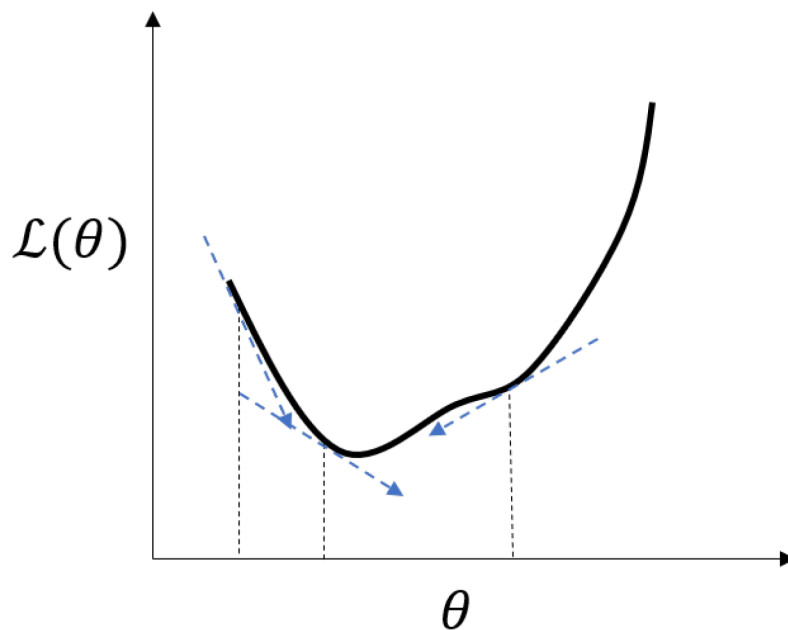


Figure 5.3: An example of a gradient descent search for a minimum.

recursive rule

$$\theta \mapsto \theta - \eta \frac{\partial \mathcal{L}}{\partial \theta}, \quad (5.5)$$

where the positive scalar η is called the learning rate that specifies the step size to take towards the minimum of the cost function. It should not be too small because many iterations will be required, which will make the training process costly. It should also not be too large because it may overshoot the actual minimum.

There is another optimization algorithm called stochastic gradient descent that is used for training a model. Similar to gradient descent, in stochastic gradient descent we update a set of parameters θ in an iterative way in order to minimize the cost function. However, while in gradient descent the network has to run through the entire training dataset to do a single update for a parameter in each iteration, in stochastic gradient descent method, we use only one example from the training set to update a parameter⁴. For a large training set, the

4. Sometimes a subset of the training set is used in order to update a parameter. This subset is called

optimization process may take longer because in every iteration we have to run through the complete training set in order to update θ . Stochastic gradient descent on the other hand, improves itself for each training sample in a shorter time moreover guarantees reaching a global minimum [158].

5.4 Backpropagation

The back-propagation algorithm allows the information from the cost function to flow backwards through the network (using a chain rule for derivatives of nested functions), in order to compute the gradient [122]. For simplicity, I compute the cost function for a neural network with only one hidden layer which is shown in Fig. (5.1). This network has a set of weights $W^{(1)}$ and biases $b^{(1)}$ that connect the input layer to the hidden layer, and another set of weights $W^{(2)}$ and biases $b^{(2)}$, that connect the middle layer to the input layer. We can write the inputs and outputs of each layer as follows

$$\begin{aligned}
z^{(1)} &= W^{(1)} x + b^{(1)} \quad \rightarrow \text{input to hidden layer} \\
a^{(1)} &= g\left(z^{(1)}\right) \quad \rightarrow \text{output of hidden layer} \\
z^{(2)} &= W^{(2)} a^{(1)} + b^{(2)} \quad \rightarrow \text{input to output layer} \\
\hat{y} &= g\left(z^{(2)}\right) \quad \rightarrow \text{output}
\end{aligned} \tag{5.6}$$

where $\hat{y}$ is the computed output from the input x , and g is the activation function.

To perform backpropagation, we start from the MSE loss function defined in Eq. (5.2). The error between the estimated and real outputs for M samples is as follows (for simplicity we

a mini-batch and the optimization method is called mini-batch stochastic gradient descent. The mini-batch size defines the number of samples that will be propagated through the network. For example if we have 1000 training sample and we decide to have a mini-batch size of 100, the optimization algorithm takes the first 100 samples from the training dataset and trains the network, then it takes the second 100 samples and trains the network again. The network keeps doing this until taking the last mini-batch set of the dataset.

remove the sum),

$$\frac{\partial \mathcal{L}}{\partial y(x)} = \frac{1}{M}(y_i - \hat{y}_i). \quad (5.7)$$

Using the chain rule iteratively and Eq. (5.7), we compute

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial z^{(2)}} &= \frac{\partial \mathcal{L}}{\partial y(x)} \frac{\partial y(x)}{\partial z^{(2)}} = \frac{\partial \mathcal{L}}{\partial y(x)} g'(z^{(2)}) \\ \frac{\partial \mathcal{L}}{\partial z^{(1)}} &= \frac{\partial \mathcal{L}}{\partial y(x)} \frac{\partial y(x)}{\partial z^{(2)}} \frac{\partial z^{(2)}}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} = \frac{\partial \mathcal{L}}{\partial y(x)} g'(z^{(2)}) W^{(2)} g'(z^{(1)}) \\ \frac{\partial \mathcal{L}}{\partial W^{(2)}} &= \frac{\partial \mathcal{L}}{\partial z^{(2)}} \frac{\partial z^{(2)}}{\partial W^{(2)}} = \frac{\partial \mathcal{L}}{\partial z^{(2)}} a^{(1)} \\ \frac{\partial \mathcal{L}}{\partial W^{(1)}} &= \frac{\partial \mathcal{L}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial W^{(1)}} = \frac{\partial \mathcal{L}}{\partial z^{(1)}} x. \end{aligned} \quad (5.8)$$

Similarly, for the biases we will have

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial b^{(2)}} &= \frac{\partial \mathcal{L}}{\partial z^{(2)}} \frac{\partial z^{(2)}}{\partial b^{(2)}}, \\ \frac{\partial \mathcal{L}}{\partial b^{(1)}} &= \frac{\partial \mathcal{L}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial b^{(1)}}. \end{aligned} \quad (5.9)$$

This can be generalized for the networks having more than one hidden layer. Substituting the weight and the bias gradients of Eq. (5.8) and Eq. (5.9) into Eq. (5.5), the network parameters in each layer we updated.

5.5 Cross Validation

In order to train and test the model through a neural network while avoiding biases in the learning process, we must divide our dataset to three sets: the training set, which is used to fit the model; the validation set, which is used to evaluate the model built using a training set; and, the test set, which is used to provide an unbiased evaluation of a final model fit. Unlike the training and validation set that are used during the training process, the testing

set is used only to assess the performance of a trained model. There are different ways to split the dataset into these three different categories but one of the most common is k-fold cross-validation, which is used throughout this thesis.

Cross-validation is used in training neural networks to estimate the skill of a model on unseen data. A k-fold cross-validation is a type of cross-validation in which we iterate training then validation k times over a dataset, using different data subsets on each iteration, then average the final results. In each round, we split the dataset into k parts, one part is used for validation, and the remaining $k - 1$ parts are merged into a training subset for the training of the model.

5.6 Overfitting

Overfitting has occurred when a model performs well only on training data and fails to generalize to other unseen data. This is a very common problem in applied machine learning due to the large number of tunable parameters, so many methods have been introduced to avoid this problem, such as

- **Simplifying the model.** The complexity can be reduced by removing layers or reducing the number of neurons. Sometimes, especially for a small training set, when a network has many hidden layers, it is possible that the network learns unnecessary details (e.g., noise) about the training set, making it incapable of generalizing these learned information to predict the output of unseen data.
- **Regularization.** This technique adds a penalty term to the loss function to encourage fewer non-zero parameter values and avoid irrelevant added structure. The most common techniques are known as $L1$ and $L2$ regularization. $L1$ penalty aims at minimizing

the weights' absolute value as,

$$\mathcal{L}'(\theta) = \mathcal{L}(\theta) + \lambda \sum_{i=1}^M \|\theta_i\|_1. \quad (5.10)$$

where λ is a hyperparameter. The $L2$ penalty aims to minimize the squared magnitude of the weights,

$$\mathcal{L}'(\theta) = \mathcal{L}(\theta) + \lambda \sum_{i=1}^M \|\theta_i\|_2^2, \quad (5.11)$$

The goal of regularization is to minimize variance while compromising on bias a little bit so that the overall model depends on fewer non-trivial parameters [159].

- **Dropouts.** During training, neurons are randomly dropped from the network's layers. Dropping neurons is equivalent to training different neural networks. Therefore the different networks will overfit in different ways, so the net effect of dropout will be to reduce overfitting.

5.7 Long Short Term Memory Recurrent Neural Networks

The previous sections focused on feed-forward types of neural networks with simple inputs and outputs, such as real values or discrete categories. In this section a new model will be introduced that is more suitable for sequential data. Sequential data includes text streams, audio clips, video clips, time-series data, etc. When we want to build a model for sequential data, the model explicitly accounts for the sequential nature of input data and its long-term dependencies. With feed-forward networks, the information only flows in one direction and never touches a node (neuron) twice, so nodes have no memory of any prior input. These models have no notion of temporal order since they consider all input simultaneously, so are not suitable for modeling sequential data.

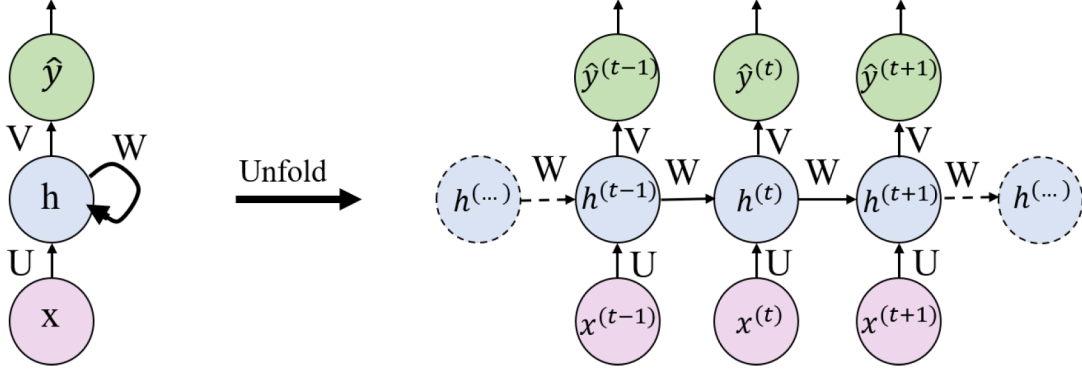


Figure 5.4: The left side of the diagram shows a RNN and on the right side it has been unfolded. $x^{(t)}$ is the network's input at time step t . $h^{(t)}$ represents a hidden state at time t and its output is calculated based on the current input and previous time step's hidden state, and finally $\hat{y}^{(t)}$ illustrates the output of the network at time t .

Recurrent Neural Networks (RNN) are a popular architecture that preserves the ordered structure of sequenced data. In an RNN, the information cycles through a recurring loop. Fig. (5.4) illustrates the information flow in a recurrent neural network. $x^{(t)}$ is taken as the input to the network at time step t . When one step of the RNN forward pass occurs, the network generates what is known as 'hidden state'. We denote the hidden state at step t as $h^{(t)}$, and it is calculated based on the current input and the previous time step's hidden state,

$$h^{(t)} = g(U_{xh} x^{(t)} + W_{hh} h^{(t-1)} + b_h). \quad (5.12)$$

Recall that g is the activation function, and b_h is the bias value assigned to hidden state at time step t .

There are three sets of weights as denoted in Fig. (5.4): The input to hidden connections parametrized by weight U_{xh} , hidden to hidden recurrent connections parametrized by a weight W_{hh} . The output of the network in time step t is

$$\hat{y}^{(t)} = V_{hy} h^{(t)} + b_y, \quad (5.13)$$

where hidden to output connections are parametrized by a weight matrix V_{hy} . All these weights are shared across time.

The hidden states here serve as the memory of the RNN, but they do not actually remember very much. In models with many hidden states, an RNN will suffer from the vanishing/exploding gradient problem [160]. By the time we reach a later hidden state, the RNN has forgotten most of the history from the first hidden states. To fix this problem, a variation of an RNN is developed to include additional explicit memory, known as a Long Short Term Memory (LSTM) network. An RNN already has a short-term memory, but by adding an extra input for persistent information, an LSTM also enables a long-term memory.

LSTM are a type of RNN architecture that addresses the vanishing/exploding gradient problem and allows learning of long-term dependencies. LSTMs are a variant of recurrent neural networks with persistent memory that were originally developed in the context of natural language processing to learn longer temporal correlations appearing in time series data. The main idea is to add a memory cell (or cell state) to maintain information over longer times with gating units inside that regulate the information flow into and out of the memory. Fig. (5.5) illustrates the inside of an LSTM memory cell in more detail.

Each memory cell has different gates that act as follows [161]:

- **Forget gates**, controls that what information to throw away from memory (cell state), $c^{(t-1)}$. By looking at the values of $h^{(t-1)}$ and $x^{(t)}$ the gate outputs a number between 0 and 1 for each cell state $c^{(t-1)}$. The Sigmoid specifically has been chosen here since it outputs a value between 0 and 1, and it can either let no flow or complete flow of information through the gates,

$$f^{(t)} = \text{Sigmoid}(W_{xf} x^{(t)} + W_{hf} h^{(t-1)} + b_f), \quad (5.14)$$

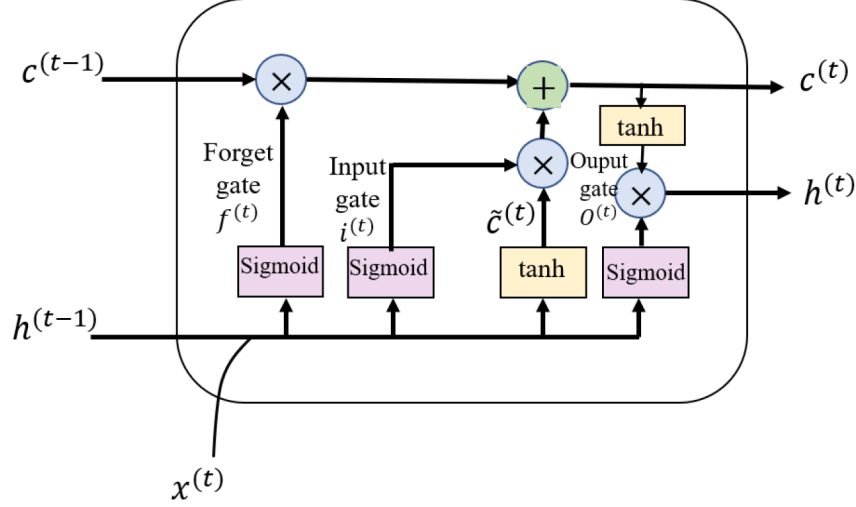


Figure 5.5: Long Short Term Memory (LSTM) cell. Fundamental components of an LSTM cell are a forget gate, input gate, output gate and a cell state. **Forget gates**, controls that what information to throw away from memory (cell state), $c^{(t-1)}$. **Input gate** controls what new information is added to cell state from current input. **Cell state** aggregates the two components, old memory via the forget gate and new memory via the input gate. **Output gate** conditionally decides what to update from the memory.

where W_{xf} is the weight assigned to the connection of input $x^{(t)}$ to the forget layer, W_{hf} is the assigned to the connection of the hidden node in the previous time step $h^{(t-1)}$ to the forget layer, and b_f is the bias value assigned to the forget layer.

- **Input gate** controls what new information is added to cell state from current input. This step has two parts: First the input gate decides what values should be updated and then a tanh layer creates a new value ($\tilde{c}^{(t)}$) for updating the memory. As we saw in the standard LSTM network, Sigmoid is used as the gating function and the tanh is used as the output activation function.

By keeping the values between -1 and 1 , a tanh function regulates the output of the network. Also, because its derivative can sustain for a longer range before going to zero, it can help us to overcome the vanishing gradient problem. Even though the activation functions used in standard LSTM cell here has this specific design, there

is no restriction on using different types of activation functions. Most of the time, to produce an accurate model that matches our data, it is best to design different LSTM cells and train the network several times. The calculation to update the input value and the memory cell is as follows

$$i^{(t)} = \text{Sigmoid}(W_{xi}x^{(t)} + W_{hi}h^{(t-1)} + b_i) \quad (5.15)$$

$$\tilde{c}^{(t)} = \tanh(W_{xc}x^{(t)} + W_{hc}h^{(t-1)} + b_c). \quad (5.16)$$

- **Memory update**, the cell state aggregates the two components, old memory via the forget gate and new memory via the input gate. In this step the old state is multiplied by the output of forget gate $f^{(t)}$, to forget the part of information before carrying it into the memory cell. Then by adding $i^{(t)} \tilde{c}^{(t)}$ to it we decide how much we want to update the memory cell. This step's calculation is as follows

$$c^{(t)} = f^{(t)} c^{(t-1)} + i^{(t)} \tilde{c}^{(t)}. \quad (5.17)$$

- **Output gate** conditionally decides what to update from the memory. This part has two steps as follows

$$o^{(t)} = \text{Sigmoid}(W_{xo}x^{(t)} + W_{ho}h^{(t-1)} + b_o) \quad (5.18)$$

$$h^{(t)} = o^{(t)} \tanh(c^{(t)}). \quad (5.19)$$

5.8 Autoencoders

The LSTM RNN architecture in the previous section is an example of supervised learning, in which both inputs and their correct outputs are provided as labeled training data. However,

suppose you just have a large amount of data and you want to know if there are specific patterns in this unlabeled training data. There is a simple architecture to compress important characteristics of the input data into a simplified internal representation, called an autoencoder.

An autoencoder [162] is an unsupervised learning type of feed-forward neural network which encodes its input x_i into a compressed hidden representation h , then decodes the input again from this hidden representation. The model is trained to minimize a cost function that ensures that the output $\hat{x}_i$ is close to the original input x_i . To ensure compression the hidden representation should be a bottleneck layer with dimension much smaller than the original input.

Autoencoders have been used widely for feature selection ⁵, mapping high-dimensional data to two dimensions for visualization, and for other tasks involving information compression, such as reducing the sizes of files. Autoencoders are also becoming increasingly important for quantum machine learning [163, 164] because of the limited size of near-term quantum processors; extra input-dependent compression will be important to down-size the inputs to tractable dimensions.

There are different types of autoencoders such as a denoising autoencoder which is a supervised learning model, a sparse autoencoder, a convolutional autoencoder, and a variational autoencoder. Denoising autoencoders are the most relevant for Chapter 7 where we will show how to apply them to the task of quantum state tomography. A denoising autoencoder is an autoencoder that not only downsizes the input data into the hidden layer (the bottleneck) but also removes unstructured noise from the original input data to recover a clean version as output. Fig. (5.6) illustrates the architecture of a denoising autoencoder.

5. The process of selecting relevant features for use in model construction is called feature selection.

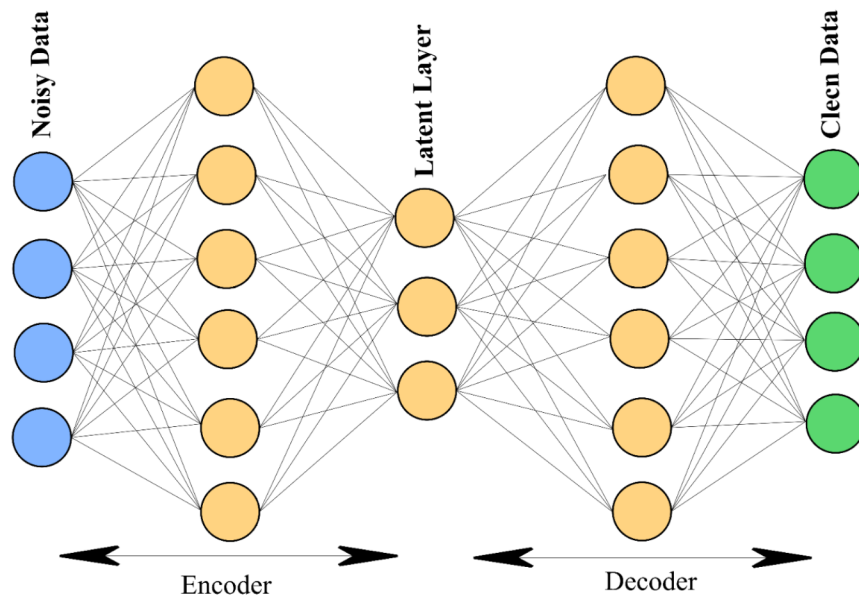


Figure 5.6: Architecture of a denoising autoencoder. Denoising autoencoders are an important tool for identifying key structures in the data by compressing it to a minimal representation using a bottleneck structure in the middle layer.

6 Monitoring Fast Superconducting Qubit Dynamics Using a Neural Network

6.1 Motivation

In Section 3.4 we saw that dispersively measuring a superconducting qubit (transmon) produces noisy homodyne signals that are weakly correlated with the qubit state. Since these signals contain qubit information, they can be used to monitor or conditionally control the qubit dynamics. These tasks require us to reconstruct a best estimate of the quantum state as it undergoes coherent and stochastic dynamics due to the knowledge obtained by the detector. This measurement backaction is proportional to the amount of information learned during the continuous measurement. This stochastic evolution is known as a quantum trajectory.

Traditional methods to recover a quantum trajectory from a noisy measurement record, such as the stochastic master equations [110, 112] and Bayesian filters [39, 118] discussed in Chapter 4, require precise calibration of system parameters, as well as a measurement record that approximates stationary Gaussian white noise [165] and a weak system-environment coupling. The last assumption of record Markovianity breaks down when the quantum state changes quickly compared with the detector bandwidth. This fast dynamics can prevent the conditioned coherent steady states in the resonator from adiabatically following their associated qubit states, producing nontrivially entangled qubit-resonator states and a measurement record with longer temporal correlations [118]. Therefore, these traditional methods will not be sufficient to accurately recover rapidly evolving quantum trajectories in modern quantum processors with limited bandwidth. Finding a method that enables us to continuously monitor these fast dynamics may enable novel experimental techniques with concurrent gate monitoring for error diagnostics or measurement-based feedback for continuous quantum

error correction [166, 167].

We already know that RNNs can successfully reconstruct quantum trajectories when the trajectory changes slowly compared with the detector bandwidth [43]. However, it is not immediately clear whether a similar RNN would be able to produce accurate physical trajectories in the more challenging regime with rapid qubit dynamics.

In this chapter we show that an LSTM recurrent neural network is in fact able to accurately reconstruct physical trajectories of a strongly driven superconducting qubit coupled to a narrow linewidth resonator. The LSTM trains entirely on experimental observations, so auto-calibrates parameters without the need for additional prior information, and compensates for the narrow-bandwidth detector thus outperforming conventional reconstruction methods in the strongly driven regime. We observe strong corrections to the measurement backaction caused by the narrow bandwidth detector averaging the rapid qubit dynamics during the detector response time, which agree well with independently derived physical models that include detailed information about resonator as well as numerically simulated trajectories of significantly higher dimension that include both the qubit and the resonator dynamics explicitly. Finally, we demonstrate the RNN’s ability to uncover hidden time-dependencies during a continuous measurement.

This chapter is adapted from the work presented in [168].

6.2 Circuit QED Model

The qubit-resonator system that is explained in this chapter and is shown in Fig. (6.1a) consists of a superconducting transmon qubit capacitively coupled to a narrow linewidth superconducting resonator, yielding a dispersive coupling interaction. The qubit is driven to induce Rabi oscillations with frequency Ω_R around the x-axis of Bloch sphere. The Hamiltonian that describes this system is the same as the Hamiltonian (3.14) in Chapter 3,

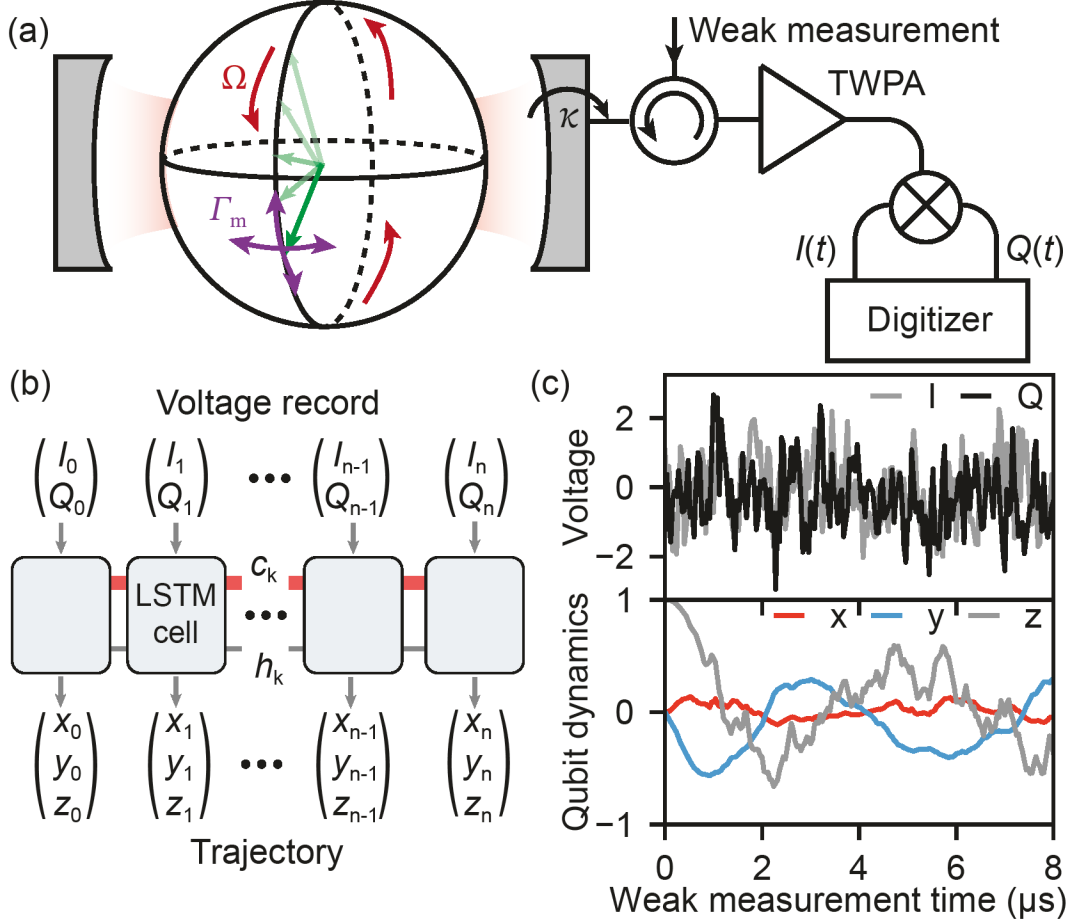


Figure 6.1: Schematic setup of a driven superconducting qubit subject to continuous measurement. (a) A weak measurement tone populates the cavity with linewidth $\kappa/2\pi = 1.56$ MHz and a phase preserving traveling-wave parametric amplifier (TWPA) amplifies the reflected cavity field before both field quadratures $I(t)$ and $Q(t)$ output by the heterodyne mixer are recorded at a digitizer. (b) Recurrent neural network composed of long-short term memory (LSTM) cells learn a representation of the stochastic, dissipative qubit dynamics, mapping elements of a noisy voltage record to qubit trajectories (x, y, z) . The persistent cell state c_k helps the hidden state h_k to track possible long-term correlations in the voltage record, which may become important when the qubit dynamics is fast compared with the rate at which photons escape the cavity ($\kappa^{-1} \approx 0.1 \mu\text{s}$). (c) Sample record with corresponding qubit trajectory output by the trained LSTM.

but with the Rabi oscillation included as well,

$$\frac{\hat{H}_{qr}}{\hbar} = \frac{\omega_q}{2} \hat{\sigma}_z + (\omega_r + \chi \hat{\sigma}_z) \hat{a}^\dagger \hat{a} + \frac{\Omega_R}{2} \hat{\sigma}_x. \quad (6.1)$$

Each run of the experiment begins by heralding¹ the ground state of the qubit. Then the qubit is prepared along one of the six cardinal points of the Bloch sphere by applying a preparation pulse. Since the resonator frequency shifts by 2χ when the qubit switches from the ground state to the excited state, a microwave measurement tone probing the resonator near the resonator frequency in reflection will accumulate a qubit state-dependent phase shift. If the amplitude of the measurement tone is small with respect to the quantum fluctuations of the electromagnetic mode, the phase shift will be obscured, resulting in a weak measurement of the qubit state that provides less information than a projective measurement but also imparts smaller measurement backaction on the qubit state.

To continuously track the qubit state we inject a weak measurement tone (Fig. 6.1a), which populates the resonator to a steady state. The injected field, after interacting with the qubit, leaves the resonator at a rate $\kappa/2 \approx 0.8$ MHz and then is amplified using a phase-preserving amplifier (which uniformly amplifies both in-phase $I(t)$ and quadrature $Q(t)$ parts of the signal), and collected with a total quantum efficiency of $\eta \sim 0.19$. The collected signal is then digitized with sampling time $\Delta t = 1$ ns. The reason that we choose a phase-preserving amplifier over phase-sensitive design is that its broad amplification bandwidth can simultaneously amplify signals from several qubits on the used 8-qubit chip, though only one qubit was needed for this particular experiment.

At the end of each long weak measurement we turn off the measurement tone and the dynamical control, then project the qubit state onto one of the three Pauli bases, chosen randomly, by applying a second strong measurement tone that implements a projective measurement. Therefore by the end of each measurement we record the initial state of the qubit along with the continuous voltage records for both quadratures as well as the randomly chosen basis and its projective measurement result. The final projective measurement is done

1. This technique of heralded state preparation uses a measurement to prepare the qubit in its ground state with high accuracy. This technique effectively eliminates state preparation errors due to thermal heating of the excited state population observed in superconducting qubits [169].

to provide reference information that is used both for training labeled data and for verification of the reconstructed trajectories after training. After the LSTM is trained, the coarse-grained digital records $\{(I(t), Q(t))\}$ are passed then into both the LSTM filter and a more traditional Bayesian filter, and the reduced resulting qubit Bloch vector trajectories are compared with each other as well as the ensemble statistics of the final projective measurements.

6.3 Analytic Quantum Trajectory Reconstruction

Before analyzing the performance of the LSTM, we first introduce the analytical Bayesian filter that we used to reconstruct comparison trajectories. As discussed in Section 3.4, for each signal realization at the end of homodyne detection in an ensemble, the observed quadrature signals will be proportional to the mean value of the measured qubit observable but modified by additive Gaussian noise $\xi(t)$, with zero ensemble-averaged mean. Because of the uncertainty from the noise, the two qubit states will not be perfectly distinguished by the signal collected over a time T ; however, one state will generally be more likely given a particular observed signal, which allows partial information to be inferred about the qubit state. Partial information collection produces a generalized measurement of the qubit state. A collected temporal sequence of generalized measurements produces a conditioned state trajectory for the qubit.

The measurement record then is translated to quantum state evolution by first applying the unitary evolution and then updating the state with the measurement record at each time step [170]. In other words, knowing the initial state of the qubit, and the Hamiltonian driving unitary evolution, the density matrix $\hat{\rho}_{ij}(t_k)$, where $t_k = k \Delta t$, can be repeatedly updated as we saw in Chapter 4. Following the procedure outlined in [118], the density matrix is updated by taking into account the measurement record $(I(t_i), Q(t_i))$ and the

partially updated density matrix $\hat{\rho}(t_i)$,

$$\rho_{11}(t_{i+1}) = \frac{\rho_{11}(t_i)/\rho_{00}(t_i) e^{A_i}}{1 + \rho_{11}(t_i)/\rho_{00}(t_i) e^{A_i}}, \quad (6.2)$$

$$\rho_{00}(t_{i+1}) = 1 - \rho_{11}(t_{i+1}), \quad (6.3)$$

$$\rho_{10}(t_{i+1}) = \rho_{10}(t_i) \sqrt{\frac{\rho_{11}(t_{i+1})\rho_{00}(t_{i+1})}{\rho_{11}(t_i)\rho_{00}(t_i)}} e^{-iB_i} e^{-2(1-\eta)\Gamma_m\Delta t}, \quad (6.4)$$

$$\rho_{01}(t_{i+1}) = \rho_{10}^*(t_{i+1}), \quad (6.5)$$

where $A_i = \tilde{I}_i\Delta I/\sigma^2$, $B_i = \tilde{Q}_i\Delta I/2\sigma^2$, σ^2 is the variance of the noise in the measurement record and

$$\tilde{I}_i = I(t_i) - \frac{(I_0 + I_1)}{2}, \quad (6.6)$$

$$\tilde{Q}_i = Q(t_i) - Q_0, \quad (6.7)$$

$$\Delta I = I_1 - I_0. \quad (6.8)$$

I_0 and I_1 are the steady state coherent state amplitudes conditioned on the qubit in definite ground state and excited state, respectively.

Γ_m in Eq. (6.4) is [118]

$$\Gamma_m \equiv \frac{\kappa}{2} |\alpha_+(t) - \alpha_-(t)|^2, \quad (6.9)$$

which is the ensemble measurement-dephasing rate and is related to distinguishability of the two qubit-state-conditioned output fields, $\alpha_{\pm}$, that propagate into the transmission line, which in turn distinguishes what the qubit state must have been to produce the observed field².

Note that these state update equations are exact for uncorrelated Gaussian noise in the absence of a Rabi drive, and if the qubit decay rates are small compared with κ [118]. To

2. Γ_m in the stochastic master equation [90] is also the maximum rate of the measurement backaction.

reconstruct the qubit trajectories for fast Rabi drive, the resonator memory effects should be considered in the derivations.

The performance of the Bayesian filter can improve when adding analytical corrections to the measurement backaction, and including calibrated parameters such as χ and κ . For adding analytical corrections we need to go back to the qubit-resonator system that was explained in Chapter 3. Given an input drive $\hat{d}(t)e^{-i\omega_d t}$ tuned to the midpoint between qubit resonances, photons in the resonator mode $\hat{a}$ encode this qubit information in their relative phase, which follows from the Heisenberg evolution of the resonator mode in the rotating frame of the drive,

$$\dot{\hat{a}} = -(\kappa/2)[1 + i(2\chi/\kappa) \hat{z}(t)] \hat{a}(t) + \sqrt{\kappa} \hat{d}(t). \quad (6.10)$$

When the qubit $\hat{z}$ is stationary and the drive $\langle \hat{d} \rangle = -i\varepsilon/\sqrt{\kappa}$ fluctuates around a constant mean, the resulting steady state $\hat{a}^{\text{s.s.}} = -i\sqrt{\bar{n}} \exp(-i\hat{\phi})$ has a Lorentzian mean photon number $\bar{n} = |2\varepsilon/\kappa|^2/(1 + (2\chi/\kappa)^2)$ and qubit-dependent phase $\hat{\phi} = \arctan[(2\chi/\kappa) \hat{z}]$, with a maximum phase contrast of $\Delta\phi_{\text{max}} = 2\arctan(2\chi/\kappa)$. A homodyne measurement aligned with the quadrature of maximum separation thus has a maximum amplitude contrast $\Delta\bar{a}_{\text{max}} = 2\sqrt{\bar{n}} \sin(\Delta\phi_{\text{max}}/2) = \sqrt{\bar{n}} (4\chi/\kappa)/\sqrt{1 + (2\chi/\kappa)^2}$ that sets the rate $\gamma_m = (\eta\kappa/2)|\Delta\bar{a}_{\text{max}}|^2 = \eta(8\chi^2\bar{n}/\kappa)/(1 + (2\chi/\kappa)^2)$ at which maximally separated steady-states can be distinguished by the photon amplitudes escaping the resonator at rate $\kappa/2$ and being successfully collected with efficiency η . The uncollected photons and residual qubit-resonator entanglement further contribute to a total qubit ensemble-dephasing rate due to measurement rate $\Gamma_m \approx \gamma_m/\eta$.

In the presence of a qubit drive, the resonator response additionally filters the evolution of $\hat{z}(t)$ to produce an effectively adiabatic response to a *time-ordered geometric series* of its

delay-averages. That is, with a similarly constant drive $\langle \hat{d} \rangle = -i\varepsilon/\sqrt{\kappa}$ and $t \gg 2/\kappa$ to let transients decay, the resonator evolution has the recurrence relation

$$\hat{a}(t) = -i\frac{2\varepsilon}{\kappa} - i\frac{2\chi}{\kappa} \int_0^t \hat{z}(t-\tau) \hat{a}(t-\tau) \frac{\kappa e^{-\kappa\tau/2} d\tau}{2}, \quad (6.11)$$

which has the formal solution with time-ordering operator $\mathcal{T}$,

$$\hat{a}(t) = -i\frac{2\varepsilon}{\kappa} \mathcal{T} \sum_{n=0}^{\infty} \left[-i\frac{2\chi}{\kappa} \int_0^t \hat{z}(t-\tau) dP(\tau) \right]^n. \quad (6.12)$$

The convolution kernel in the Green's function is an exponential probability distribution $dP(\tau) = \kappa e^{-\kappa\tau/2} d\tau/2$ over delay-times τ , normalized as $\int_0^{\infty} dP(\tau) = 1$ with mean and variance both equal to the time constant $\bar{\tau} = 2/\kappa$.

When $\hat{z}$ varies slowly on the timescale $\bar{\tau}$ it can be approximately pulled outside the integral of Eq. (6.12) to yield the standard steady-state solution but with a time-dependent phase $\hat{\phi}(t) = \arctan[(2\chi/\kappa) \hat{z}(t)]$ that adiabatically tracks the qubit evolution. The next-order approximation treats the evolution as approximately linear within the exponential envelope $\hat{z}(t-\tau) \approx \hat{z}(t) - \hat{z}'(t)\tau$, which additionally *delays* the response to the qubit by the mean delay time $\bar{\tau} = 2/\kappa$ to produce the effective phase $\hat{\phi}(t) = \arctan[(2\chi/\kappa) \hat{z}(t-\bar{\tau})]$. For more rapid evolution, part of the evolution is averaged, thus reducing the measurement contrast while rotating the measurement basis.

In the case of a constant Rabi drive, the delay-average in Eq. (6.12) can be computed directly. Assuming dominant harmonic evolution $\dot{\hat{z}} = \Omega_R \hat{y}$, $\dot{\hat{y}} = -\Omega_R \hat{z}$, repeated integration-by-parts of the delay-averaged $\hat{z}(t)$ when $t \gg \bar{\tau}$ yields a pair of geometric series defining an effective

$\hat{z}_{\text{eff}}(t)$ characterized by an adiabaticity parameter $(2\Omega_R/\kappa)$,

$$\hat{z}_{\text{eff}}(t) = \int_0^t \hat{z}(t - \tau) dP(\tau) \quad (6.13)$$

$$= \frac{1}{1 + (2\Omega_R/\kappa)^2} \hat{z}(t) - \frac{(2\Omega_R/\kappa)}{1 + (2\Omega_R/\kappa)^2} \hat{y}(t) \quad (6.14)$$

$$= \sqrt{\eta_{\text{avg}}} [\cos \theta_{\text{tilt}} \hat{z}(t) - \sin \theta_{\text{tilt}} \hat{y}(t)]. \quad (6.15)$$

The averaging both attenuates the eigenvalue contrast of $\hat{z}$ by an efficiency factor η_{avg} and rotates the observable coupled to the resonator by an angle θ_{tilt} in the y - z oscillation plane. The tilt angle and efficiency are thus

$$\theta_{\text{tilt}} = \arctan(\Omega_R \bar{\tau}) = \arctan(2\Omega_R/\kappa), \quad \sqrt{\eta_{\text{avg}}} = \cos \theta_{\text{tilt}} = \frac{1}{\sqrt{1 + (2\Omega_R/\kappa)^2}}. \quad (6.16)$$

At longer times $t \gg \bar{\tau}$ the geometric series in Eq. (6.12) then yields the standard steady state, but with a phase angle that depends upon the effective delay-averaged observable $\hat{z}_{\text{eff}}(t)$ that is rotated by θ_{tilt} and with eigenvalues reduced by $\hat{z}_{\text{eff}}^2 = \eta_{\text{avg}}$. This tilt can be understood equivalently as two simultaneous measurements along z and y with differing measurement rates $\gamma_z = \eta_{\text{avg}} \cos^2 \theta_{\text{tilt}} \gamma_m$ and $\gamma_y = \eta_{\text{avg}} \sin^2 \theta_{\text{tilt}} \gamma_m$, that compete to rotate the effective measurement poles. The reduction in the measurement rate in the yz plane which is captured by η_{avg} is

$$\Gamma_m(\Omega_R) = \eta_{\text{avg}} \Gamma_m(0) = \frac{\Gamma_m(0)}{1 + (\Omega_R \bar{\tau})^2}. \quad (6.17)$$

6.4 Training of LSTM RNN

An LSTM RNN is trained to reconstruct trajectories by feeding it coarse-grained voltage records for the two measured signal quadratures (I_n, Q_n) of variable length $0 < T_m <$

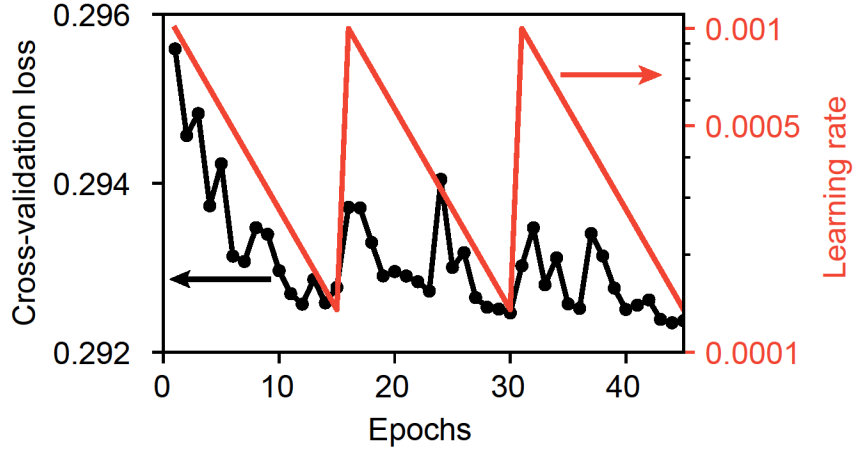


Figure 6.2: Cost function (black) of cross-validation dataset during training. Temporary increases in the cost function are expected as the learning rate (red) increases periodically.

$8\mu s$, and computing the cross-entropy loss with respect to the associated final projective measurement result labels at $t = T_m$.

We train the LSTM using different Rabi frequencies Ω_R ranging from slow dynamics $2\Omega_R/\kappa \ll 1$ to fast dynamics that are outside of the adiabatic regime $2\Omega_R/\kappa \gg 1$. For a single Rabi frequency we have $N \approx 5 \times 10^3$ repetitions of 40 different measurement records for each of three tomography axes, yielding a total of 0.6×10^6 voltage records. Out of this total dataset, we used 10% for cross validation (90% for training).

The cost function that receives mini-batches of size $N_b = 512$ in each step has three components,

1. A cross-entropy loss, Eq. (5.4), due to tomography at the end of a voltage record,

$$\mathcal{L}_{CE} = -\frac{1}{N_b} \sum_{i=1}^{N_b} t_i \log(p_i) + (1 - t_i) \log(1 - p_i), \quad (6.18)$$

where N_b is the mini-batch size, t_i is the binary tomography result (0 or 1) and $0 \leq$

$p_i \leq 1$ is the probability predicted by the final state output by the neural network for voltage record i (in training, bits average to state probabilities).

2. A mean-squared-error for deviating from the known initial state at time $t = 0$, $\mathcal{L}_{init}$.
3. A physical constraint on the purity of the quantum state $\mathcal{L}_{Bloch}$, which tries to enforce that the Bloch vector does not lie outside the Bloch sphere for all t , $\mathcal{L}_{purity}$.

The total cost function is a weighted sum of each component, but the main contribution comes from $\mathcal{L}_{CE}$. The optimization method we use here is the ADAM optimizer [157] and to prevent getting stuck in local minima we use stochastic gradient descent with a cyclical learning rate η_{LSTM} , see Fig. (6.2). One of the reason to use the cyclical learning rate is that it eliminates the need to perform numerous expensive experiments (training) to find the best values and schedule with no additional computation [171].

6.5 Qubit Trajectory Reconstruction with a Neural Network vs. a Bayesian Update

First the LSTM RNN is used to reconstruct qubit dynamics with a weak Rabi drive ($2\Omega_R/\kappa = 0.2$), where conventional methods can still accurately reconstruct trajectories. A histogram of reconstructed trajectories (Fig. (6.3)) shows oscillations due to the Rabi drive as well as diffusion due to measurement back-action, which purifies the trajectories towards $|\pm z\rangle$. The competition between the Rabi drive, trajectory purification at rate $\eta\Gamma_m$, and trajectory dephasing at rate $2(1-\eta)\Gamma_m$ confines trajectories to a Bloch sphere with reduced radius set by the measurement efficiency η , which is shown by the white dashed line in Fig. (6.3a). When comparing individual trajectories, the RNN produces trajectories similar to the conventional Bayesian filter approach.

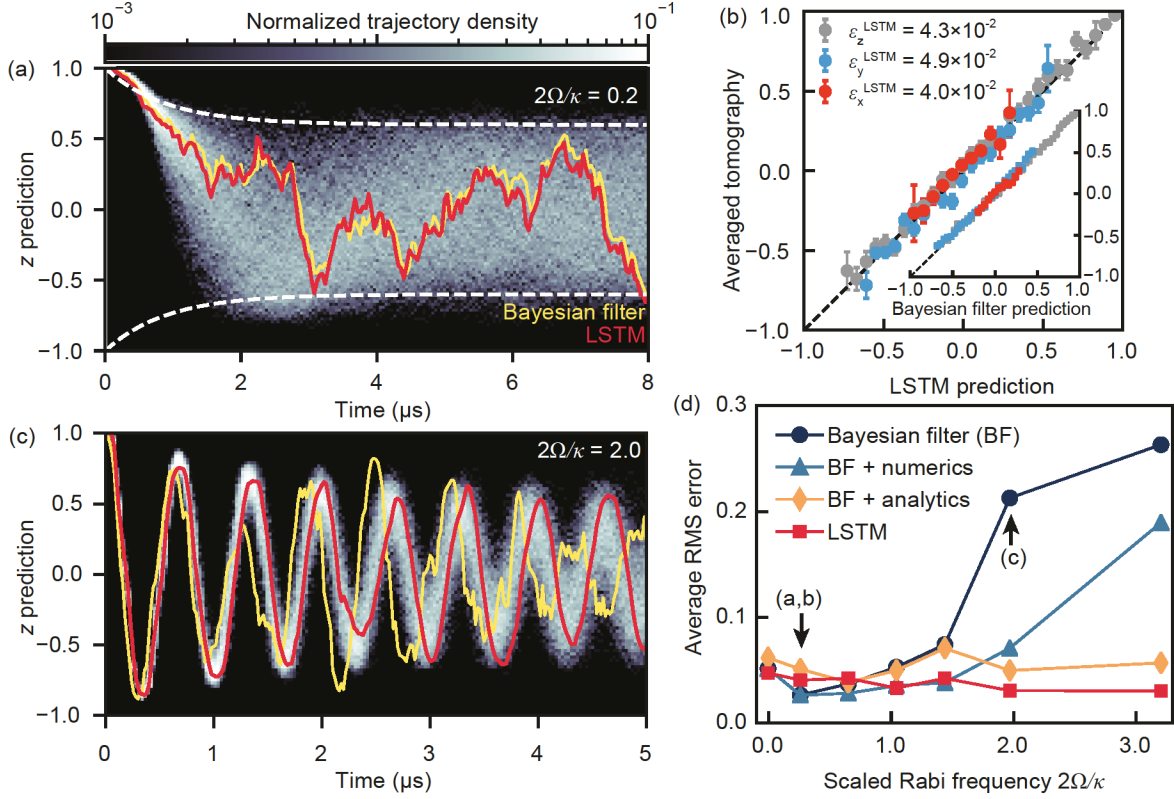


Figure 6.3: Breakdown of adiabatic trajectory reconstruction (a) The histogram of weakly driven trajectories of length $T_m = 8.0 \mu\text{s}$ (reconstructed by the LSTM with $dt = 40 \text{ ns}$) shows rapid trajectory diffusion due to measurement backaction. The colorbar represents the trajectory probability density at each timestep. The LSTM produces trajectories (example in red) comparable to those from a steady-state Bayesian filter (yellow). (b) LSTM validation based on tomographic measurements immediately following the LSTM prediction, for the trajectories shown in (a) where $2\Omega_R/\kappa = 0.2$. The dashed line with slope 1 indicates perfect validation. The inset shows the Bayesian filter validation for the same data set with slightly smaller RMS error $(\varepsilon_x, \varepsilon_y, \varepsilon_z) = (2.3, 3.2, 2.5) \times 10^{-2}$ due to larger number of available trajectories. (c) For fast qubit dynamics ($2\Omega_R/\kappa = 2.0$) outside the adiabatic regime the predictions of the steady-state Bayesian filter and LSTM diverge (trajectory $dt = 20 \text{ ns}$). (d) Validation errors averaged over the three qubit coordinates vs. Rabi frequency $2\Omega_R/\kappa$, showing a breakdown of the steady-state Bayesian filter for $2\Omega_R/\kappa > 1$ (dots), while the LSTM validation error (squares) remains small. As discussed in Section 6.3, the performance of the Bayesian filter improves with additional prior information of the expected evolution of the z -conditioned resonator fields, and improves even further when adding analytical corrections to the measurement backaction. Importantly, the LSTM performance stays consistent without additional prior information. Arrows mark the data shown in (a), (b) and (c).

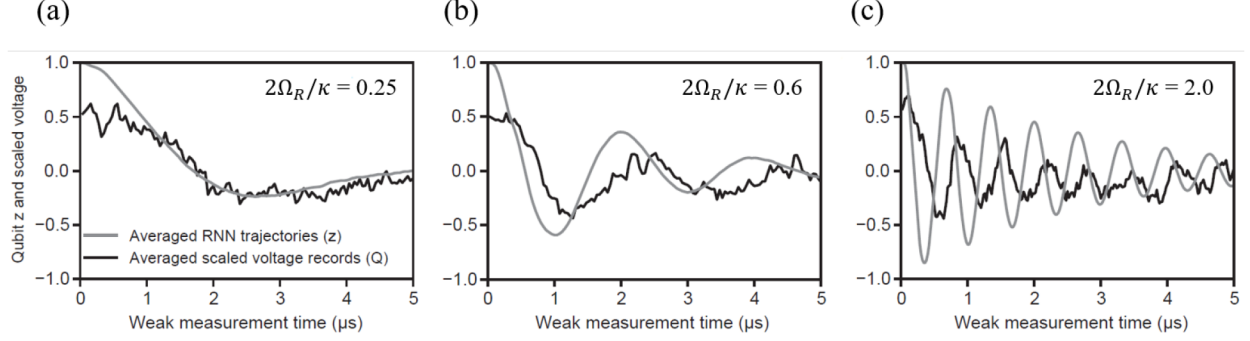


Figure 6.4: Phase Difference in the Voltage Records. Comparison between averaged voltage records and averaged z trajectories reconstructed by the LSTM for increasing Rabi frequencies (from left to right). The signal phase is delayed from the qubit oscillation phase by $2/\kappa$. For slow Rabi drives that phase delay is small compared to the oscillation period, but for fast Rabi drives it becomes significant. Nevertheless, the RNN correctly accounts for this delay to reconstruct the qubit state evolution correctly.

The error is quantified for both methods by averaging projective measurement results of trajectories with similar predictions [172, 173]. The averaged tomography results closely follow the RNN predictions for all three Bloch coordinates, Fig. (6.3b). This error largely reflects our imperfect knowledge of the the true quantum state from a finite number of projective measurements. Because the trajectories of both methods are similar, and the RNN prediction correlates strongly with the averaged projective measurements, the neurons of the RNN accurately encode the reduced qubit dynamics.

For large Rabi frequencies the RNN trajectories remain faithful, even when the qubit dynamics exceeds the detector bandwidth, Fig. (6.3 c) and (6.3 d). In contrast, the Bayesian filter's validation error increases sharply past $2\Omega_R/\kappa \approx 1$.

Experimentally the breakdown of the Bayesian filter coincides with a large phase difference between oscillations in the measured signal records and the qubit coordinate $z(t)$, see Fig. (6.4). For fast qubit dynamics, we observe a large phase difference between the oscillations in the measurement record and those in $z(t)$, because the resonator memory delays photons escaping to the transmission line while the qubit rotates quickly. Stochastic master equation or

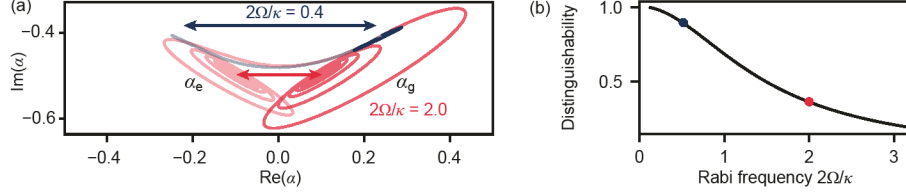


Figure 6.5: (a) QuTiP master equation simulation of a driven qubit dispersively coupled to the readout cavity, showing coherent state amplitudes vs. time, conditioned on the qubit state. We show two pairs of α_g and α_e for $2\Omega_R/\kappa = 0.4$ and $2\Omega_R/\kappa = 2.0$ obtained from the expectation value $\alpha_- \approx \text{Tr}(\hat{\rho}_{\text{qr}} a |0\rangle\langle 0|/P_0)$, where a is the cavity photon annihilation operator and ρ_{qr} is the joint qubit-resonator density matrix, and P_0 is the probability of finding the qubit in the state $|0\rangle$. (b) We use the relative distinguishability, defined as $\alpha_-(\Omega_R) - \alpha_+(\Omega_R)/\alpha_-(0) - \alpha_+(0)$, to scale ΔI and partially correct for cavity effects in the Bayesian filter method.

Bayesian filters that do not include the resonator memory assume photons measure the qubit state $z(t)$ instantaneously and therefore, this phase difference signals the breakdown of those methods. While it is possible to construct more complicated non-Markovian SMEs [174] or use a Bayesian filter that includes the resonator memory [118, 175], the LSTM RNN offers an accurate, more flexible reconstruction method requiring no prior knowledge of coupling rates to the environment and memory kernels which are hard to calibrate experimentally.

As a first order correction of the resonator memory effects before applying an analytical treatment, we can improve the performance of the Bayesian filter reconstruction numerically by adjusting ΔI based on a simulation which includes qubit and coupling to the resonator. From these simulations Fig. (6.5 a), we see that the steady-state amplitudes conditioned on the qubit states $\alpha_{\pm}$ decrease as the Rabi frequency increases. Therefore, to correct for the resonator effect, the relative distinguishability is defined as $(\alpha_+(\Omega_R) - \alpha_-(\Omega_R))/(\alpha_+(0) - \alpha_-(0))$ (Fig. (6.5 b)), to scale ΔI , after applying this correction the average error decreases (see Fig. (6.3 c, BF + numerics)) compared to average error of the conventional Bayesian filter in Fig. (6.3 c, Bayesian filter (BF)) but still the average error remains large for $2\Omega_R/\kappa > 1$.

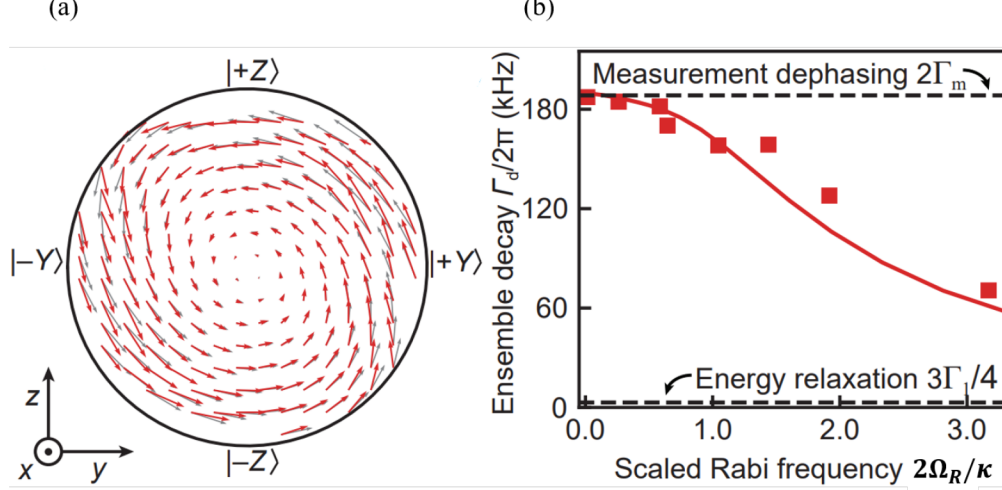


Figure 6.6: Unraveling Lindbladian trajectory dynamics from LSTM trajectories (a) The average drift (dy, dz) of trajectories binned in the yz -plane ($2\Omega_R/\kappa = 0.6$, red arrows) reveals dynamics consistent with the applied Hamiltonian $\hat{H} = 1/2\Omega_R\hat{\sigma}_x$ and decay towards $y = 0$ at a rate Γ_d (gray arrows). (b) The decay rate $\Gamma_d/2\pi$, extracted from fits to Lindbladian maps such as (a), (red squares) falls from the expected measurement dephasing rate $2\Gamma_m$ towards the bare qubit relaxation rate [176], as the qubit and resonator decouple. A master-equation simulation of the qubit-resonator system (solid line) agrees well with the LSTM trajectories.

Only the analytic treatment discussed in Section 6.3, which adds extra prior information about calibrated parameters, such as χ and κ , to the Bayesian filter achieves error similar to the LSTM RNN for all Rabi frequencies, see Fig. (6.3 d, BF + analytics). Note that while the Bayesian filter needs enough prior information about the system the LSTM RNN offers an accurate reconstruction method requiring no prior knowledge.

6.6 Resonator Memory Corrections To Qubit Trajectories

Extracting physical information encoded in neural networks is generally non-trivial. To gain further insight into the LSTM trajectories in the regime inaccessible to the standard Bayesian filter, we decompose the LSTM trajectories into deterministic and stochastic parts, to expose

Lindbladian dynamics and measurement backaction, respectively. This decomposition is motivated by the stochastic master equation [177],

$$d\mathbf{r} = \underbrace{\mathcal{L}(\Omega_R, \Gamma_d) \mathbf{r} dt}_{\text{Lindbladian dynamics}} + \underbrace{\sqrt{2\eta\Gamma_m} dW_1 \begin{pmatrix} -xz \\ -yz \\ 1 - z^2 \end{pmatrix}}_{\text{Informational backaction}} + \underbrace{\sqrt{2\eta\Gamma_m} dW_2 \begin{pmatrix} -y \\ x \\ 0 \end{pmatrix}}_{\text{Phase backaction}} \quad (6.19)$$

where $\hat{\mathbf{r}} = (x, y, z)$ is the Bloch vector, η is the total efficiency of the measurement chain, $\mathcal{L}(\Omega_R, \Gamma_d)$ is a 3×3 matrix describing Lindbladian dynamics,

$$\mathcal{L}(\Omega_R, \Gamma_d) = \begin{pmatrix} -\Gamma_d & 0 & 0 \\ 0 & -\Gamma_d & -\Omega_R \\ 0 & \Omega_R & 0 \end{pmatrix} \quad (6.20)$$

and $dW_{1,2}$ are zero-mean uncorrelated Wiener noise terms. While this decomposition still assumes Markovianity, small deviations from Eq. (6.19) can help us understand cavity effects on the trajectories.

From the Lindbladian decomposition we observe dynamics consistent with the Hamiltonian $H = \Omega_R \hat{\sigma}_x$ and dephasing pulling trajectories towards the center of the Bloch sphere at a rate $\Gamma_d/2\pi$, see Fig. (6.6 a). For large Rabi frequencies the decay rate Γ_d decreases³ substantially, it is shown in Fig. (6.6 b), because the strong Rabi drive decouples the qubit and cavity which decreases the cavity photon-induced dephasing, a phenomenon described as quantum rifling [179].

As the LSTM produces a sequence of qubit states, we choose a subensemble of states such

3. The Rabi decay rate is caused by combination of both measurement dephasing [112] and relaxation [178]. When the Rabi frequency is small, measurement dephasing dominates Γ_d , while for fast Rabi drive Γ_d approaches the qubit relaxation rate $3\Gamma_1/4$, see Fig. (6.6 b).

that the state is at a particular point in the Bloch sphere. Then for each trajectory in that ensemble we look one time-step ahead and compute the difference vector. Then we average and take the covariance of that ensemble of differences. We extract the experimental measurement backaction from the eigenvectors of the covariance matrix computed on the LSTM trajectories.

At low Rabi frequencies, we find both informational backaction, vanishing at the poles $|\pm z\rangle$ of the Bloch sphere (see fig. (6.6 a)), and phase backaction, consistent with our heterodyne measurement.

For larger Rabi drives, we observe corrections to the backaction, a tilt of the measurement poles towards $|\pm y\rangle$ in Fig. (6.7 a,c) and a reduced diffusion rate $2\eta\Gamma_m$ Fig. (6.7c). Intuitively, this tilt occurs because the Rabi drive drags the qubit state counter clockwise while photons in the cavity measure the qubit z coordinate for a characteristic time κ^{-1} . This tilt was reproduced analytically in Section (6.3). The second correction, a reduction of the measurement rate is a manifestation of the same quantum rifling effect and shows that a fast Rabi frequency protects the qubit from the measurement, effectively decoupling the qubit and cavity.

Based on results of Fig. (6.6) and (6.7), we find good agreement between experimentally reconstructed trajectories and theory over a wide range of Ω_R/κ .

6.7 Time-Dependent Rabi Drive

Our experience with the LSTM has proven that LSTM requires less explicit prior information about calibrated parameters in a specific model. In other words the LSTM autocalibrates the model and all parameters internally so that parameter information is not needed in addition. This feature becomes important as sometimes these parameters can fluctuate over time and make calibration challenging. To highlight this advantage, we perform a new set

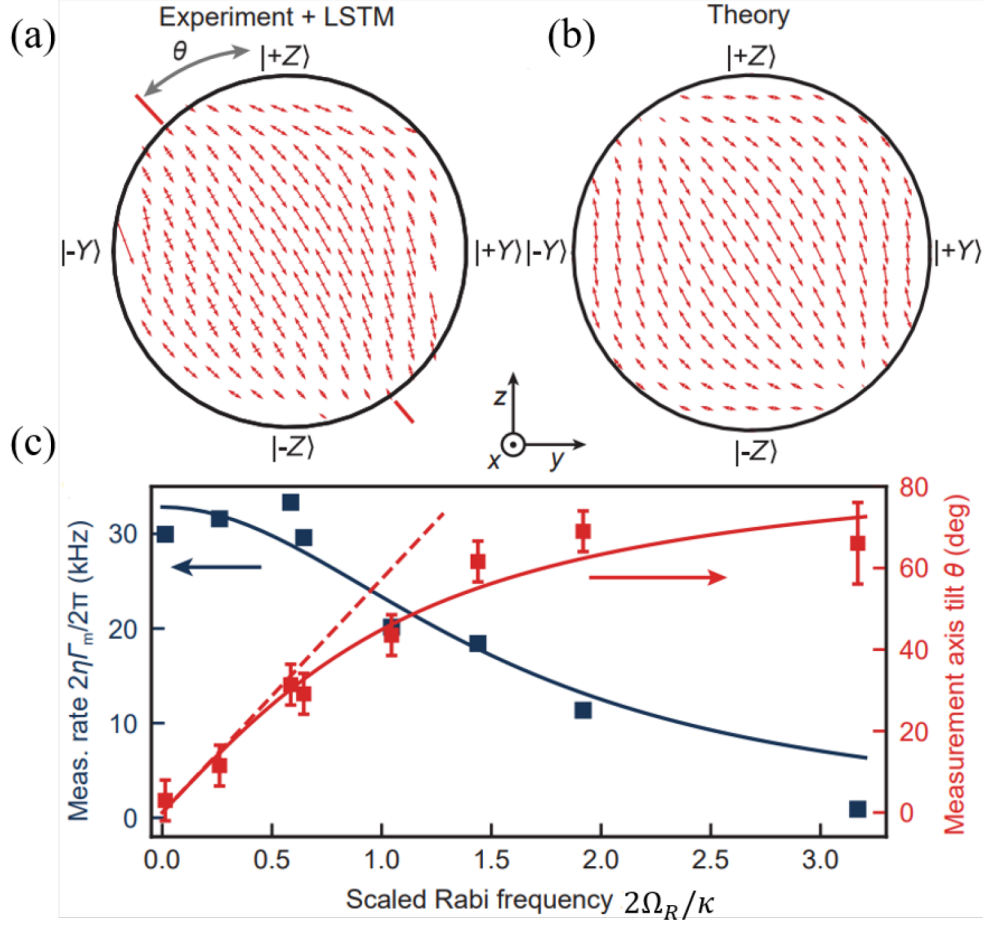


Figure 6.7: Resonator memory corrections to the measurement backaction from LSTM trajectories (a) Trajectory diffusion map obtained from the eigenvectors $\mathbf{v}(y, z)$ of the covariance matrix for $2\Omega_R/\kappa = 0.6$. The tilt towards $|\pm y\rangle$ in plane of the Rabi drive is in contrast to the prediction from Eq. (6.19). (b) The theory prediction which includes resonator memory effects (Eq. (6.15)) reproduces the tilt. (c) For small Ω_R/κ the measurement axis tilts linearly with the Rabi frequency $\theta = 2\Omega_R/\kappa$ (dashed line). The experimental tilt of the measurement eigenstates in the yz plane (red squares) is accompanied by a decrease in measurement rate (blue squares), extracted from the magnitude of $\mathbf{v}(y, z)$. Solid lines are fits to $\theta = \arctan(\Omega_R\bar{\tau})$ (red) and Eq. (6.17) (blue), respectively. The error bars for θ are estimated from imprecision in determining the tilt angle.

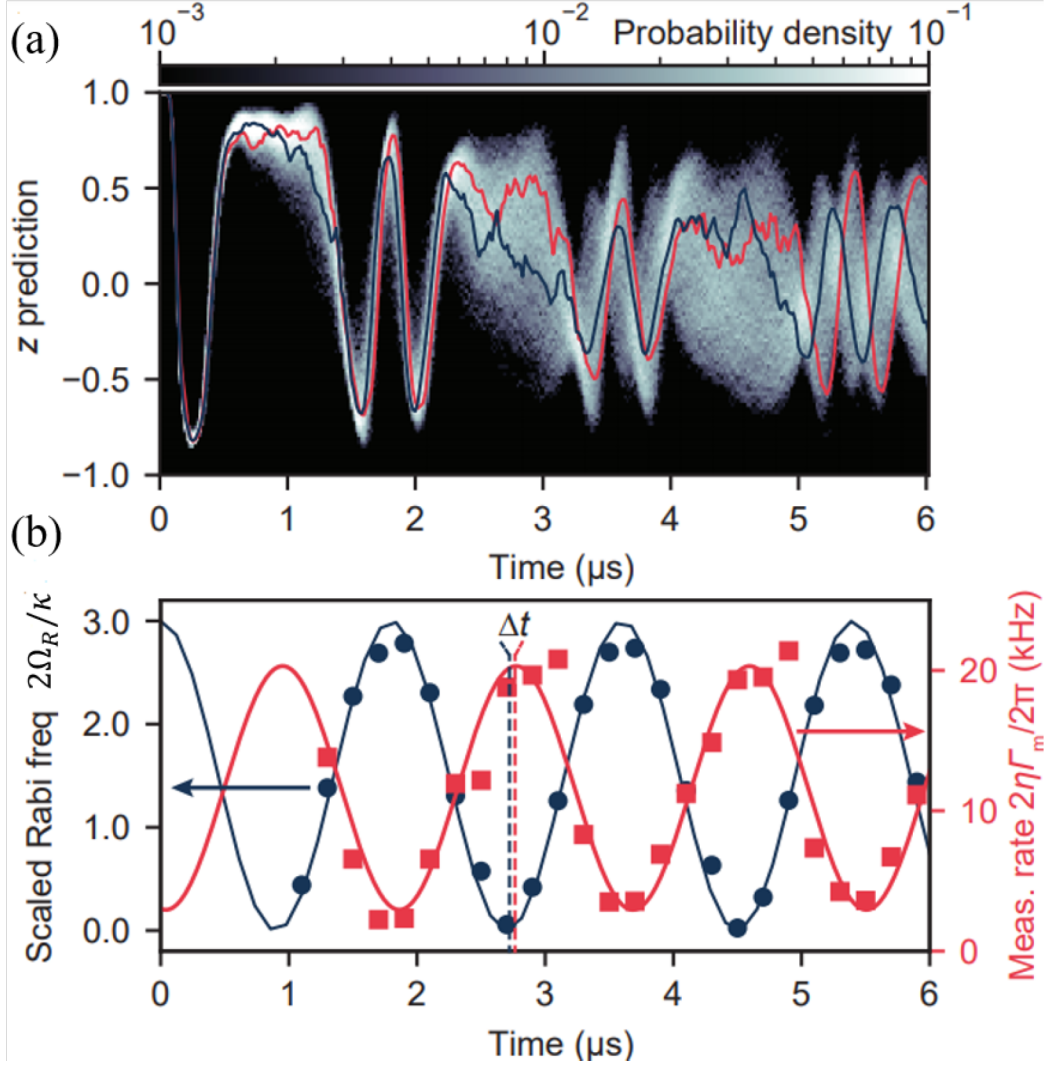


Figure 6.8: Uncovering hidden time-dependencies from trajectories subject to sinusoidal Rabi drive modulation. (a) A histogram of the reconstructed z coordinate shows periods of increased and decreased diffusion, also visible in two sample trajectories (red and blue). (b) Time-windowed trajectories analysis for the data in (a), showing the instantaneous Rabi frequency and instantaneous measurement rate

of weak measurements where we vary Rabi frequency sinusoidally in time. In this case the underlying control parameters are no longer constant in time. Reconstructing trajectories with time-dependent control parameters is important for example when monitoring to diagnose unexpected behavior that occurs during gates, where the drive strength typically has a Gaussian envelope in time or weak measurement of a highly non-markovian environment

[180].

The histogram of LSTM trajectories in Fig. (6.8a) shows alternating periods of trajectory bunching, where the trajectories are protected from measurement back-action, and diffusion, where the weak measurement clearly imparts stochastic kicks to the qubit. We quantify this non-trivial dynamics by fitting to the deterministic and stochastic parts of trajectories in $0.2 \mu\text{s}$ wide time windows, Fig. (6.8b). Without any prior knowledge about the time dependency of the parameters, we correctly recover the sinusoidal applied Rabi frequency, Fig. (6.8b, dark blue) from fits to the Lindbladian, and find that it anti-correlates strongly with the extracted measurement rate in Fig (6.8b, red), which is consistent with our previous results. A sinusoidal fit to the instantaneous measurement rate shows a delay Δt (see Fig. (6.8) b), which is consistent with the relaxation time of the cavity $2/\kappa$. These results demonstrates that LSTM correctly reconstruct trajectories with hidden time-dependencies.

Our results open up a new area of weak measurement where the qubit dynamics is fast compared with the cavity linewidth. Therefore, it can be used to reconstruct individual trajectories during gates, which is useful to diagnose coherent gate errors with high temporal resolution. In addition, since our reconstruction method requires no prior knowledge of a system and its parameters, it can be used to discover new physics in regimes where a theoretical description of the measurement backaction is unavailable. Finally, the LSTM approach shows promise for detection and classification of stochastic errors in continuous error correction experiments.

7 State Tomography with a Denoising Autoencoder

7.1 Motivation

An experimenter must always verify that components of a quantum experiment are characterized correctly for comparisons of experimental results with theoretical predictions. Loosely, the experimental components can be organized into three logical stages of the experiment: state preparation, transformation, and measurement. To make sure that the theory correctly describes the experiment, it is important to check that the state actually prepared after the first stage matches expectations. Unfortunately, experimentalists are not able to directly check a quantum state. Instead, they must repeat the same preparation many times and measure it in different ways. From all these characterization measurements, they can reconstruct the state (or channel, or measurement in the other two experimental stages). This important characterization procedure is called quantum state tomography (QST) [181–186]. QST can be characterized as an optimization problem, where the goal is to identify the quantum state $\hat{\rho}$ which is most likely to result in the observed data. Different methods have been proposed to solve this optimization problem, which we will review shortly. This chapter explores how modern machine learning methods can offer another approach that may have advantages as quantum systems get increasingly complex.

Specifically, this chapter will investigate the use of supervised machine learning, in the form of a modified denoising autoencoder, to simultaneously remove experimental noise while encoding one and two-qubit quantum state estimates into a minimum number of nodes within the latent layer of a neural network. I automate the decoding of these latent representations into positive density matrices and compare them to similar estimates obtained via linear inversion and maximum likelihood estimation. Using a superconducting multiqubit chip

I experimentally verify that the neural network estimates the quantum state with greater fidelity than either traditional method. Furthermore, the network can be trained using only product states and still achieve high fidelity for entangled states. This simplification of the training overhead permits the network to aid experimental calibration, such as the diagnosis of multi-qubit crosstalk [187]. As quantum processors increase in size and complexity, I expect automated methods such as those presented in this chapter to become increasingly attractive.

7.2 Traditional State Tomography Strategies

QST consists of two steps: performing measurements on an ensemble of identically prepared quantum states in different bases, and then reconstructing the quantum state’s density matrix from the collected measurement results. The goal is to be able to predict the statistics of future measurements made after the same preparation procedure described by the reconstructed quantum state. This reconstruction can be accomplished using a variety of methods such as linear inversion [188], Bayesian inference [189, 190], maximum likelihood estimation [182, 191–194], and maximum entropy estimation [181, 195, 196]. In order to compare the performance of the denoising autoencoder with traditional methods, I first provide a brief explanation of linear inversion and maximum likelihood estimation, which are the two most common methods used for QST in practice.

7.2.1 Preliminary Definitions

As discussed at Chapter 2, the quantum state of any two-level system can be represented as a 2×2 density matrix,

$$\hat{\rho} = \frac{1}{2}(\hat{\mathbb{I}} + x \hat{\sigma}_x + y \hat{\sigma}_y + z \hat{\sigma}_z) = \frac{1}{2}(\hat{\mathbb{I}} + \hat{\mathbf{r}} \cdot \hat{\boldsymbol{\sigma}}), \quad (7.1)$$

where, $\hat{\mathbf{r}} = (x, y, z) \in \mathbb{R}^3$ and $\hat{\boldsymbol{\sigma}} = (\hat{\sigma}_x, \hat{\sigma}_y, \hat{\sigma}_z)$. Since Eq. (7.1) is constrained to have positive eigenvalues, therefore the Bloch vector $\hat{\mathbf{r}}$ only represents a physical state if $\|\mathbf{r}\|^2 = x^2 + y^2 + z^2 \leq 1$. Recall from Chapter 2 we saw that a projective measurement along an axis $\mathbf{n}$ is represented by the observable $\hat{\sigma}_n = \mathbf{n} \cdot \hat{\boldsymbol{\sigma}}$ and its expectation value is $\langle \hat{\sigma}_n \rangle = \text{Tr}(\hat{\sigma}_n \hat{\rho}) = \mathbf{n} \cdot \hat{\mathbf{r}}$. If one identically prepares N_n qubits and measures the observable $\hat{\sigma}_n$ on each of them, $N_{n\uparrow}$ of these qubits will be found in the $|n \uparrow\rangle$ state and $N_{n\downarrow}$ qubits in the $|n \downarrow\rangle$ state. Therefore the observable's expectation value is statistically approximated by

$$\langle \hat{\sigma}_n \rangle = \frac{N_{n\uparrow} - N_{n\downarrow}}{N_n} \quad (7.2)$$

The statistical error of this expectation value¹ is [197],

$$\Delta \langle \hat{\sigma}_n \rangle = \frac{2\sqrt{N_{n\uparrow}N_{n\downarrow}}}{N_n^{3/2}}. \quad (7.3)$$

For the rest of this chapter $\vec{V}_{\text{exp}} = \left((N_{n_1\uparrow} - N_{n_1\downarrow})/N_{n_1}, (N_{n_2\uparrow} - N_{n_2\downarrow})/N_{n_2}, (N_{n_3\uparrow} - N_{n_3\downarrow})/N_{n_3}, \dots \right)$, where $\{n_1, n_2, n_3, \dots\}$ are the measurement bases, will represent an experimentally collected dataset estimating the three independent observables, with the binary projective readout affected by additive bit-flip noise in the form $\begin{cases} p_{0 \rightarrow 1} = p(1 - \epsilon) \\ p_{1 \rightarrow 0} = p(1 + \epsilon) \end{cases}$ where p is a simulated readout error probability, and ϵ is an additive asymmetry to mimic

1. By the width of a binomial distribution

the bias toward relaxation to the ground state seen in the experimental data, Analogously, $\vec{V}_{\text{ideal}} = \left(\text{Tr}(\hat{\sigma}_{n_1}\hat{\rho}), \text{Tr}(\hat{\sigma}_{n_2}\hat{\rho}), \text{Tr}(\hat{\sigma}_{n_3}\hat{\rho}), \dots \right)$ will represent a corresponding true (theoretical, or ideal) set of expectation values that the experimental data is statistically approximating.

7.2.2 Linear inversion (LI)

Starting with Born's rule, the probability to obtain the specific result i is $P(i|\hat{\rho}) = \text{Tr}(\hat{P}_i \hat{\rho}) = p_i$, where $\hat{P}_i$ is a particular measurement outcome projector. Next, a matrix M is defined as

$$M = \begin{pmatrix} \vec{P}_1^\dagger \\ \vec{P}_2^\dagger \\ \vec{P}_3^\dagger \\ \vdots \end{pmatrix} \quad (7.4)$$

Where, $\vec{P}_i$ is a list of individual measurements with binary outcomes. Applying matrix M to $\vec{\rho}$ (the representation of the density matrix $\hat{\rho}$ as a column vector), will give us the probability,

$$M\vec{\rho} = \begin{pmatrix} \vec{P}_1^\dagger \vec{\rho} \\ \vec{P}_2^\dagger \vec{\rho} \\ \vec{P}_3^\dagger \vec{\rho} \\ \vdots \end{pmatrix} = \begin{pmatrix} \text{Tr}(\hat{P}_1 \hat{\rho}) \\ \text{Tr}(\hat{P}_2 \hat{\rho}) \\ \text{Tr}(\hat{P}_3 \hat{\rho}) \\ \vdots \end{pmatrix} \approx \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ \vdots \end{pmatrix} = \vec{p}, \quad (7.5)$$

where $\vec{p}$ is a vector of ideal probabilities. To reconstruct the density matrix from the observed frequencies p_i , one would need to multiply M^T on the left, $M^T M \hat{\rho} = M^T \vec{p}$, which leads to pseudoinverse [198] of tomographically complete set of probabilities (or expectation values), $\vec{\rho}_{\text{LI}} = (M^T M)^{-1} M^T \vec{p}$, then we get the actual density matrix $\hat{\rho}$ from this vector $\vec{\rho}$.

In spite of LI being a simple method, certain constraints in these method create challenges for practical applications. For example the obtained $\vec{\rho}_{\text{LI}}$ is typically not a valid (positive

semi-definite, Hermitian, trace-one) density matrix [199].

The reason that linear inversion method can fail is that the reconstruction uncertainty gives a small uncertainty volume around the reconstructed state. If the state lies near the pure state boundary of the state space then that uncertainty volume can be partially outside the region of physical states. In this case is it possible for the LI construction to accidentally find a state within that uncertainty volume but outside the physical state region. The more mixed a state is, the further from the state space boundary it is, and the less likely it will be for the uncertainty volume to leak outside the physical state space, so all reconstructed states will be valid.

7.2.3 *Maximum Likelihood Estimation (MLE)*

The problem of reconstructing unphysical density (for the states near to the surface of the Bloch sphere) matrices can be solved by maximizing the probabilities to yield observed data, which is known as maximum likelihood estimation. The likelihood L that the prepared state was $\hat{\rho}$, given a particular set of counts $\{N_1, N_2, \dots, N_K\}$ observed from the measurements is,

$$L(\hat{\rho}|\{N_i\}) = P(\{N_i\}|\hat{\rho}) = \prod_{i=1}^K P(N_i|\hat{\rho}), \quad (7.6)$$

where $P(N_i|\hat{\rho})$ is the probability that N_i counts of outcome i occurred in an ensemble of size $\sum_i N_i$. Maximizing the likelihood function in Eq. (7.6) over all possible preparation states $\hat{\rho}$ yields a maximum likelihood estimate (MLE) for the state $\hat{\rho}_{\text{MLE}}$. This optimization is equivalent to minimizing the negative log likelihood function which turns out to be computationally more stable,

$$\hat{\rho}_{\text{MLE}} = \arg \min[-\log L(\rho|\{N_i\})]. \quad (7.7)$$

Making a Gaussian assumption for the statistics of the $P(N_i|\hat{\rho})$ probabilities such that the log pulls down a quadratic difference from the mean from the exponent, this minimization can be seen as a least-squared error optimization which is subjected to $\hat{\rho} \geq 0$, $\text{Tr}[\hat{\rho}] = 1$. The optimization process is done by parameterizing a physical state. physical means that the density matrix is normalized, Hermitian and positive in the following way by defining a triangular matrix $\hat{T}$:

- To account for non-negativity, $\hat{\rho} = \hat{T}^\dagger \hat{T}$ must holds, as $\langle \psi | \hat{T}^\dagger \hat{T} | \psi \rangle = \langle \psi' | \psi' \rangle \geq 0$.
- We can show that this construction is Hermitian, $\hat{\rho}^\dagger = \left(\hat{T}^\dagger \hat{T} \right)^\dagger = \hat{T} \left(\hat{T}^\dagger \right)^\dagger = \hat{T}^\dagger \hat{T} = \hat{\rho}$.
- The normalization is guaranteed by dividing it by its trace, $\hat{\rho} = \hat{T}^\dagger \hat{T} / \text{Tr}[\hat{T}^\dagger \hat{T}]$.

Though MLE guarantees a physical state estimate by parameterization of the elements of triangular matrix $\hat{T}$, it can be computationally expensive because the minimization process has to search over all possible states, and this minimization must be recomputed for each state that needs to be estimated. Moreover, if the initial guess is too far from the true optimal state, then the search process may find an incorrect local optimum instead. There is thus strong motivation to improve this process for higher-dimensional states that compound both these weaknesses of MLE.

7.3 Denoising Autoencoder Model and Methods

A generic motivation for considering neural network models for state tomography is their ability to automatically learn dominant systematic error sources with no prior information about the measured system and its environment. In addition, once these systematic errors are learned, the NN automatically corrects them. An interesting aspect of this technique is

that after training the NN, one can unpack these automatic sources and extract the model's parameters to improve the analytical model of the system. The main idea is to use known quantum states that one can prepare with high fidelity in order to train an algorithm that is able to interpret raw measurement data and output the state estimate of unknown states.

In the case of state tomography, it is instructive to frame the task as a traditional Alice and Bob set up in the following manner. This task can be broken down into two parts, **training**: where Alice prepares a fixed set of known quantum states (i.e. qubits) and sends them to Bob. Bob's task is to perform a complete set of measurements on those quantum states and pass the measurement results (i.e. as a set of expectation values) into an untrained tomography algorithm in order to estimate the states. In the end, Bob trains the tomographic model based on the feedback he receives from Alice, see Fig. (7.1a). Once the training is done, the **testing** phase begins, see Fig. (7.1b), where Alice sends random quantum states to Bob. Bob estimates the states and sends them to Alice, and finally, Alice computes the fidelity of the states to verify the accuracy of Bob's tomographic model.

A denoising autoencoder (DAE) is a type of neural network that is particularly well-suited for this task. During the training process the raw tomographic data is compressed into a minimal state representation using a bottleneck structure (through an encoder). Decompression is then performed in such a way that it outputs a cleaned version of input data (through a decoder), see Fig. (7.2). Such a DAE removes the extraneous measurement noise by comparing the outputs of the network with the ideal data $\vec{V}_{ideal}$ provided by theory, employing the following mean square error as a loss function,

$$\begin{aligned}\mathcal{L}(\vec{V}_{ideal}, \vec{V}_{clean}) &= \|\vec{V}_{ideal} - \vec{V}_{clean}\|^2 \\ &= \|\vec{V}_{ideal} - \text{ReLU}(W'(\text{ReLU}(W\vec{V}_{exp} + \vec{b})) + \vec{b}')\|^2\end{aligned}\tag{7.8}$$

where $W(W')$ and $b(b')$ are the weight matrix and bias vector of the encoder (decoder),

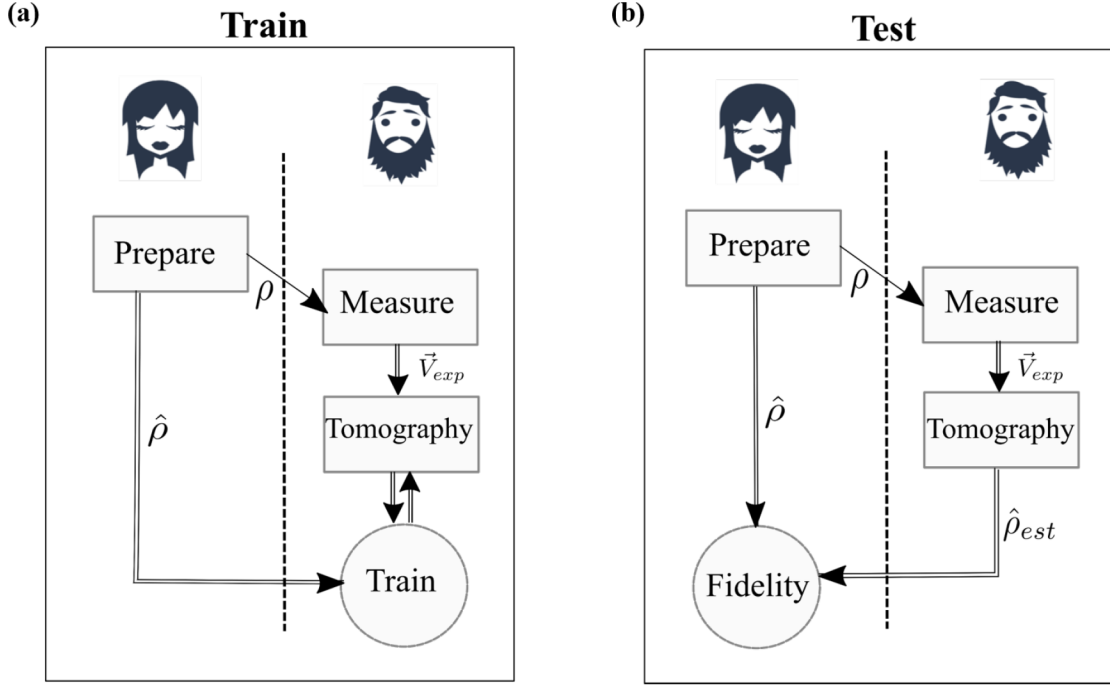


Figure 7.1: State tomography as a learned task framed as a traditional Alice and Bob setup. (a) Alice sends fixed set of known states, Bob performs complete sets of measurements and estimates states, then he uses known state information to train a tomographic model. (b) Alice sends random testing states, Bob estimates these states and sends it to Alice. Alice computes state fidelity to verify the accuracy of Bob's tomographic model.

respectively, and ReLU is the applied activation function. The pros of this approach are that it uses minimal assumptions about the system², learns and removes systematic noise *in situ*, unlike the MLE method, the training process is not as sensitive to being trapped in local minima, and the operation of estimating many different test states becomes very fast after the network is trained once. Thus, though the training process has a high overhead due to the large amount of training data needed initially, that overhead is constant unlike the overhead of MLE that grows linearly with the number of estimated states.

2. The only things we use here from the quantum mechanics are the ideal expectation values of observables in the loss function.

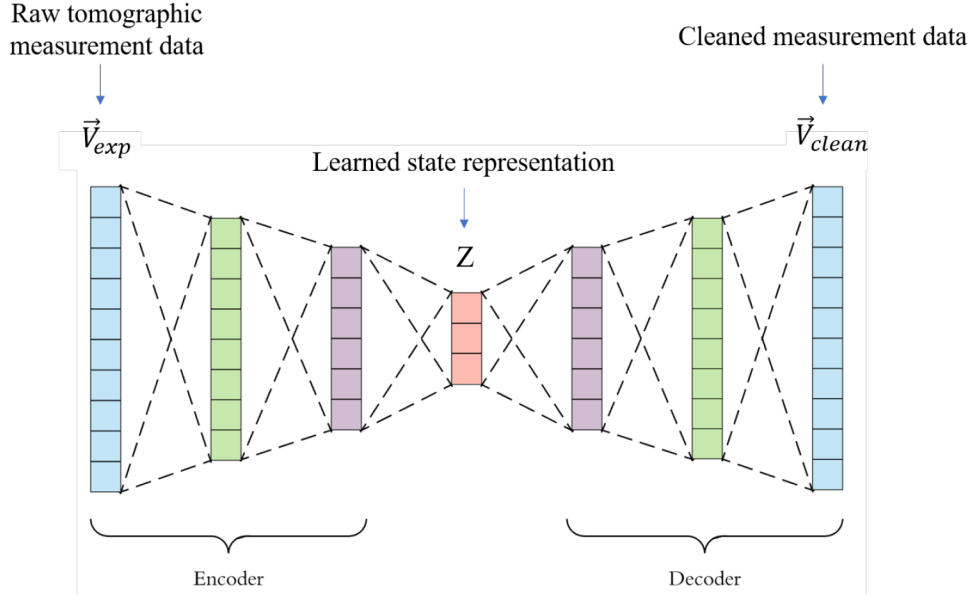


Figure 7.2: The idea is using a bottlenecked neural network to force compression of raw tomographic data into a minimal state representation. We use raw tomographic measurement data as a noisy input of DAE, $\vec{V}_{exp}$, and in order to train it we use ideal measurement data, $\vec{V}_{ideal}$ as a reference.

7.4 Data Preparation for Denoising Autoencoder

In order to prepare the data for the DAE, the first question is which choices of tomographic bases are optimal for accurately reconstructing an unknown qubit state? In other words, while any tomographically complete set in principal should work, does the training process behave differently for different choices of the bases? To answer this question we examined three different measurement bases:

1. The Bloch measurement, which is the most natural choice, see Fig. (7.3 a), whose observables correspond to normalized projectors along the directions: $n_1 = (1, 0, 0)$, $n_2 = (0, 1, 0)$, and $n_3 = (0, 0, 1)$. Each observable has two projectors and the coordinates are for the positive eigenvalue projector only, with the antipodal points implied for the negative projectors.

2. The Bloch measurement with four additional bases added at the diagonals, see Fig. (7.3 b), whose observables correspond to normalized projectors along the directions $n_1 = (1, 0, 0)$, $n_2 = (0, 1, 0)$, $n_3 = (0, 0, 1)$, $n_4 = (1/2, 1/2, 1/\sqrt{2})$, $n_5 = (-1/2, 1/2, 1/\sqrt{2})$, $n_6 = (-1/2, -1/2, 1/\sqrt{2})$, and $n_7 = (1/2, -1/2, 1/\sqrt{2})$.
3. A tetrahedron with a symmetric over-complete set of bases, see Fig(7.3 c), whose observables correspond to normalized projectors along the directions: $n_1 = (0, 0, 1)$, $n_2 = (2\sqrt{2}/3, 0, -1/3)$, $n_3 = (-\sqrt{2}/3, \sqrt{2}/3, -1/3)$, and $n_4 = (-\sqrt{2}/3, -\sqrt{2}/3, -1/3)$. This is similar to symmetric informationally complete positive operator valued measures (SIC-POVM). SIC-POVM is a special case of a generalized measurement on a Hilbert space, that is informationally complete. In other words, it possesses the characteristic that, when acting on a particular state, their statistics completely determine the quantum state. However, because of the difficulty of implementing an actual 4-outcome POVM with high fidelity, what we use here is actually an 8-outcome measurement implemented as a random sample of each of the 4 projective bases containing a point of the SIC-POVM tetrahedron, averaged together. The resulting 8 eigenstate points produce two complementary tetrahedrons on the Bloch sphere, and can be implemented with high fidelity (because all measurements are projective). It retains the main attractive features of the SIC-POVM without its downsides, at the cost of having 8 possible outcomes rather than 4.

In the next step, high fidelity training preparations are necessary in order for the collected training data to properly correspond to the ideal references predicted theoretically by the intended preparation state. We prepare the training set using the measurement eigenstates, which are by definition the highest fidelity states one can prepare in the lab. Because when we perform a projective measurement, as long as the measurement is reasonably good we will get a high preparation fidelity for that particular state. The DAE is tested then with 2000 test states that are distributed on the Bloch sphere with varying the polar θ and azimuthal

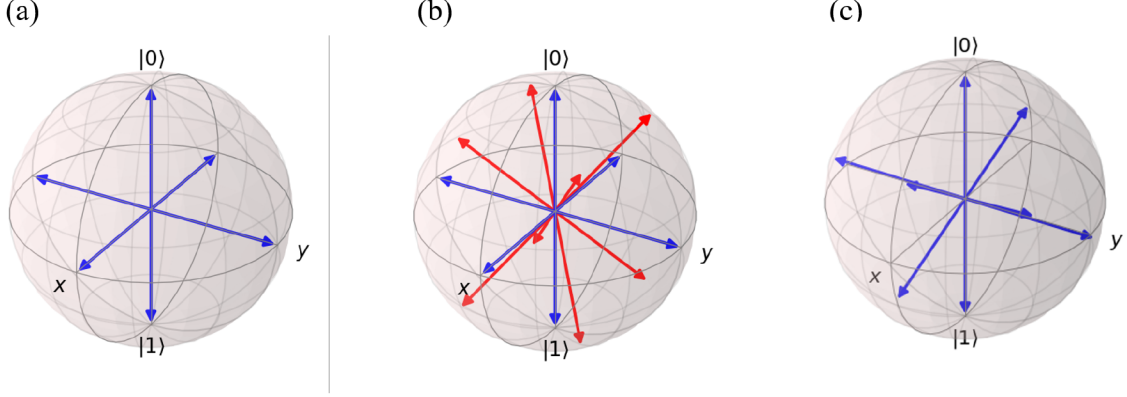


Figure 7.3: The three different measurement bases are: (a)The Bloch measurement (b)The Bloch measurement with four additional bases added at the diagonals (c)A tetrahedron with a symmetric over-complete set of bases.

φ angles of the Bloch sphere.

The training ensemble size to compute the simulated experimental expectation values $\vec{V}_{\text{exp}}$ vary from $N_n^{(\text{train})} = 10^2$ to $N_n^{(\text{train})} = 10^4$, and different DAE networks are trained independently for each training ensemble size. The ensemble sizes used in the lab are usually $N_n \approx 10^5 - 10^6$ which is large compared to the ensemble size that we use in DAE. In the next step we train and test the network with the experimental data obtained from measuring a planar multi-transmon qubit.

In order to determine how close the reconstructed states ($\hat{\rho}_{\text{est}}$) are to the ideal theoretical states ($\hat{\rho}_{\text{ideal}}$), we measure the fidelity using [200],

$$F(\hat{\rho}_{\text{ideal}}, \hat{\rho}_{\text{est}}) = \text{Tr} \left(\sqrt{\sqrt{\hat{\rho}_{\text{ideal}}} \hat{\rho}_{\text{est}} \sqrt{\hat{\rho}_{\text{ideal}}}} \right)^2 \quad (7.9)$$

For the pure states, Eq. (7.9) takes this form, $F(\hat{\rho}_{\text{ideal}}, \hat{\rho}_{\text{est}}) = \text{Tr}(\hat{\rho}_{\text{est}} \hat{\rho}_{\text{ideal}})$ [101].

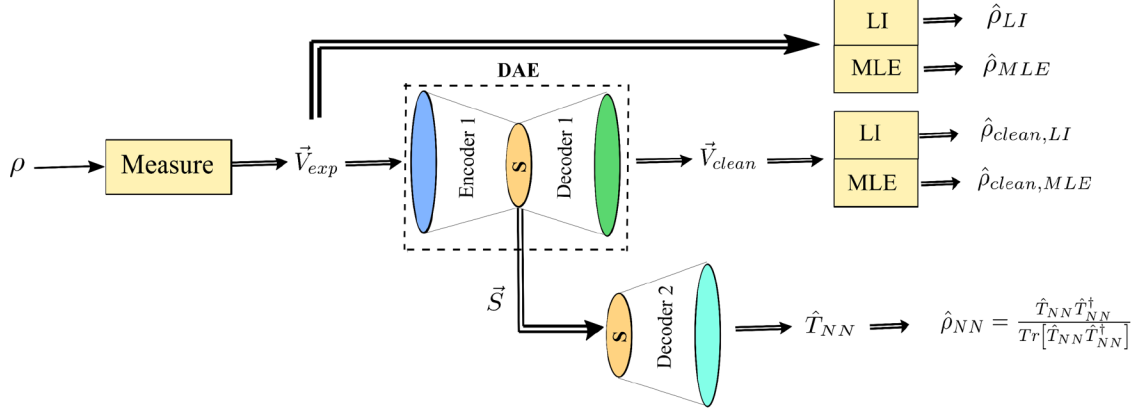


Figure 7.4: Comparing five different strategies: linear inversion of the raw data, maximum likelihood estimation of the raw data, linear inversion of cleaned data, maximum likelihood estimation of the cleaned data, and direct positive state estimate of neural network.

7.5 Measurement Strategies

Although the DAE could in principle speed up the state tomography process for a large number of estimated states, there are other important questions to answer regarding its performance before it could be adopted as a general purpose tomographic tool. We must determine whether it can resolve the issues associated with traditional methods, like ensuring a physical state at the end of estimation process. Similarly, we must determine whether the learned state representation is encoded within the bottleneck in a form that can be readily extracted and interpreted as a quantum state. The basic design of a DAE has an output that is just a cleaned version of the input data, so obtaining the associated quantum state representing the cleaned data is not automatic without additional processing. In order to address these questions we compared five different strategies for extracting a quantum state estimate from a DAE, or a suitably modified design based on a DAE, see Fig. (7.4). The five chosen estimation strategies were:

1. Linear inversion of raw data
2. Maximum likelihood estimation of raw data

3. Linear inversion of cleaned data
4. Maximum likelihood estimation of cleaned data
5. Direct positive state estimate of neural network

Numbers 1 and 2 are the reference cases of applying the standard tomographic procedures to the raw data. Numbers 3 and 4 use the DAE to clean the noise from the raw data before applying the standard tomographic methods, assuming that the noise reduction will improve the estimate fidelity. And number 5 is a modified design for the DAE that uses a second trained decoder to directly extract the compressed internal representation and output the associated quantum state in a useful form.

To explain what is done in the 5th method in more detail, after performing measurements on the eigenstates, the noisy measurement results are passed into the DAE in the form of a set of noisy expectation values. Once the DAE is trained we take the latent layer and redirect it to a second decoder. The second decoder, which is basically a feed-forward neural network, outputs the parameters of the triangular matrices. For training purposes of the second decoder, the reference parameters of the triangular matrix are determined using Cholesky decomposition of the eigenstates [201]. In linear algebra, the Cholesky decomposition is a decomposition of a Hermitian, positive-definite matrix A into the product of a lower triangular matrix T , which is discussed in Section 7.2.3, and its conjugate transpose $T^\dagger$, as $A = T T^\dagger$, the general formula to determine the components of the Cholesky factor T is,

$$t_{kk} = \sqrt{a_{kk} - \sum_{j=1}^{k-1} t_{kj}^2}, \quad (7.10)$$

$$t_{ik} = \frac{1}{t_{kk}} \left(a_{ik} - \sum_{j=1}^{k-1} t_{ij} t_{kj} \right), \quad (7.11)$$

where t_{kk} (a_{kk}) are the diagonal elements of matrix T (A), and t_{ik} (a_{ik}) are the off-diagonal elements of T (A).

For density matrix decomposition, in addition to the conditions that are discussed above, it should be normalized, therefore the final decomposition relation for the density matrix takes the form

$$\hat{\rho} = \frac{\hat{T} \hat{T}^\dagger}{\text{Tr}[\hat{T} \hat{T}^\dagger]}, \quad \hat{T} = \begin{bmatrix} t_1 & 0 \\ t_3 + it_4 & t_2 \end{bmatrix}. \quad (7.12)$$

The elements of Cholesky factor of density matrix T are used (as an array) as training labels for the second decoder, $[t_1, t_2, t_3, t_4]$. This output design guarantees that the estimated state will be a physical state while removing the difficulty of manually interpreting how information is encoded in the learned state representation within the bottleneck. Note that the DAE decoder and the second decoder are independently trained.

7.6 Results

7.6.1 Numerical Simulation of One-Qubit System

Fidelity of Reconstructed States

Having trained three separate DAEs with each of the measurement schemes, Fig. (7.3 a-c), where the training and testing ensemble sizes are $N_n^{\text{train}} = N_n^{\text{test}} = 1000$, we tested each model with 2000 test states that are distributed on the Bloch sphere. Fig. (7.5) illustrates the fidelity heat map for each cases. Linear inversion method is used to reconstruct the test states.

The angles θ and φ are the polar and azimuthal angle of the Bloch sphere, respectively. These results are achieved by training the DAE with only eigenstates. At the beginning

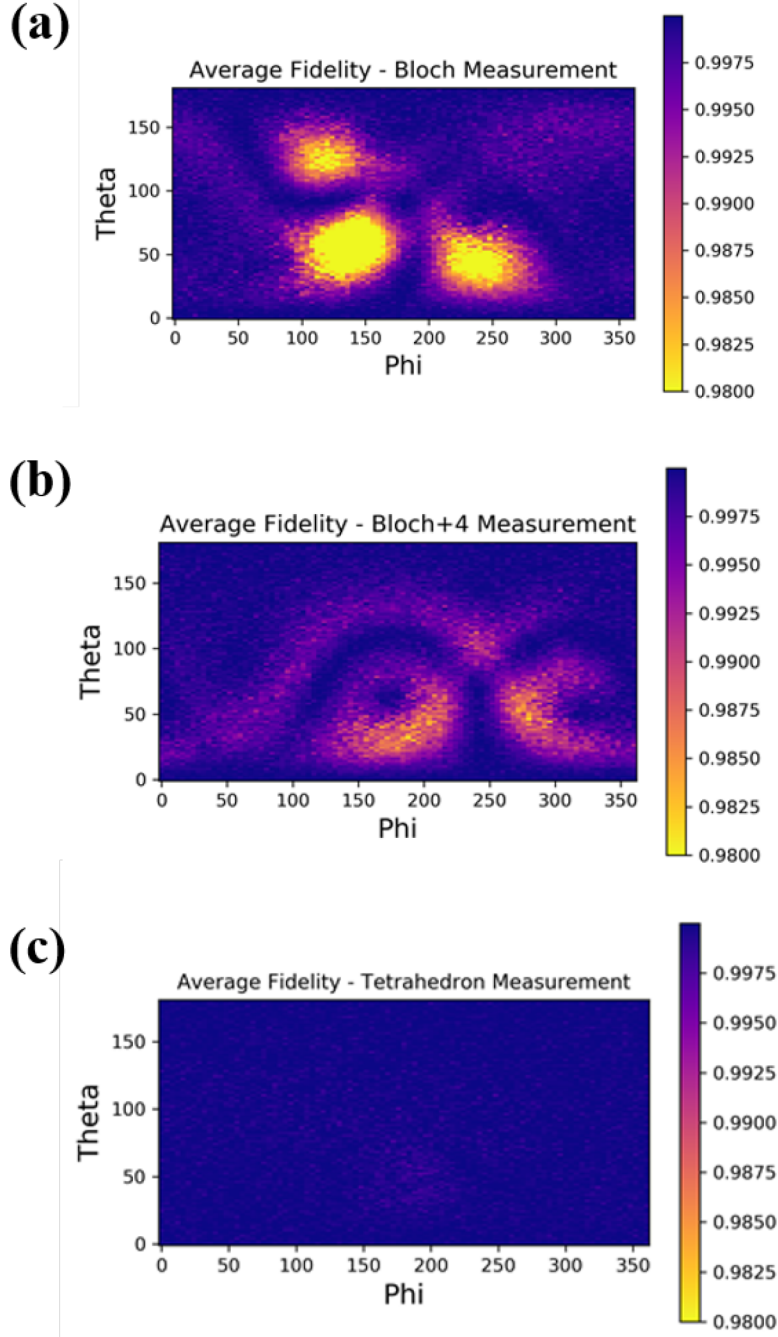


Figure 7.5: The fidelity heatmap for three separate DAEs which are trained with three different measurement schemes: Bloch measurement, supplemented Bloch measurement, symmetric tetrahedron measurement. The training and testing ensemble sizes are $N_n^{\text{train}} = N_n^{\text{test}} = 1000$. The additive noise to simulate the experimental error was 5% simulated readout error, with 2% asymmetry. The model is tested with 2000 states that are distributed on the Bloch sphere. These results are achieved by training the DAE with only measurement eigenstates.

the eigenstates were chosen because by measuring them one of the elements of expectation values becomes 1 and it can be faster for neural network to learn the structure and converge fast, also, in the laboratory the eigenstates are already being prepared by the measurements, and re-using them as training preparation shortens the experimental training procedure. However, achieving an average fidelity over 0.99% by DAE that is trained only with the eigenstates and for such a small training and testing ensemble sizes were not expected.

The additive noise to simulate the experimental error was 5% simulated readout error, with 2% asymmetry ³. Although the Bloch (Fig. (7.5a)) and the supplemented Bloch (Fig. (7.5b)) measurements do reconstruct the states with high average fidelity (the number of realizations for computing the average fidelity is 1000), the heat map still shows hot spots in the middle, indicating that reconstruction fidelity is state-dependent. In contrast, the doubled tetrahedron basis (Fig. (7.5c)) has uniform performance over the entire sphere. In other words the symmetric and overcomplete structure of the tetrahedron basis cancels noise most uniformly for all states.

In the next step, to see how the training ensemble size and the amount of the additive noise affect the fidelity of the test states, we choose the symmetric tetrahedron measurement scheme, which has the highest average fidelity, and train it independently for different training ensemble size N_n^{train} and the readout noise p , see Fig. (7.6) As we can see in Fig. (7.6), increasing training ensemble size boosts learned state fidelity

Latent Layer of Denoising Autoencoder

The DAE with only one hidden layer (which is the latent layer) automatically selects three nodes to reconstruct the high fidelity test states, we checked this by adding more nodes to the latent layer that does not increase the average fidelity, see Table (7.1), for computing

3. For the rest of this chapter these values will be used as the simulated readout error.

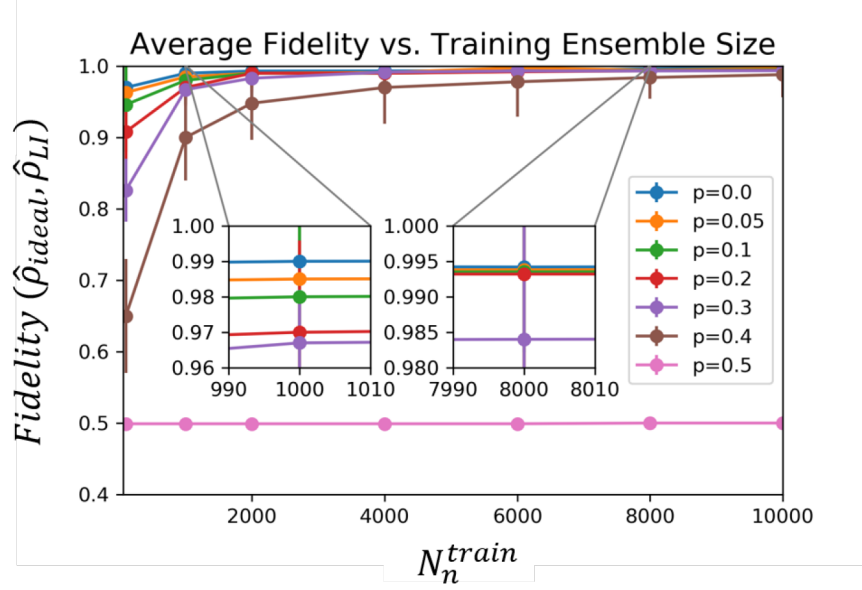


Figure 7.6: The reconstructed fidelity vs. training ensemble size for different value of readout noise p (with constant additive asymmetry $\epsilon = 0.02$) shows that even for small $N_n^{train} = 1000$, which is small compared to the ensemble sizes used in the lab ($N_n^{lab} = 10^5 - 10^6$), the fidelity of the estimated state reconstructed by the clean outputs of the DAE is over 0.98 for small amount of noise (The states here, are reconstructed by applying linear inversion method on the clean output of the DAE).

the average fidelity we choose the same 2000 test states discussed in previous section with tetrahedron measurement scheme. Table (7.1) demonstrates that the DAE is able to figure

Average Fidelity of 2000 test states - Symmetric Tetrahedron						
number of nodes	1 node	2 nodes	3 nodes	4 nodes	5 nodes	6 nodes
Average Fidelity	0.375	0.775	0.987	0.987	0.986	0.987

Table 7.1: The average fidelity vs. number of nodes in latent layer.

out the correct dimensionality of the quantum state space and discards superfluous nodes, even though it has very little prior information and nothing specific about quantum built into its design.

Knowing that denoising autoencoder learns the qubit states with high fidelity, we can peek into neurons of the latent layer to see if the network encoded the learned state within the bottleneck in way that can be easily interpreted by a human. In order to do it I chose the

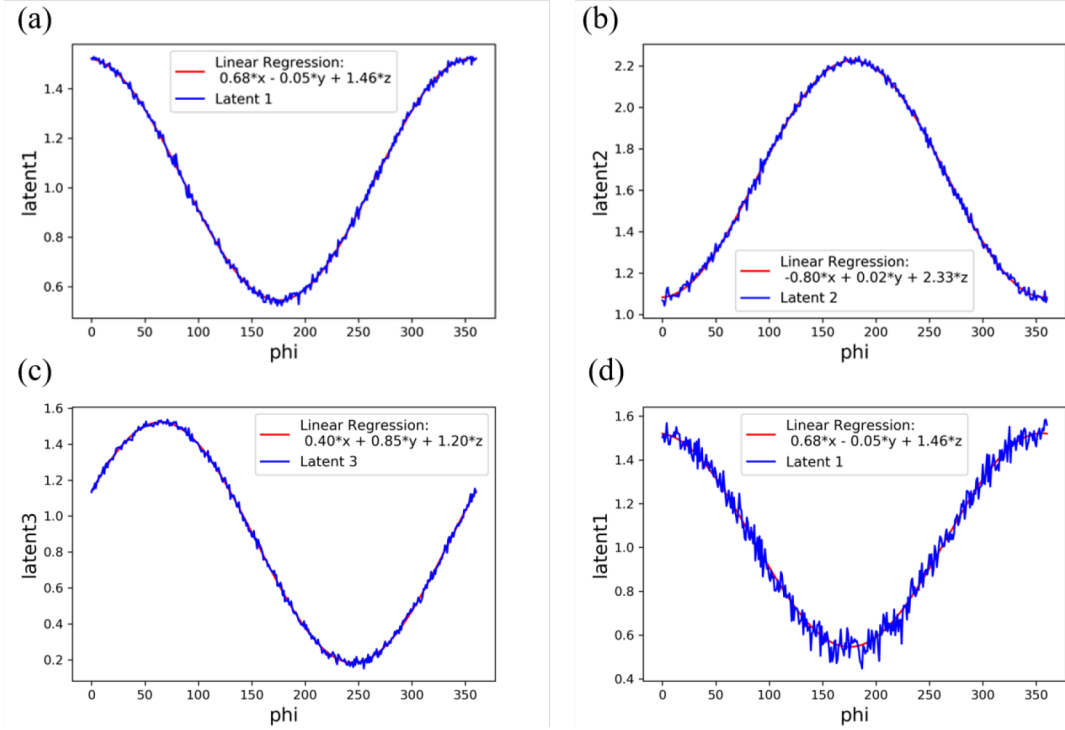


Figure 7.7: The output of three neurons in the latent layer for the states that are picked from one of the parallel azimuthal circles of the Bloch sphere with constant polar angle $\theta = \pi/4$ and varying azimuthal angle $\varphi \in [0, 2\pi)$, and testing ensemble size of $N_n^{\text{test}} = 10^4$. (d) To compare the results of different testing ensemble size we reproduced the first neuron's output using only $N_n^{\text{test}} = 1000$. A linear regression model is fitted to manually extracted outputs of the latent nodes, in red curve. In figure (d), the average deviation of this fit increases by reducing the testing ensemble size from 0.01 to 0.03, but its result is still comparable with the usual projective state tomography.

trained model of the tetrahedron measurement due to having the highest average fidelity compared to the other two measurement schemes (Fig. (7.5)). Fig. (7.7 a-c) shows the output of three neurons in the latent layer for the states that are picked from one of the parallel azimuthal circles of the Bloch sphere with constant polar angle $\theta = \pi/4$ and varying azimuthal angle $\varphi \in [0, 2\pi)$, and testing ensemble size of $N_n^{\text{test}} = 10^4$. To compare the results of different testing ensemble size we reproduced the first neuron's output using only $N_n^{\text{test}} = 1000$ testing ensemble size, in Fig. (7.7d).

Fitting a simple linear regression model to manually extracted outputs of the latent nodes,

in red curve, we can see that the DAE reproduces a scaled and shifted version of the usual orthogonal Bloch vector representation as a minimal encoding. The average deviation of this fit increases by reducing the testing ensemble size from 0.01 to 0.03, but its result is still comparable with the usual projective state tomography. Thus once the representation correspondence is calibrated with such a fit after training the network, the same fit can be used to directly translate the latent node data into human-readable quantum states without the need for linear inversion.

Comparing Five Estimation Strategies

So far we only reconstructed the test states from the DAE output layer using the standard linear regression model, to compare the performance of all five estimation strategies discussed in Section 7.5, the fidelity of 2000 test states with $N_n^{\text{train}} = N_n^{\text{test}} = 1000$ and tetrahedron measurement scheme are computed, see Table (7.2) The last column of Table (7.2) is the

Fidelity of Five Different Measurement Strategies - Simulated Data					
Method	LI	MLE	Clean-LI	Clean-MLE	NN
Average	0.970	0.970	0.982	0.983	0.997
Median	0.970	0.969	0.982	0.982	0.999
Min	0.772	0.792	0.832	0.888	0.9910
Max	0.993	0.995	0.999	0.999	0.999

Table 7.2: Fidelity of five different measurement strategies of simulated data.

fidelity measure for the second decoder, which outperforms the other four strategies. This could be because training the second decoder removes the noise for the second time (after training with DAE), and direct positive state estimation ensures that the state is not outside the Bloch sphere.

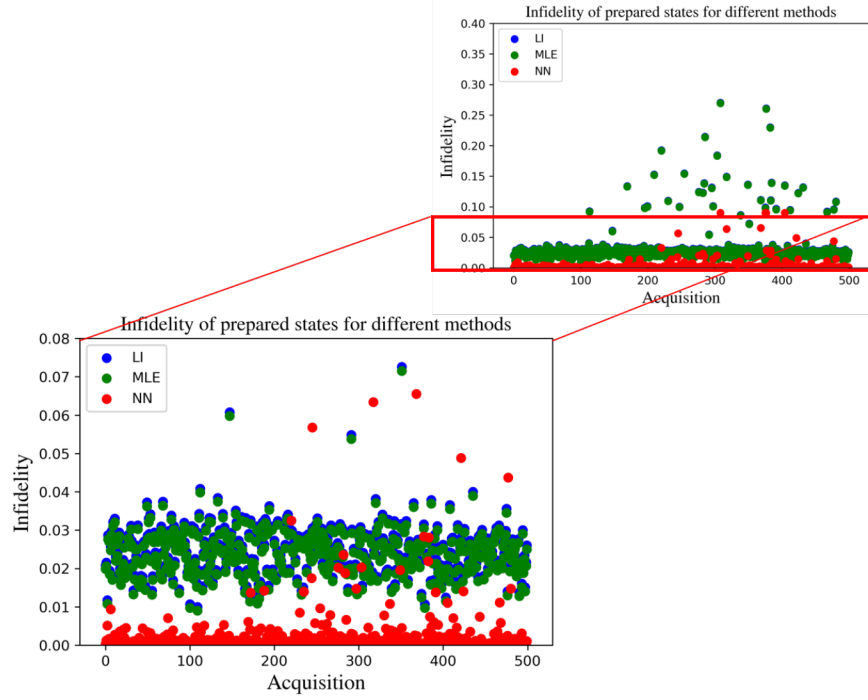


Figure 7.8: 500 random states are reconstructed with three different strategies: linear inversion, maximum likelihood estimation and the denoising autoencoder with second decoder. the MLE method only slightly improves LI in practice which was expected since we use LI to initialize the MLE algorithm. The denoising autoencoder with a second decoder on the other hand substantially improves fidelity over the usual methods ($\sim 2\%$).

7.6.2 Experimental Data of a Planar Multi-Transmon Qubit

These results are verified with the experimental data obtained from measuring a planar multi-transmon qubit, in Fig. (7.8). 500 random states are reconstructed with all five different strategies that were discussed, see Fig. (7.5). In Fig. (7.8), the infidelity of three of them (to keep the plot simple, and also compare the performance of the second decoder with the standard methods that are used in laboratory) are compared which are: linear inversion, maximum likelihood estimation and the denoising autoencoder with second decoder. As we can see MLE only slightly improves LI in practice which was expected since we use LI to initialize the MLE algorithm. To compare the performances of all five estimation strategies, see Table 7.3, This table shows that the denoising autoencoder with a second

Fidelity of Five Different Measurement Strategies - Experimental Data					
Method	LI	MLE	Clean-LI	Clean-MLE	NN
Average	0.971	0.975	0.980	0.984	0.996
Median	0.970	0.973	0.981	0.981	0.999
Min	0.731	0.733	0.868	0.873	0.910
Max	0.991	0.994	0.999	0.999	0.999
Std	0.12	0.12	0.06	0.06	0.02

Table 7.3: Fidelity of five different measurement strategies of experimental data.

decoder substantially improves fidelity over the usual methods ($\sim 2\%$).

7.6.3 Two-Qubit System

In order to scale up to multi-qubits, there are different neural network design choices that one will need to make: A shallow network with only one hidden latent layer, or a deeper network with several hidden layers. These choices can be made independently for either the encoder or the decoder since the two have distinct purposes. As discussed in Chapter 5, adding extra hidden layers introduces more nested non-linear activation functions which can in principle emulate more complicated nonlinear functions using fewer nodes, which may have advantages during training.

One of the biggest benefits of the shallow design is that it forces an embedding space that is linearly related to the orthogonal coordinates of quantum states. It means that the learned embedding space will contain the entire physical state space and identify its dimension, even though it has been trained with very small subset of the state space: The eigenstates of the measurement. For example, the standard Bloch measurement uses only the 6 cardinal points for training, but the NN is able to infer that the minimum-dimension embedding space for all states must have dimension 3, and in the latent nodes correctly reconstructs physical states as appearing within a unit sphere volume in that embedding space, up to the scaling and shifting identified in the linear regression of the latent nodes. The second decoder further

improves this latent node estimate to force a physical state output, much like MLE does for LI, but even without this second decoder the NN has already identified the essential state space structure of the qubit without having that information a priori. The con of the shallow design is it will need to learn the maximum number of parameters to reconstruct the state. For example in the case of one and two qubits the latent layer of the shallow network will need to have 3 and 15 nodes respectively during the training, in order to reconstruct the high fidelity states. This means if we increase the number of qubits the numbers of parameters will scale up exponentially, therefore, will not solve the scaling problem.

The deep design on the other hand, can learn a submanifold of the embedding space that is specified to the chosen set of training states. For example for a single qubit, rather than learning 3D space the Bloch sphere, the added nonlinear complexity of the network allows it to learn the 2D curved manifold of the Bloch sphere, which reduces the dimensionality of the compressed representation in the latent nodes by one. This number will be 6 for two qubit pure states. In order to train such a deep network that is able to learn the curved structure of the space, the sample set of states for training needs to be evenly chosen from the surface, i.e., the surfaces of Bloch sphere. This means that we will need to have high fidelity preparations that cover the submanifold evenly. In this work these states are randomly sampled according to the Haar measure [202].

For two-qubit state tomography, we trained a shallow network with only 36 product states and tested with a set of states in the form

$$|\psi(\varphi)\rangle \propto |00\rangle + |01\rangle + |10\rangle + e^{i\varphi} |11\rangle, \quad (7.13)$$

that are parametrized by a single phase parameter φ . 2000 test states are prepared by varying the phase parameter φ between 0 and 2π . These test states are product states only for $\varphi = 0$ and $\varphi = 2\pi$, and are maximally entangled for $\varphi = \pi$. Therefore the test states are

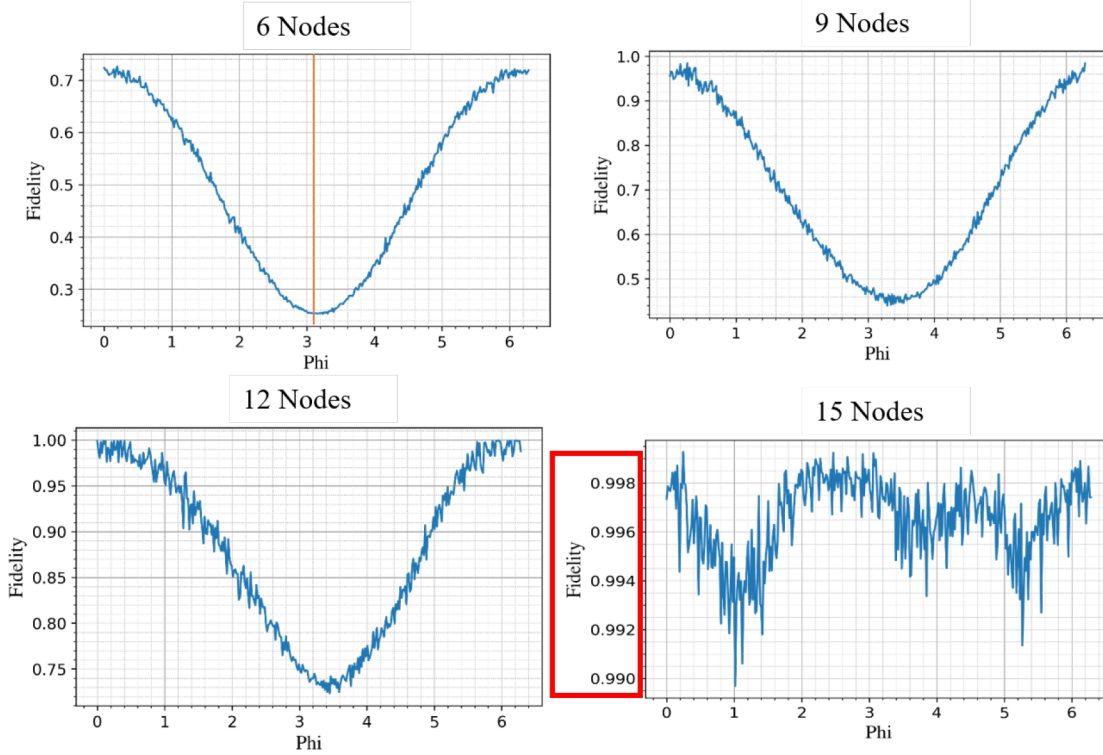


Figure 7.9: The fidelity of the reconstructed 2000 test states versus the phase variable of $|\psi(\varphi)\rangle \propto |00\rangle + |01\rangle + |10\rangle + e^{i\varphi} |11\rangle$, for different number of nodes in the latent layer.

varying between product and maximally entangled states.

The fidelity of the reconstructed test states versus the phase variable is shown in Fig. (7.9), which compares its performance for different choices of latent node number (6,9,12,15) ranging from the minimum number needed to represent all pure entangled states to the number required for arbitrary 2-qubit density matrices. The NN was trained independently for each number of nodes to assess whether it was able to identify the nonlinear structure of the specific submanifold containing the training data. For numbers of nodes less than 15, the product states are reconstructed with a reasonable fidelity but the fidelity degrades with increasing entanglement. The fidelity improves as more latent nodes are added, but, entangled states only achieve fidelity comparable to the pure state fidelity when the latent layer has all 15 nodes needed to describe any density matrix. Although the representation dimensionality

is not compressed by the shallow network, it is notable that high fidelity for all entangled states can still be achieved with 15 nodes despite using only product eigenstates during training. This economy of training data could be very useful for diagnosing two-qubit gate errors and multi-qubit cross-talk, since the 36 product eigenstates used for training can all be experimentally prepared with high-fidelity using only optimized single-qubit gates and projective readout. Two-qubit gates are significantly more difficult to calibrate, whereas single-qubit gates can be reliably calibrated to high-fidelity. Thus, a NN trained using only product eigenstates will have high-quality training preparations that do not depend upon any imperfections of the two-qubit control. Nevertheless, the trained NN will still be able to reconstruct all entangled states and thus help identify where there are mismatches between intended control and the actual state being produced.

For the deep network, with 3 hidden layer in the middle including the latent layer, with the only important difference being that 500 product states sampled from Haar measure have been used for training the network. Table 7.4 compares the fidelity for different nodes, The

Fidelity vs. Number of Nodes		
Fidelity	$\varphi = 0$	$\varphi = \pi$
6 Nodes	0.992	0.280
9 Nodes	0.992	0.412
12 Nodes	0.993	0.563
15 Nodes	0.993	0.584

Table 7.4: Fidelity vs. Number of Nodes in Latent Layer for Deep Network.

deep network is able to reconstruct product states with high fidelity using any of the tested node numbers, it can no longer reconstruct entangled states with high fidelity, even with 15 nodes. This means that the deep design of the denoising autoencoder has learned the product state space that matches the structure of the chosen training data, but does not preserve the full embedding space structure like the shallow design. Instead, it specializes its learned representation to more closely match the training data, so can no longer encode states that fall outside that specialized internal representation. The benefit of this specialization is that

fewer latent nodes are needed to encode an appropriate submanifold that includes the training data, making the network operation more efficient once trained. Moreover, if the structure of a set of training data is unknown, the NN will identify the relevant structure internally. The latent nodes can then be examined to identify how the appropriate manifold is being efficiently parameterized by the trained network. This feature of the deep network may be useful for auto-learning optimal parameterizations for physically relevant submanifolds in multi-qubit systems. The structure of these parameterizations may provide insight into the relevant many-body states while drastically lowering the dimension necessary for practical characterization of the many-body system.

8 Conclusion and Outlook

Physicists are now applying machine learning, especially deep learning, to a wide variety of topics, from state estimation to analysis of measurement data and optimization of control strategies. All of these applications automatically benefit from the astonishing rate of progress in the machine learning community. The purpose of this thesis was to compare the performance of traditional methods and modern deep learning models for estimating and tracking the quantum states of superconducting processors.

For quantum state tracking, we have shown that LSTM recurrent neural networks produce accurate trajectories of quickly driven qubit dynamics even in parameter regimes where conventional reconstruction methods are less accurate. Our results open up a new area of continuous qubit monitoring where the dynamics of individual qubit trajectories is allowed to change rapidly compared with the detection bandwidth, which in principle can enable tracking of qubit trajectories to diagnose errors during quantum gates, and other exciting feedback control or closed-loop optimization applications.

As another example, our LSTM approach shows promise for detection and classification of stochastic errors in continuous quantum error correction experiments [46, 166, 167]. An interesting future direction to this work could be improving the LSTM accuracy for parameter estimation by adding physical constraints to the LSTM loss function, which can facilitate extracting interesting system parameters such as the memory kernel learned automatically during the training process. In addition, other neural network architectures, such as tensor networks [203–205] or wavenets [206], may improve reconstruction accuracy, or reconstruct trajectories with fewer training data.

For the task of quantum state tomography we used a state-of-the-art autoencoder model, inspired by the denoising autoencoder. We showed that our method robustly reconstructs quantum states even in the presence of experimental imperfections, with accuracy as good or

better than methods such as linear inversion and maximum likelihood estimation. Notably, its mitigation of troublesome state preparation and measurement (SPAM) errors performs particularly well due to the auto-calibration of the typical readout error during the training process. While we have shown preliminary data for how to extend this technique to two-qubit state tomography, the next step in this line of research is to investigate how to scale the method to while keeping data collection and training overhead sufficiently low to remain practical larger systems.

It has become apparent in recent years that the exchange of ideas between quantum computing and machine learning has its own genuine questions and promises as we also saw in this work. There are still a lot of areas that machine learning might be applied to improve the methods that have been used to control and manipulate the quantum systems and even help with the scaling problem using generative models such as different type of autoencoders to extract a minimal representation of the quantum states. Also, neural networks can be used for characterization of experimental devices, specifically when the relationship between the observations and the underlying parameters is more complex and can not be done easily experimentally or numerically.

It is also possible that rather than only interpreting a given set of measurement outcomes, similar to what we do in Chapter 6, one can use neural networks to choose the most informative measurements instead. Given a sequence of prior observations, the observable for the next measurement can be selected to maximize the information. In other words, we are looking for an adaptive-measurement strategy that represents a high-dimensional optimization problem. Machine learning tools can help discovering such strategies.

References

- [1] P. Benioff, The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines, *Journal of statistical physics* **22**, 563–591 (1980).
- [2] R. P. Feynman, Simulating physics with computers, in *Feynman and computation* (CRC Press, 2018) pp. 133–153.
- [3] D. Deutsch, Quantum theory, the church–turing principle and the universal quantum computer, *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* **400**, 97–117 (1985).
- [4] D. Deutsch and R. Jozsa, Rapid solution of problems by quantum computation, *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* **439**, 553–558 (1992).
- [5] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (1996) pp. 212–219.
- [6] E. Bernstein and U. Vazirani, Quantum complexity theory, *SIAM Journal on computing* **26**, 1411–1473 (1997).
- [7] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM review* **41**, 303–332 (1999).
- [8] S. Lloyd, Universal quantum simulators, *Science* , 1073–1078 (1996).
- [9] K. Raha, M. B. Peters, B. Wang, N. Yu, A. M. Wollacott, L. M. Westerhoff, and K. M. Merz Jr, The role of quantum mechanics in structure-based drug design, *Drug Discovery Today* **12**, 725–731 (2007).
- [10] M. B. Peters, K. Raha, and K. Merz, Quantum mechanics in structure-based drug design, *Current Opinion in Drug Discovery and Development* **9**, 370 (2006).
- [11] J. Singh, *Quantum Mechanics: Fundamentals and applications to technology* (John Wiley & Sons, 2008).
- [12] R. M. Weiner, Boson interferometry in high-energy physics, *Physics Reports* **327**, 249–346 (2000).
- [13] M. A. Nielsen and I. Chuang, *Quantum computation and quantum information* (2002).

- [14] C. G. Almudever, L. Lao, X. Fu, N. Khammassi, I. Ashraf, D. Iorga, S. Varsamopoulos, C. Eichler, A. Wallraff, L. Geck, *et al.*, The engineering challenges in quantum computing, in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017* (IEEE, 2017) pp. 836–845.
- [15] R. Biswas, Z. Jiang, K. Kechezhi, S. Knysh, S. Mandra, B. O’Gorman, A. Perdomo-Ortiz, A. Petukhov, J. Realpe-Gómez, E. Rieffel, *et al.*, A nasa perspective on quantum computing: Opportunities and challenges, *Parallel Computing* **64**, 81–98 (2017).
- [16] A. D. Córcoles, A. Kandala, A. Javadi-Abhari, D. T. McClure, A. W. Cross, K. Temme, P. D. Nation, M. Steffen, and J. M. Gambetta, Challenges and opportunities of near-term quantum computing systems, arXiv preprint arXiv:1910.02894 (2019).
- [17] M. H. Devoret, A. Wallraff, and J. M. Martinis, Superconducting qubits: A short review, arXiv preprint cond-mat/0411174 (2004).
- [18] J. Clarke and F. K. Wilhelm, Superconducting quantum bits, *Nature* **453**, 1031–1042 (2008).
- [19] R. Schoelkopf and S. Girvin, Wiring up quantum systems, *Nature* **451**, 664–669 (2008).
- [20] M. H. Devoret and R. J. Schoelkopf, Superconducting circuits for quantum information: an outlook, *Science* **339**, 1169–1174 (2013).
- [21] Y. Nakamura, Y. A. Pashkin, and J. S. Tsai, Coherent control of macroscopic quantum states in a single-cooper-pair box, *nature* **398**, 786–788 (1999).
- [22] S. Gulde, M. Riebe, G. P. Lancaster, C. Becher, J. Eschner, H. Häffner, F. Schmidt-Kaler, I. L. Chuang, and R. Blatt, Implementation of the deutsch–jozsa algorithm on an ion-trap quantum computer, *Nature* **421**, 48–50 (2003).
- [23] A. Blais, R.-S. Huang, A. Wallraff, S. M. Girvin, and R. J. Schoelkopf, Cavity quantum electrodynamics for superconducting electrical circuits: An architecture for quantum computation, *Physical Review A* **69**, 062320 (2004).
- [24] A. Wallraff, D. I. Schuster, A. Blais, L. Frunzio, R.-S. Huang, J. Majer, S. Kumar, S. M. Girvin, and R. J. Schoelkopf, Strong coupling of a single photon to a superconducting qubit using circuit quantum electrodynamics, *Nature* **431**, 162–167 (2004).
- [25] J.-M. Raimond, M. Brune, and S. Haroche, Manipulating quantum entanglement with atoms and photons in a cavity, *Reviews of Modern Physics* **73**, 565 (2001).
- [26] S. Haroche and J.-M. Raimond, *Exploring the quantum: atoms, cavities, and photons* (Oxford university press, 2006).
- [27] H. Paik, D. Schuster, L. S. Bishop, G. Kirchmair, G. Catelani, A. Sears, B. Johnson, M. Reagor, L. Frunzio, L. Glazman, *et al.*, Observation of high coherence in josephson junction qubits measured in a three-dimensional circuit qed architecture, *Physical Review Letters* **107**, 240501 (2011).

- [28] C. Rigetti, J. M. Gambetta, S. Poletto, B. L. Plourde, J. M. Chow, A. D. Córcoles, J. A. Smolin, S. T. Merkel, J. R. Rozen, G. A. Keefe, *et al.*, Superconducting qubit in a waveguide cavity with a coherence time approaching 0.1 ms, *Physical Review B* **86**, 100506 (2012).
- [29] A. Rivas and S. F. Huelga, *Open quantum systems*, Vol. 10 (Springer, 2012).
- [30] I. Rotter and J. Bird, A review of progress in the physics of open quantum systems: theory and experiment, *Reports on Progress in Physics* **78**, 114001 (2015).
- [31] H.-P. Breuer, F. Petruccione, *et al.*, *The theory of open quantum systems* (Oxford University Press on Demand, 2002).
- [32] A. Isar, A. Sandulescu, H. Scutaru, E. Stefanescu, and W. Scheid, Open quantum systems, *International Journal of Modern Physics E* **3**, 635–714 (1994).
- [33] A. N. Korotkov and K. Keane, Decoherence suppression by quantum measurement reversal, *Physical Review A* **81**, 040103 (2010).
- [34] Y.-S. Kim, J.-C. Lee, O. Kwon, and Y.-H. Kim, Protecting entanglement from decoherence using weak measurement and quantum measurement reversal, *Nature Physics* **8**, 117–120 (2012).
- [35] G. Gillett, R. Dalton, B. Lanyon, M. Almeida, M. Barbieri, G. J. Pryde, J. O’Brien, K. Resch, S. Bartlett, and A. White, Experimental feedback control of quantum systems using weak measurements, *Physical review letters* **104**, 080503 (2010).
- [36] R. Vijay, C. Macklin, D. Slichter, S. Weber, K. Murch, R. Naik, A. N. Korotkov, and I. Siddiqi, Stabilizing rabi oscillations in a superconducting qubit using quantum feedback, *Nature* **490**, 77–80 (2012).
- [37] S. Hacohen-Gourgy, L. P. García-Pintos, L. S. Martin, J. Dressel, and I. Siddiqi, Incoherent qubit control using the quantum zeno effect, *Physical review letters* **120**, 020505 (2018).
- [38] K. Murch, S. Weber, C. Macklin, and I. Siddiqi, Observing single quantum trajectories of a superconducting quantum bit, *Nature* **502**, 211–214 (2013).
- [39] S. Weber, A. Chantasri, J. Dressel, A. N. Jordan, K. Murch, and I. Siddiqi, Mapping the optimal route between two quantum states, *Nature* **511**, 570–573 (2014).
- [40] A. Korotkov and D. Averin, Continuous weak measurement of quantum coherent oscillations, *Physical Review B* **64**, 165310 (2001).
- [41] M. Schuld and N. Killoran, Quantum machine learning in feature hilbert spaces, *Physical review letters* **122**, 040504 (2019).

- [42] Y. Nomura, A. S. Darmawan, Y. Yamaji, and M. Imada, Restricted boltzmann machine learning for solving strongly correlated quantum systems, *Physical Review B* **96**, 205152 (2017).
- [43] E. Flurin, L. S. Martin, S. Hacothen-Gourgy, and I. Siddiqi, Using a recurrent neural network to reconstruct quantum dynamics of a superconducting qubit from physical observations, *Physical Review X* **10**, 011006 (2020).
- [44] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Neural-network quantum state tomography, *Nature Physics* **14**, 447–450 (2018).
- [45] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, *Science* **355**, 602–606 (2017).
- [46] P. Baireuther, T. E. O’Brien, B. Tarasinski, and C. W. Beenakker, Machine-learning-assisted correction of correlated qubit errors in a topological code, *Quantum* **2**, 48 (2018).
- [47] S.-C. Wang, Artificial neural network, in *Interdisciplinary computing in java programming* (Springer, 2003) pp. 81–100.
- [48] A. Dongare, R. Kharde, A. D. Kachare, *et al.*, Introduction to artificial neural network, *International Journal of Engineering and Innovative Technology (IJEIT)* **2**, 189–194 (2012).
- [49] J. Slawny, On factor representations and the c^* -algebra of canonical commutation relations, *Communications in Mathematical Physics* **24**, 151–170 (1972).
- [50] E. Schrödinger, About heisenberg uncertainty relation, arXiv preprint quant-ph/9903100 (1999).
- [51] J. J. Sakurai and E. D. Commins, *Modern quantum mechanics*, revised edition (1995).
- [52] M. Rieth and W. Schommers, *Handbook Of Theoretical And Computational Nanotechnology. Volume 7: Magnetic Nanostructures And Nano-Optics* (American Scientific Publishers, 2007).
- [53] N. Amir and S. Iqbal, Exact solutions of schrödinger equation for the position-dependent effective mass harmonic oscillator, *Communications in Theoretical Physics* **62**, 790 (2014).
- [54] S. Howard and S. K. Roy, Coherent states of a harmonic oscillator, *American Journal of Physics* **55**, 1109–1117 (1987).
- [55] J. G. Hartley and J. R. Ray, Coherent states for the time-dependent harmonic oscillator, *Physical Review D* **25**, 382 (1982).

- [56] G. A. Baker Jr, Formulation of quantum mechanics based on the quasi-probability distribution induced on phase space, *Physical Review* **109**, 2198 (1958).
- [57] T. L. Curtright and C. K. Zachos, Quantum mechanics in phase space, *Asia Pacific Physics Newsletter* **1**, 37–46 (2012).
- [58] H. M. Wiseman and G. J. Milburn, Quantum theory of optical feedback via homodyne detection, *Physical Review Letters* **70**, 548 (1993).
- [59] W. K. Wootters, A wigner-function formulation of finite-state quantum mechanics, *Annals of Physics* **176**, 1–21 (1987).
- [60] E. T. Jaynes and F. W. Cummings, Comparison of quantum and semiclassical radiation theories with application to the beam maser, *Proceedings of the IEEE* **51**, 89–109 (1963).
- [61] F. Cummings, Stimulated emission of radiation in a single mode, *Physical Review* **140**, A1051 (1965).
- [62] D. Braak, Integrability of the rabi model, *Physical Review Letters* **107**, 100401 (2011).
- [63] C. Gerry, P. Knight, and P. L. Knight, *Introductory quantum optics* (Cambridge university press, 2005).
- [64] P. L. Knight and P. W. Milonni, The rabi frequency in optical spectra, *Physics Reports* **66**, 21–107 (1980).
- [65] K. Andersson, P. A. Malmqvist, B. O. Roos, A. J. Sadlej, and K. Wolinski, Second-order perturbation theory with a casscf reference function, *Journal of Physical Chemistry* **94**, 5483–5488 (1990).
- [66] B. T. Matthias, T. H. Geballe, and V. B. Compton, Superconductivity, *Reviews of Modern Physics* **35**, 1 (1963).
- [67] J. Bardeen, L. N. Cooper, and J. R. Schrieffer, Theory of superconductivity, *Physical review* **108**, 1175 (1957).
- [68] V. L. Ginzburg and L. D. Landau, On the theory of superconductivity, in *On Superconductivity and Superfluidity* (Springer, 2009) pp. 113–137.
- [69] L. N. Cooper, Bound electron pairs in a degenerate fermi gas, *Physical Review* **104**, 1189 (1956).
- [70] W. Kohn and J. Luttinger, New mechanism for superconductivity, *Physical Review Letters* **15**, 524 (1965).
- [71] D. I. Schuster, *Circuit quantum electrodynamics* (Yale University, 2007).
- [72] L. S. Bishop, *Circuit quantum electrodynamics* (Yale University, 2010).

- [73] Y. Makhlin, G. Schön, and A. Shnirman, Quantum-state engineering with josephson-junction devices, *Reviews of modern physics* **73**, 357 (2001).
- [74] M. Levi, F. C. Hoppensteadt, and W. Miranker, Dynamics of the josephson junction, *Quarterly of Applied Mathematics* **36**, 167–198 (1978).
- [75] Y. Makhlin, G. Scöhn, and A. Shnirman, Josephson-junction qubits with controlled couplings, *nature* **398**, 305–307 (1999).
- [76] R. Gati and M. K. Oberthaler, A bosonic josephson junction, *Journal of Physics B: Atomic, Molecular and Optical Physics* **40**, R61 (2007).
- [77] J. Koch, M. Y. Terri, J. Gambetta, A. A. Houck, D. I. Schuster, J. Majer, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, Charge-insensitive qubit design derived from the cooper pair box, *Physical Review A* **76**, 042319 (2007).
- [78] D. K. Cheng *et al.*, *Field and wave electromagnetics* (Pearson Education India, 1989).
- [79] M. Tinkham, *Introduction to superconductivity* (Courier Corporation, 2004).
- [80] S. M. Girvin, Circuit qed: superconducting qubits coupled to microwave photons, *Quantum machines: measurement and control of engineered quantum systems* **113**, 2 (2011).
- [81] V. Bouchiat, D. Vion, P. Joyez, D. Esteve, and M. Devoret, Quantum coherence with a single cooper pair, *Physica Scripta* **1998**, 165 (1998).
- [82] A. Shnirman, G. Schön, and Z. Hermon, Quantum manipulations of small josephson junctions, *Physical Review Letters* **79**, 2371 (1997).
- [83] J. M. Martinis, S. Nam, J. Aumentado, and C. Urbina, Rabi oscillations in a large josephson-junction qubit, *Physical review letters* **89**, 117901 (2002).
- [84] J. Mooij, T. Orlando, L. Levitov, L. Tian, C. H. Van der Wal, and S. Lloyd, Josephson persistent-current qubit, *Science* **285**, 1036–1039 (1999).
- [85] T. Orlando, J. Mooij, L. Tian, C. H. Van Der Wal, L. Levitov, S. Lloyd, and J. Mazo, Superconducting persistent-current qubit, *Physical Review B* **60**, 15398 (1999).
- [86] J. Schreier, A. A. Houck, J. Koch, D. I. Schuster, B. Johnson, J. Chow, J. M. Gambetta, J. Majer, L. Frunzio, M. H. Devoret, *et al.*, Suppressing charge noise decoherence in superconducting charge qubits, *Physical Review B* **77**, 180502 (2008).
- [87] E. A. Sete, J. M. Martinis, and A. N. Korotkov, Quantum theory of a bandpass purcell filter for qubit readout, *Physical Review A* **92**, 012325 (2015).
- [88] P. Campagne-Ibarcq, *Measurement back action and feedback in superconducting circuits*, Ph.D. thesis, Ecole Normale Supérieure (ENS) (2015).

- [89] A. A. Clerk, M. H. Devoret, S. M. Girvin, F. Marquardt, and R. J. Schoelkopf, Introduction to quantum noise, measurement, and amplification, *Reviews of Modern Physics* **82**, 1155 (2010).
- [90] J. Gambetta, A. Blais, M. Boissonneault, A. A. Houck, D. Schuster, and S. M. Girvin, Quantum trajectory approach to circuit qed: Quantum jumps and the zeno effect, *Physical Review A* **77**, 012112 (2008).
- [91] M. Boissonneault, J. M. Gambetta, and A. Blais, Dispersive regime of circuit qed: Photon-dependent qubit dephasing and relaxation rates, *Physical Review A* **79**, 013819 (2009).
- [92] M. Boissonneault, J. M. Gambetta, and A. Blais, Nonlinear dispersive regime of cavity qed: The dressed dephasing model, *Physical Review A* **77**, 060305 (2008).
- [93] C. M. Caves, J. Combes, Z. Jiang, and S. Pandey, Quantum limits on phase-preserving linear amplifiers, *Physical Review A* **86**, 063802 (2012).
- [94] N. Bergeal, F. Schackert, M. Metcalfe, R. Vijay, V. Manucharyan, L. Frunzio, D. Prober, R. Schoelkopf, S. Girvin, and M. Devoret, Phase-preserving amplification near the quantum limit with a josephson ring modulator, *Nature* **465**, 64–68 (2010).
- [95] M. Fuwa, S. Takeda, M. Zwierz, H. M. Wiseman, and A. Furusawa, Experimental proof of nonlocal wavefunction collapse for a single particle using homodyne measurements, *Nature Communications* **6**, 1–6 (2015).
- [96] M. Chapman, *Heterodyne and homodyne interferometry*, Renishaw, New Mills (2002).
- [97] H. M. Wiseman and G. J. Milburn, *Quantum measurement and control* (Cambridge university press, 2009).
- [98] C. W. Helstrom, Quantum detection and estimation theory, *Journal of Statistical Physics* **1**, 231–252 (1969).
- [99] K. Jacobs, *Quantum measurement theory and its applications* (Cambridge University Press, 2014).
- [100] E. Davies, Information and quantum measurement, *IEEE Transactions on Information Theory* **24**, 596–599 (1978).
- [101] M. A. Nielsen and I. L. Chuang, *Quantum computing and quantum information* (2000).
- [102] V. Belavkin, Measurement, filtering and control in quantum open dynamical systems, *Reports on Mathematical Physics* **43**, A405–A425 (1999).
- [103] P. Bushev, D. Rotter, A. Wilson, F. Dubin, C. Becher, J. Eschner, R. Blatt, V. Steixner, P. Rabl, and P. Zoller, Feedback cooling of a single trapped ion, *Physical review letters* **96**, 043003 (2006).

- [104] J. Combes and K. Jacobs, Rapid state reduction of quantum systems using feedback control, *Physical review letters* **96**, 010504 (2006).
- [105] C. D’Helon and M. James, Stability, gain, and robustness in quantum feedback networks, *Physical Review A* **73**, 053803 (2006).
- [106] P. Rabl, V. Steixner, and P. Zoller, Quantum-limited velocity readout and quantum feedback cooling of a trapped ion via electromagnetically induced transparency, *Physical Review A* **72**, 043823 (2005).
- [107] H. Wiseman and A. Doherty, Optimal unravellings for feedback control in linear quantum systems, *Physical Review Letters* **94**, 070405 (2005).
- [108] K. W. Murch, R. Vijay, and I. Siddiqi, Weak measurement and feedback in superconducting quantum circuits, in *Superconducting Devices in Quantum Optics* (Springer, 2016) pp. 163–185.
- [109] L. Vaidman, Weak measurements, arXiv preprint hep-th/9408154 (1994).
- [110] K. Jacobs and D. A. Steck, A straightforward introduction to continuous quantum measurement, *Contemporary Physics* **47**, 279–303 (2006).
- [111] H. Carmichael, Quantum trajectory theory for cascaded open systems, *Physical review letters* **70**, 2273 (1993).
- [112] J. Gambetta, A. Blais, D. I. Schuster, A. Wallraff, L. Frunzio, J. Majer, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, Qubit-photon interactions in a cavity: Measurement-induced dephasing and number splitting, *Physical Review A* **74**, 042318 (2006).
- [113] C. W. Gardiner and M. J. Collett, Input and output in damped quantum systems: Quantum stochastic differential equations and the master equation, *Physical Review A* **31**, 3761 (1985).
- [114] A. Barchielli, Measurement theory and stochastic differential equations in quantum mechanics, *Physical review A* **34**, 1642 (1986).
- [115] L. Diósi, Continuous quantum measurement and itô formalism, *Physics Letters A* **129**, 419–423 (1988).
- [116] V. P. Belavkin, Quantum continual measurements and a posteriori collapse on ccr, *Communications in mathematical physics* **146**, 611–635 (1992).
- [117] A. N. Korotkov, Quantum bayesian approach to circuit qed measurement, arXiv preprint arXiv:1111.4016 (2011).
- [118] A. N. Korotkov, Quantum bayesian approach to circuit qed measurement with moderate bandwidth, *Physical Review A* **94**, 042326 (2016).

- [119] P. Rouchon and J. F. Ralph, Efficient quantum filtering for quantum feedback control, *Physical Review A* **91**, 012118 (2015).
- [120] M. Anthony and P. L. Bartlett, *Neural network learning: Theoretical foundations* (cambridge university press, 2009).
- [121] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, *nature* **521**, 436–444 (2015).
- [122] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, Vol. 1 (MIT press Cambridge, 2016).
- [123] I. Glasser, N. Pancotti, M. August, I. D. Rodriguez, and J. I. Cirac, Neural-network quantum states, string-bond states, and chiral topological states, *Physical Review X* **8**, 011006 (2018).
- [124] J. Gao, L.-F. Qiao, Z.-Q. Jiao, Y.-C. Ma, C.-Q. Hu, R.-J. Ren, A.-L. Yang, H. Tang, M.-H. Yung, and X.-M. Jin, Experimental machine learning of quantum states, *Physical review letters* **120**, 240501 (2018).
- [125] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Many-body quantum state tomography with neural networks, *arXiv preprint arXiv:1703.05334* (2017).
- [126] A. Seif, K. A. Landsman, N. M. Linke, C. Figgatt, C. Monroe, and M. Hafezi, Machine learning assisted readout of trapped-ion qubits, *Journal of Physics B: Atomic, Molecular and Optical Physics* **51**, 174006 (2018).
- [127] G. Torlai and R. G. Melko, Neural decoder for topological codes, *Physical review letters* **119**, 030501 (2017).
- [128] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, Reinforcement learning with neural networks for quantum feedback, *Physical Review X* **8**, 031084 (2018).
- [129] H. P. Nautrup, N. Delfosse, V. Dunjko, H. J. Briegel, and N. Friis, Optimizing quantum error correction codes with reinforcement learning, *Quantum* **3**, 215 (2019).
- [130] L. Wang, Discovering phase transitions with unsupervised learning, *Physical Review B* **94**, 195105 (2016).
- [131] J. Carrasquilla and R. G. Melko, Machine learning phases of matter, *Nature Physics* **13**, 431–434 (2017).
- [132] M. J. Beach, A. Golubeva, and R. G. Melko, Machine learning vortices at the kosterlitz-thouless transition, *Physical Review B* **97**, 045207 (2018).
- [133] X.-Y. Dong, F. Pollmann, X.-F. Zhang, *et al.*, Machine learning of quantum phase transitions, *Physical Review B* **99**, 121104 (2019).

- [134] D.-L. Deng, X. Li, and S. D. Sarma, Quantum entanglement in neural network states, *Physical Review X* **7**, 021021 (2017).
- [135] Y. Levine, O. Sharir, N. Cohen, and A. Shashua, Quantum entanglement in deep learning architectures, *Physical review letters* **122**, 065301 (2019).
- [136] X. Gao and L.-M. Duan, Efficient representation of quantum many-body states with deep neural networks, *Nature communications* **8**, 1–6 (2017).
- [137] M. Schmitt and M. Heyl, Quantum many-body dynamics in two dimensions with artificial neural networks, *Physical Review Letters* **125**, 100503 (2020).
- [138] T. Hastie, R. Tibshirani, and J. Friedman, Overview of supervised learning, in *The elements of statistical learning* (Springer, 2009) pp. 9–41.
- [139] H. B. Barlow, Unsupervised learning, *Neural computation* **1**, 295–311 (1989).
- [140] Z. Ghahramani, Unsupervised learning, in *Summer School on Machine Learning* (Springer, 2003) pp. 72–112.
- [141] T. Hastie, R. Tibshirani, and J. Friedman, Unsupervised learning, in *The elements of statistical learning* (Springer, 2009) pp. 485–585.
- [142] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction* (MIT press, 2018).
- [143] M. A. Wiering and M. Van Otterlo, Reinforcement learning, *Adaptation, learning, and optimization* **12** (2012).
- [144] S. Linnainmaa, Taylor expansion of the accumulated rounding error, *BIT Numerical Mathematics* **16**, 146–160 (1976).
- [145] M. Cilimkovic, Neural networks and back propagation algorithm, *Institute of Technology Blanchardstown, Blanchardstown Road North Dublin* **15**, 1–12 (2015).
- [146] A. Van Ooyen and B. Nienhuis, Improving the convergence of the back-propagation algorithm, *Neural networks* **5**, 465–471 (1992).
- [147] F. A. Gers, J. Schmidhuber, and F. Cummins, Learning to forget: Continual prediction with lstm, *Neural computation* **12**, 2451–2471 (2000).
- [148] A. Sherstinsky, Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network, *Physica D: Nonlinear Phenomena* **404**, 132306 (2020).
- [149] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in *Proceedings of the 25th international conference on Machine learning* (2008) pp. 1096–1103.

- [150] P. Baldi, Autoencoders, unsupervised learning, and deep architectures, in *Proceedings of ICML workshop on unsupervised and transfer learning* (JMLR Workshop and Conference Proceedings, 2012) pp. 37–49.
- [151] W. S. McCulloch and W. Pitts, A logical calculus of the ideas immanent in nervous activity, *The bulletin of mathematical biophysics* **5**, 115–133 (1943).
- [152] F. Rosenblatt, *The perceptron, a perceiving and recognizing automaton Project Para* (Cornell Aeronautical Laboratory, 1957).
- [153] H. J. Kelley, Gradient theory of optimal flight paths, *Ars Journal* **30**, 947–954 (1960).
- [154] S. Dreyfus, The numerical solution of variational problems, *Journal of Mathematical Analysis and Applications* **5**, 30–45 (1962).
- [155] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Learning representations by back-propagating errors, *nature* **323**, 533–536 (1986).
- [156] S. Makridakis and M. Hibon, Evaluating accuracy (or error) measures (1995).
- [157] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).
- [158] L. Bottou *et al.*, Stochastic gradient learning in neural networks, *Proceedings of Neuro-Nimes* **91**, 12 (1991).
- [159] R. Moore and J. DeNero, L1 and l2 regularization for multiclass hinge loss models, in *Symposium on machine learning in speech and language processing* (2011).
- [160] R. Pascanu, T. Mikolov, and Y. Bengio, On the difficulty of training recurrent neural networks, in *International conference on machine learning* (PMLR, 2013) pp. 1310–1318.
- [161] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural computation* **9**, 1735–1780 (1997).
- [162] Y. Bengio, *Learning deep architectures for AI* (Now Publishers Inc, 2009).
- [163] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, *Nature* **549**, 195–202 (2017).
- [164] J. Romero, J. P. Olson, and A. Aspuru-Guzik, Quantum autoencoders for efficient compression of quantum data, *Quantum Science and Technology* **2**, 045001 (2017).
- [165] A. Chantasri, J. Dressel, and A. N. Jordan, Action principle for continuous quantum measurement, *Physical Review A* **88**, 042110 (2013).
- [166] R. Mohseninia, J. Yang, I. Siddiqi, A. N. Jordan, and J. Dressel, Always-on quantum error tracking with continuous parity measurements, *Quantum* **4**, 358 (2020).

- [167] J. Atalaya, M. Bahrami, L. P. Pryadko, and A. N. Korotkov, Bacon-shor code with continuous measurement of noncommuting operators, *Physical Review A* **95**, 032317 (2017).
- [168] G. Koolstra, N. Stevenson, S. Barzili, L. Burns, K. Siva, S. Greenfield, W. Livingston, A. Hashim, R. K. Naik, J. M. Kreikebaum, K. P. O’Brien, D. I. Santiago, J. Dressel, and I. Siddiqi, Monitoring fast superconducting qubit dynamics using a neural network (2021), [arXiv:2108.12023 \[quant-ph\]](#) .
- [169] J. Johnson, C. Macklin, D. Slichter, R. Vijay, E. Weingarten, J. Clarke, and I. Siddiqi, Heralded state preparation in a superconducting qubit, *Physical review letters* **109**, 050506 (2012).
- [170] S. Hacohe-Gourgy and L. S. Martin, Continuous measurements for control of superconducting quantum circuits, *Advances in Physics: X* **5**, 1813626 (2020).
- [171] L. N. Smith, Cyclical learning rates for training neural networks, in *2017 IEEE winter conference on applications of computer vision (WACV)* (IEEE, 2017) pp. 464–472.
- [172] P. Campagne-Ibarcq, P. Six, L. Bretheau, A. Sarlette, M. Mirrahimi, P. Rouchon, and B. Huard, Observing quantum state diffusion by heterodyne detection of fluorescence, *Physical Review X* **6**, 011002 (2016).
- [173] Q. Ficheux, S. Jezouin, Z. Leghtas, and B. Huard, Dynamics of a qubit while simultaneously monitoring its relaxation and dephasing, *Nature communications* **9**, 1–6 (2018).
- [174] J. Cerrillo and J. Cao, Non-markovian dynamical maps: numerical processing of open quantum trajectories, *Physical review letters* **112**, 110401 (2014).
- [175] T. L. Patti, A. Chantasri, L. P. García-Pintos, A. N. Jordan, and J. Dressel, Linear feedback stabilization of a dispersively monitored qubit, *Physical Review A* **96**, 022311 (2017).
- [176] F. Yan, S. Gustavsson, J. Bylander, X. Jin, F. Yoshihara, D. G. Cory, Y. Nakamura, T. P. Orlando, and W. D. Oliver, Rotating-frame relaxation as a noise spectrum analyser of a superconducting qubit undergoing driven evolution, *Nature communications* **4**, 1–8 (2013).
- [177] J. Atalaya, S. Hacohe-Gourgy, I. Siddiqi, and A. N. Korotkov, Correlators exceeding one in continuous measurements of superconducting qubits, *Physical review letters* **122**, 223603 (2019).
- [178] M. Naghiloo, A. Jordan, and K. Murch, Achieving optimal quantum acceleration of frequency estimation using adaptive coherent control, *Physical review letters* **119**, 180801 (2017).

- [179] D. Szombati, A. G. Friero, C. Müller, T. Jones, M. Jerger, and A. Fedorov, Quantum rifling: protecting a qubit from measurement back action, *Physical review letters* **124**, 070401 (2020).
- [180] A. Shabani, J. Roden, and K. Whaley, Continuous measurement of a non-markovian open quantum system, *Physical review letters* **112**, 113601 (2014).
- [181] M. Paris and J. Rehacek, *Quantum state estimation*, Vol. 649 (Springer Science & Business Media, 2004).
- [182] Z. Hradil, Quantum-state estimation, *Physical Review A* **55**, R1561 (1997).
- [183] M. Cramer, M. B. Plenio, S. T. Flammia, R. Somma, D. Gross, S. D. Bartlett, O. Landon-Cardinal, D. Poulin, and Y.-K. Liu, Efficient quantum state tomography, *Nature communications* **1**, 1–7 (2010).
- [184] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert, Quantum state tomography via compressed sensing, *Physical review letters* **105**, 150401 (2010).
- [185] M. Christandl and R. Renner, Reliable quantum state tomography, *Physical Review Letters* **109**, 120403 (2012).
- [186] A. I. Lvovsky and M. G. Raymer, Continuous-variable optical quantum-state tomography, *Reviews of modern physics* **81**, 299 (2009).
- [187] J. Wenner, M. Neeley, R. C. Bialczak, M. Lenander, E. Lucero, A. D. O’Connell, D. Sank, H. Wang, M. Weides, A. N. Cleland, *et al.*, Wirebond crosstalk and cavity modes in large chip mounts for superconducting qubits, *Superconductor Science and Technology* **24**, 065001 (2011).
- [188] D. F. James, P. G. Kwiat, W. J. Munro, and A. G. White, On the measurement of qubits, in *Asymptotic Theory of Quantum Statistical Inference: Selected Papers* (World Scientific, 2005) pp. 509–538.
- [189] J. Rau, Evidence procedure for efficient quantum-state tomography, *Physical Review A* **82**, 012104 (2010).
- [190] R. Blume-Kohout, Optimal, reliable estimation of quantum states, *New Journal of Physics* **12**, 043034 (2010).
- [191] J. Řeháček, Z. Hradil, and M. Ježek, Iterative algorithm for reconstruction of entangled states, *Physical Review A* **63**, 040303 (2001).
- [192] J. Řeháček, Z. Hradil, E. Knill, and A. Lvovsky, Diluted maximum-likelihood algorithm for quantum tomography, *Physical Review A* **75**, 042108 (2007).
- [193] A. Anis and A. Lvovsky, Maximum-likelihood coherent-state quantum process tomography, *New Journal of Physics* **14**, 105021 (2012).

- [194] A. I. Lvovsky, Iterative maximum-likelihood reconstruction in quantum homodyne tomography, *Journal of Optics B: Quantum and Semiclassical Optics* **6**, S556 (2004).
- [195] V. Bužek and G. Drobný, Quantum tomography via the maxent principle, *Journal of Modern Optics* **47**, 2823–2839 (2000).
- [196] V. Bužek, R. Derka, G. Adam, and P. Knight, Reconstruction of quantum states of spin systems: From quantum bayesian inference to quantum tomography, *Annals of Physics* **266**, 454–496 (1998).
- [197] R. Schmied, Quantum state tomography of a single qubit: comparison of methods, *Journal of Modern Optics* **63**, 1744–1758 (2016).
- [198] G. Golub and W. Kahan, Calculating the singular values and pseudo-inverse of a matrix, *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis* **2**, 205–224 (1965).
- [199] C. Schwemmer, L. Knips, D. Richart, H. Weinfurter, T. Moroder, M. Kleinmann, and O. Gühne, Systematic errors in current quantum state tomography tools, *Physical review letters* **114**, 080403 (2015).
- [200] R. Jozsa, Fidelity for mixed quantum states, *Journal of modern optics* **41**, 2315–2323 (1994).
- [201] N. J. Higham, *Analysis of the Cholesky decomposition of a semi-definite matrix* (Oxford University Press, 1990).
- [202] A. Van Daele, The haar measure on a compact quantum group, *Proceedings of the American Mathematical Society* **123**, 3125–3128 (1995).
- [203] A. Cichocki, N. Lee, I. Oseledets, A.-H. Phan, Q. Zhao, and D. P. Mandic, Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions, *Foundations and Trends® in Machine Learning* **9**, 249–429 (2016).
- [204] I. Glasser, N. Pancotti, and J. I. Cirac, Supervised learning with generalized tensor networks, *arXiv preprint arXiv:1806.05964* (2018).
- [205] I. Luchnikov, S. Vintskevich, H. Ouerdane, and S. Filippov, Simulation complexity of open quantum dynamics: Connection with tensor networks, *Physical review letters* **122**, 160401 (2019).
- [206] A. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. Driessche, E. Lockhart, L. Cobo, F. Stimberg, *et al.*, Parallel wavenet: Fast high-fidelity speech synthesis, in *International conference on machine learning* (PMLR, 2018) pp. 3918–3926.