

# **Quantum Statistical Inference**

Submitted by

Zhikuan ZHAO

Thesis Advisor

Prof. Joseph FITZSIMONS

A thesis submitted to the Singapore University of Technology and Design in fulfilment of the requirement for the degree of Doctor of Philosophy.

December 13, 2018

# **Thesis Examination Committee**

TEC Chair: Ricky Ang

Thesis Advisor: Joseph Fitzsimons

Internal TEC Member: Shaowei Lin

Internal TEC Member: Dario Poletti

External TEC Member: Troy Lee<sup>12</sup>

<sup>&</sup>lt;sup>1</sup>University of Technology Sydney

<sup>&</sup>lt;sup>2</sup>Centre for Quantum Technologies, National University of Singapore

# **Declaration of Authorship**

I, Zhikuan ZHAO, declare that this thesis titled, "Quantum Statistical Inference" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the
  exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:		
Date:		

"Product of optimism and knowledge is a constant."

Lev Landau

## **List of Publications**

#### • Quantum Linear System Algorithm for Dense Matrices

L. Wossnig, Z. Zhao, & A. Prakash. Phys. Rev. Lett. 120, 050502 (2018). (Contains work used in Chapter 3)

#### • A note on state preparation for quantum machine learning

Z. Zhao, V. Dunjko, J. K. Fitzsimons, P. Rebentrost, & J. F. Fitzsimons. arXiv preprint arXiv:1804.00281 (2018). (Contains work used in Chapter 5)

#### • Quantum assisted Gaussian process regression

Z. Zhao, J. K. Fitzsimons, & J. F. Fitzsimons, arXiv preprint arXiv:1512.03929 (2015). (Contains work used in Chapter 5)

#### • Quantum algorithms for training Gaussian Processes

Z. Zhao, J. K. Fitzsimons, M. A. Osborne, S. J. Roberts, & J. F. Fitzsimons. arXiv preprint arXiv:1803.10520 (2018). (Contains work used in Chapter 6)

#### • Bayesian Deep Learning on a Quantum Computer

Z. Zhao, A. Pozas-Kerstjens, P. Rebentrost, & P. Wittek. arXiv preprint arXiv: 1806.11463 (2018). (Contains work used in Chapter 7)

#### • Geometry of quantum correlations in space-time

Z. Zhao, R. Pisarczyk, J. Thompson, M. Gu, V. Vedral, & J. F. Fitzsimons, Phys. Rev. A 98, 052312 (2018). (Contains work used in Chapter 8)

#### • Causal limit on quantum communication

R. Pisarczyk, Z. Zhao, Y. Ouyang, V. Vedral, & J. F. Fitzsimons. arXiv preprint arXiv:1804.02594 (2018). (Contains work used in Chapter 9)

## **Statement on Collaborative Work**

The results presented in this thesis came from several fruitful research collaborations.

The quantum linear system algorithm for dense matrices presented in Chapter 3 was developed in collaboration with Leonard Wossnig and Anupam Prakash. I initiated the project, led and jointly contributed to the analytical work, and took the role as the corresponding author of the paper.

The state preparation technique for quantum machine learning used in Chapter 5 was developed together with Vedran Dunjko, Jack Fitzsimons, Patrick Rebentrost and my supervisor, Joseph Fitzsimons who came up with the initial idea. I contributed to working out the technical details of the research.

The quantum assisted Gaussian process algorithm in Chapter 5 came from a collaboration with Jack Fitzsimons, and Joseph Fitzsimons, with whom a discussion inspired the initial idea of the project. I contributed to a large part of the detailed algorithm design and most of the analysis involved.

The quantum algorithms for training Gaussian processes discussed in Chapter 6 came from the collaborative work with Jack Fitzsimons, Michael Osborne, Stephen Roberts, who collectively provided expertise on classical machine learning, and Joseph Fitzsimons who initiated the research. I contributed to the algorithm design and most of the analytical work.

The quantum Bayesian deep learning algorithm described in Chapter 7 was developed in collaboration with Alejandro Pozas-Kerstjens, Patrick Rebentrost, and Peter Wittek, with whom I jointly initiated the project and contributed to the programming of numerical simulations. I did most of the theoretical work, and proved the main theorem with the help of Patrick Rebentrost. Alejandro Pozas-Kerstjens completed the experimental part of the research.

The work presented in Chapter 8 on the geometry of quantum correlations in space-time was done together with Robert Pisarczyk, Jayne Thompson, Mile Gu, Vlatko Vedral and Joseph Fitzsimons. The project originated from discussions with Jayne Thompson, Mile Gu, Vlatko Vedral and Joseph Fitzsimons. I proved the main results, and completed the theoretical details jointly with Robert Pisarczyk.

The work presented in Chapter 9 on bounding channel capacities with quantum causality was done in collaboration with Robert Pisarczyk, Yingkai Ouyang, Vlatko Vedral and Joseph Fitzsimons. Vlatko Vedral and Joseph Fitzsimons initiated the research. I contributed to proving the theoretical results jointly with Robert Pisarczyk and Yingkai Ouyang.

#### SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

## **Abstract**

Doctor of Philosophy

#### **Quantum Statistical Inference**

by Zhikuan ZHAO

In this thesis, I present several results on quantum statistical inference in the following two directions. Firstly, I demonstrate that quantum algorithms can be applied to enhance the computing and training of Gaussian processes (GPs), a powerful model widely used in classical statistical inference and supervised machine learning. A crucial component of the quantum GP algorithm is solving linear systems with quantum computers, for which I present a novel algorithm that achieves a provable advantage over previously known methods. I will also explicitly address the task of encoding the classical data into a quantum state for machine learning applications. I then apply the quantum enhanced GPs to Bayesian deep learning and present an experimental demonstration on contemporary hardware and simulators. Secondly, I look into the notion of quantum causality and apply it to inferring spatial and temporal quantum correlations, and present an analytical toolkit for causal inference in quantum data. I will also make the connection between causality and quantum communications, and present a general bound for the quantum capacity of noisy communication channels.

## Acknowledgements

First and foremost, I would like to express my most sincere gratitude and appreciation to my supervisor, Joseph Fitzsimons for providing continuous support, patient guidance and perhaps most vitally, role model through his most rigorous attitude toward science, which has kept me going throughout the past four years of intellectual journey. Thank you, Joe. Without your tutorship and mentorship, none of these would have been possible.

I would like to thank to Ricky Ang, Shaowei Lin, Dario Poletti and Troy Lee for kindly agreeing to serve as the examination committee for this thesis. I am most grateful to my collaborators: Vedran Dunjko, Jack Fitzsimons, Mile Gu, Michael Osborne, Robert Pisarczyk, Alejandro Pozas-Kerstjens, Anupam Prakash, Patrick Rebentrost, Stephen Roberts, Jayne Thompson, Vlatko Vedral, Peter Wittek, Leonard Wossnig and Ouyang Yingkai. It has been a great pleasure working together. I am also thankful to my peers, Joshua Kettlewell, Atul Mantri and Liming Zhao for the most memorable experience of growing up together, and my seniors in the group, especially Tiago Batalhão, Tommaso Demarie, Michal Hajdušek, Nana Liu and Si-Hui Tan for the care, support and all the fun we had during the past years.

Last but not least, I could never have made it through without the love and support of my families who have always been the heroes by my side during times of struggle. My most profound gratitude goes well beyond the scope of this thesis.

# **Contents**

Declaration of Authorship				i	
Li	ist of l	Publicat	tions	iv	
St	ateme	ent on C	Collaborative Work	v	
Al	bstrac	:t		vii	
A	cknow	vledgem	nents	viii	
Ι	Qu	antum	n computation and algorithms	1	
1	Introduction				
	1.1	Quanti	um mechanics preliminaries	. 4	
		1.1.1	The state space	. 4	
		1.1.2	Evolution of states	. 4	
		1.1.3	Quantum measurements	. 5	
		1.1.4	Composite systems	. 6	
	1.2	Eleme	nts of quantum computation	. 6	
		1.2.1	The qubit	. 6	
		1.2.2	Pauli operators	. 7	
		1.2.3	Quantum gates	. 7	
	1.3	Statisti	ical ensemble of states	. 8	

		1.3.1	The density matrix	8
		1.3.2	Quantum operations	9
2	Esse	ential qu	antum algorithms	10
	2.1	Basic o	quantum algorithms	10
		2.1.1	Quantum Fourier transform	10
		2.1.2	Quantum phase estimation	12
		2.1.3	Black-box Hamiltonian simulation	13
	2.2	Quanti	um linear system algorithm	14
		2.2.1	Quantum formulation of linear systems	15
		2.2.2	The HHL algorithm	16
3	Qua	ntum d	ense linear system algorithm	20
	3.1	Memo	ry model	21
	3.2	Quanti	um singular value estimation	22
		3.2.1	Overview	23
		3.2.2	Procedures	25
		3.2.3	Brief analysis	26
	3.3	The Q	DLS algorithm	28
		3.3.1	Procedures	29
		3.3.2	Analysis	31
	3.4	Summ	ary and discussions	35
II	Ga	aussiai	n processes with quantum algorithms	38
4	Gau	ssian pı	rocesses in classical machine learning	39
	4.1	Introdu	action	40
		4.1.1	Preliminaries	41

	4.2	2 Gaussian process regression			
		4.2.1	Linear model with Gaussian noise	43	
		4.2.2	Feature space projection	44	
		4.2.3	Classical computation and complexity	46	
	4.3	Trainir	ng Gaussian processes	47	
		4.3.1	Log marginal likelihood	47	
		4.3.2	Implementations and complexity	48	
		4.3.3	Stochastic trace estimation	49	
	4.4	Conne	ction with deep learning	50	
5	Oua	ntum ei	nhanced Gaussian processes	52	
J			•		
	5.1	_	preparation	53	
		5.1.1	Quantum random access memory	53	
		5.1.2	Robustness and rounding conventions	54	
		5.1.3	State preparation for GPR	56	
	5.2	Quantu	um Gaussian process algorithm	57	
		5.2.1	Inner product estimation	58	
		5.2.2	Procedures	58	
		5.2.3	Mean predictor	61	
		5.2.4	Variance predictor	62	
	5.3	Discus	sions	63	
6	Trai	ning au	antum Gaussian processes	66	
Ū	6.1	•	um LML algorithm	66	
	0.1	6.1.1	Augmented linear algorithm	67	
		6.1.2	Log determinant algorithm	68	
	6.2	Variati	on estimation	71	

	6.3	Summary			
7	Qua	Sayesian Deep Learning	<b>7</b> 3		
	7.1	Quantum Bayesian training of neural networks			
		7.1.1	Single-layer case	. 76	
		7.1.2	Multi-layer case	. 77	
		7.1.3	Coherent element-wise operations	. 79	
	7.2	Experi	ments	. 84	
		7.2.1	Simulations on a quantum virtual machine	. 85	
		7.2.2	Implementations on quantum processing units	. 87	
	7.3	Summ	ary	. 88	
II	I Q	uantu	ım correlations and causality	90	
8	Geo	metry o	of quantum correlations	<b>9</b> 1	
	8.1	Introdu	uction	. 91	
		8.1.1	Density matrices and spatial correlations	. 92	
		8.1.2	The pseudo-density matrix formalism	. 93	
	8.2	Genera	al two-time quantum correlations	. 94	
		8.2.1	The two-point temporal PDM	. 95	
		8.2.2	Single-qubit quantum channels	. 96	
		8.2.3	Convex closure	. 97	
	8.3	3 Two-point correlations in space-time		. 99	
		8.3.1	General PDM Pauli components	. 101	
	8.4	Discus	ssions	. 103	
9	Can	cality in	n quantum communication	105	
I		•	•		
	9.1	Dound	ling quantum channel capacities	. 100	

		9.1.1	Logarithmic Causality	107
		9.1.2	PDM representation of quantum channels	108
		9.1.3	Causal bound	109
		9.1.4	Proof	110
	9.2	Mather	matical details	112
		9.2.1	Non-increasing property	112
		9.2.2	Large-n limit	116
	9.3	ation of causal bound	118	
		9.3.1	Comparison with Holevo and Werner bound	118
		9.3.2	Shifted depolarising channel	119
	9.4	Summa	ary and discussions	122
10	Conc	clusion		123
	10.1	Summa	ary	123
	10.2	Outloo	k	125

Dedicated to my beloved families

# Part I

Quantum computation and algorithms

# **Chapter 1**

## Introduction

Quantum mechanics is the theoretical framework that underpins our understanding of the physical world at the most fundamental level. Since its discovery in the early 20th century, quantum mechanics has proved to be tremendously successful in predicting physical phenomena at the microscopic scale, providing unprecedented insights ranging from the fundamental particles in nature to the origin of cosmos. Throughout history, our society has held a track record of coupling scientific discoveries with the invention of technologies that reshape everyday life. Quantum mechanics is no exception. Perhaps most pronouncedly, the understanding of the quantum nature of electronic structures in matter played the vital role in giving birth to the entire semiconductor industry, which is in turn responsible for the dawn of the information era, an era in which computation has taken centre stage and revolutionised the world. Broadly speaking, the conventional digital computer is called "classical" since it processes information in the form of logical bits, which omits the possibility of superposition and entanglement allowed by quantum mechanics. As such, despite its almost universal success, when classical computer is used for the task of simulating complex quantum mechanical systems, significant difficulties arise due to the memory requirement for keeping track of the exponentially large state space of the system.

Motivated initially by the problem of simulating physics, Feynman proposed to design and build computers that directly leverage the exponential state space in quantum mechanics [1]. Since this original vision, progress in finding algorithms for future quantum computers has come a long way, and well beyond the domain of quantum simulation alone. Among the most celebrated results are Grover's search algorithm [2] which shows a quadratic advantageous over its classical counter-part and Shor's factoring algorithm [3] which has the potential to break the (to our best knowledge) classically secure RSA cryptosystem. More recently, machine learning has rapidly emerged as an area where quantum algorithms can display dramatic advantages [4–9].

In this thesis, I will present several new results in the more general context of quantum statistical inference, a term used here with two-fold meanings. Firstly, we demonstrate the power of applying quantum computation to statistical models for supervised machine learning with classical datasets. Secondly, we address the notion of causality in quantum information and present an analytical toolkit for inferring causal correlations when the data is itself inherently quantum. In Part I of the thesis, I will start by introducing the basic concepts of quantum mechanics and quantum computation, then move on to review several essential quantum algorithms in Chapter 2. In Chapter 3, I will present a new algorithm for the quantum version of the linear system problem, which shows an advantage over the existing approaches, particularly when the matrix involved is inherently dense. In Part II, we will see that quantum algorithms can be applied to improve the efficiency of supervised learning with Gaussian processes, with a novel application to deep learning. In Part III, we look into quantum causality. I will present results on the geometry of spatial and temporal quantum correlations and the operational role of causality in quantum communication.

## 1.1 Quantum mechanics preliminaries

Here we start by reviewing the fundamental postulates of quantum mechanics and introduce the notation and concepts elementary to the presentation of this thesis. These postulates underline the mathematical framework of quantum physics. Hence they hold a foundational role to future discussions about quantum computation and quantum statistical inference. We will keep our presentation at a basic level. An in-depth discussion of the postulates and a detailed introduction to quantum mechanics is presented in the canonical text of Ref. [10].

#### 1.1.1 The state space

**Postulate 1** Any isolated physical system is associated with a complex vector space with inner product, which is known as the state space (also known as the Hilbert space) of the system. The system is fully described by a unit vector in its state space, which is known as its state vector.

Dirac notation and superposition The state vectors in quantum mechanics are commonly denoted by a "ket", e.g.,  $|\psi\rangle$ . Their Hermitian transpose is denoted by a "bra", so that  $|\psi\rangle^{\dagger} = \langle \psi|$ . The inner product between two state vectors,  $|\psi\rangle$  and  $|\phi\rangle$  is denoted as the "braket",  $\langle \psi | \phi \rangle$ . It follows directly from Postulate 1 that any valid quantum state vector,  $|\psi\rangle$ , satisfies  $\langle \psi | \psi \rangle = 1$ . A quantum state  $|\psi\rangle$  is in a superposition of the states  $\{|\phi_i\rangle\}$  if it can be written as a set of mutually orthogonal states,  $|\psi\rangle = \sum_i \alpha_i |\phi_i\rangle$ , where  $\sum_i |\alpha_i|^2 = 1$ 

#### 1.1.2 Evolution of states

**Postulate 2** The evolution of closed quantum systems is linear, and described by unitary transformations. The state,  $|\psi(t_2)\rangle$  of a quantum system at time  $t_2$  is related to

the state,  $|\psi(t_1)\rangle$ , at an earlier time  $t_1$  via a unitary transformation U that only depends on  $t_1$  and  $t_2$ , so that  $|\psi(t_2)\rangle = U |\psi(t_1)\rangle$ .

**Schrödinger equation** The time-dependent Schrödinger equation describes the time evolution of a closed quantum system,

$$i\hbar \frac{d}{dt} |\psi(t)\rangle = H |\psi(t)\rangle,$$
 (1.1)

where the Hermitian operator H is known as the Hamiltonian. The factor  $\hbar$  is the Planck's constant. We work in units such that  $\hbar = 1$ .

#### 1.1.3 Quantum measurements

**Postulate 3** Quantum measurements are described by a set of measurement operators,  $\{M_m\}$ , where  $\sum_m M_m^\dagger M_m = I$ . If the system is in the quantum state  $|\psi\rangle$  immediately before the measurement, then the probability of the measurement result m occurring is given by  $p(m) = \langle \psi | M_m^\dagger M_m | \psi \rangle$ , and the post-measurement state of the system after is given by  $\frac{M_m |\psi\rangle}{\sqrt{\langle \psi | M_m^\dagger M_m | \psi\rangle}}$ .

**Projective measurements** An important special case of the quantum measurements is the projective measurement. In a projective measurement, the measurement operators are taken to be  $M_m = |\phi_m\rangle \langle \phi_m|$ , where the set of state vectors  $\{|\phi_m\rangle\}$  form an orthonormal basis for the system's Hilbert space. The corresponding probability of an outcome m occurring is then given by  $p(m) = |\langle \psi | \phi_m \rangle|^2$ . Every projective measurement is associated with an observable,  $M = \sum_m |\phi_m\rangle \langle \phi_m|$ . The expectation value of the observable given by  $\langle M \rangle = \langle \psi | M | \psi \rangle$ .

#### 1.1.4 Composite systems

**Postulate 4** The Hilbert space of a composite quantum system is given by the tensor product of the Hilbert spaces of the individual components. For a set of n component systems initialised in the states  $\{|\psi_i\rangle\}_{i=1}^n$ , the state of the composite system is given by  $\bigotimes_{i=1}^n |\psi_i\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes ... \otimes |\psi_n\rangle$ .

**Entanglement** If a composite system has the state as a tensor product of the states of its subsystems, we say the composite system is in a product state. Note that, however, the superposition of product states will not, in general, be in a product state. If the state of a system cannot be written as the tensor product of the states of its subsystems, we say it is entangled. For instance, if  $\langle \psi_1 | \psi_2 \rangle = 0$ , the state  $\frac{1}{\sqrt{2}} (|\psi_1\rangle |\psi_2\rangle + |\psi_2\rangle |\psi_1\rangle)$  is maximally entangled.

## 1.2 Elements of quantum computation

Having reviewed the fundamentals of quantum physics, we now move on to introduce the elementary concepts used in quantum computation. These include the basic unit of quantum computation, the qubit, the important observables given by the Pauli operators, and the unitary gates used to process quantum information.

## **1.2.1** The qubit

The qubit is the most basic non-trivial quantum system. It is also the smallest unit of quantum computation. A single qubit in a "pure" quantum state is a two dimensional complex vector, and can be written as  $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ , where  $|0\rangle = (1,0)^T$  and  $|1\rangle = (0,1)^T$  are, and  $|\alpha|^2 + |\beta|^2 = 1$ . Note that since the probability of a measurement outcome, by postulate 3 is invariant under  $|\psi\rangle \to \mathrm{e}^{i\phi} |\psi\rangle$ , a global phase factor

 $\mathrm{e}^{i\phi}$  is not an observable in quantum mechanics, and we can parameterise the single qubit state as  $|\psi\rangle=\cos(\theta)\,|0\rangle+\mathrm{e}^{i\phi}\sin(\theta)\,|1\rangle$ . As such the qubit can be visualised as a point lying on the surface of a unit sphere, known as the Bloch sphere. The vectors  $\{|0\rangle\,,|1\rangle\}$  forms the Z basis (computational basis) of the single qubit state space. Alternatively, the basis can be chosen as any pair of orthogonal states, e.g. the X basis,  $\left\{|+_x\rangle=\frac{|0\rangle+|1\rangle}{\sqrt{2}},|-_x\rangle=\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right\}$  and the Y basis,  $\left\{|+_y\rangle=\frac{|0\rangle+i|1\rangle}{\sqrt{2}},|-_y\rangle=\frac{|0\rangle-i|1\rangle}{\sqrt{2}}\right\}$ .

#### 1.2.2 Pauli operators

The Pauli operators are observables corresponding to the projectors in the X,Y and Z bases. They are given by  $\sigma_1=X=|+_x\rangle\,\langle +_x|-|-_x\rangle\,\langle -_x|,\,\sigma_2=Y=|+_y\rangle\,\langle +_y|-|-_y\rangle\,\langle -_y|$  and  $\sigma_3=Z=|0\rangle\,\langle 0|-|1\rangle\,\langle 1|$ . We will also use  $\sigma_0=I$  to denote the  $2\times 2$  identity operator. Note that the Pauli operators are traceless, Hermitian and unitary, i.e.  ${\rm Tr}[\sigma_i]=0,\,\sigma_i=\sigma_i^\dagger$  and  $\sigma_i^2=\sigma_0$  for i=1,2,3.

### 1.2.3 Quantum gates

An important part of quantum computation amounts to composing unitary operations acting on collections of qubits. These operation are known as quantum gates. Single-qubit gates correspond to unitary operators acting locally on one qubit, e.g. the Hadamard gate,  $H|j\rangle = \frac{|0\rangle + (-1)^j |1\rangle}{\sqrt{2}}, j \in \{0,1\}$ . In general, the Pauli operators can be used to construct arbitrary single qubit unitary rotations,  $R_{\sigma_i}(\theta) = \mathrm{e}^{i\theta\sigma_i}$  around each respective axis. Many-qubit gates are unitary operations acting on more than one qubit. These operations are capable of generating quantum entanglement, e.g. the controlled-not gate,  $CNOT|i\rangle|j\rangle = |i\rangle|i\oplus j\rangle$ , where  $i,j\in\{0,1\}$  and  $\oplus$  denotes the addition modulo 2.

## 1.3 Statistical ensemble of states

#### 1.3.1 The density matrix

The density matrix is a formalism to describe a probability mixture of pure quantum states. Suppose we are given a system which has a probability  $p_i$  to be in the state  $|\psi_i\rangle$ , we say the system is in a statistical ensemble of pure states,  $\{p_i, |\psi_i\rangle\}$ . The density matrix (or density operator) of the system is then defined as

$$\rho = \sum_{i} p_{i} |\psi_{i}\rangle \langle \psi_{i}|, \qquad (1.2)$$

where  $\sum_i p_i = 1$ . The density matrix is an operator acting on the system's Hilbert space. In the special case when the state of the system is in  $|\psi_j\rangle$  with unit probability, we say the system is in a pure state, and the density matrix is simply given by the projector,  $|\psi_j\rangle\langle\psi_j|$ . Otherwise, we say the system is in a mixed state with a probability distribution  $\{p_i\}$ . The density matrix can be used to calculate the expectation value of any observable M on the system as follows,

$$\langle M \rangle = \sum_{i} p_{i} \langle \psi_{i} | M | \psi_{i} \rangle = \text{Tr}[\rho M].$$
 (1.3)

Since the eigenvalues of the density matrix physically correspond to a probability distribution over the eigenvectors of  $\rho$  which are themselves pure quantum state vectors, the density matrix is necessarily positive semi-definite Hermitian operators with unit trace. On the other hand, any given  $2^N \times 2^N$  matrix that satisfies the Hermitian, positive semi-definite and unit trace properties have the physical interpretation of an N-qubit density matrix. In Part III of this thesis, we will consider a natural extension of the density matrix formalism where the multi-qubit observables on the mixed state are allowed to extend across the temporal domain.

#### 1.3.2 Quantum operations

In the case of a closed system, the evolution of the density matrix translates straightforwardly from the unitary and linear dynamics for pure states, i.e., if a unitary U is applied on the ensemble  $\{p_i, |\psi_i\rangle\}$ , the corresponding density matrix transforms as  $\rho \to U \rho U^{\dagger}$ . In this section, we describe the general quantum operation on open quantum systems.

Suppose now an initial system described by  $\rho$  is coupled with an environment described (without loss of generality) by the pure state  $\rho_e = |e_0\rangle \langle e_0|$ . Since the joint system,  $\rho \otimes \rho_e$  is now a closed system, its general dynamics can be described by the unitary transformation,  $U(\rho \otimes |e_0\rangle \langle e_0|)U^{\dagger}$ . The resultant transformation on the initial system,  $\varepsilon(\rho)$  is then given by a partial trace over the environment,

$$\varepsilon(\rho) = \operatorname{Tr}_{e} \left[ U(\rho \otimes |e_{0}\rangle \langle e_{0}|) U^{\dagger} \right]$$

$$= \sum_{k} \langle e_{k} | U(\rho \otimes |e_{0}\rangle \langle e_{0}|) U^{\dagger} |e_{k}\rangle$$

$$= \sum_{k} E_{k} \rho E_{k}^{\dagger}, \qquad (1.4)$$

where  $|e_k\rangle$  denotes an orthonormal basis for the environment's state space. We have defined  $E_k = \langle e_k | U | e_0 \rangle$  which are known as the Kraus operators of the quantum operation  $\varepsilon$ . Trace preserving quantum operations are also known as quantum channels. A channel mathematically corresponds to a completely positive trace preserving (CPTP) map. In this case, the Kraus operators satisfy the completeness relation,  $\sum_k E_k^\dagger E_k = I$ . In general, when measurements are involved and extra information is obtained about the process, the quantum operation is not necessarily trace preserving, and the Kraus operators instead satisfy  $\sum_k E_k^\dagger E_k \leq I$ . The trace preserving cases (quantum channels) will be more relevant to the materials presented in Part III of this thesis.

# **Chapter 2**

# **Essential quantum algorithms**

In this chapter, I introduce some essential quantum algorithms which will serve as building blocks later in the thesis. We start with the more basic algorithms: The quantum Fourier transform which is regarded as the root of quantum advantage in many higher-level algorithms, quantum phase estimation which approximately computes the eigenvalues of a Hamiltonian matrix in a superposition, and quantum Hamiltonian simulation which amounts to constructing a unitary operator corresponding to the time evolution under a Hamiltonian. We then review a quantum algorithm that combines these basic techniques and provides an advantage in solving systems of linear equations under a quantum formulation of the problem.

## 2.1 Basic quantum algorithms

#### 2.1.1 Quantum Fourier transform

The quantum Fourier transform (QFT) is the foundation of many quantum algorithms, including the celebrated quantum factoring algorithm [11]. It can be seen as the quantum analog of the discrete Fourier transform in classical computation. Here we briefly introduce QFT and describe the unitary operator for its implementation. A detailed

description can be found in all canonical texts of quantum information, such as Ref. [10, 12].

The normalised discrete Fourier transform of a vector  $\mathbf{v}=(v_1...v_n)^T$  is given by the vector  $\hat{\mathbf{v}}$  with entries,  $\hat{v}_y=\frac{1}{\sqrt{n}}\sum_{x=1}^n v_x\mathrm{e}^{-\frac{2\pi xyi}{n}}$ . For  $v_x$  with periodicity P, such that  $v_x=v_{x+P}$ , we have

$$\hat{v}_{y} = \frac{1}{\sqrt{n}} \left( \sum_{x=1}^{P} \sum_{m=0}^{\lfloor nP^{-1} \rfloor - 1} v_{x+mP} e^{-\frac{2\pi(x+mP)yi}{n}} + \sum_{x=\lfloor nP^{-1} \rfloor + 1}^{n} v_{x} e^{-\frac{2\pi xyi}{n}} \right) 
= \frac{1}{\sqrt{n}} \left( \sum_{m=0}^{\lfloor nP^{-1} \rfloor - 1} e^{-\frac{2\pi mPyi}{n}} \sum_{x=1}^{P} v_{x+mP} e^{-\frac{2\pi xyi}{n}} + \sum_{x=\lfloor nP^{-1} \rfloor + 1}^{n} v_{x} e^{-\frac{2\pi xyi}{n}} \right).$$
(2.1)

Note that for  $n \gg P$ , the above expression only consists of small oscillations around zero unless Py is an integer. Therefore the only surviving terms correspond to y being an integer multiple of the frequency.

The QFT is the discrete Fourier transform applied to quantum state vectors, and it is implemented by the unitary operator,

$$U_{QFT} = \frac{1}{\sqrt{n}} \sum_{y=1}^{n} \sum_{x=1}^{n} e^{-\frac{2\pi xy^{i}}{n}} |y\rangle \langle x|.$$
 (2.2)

One can easily verify the above indeed corresponds to the discrete Fourier transform of a quantum state by applying it to an arbitrary state vector  $|\mathbf{v}\rangle$ ,

$$\langle z|U_{QFT}|\mathbf{v}\rangle = \langle z|\frac{1}{\sqrt{n}}\sum_{y=1}^{n}\sum_{x=1}^{n}e^{-\frac{2\pi xyi}{n}}|y\rangle\langle x|\mathbf{v}\rangle$$

$$=\frac{1}{\sqrt{n}}\sum_{x=1}^{n}\langle x|\mathbf{v}\rangle e^{-\frac{2\pi xyi}{n}}$$

$$=\langle z|\hat{\mathbf{v}}\rangle. \tag{2.3}$$

Given access to a set of basic unitary gates and n qubits, a quantum computer can

perform the discrete Fourier transform on  $2^n$  amplitudes with only  $\mathcal{O}(n^2)$  Hadamard and controlled phase gates, providing an exponential advantage over the classical counterpart that takes  $\mathcal{O}(n2^n)$  gates [10]. It is worth noting that an improved version of the QFT presented in Ref. [13] has further suppressed the cost to  $\mathcal{O}(n \log n)$ .

#### 2.1.2 Quantum phase estimation

The Quantum phase estimation, first introduced in Ref. [14] is a quantum algorithm that takes as input an eigenvector of a unitary operator and estimates the corresponding eigenvalue to a certain additive error. It is the root of the quantum advantage in many machine learning and linear algebraic applications. Here we define the quantum phase estimation algorithm for future reference. A detailed description of its procedures can be found for example in section 5.2 of Ref. [10].

Let the unitary operator  $U \in \mathbb{C}^{n \times n}$  have eigenvectors  $\{|v_j\rangle\}$  with corresponding eigenvalues  $\{e^{i\theta_j}\}$ , such that  $|v_j\rangle = e^{i\theta_j}\,|v_j\rangle$ , where  $\theta_j \in [-\pi,\pi]$  for  $j \in [n]$ . Further define the precision parameter  $\delta$  to denote an additive error. Given an oracle for implementing  $U^l$  for  $l = \mathcal{O}(1/\delta)$ , the quantum phase estimation algorithm performs the following transformation,

$$\sum_{j \in [n]} \alpha_j |v_j\rangle \to \sum_{j \in [n]} \alpha_j |v_j\rangle |\overline{\theta}_j\rangle, \qquad (2.4)$$

such that  $|\overline{\theta_j} - \theta_j| \le \delta$  for all  $j \in [n]$  with probability 1 - 1/poly(n) in time that scales as  $\mathcal{O}(T_U \log(n)/\delta)$ , where  $T_U$  denotes the time required to implement U.

#### 2.1.3 Black-box Hamiltonian simulation

Given a Hermitian Hamiltonian operator  $H \in \mathbb{C}^{n \times n}$ , the black-box access to the matrix elements  $H_{jk}$  is an oracle  $O_H$  that allows for the operation,

$$O_H |j,k\rangle |z\rangle \to |j,k\rangle |z \oplus H_{ik}\rangle,$$
 (2.5)

for an arbitrary input  $|z\rangle$ , where  $j,k\in\{1,2,...,n\}$  and  $\oplus$  denotes the bitwise addition modulo two operation. The time evolution of a quantum state  $|\psi(t)\rangle$  under H is described by the time-dependent Schrödinger equation,

$$i\frac{d}{dt}|\psi(t)\rangle = H|\psi(t)\rangle.$$
 (2.6)

The solution is given by  $|\psi(t)\rangle = U(H,t) |\psi(0)\rangle$ , where the unitary operator  $U(H,t) = \exp(-iHt)$ . Black-box Hamiltonian simulation amounts to constructing a quantum circuit that implements U(H,t) given access to the oracle  $O_H$ .

In the general case, the results of [15] shows that the black-box Hamiltonian simulation can be performed in time  $\mathcal{O}\left(n^{2/3} \cdot \operatorname{polylog}(n)/\delta_h^{1/3}\right)$  with an  $\delta_h$  error in the trace distance using a method based on discrete time quantum walks [16]. Empirical results of [15] suggested black-box Hamiltonian simulation can be implemented in time  $\mathcal{O}\left(\sqrt{n} \cdot \operatorname{polylog}(n)/\delta_h^{1/2}\right)$  for several classes of Hamiltonians. However, the  $\tilde{\mathcal{O}}(\sqrt{n})$  runtime is known to not hold in the worst case. The notation  $\tilde{\mathcal{O}}(.)$  is used here to suppress slower growing factors in the runtime scaling. In special cases, properties of H such as sparsity can be leveraged to implement Hamiltonian simulation more efficiently. It was shown in Ref. [17] that combing techniques from quantum walk [16] and fractional query simulation [18], Hamiltonian simulation on an s-sparse matrix (that is, the maximum number of non-zero entries on any rows or columns is s) can be performed in time  $\tilde{\mathcal{O}}(s \cdot \operatorname{polylog}(n)/\delta_h^{1/2})$ .

It is worth mentioning that the black-box model is not the uniquely interesting setting to consider. Other important models include the quantum signal processor [19] and the density matrix encoding mode [20,21]. Detailed descriptions of quantum Hamiltonian simulation algorithms and a comprehensive review on this subject is beyond the scope of this thesis. Interested readers are referred to the Chapters 25 and 26 of Ref. [22].

## 2.2 Quantum linear system algorithm

Solving a linear system of equations is a problem that appears in many disciplines across science and engineering. Given a set of n linear equations with n unknown variables, we wish to find the n dimensional vector x which satisfies Ax = b, where A and b a are known  $n \times n$  dimensional matrix and a known n dimensional vector respectively. The solution of the linear system can be written as  $x = A^{-1}b$  for an invertible matrix A. In special cases, A has convenient properties such as sparsity, of which one can take advantage and compute  $A^{-1}$  in time proportional to n with the conjugate gradient method [23]. In general, the best known classical method for matrix inversion scales as  $\mathcal{O}(n^{2.373})$ , with the optimised CW-like algorithms [24, 25]. However, this sub-cubic scaling is practically difficult to achieve. A more typical implementation amounts to using the Cholesky decomposition which has a runtime that scales as  $\mathcal{O}(n^3)$  for dense matrices. In modern statistical inference and machine learning applications, matrix inversion presents a computational bottleneck when the dimensionality n of the underlying problem grows. Recent discoveries in quantum algorithms have shown promises for a more efficient solution of high-dimensional linear systems. Given the importance and generality of the problem, quantum linear system algorithms may manifest as the cornerstone of quantum advantage in many use cases. In this section, we review some of the earlier progress in this subject. In the next chapter, we will present a new result along the same line of research.

#### 2.2.1 Quantum formulation of linear systems

Let  $A \in \mathbb{R}^{n \times n}$  be a Hermitian matrix, with  $||A||_* \leq 1$ . Here  $||.||_*$  denotes the spectral norm which corresponds to the largest absolute value of the eigenvalues in the case of Hermitian matrices. Let  $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$ , such that  $A\mathbf{x} = \mathbf{b}$ . We define the following quantum formulation of the linear system problem:

Given access to the elements of A and an input quantum state vector  $|\mathbf{b}\rangle$  of  $\log n$  qubits which encodes the entries in  $\mathbf{b}$  as

$$|\mathbf{b}\rangle = \frac{\sum_{j} b_{j} |j\rangle}{\|\sum_{j} b_{j} |j\rangle\|_{2}},\tag{2.7}$$

the quantum linear system problem amounts to finding the state vector  $|\mathbf{x}\rangle$  of  $\log n$  qubits which encodes the entries in solution vector  $\mathbf{x}$  as

$$|\mathbf{x}\rangle = \frac{\sum_{j} x_{j} |j\rangle}{\|\sum_{j} x_{j} |j\rangle\|_{2}}.$$
(2.8)

#### **Remarks:**

- Note that the input and output of the quantum linear system problem are both quantum states. Therefore the initial state preparation and final solution readout procedures will need to be explicitly addressed for any applications that have classical vectors as inputs and outputs. This point has been discussed in Ref. [26] and will be revisited later in this thesis.
- Defining A to be a Hermitian matrix is in fact without loss of generality. As pointed
  out in Ref. [27], a general matrix M can be embedded into a Hermitian matrix with
  a constant memory overhead by constructing a block-wise anti-diagonal matrix A

as follows,

$$A = \begin{pmatrix} 0 & M^{\dagger} \\ M & 0 \end{pmatrix}. \tag{2.9}$$

• The requirement on bounded spectral norm is not a strong restriction in practice since it can often be satisfied with a suitable choice of normalisation factor.

#### 2.2.2 The HHL algorithm

In the breakthrough work of Ref. [27], Harrow, Hassidim and Lloyd (HHL) introduced the first quantum linear system algorithm (QLSA) that computes the quantum state  $|\mathbf{x}\rangle = |A^{-1}\mathbf{b}\rangle$  which corresponds to the solution of the linear system  $A\mathbf{x} = \mathbf{b}$  in time  $\mathcal{O}(\text{polylog}(n))$  for a sparse and well-conditioned A. In this section, we review this seminal algorithm and discuss its implications. The procedure of the original quantum linear systems solver provided in Ref. [27] can be summarised in the following five steps:

1. To start with, prepare a quantum state  $|\mathbf{b}\rangle$  which encodes the vector  $\mathbf{b} \in \mathbb{R}^n$  as  $|\mathbf{b}\rangle = (\mathbf{b}^T\mathbf{b})^{-1/2}\sum_{i=0}^{n-1}b_i\,|i\rangle$ . Then append to  $|\mathbf{b}\rangle$  an ancillary register in a superposition state  $\frac{1}{\sqrt{T}}\sum_{\tau=0}^{T}|\tau\rangle$ . The time period T is chosen to be some large value as required in the variant of phase-estimation described in Ref. [28], so that after Step 1 we have the quantum state,

$$|\phi_1\rangle = \frac{1}{\sqrt{\mathbf{b}^T \mathbf{b}}} \frac{1}{\sqrt{T}} \sum_{i=0}^{n-1} \sum_{\tau=0}^{T} b_i |i\rangle |\tau\rangle.$$
 (2.10)

2. Perform Hamiltonian simulation treating the matrix A as the Hamiltonian at time  $\tau$ . Apply the resultant controlled unitary operation to  $|\mathbf{b}\rangle$  using techniques described

in Ref. [29]. By writing  $|\mathbf{b}\rangle$  in the eigenbasis of A after evolution, we obtain the state,

$$|\phi_2\rangle = \frac{1}{\sqrt{\mathbf{b}^T \mathbf{b}}} \frac{1}{\sqrt{T}} \sum_{i=0}^{n-1} \sum_{\tau=0}^{T-1} |\tau\rangle e^{i\lambda_i t_0 \tau/T} \beta_i |\mu_i\rangle, \qquad (2.11)$$

where  $\lambda_i$  are the eigenvalues and  $|\mu_i\rangle$  are the eigenvectors of A. The complex numbers  $\beta_i$  are the probability amplitudes associated with  $|\mu_i\rangle$ . For some precision parameter  $\epsilon$  which will feature as an additive error of the final result in the trace norm, we choose the time scale  $t_0 = \mathcal{O}(\kappa/\epsilon)$  where  $\kappa$  denotes the condition number, the ratio between the largest and the smallest eigenvalues of A.

3. Complete phase estimation [14,28] by applying the quantum Fourier transform (QFT) to the first register in  $|\phi_2\rangle$ , which leads to

$$|\phi_3\rangle = \frac{1}{\sqrt{\mathbf{b}^T \mathbf{b}}} \sum_{i=0}^{n-1} \beta_i |t\bar{\lambda}_i\rangle |\mu_i\rangle,$$
 (2.12)

where the first register now stores the estimated eigenvalues  $\bar{\lambda}_i$  up to a constant multiplicative factor t.

4. Introduce another ancillary qubit and perform a controlled rotation on it based on the value in the first register, and obtain the extended state

$$|\phi_4\rangle = \frac{1}{\sqrt{\mathbf{b}^T \mathbf{b}}} \sum_{i=0}^{n-1} \beta_i |t\bar{\lambda}_i\rangle |\mu_i\rangle \left(\sqrt{1 - \frac{c_\lambda^2}{\bar{\lambda}_i^2}} |0\rangle + \frac{c_\lambda}{\bar{\lambda}_i} |1\rangle\right). \tag{2.13}$$

Here the constant  $c_{\lambda}$  is chosen such that the resultant probability amplitude is bounded by unity.

5. Reverse the phase estimation step on the first register to uncompute  $|t\bar{\lambda}_i\rangle$ . Measure the final ancillary qubit. Conditioned on obtaining  $|1\rangle$  as the measurement result,

an approximated solution of  $A | \mathbf{x} \rangle = | \mathbf{b} \rangle$  is obtained,

$$|\bar{\mathbf{x}}\rangle = |\phi_5\rangle = \frac{1}{\sqrt{\mathbf{b}^T \mathbf{b}}} \sum_{i=0}^{n-1} \frac{\beta_i}{\bar{\lambda}_i} |\mu_i\rangle.$$
 (2.14)

For a precision parameter  $\epsilon$ , the additive error in the trace norm of the output state is bounded as  $\| |\bar{\mathbf{x}}\rangle - |\mathbf{x}\rangle \| \le \epsilon$ . Note that a post-selection of measurement outcomes is involved in this final step, and as a consequence multiple repetitions of the procedure may be needed in order to successfully obtain the desired outcome.

Runtime and errors The required Hamiltonian simulation subroutine runs nearly linearly with the sparsity, s, with the black-box Hamiltonian simulation technique of [17]. The time scale parameter  $t_0$  of phase estimation is chosen to be  $\mathcal{O}(\kappa/\epsilon)$  to ensure the desired precision. Furthermore,  $\mathcal{O}(\kappa)$  repetitions of the procedure are needed to obtain the desired outcome on the final measurement of the ancillary qubit, making use of the amplitude amplification based techniques of [30]. From the above rough account, the total runtime scales as  $\tilde{O}(\log(n)\kappa^2s^2/\epsilon)$ . A detailed error and runtime analysis can be found in the supplementary material of [27].

Potential caveats The quantum linear algorithm described above can potentially provide a promising exponential speed-up. However, one needs to apply it with care. As Aaronson accurately described in Ref. [26], there are four potential caveats that need particular care in any applications: (1) The time consumption of preparing  $|\mathbf{b}\rangle$  encoding b needs to be taken into account; (2) the matrix A has to be robustly invertible, meaning that the condition number  $\kappa$  needs to grow at most polylogarithmically in n in order to retain a polylogarithmic overall runtime; (3) one also needs to address the sparsity contribution to the total runtime, since the general phase estimation sub-routine costs

time polynomial in s; (4) although the output of QLSA is the state  $|\mathbf{x}\rangle$ , there is no efficient procedure to extract every entry of  $\mathbf{x}$ . The quantum advantage only presents when the matter of practical interest does not require the full  $\mathbf{x}$  but requires only information accessible with a few copies of  $|\mathbf{x}\rangle$ . For instance, if a known Hermitian matrix M is of interest, one can efficiently estimate quantities such as  $\langle \mathbf{x} | M | \mathbf{x} \rangle$ , since this amounts to the expectation value of the observable M on  $|\mathbf{x}\rangle$ .

**Developments** There have been several improvements to the QLSA since the original HHL proposal that have improved the running time to linear in the condition number  $\kappa$  and the sparsity s, and to poly-logarithmic in the precision parameter  $\epsilon$  [30,31]. The work of Ref. [32] further introduced pre-conditioning for the QLSA and extended its applicability. In the next chapter, we build upon this line of research and present a linear system algorithm that circumvents the expensive Hamiltonian simulation step and has a provably better performance than the existing algorithms when applied to linear systems with dense matrices.

# **Chapter 3**

# Quantum dense linear system algorithm

In this chapter, I present an alternative approach to solving the quantum linear systems problem, which is based on a quantum subroutine for singular value estimation (SVE). The SVE-based linear system algorithm, introduced in Ref. [33] has a runtime scaling of  $\mathcal{O}\left(\kappa^2\|A\|_F\cdot \operatorname{polylog}(n)/\epsilon\right)$  for an  $n\times n$  dimensional Hermitian matrix A with a Frobenius norm  $\|A\|_F$  and condition number  $\kappa$ . As before,  $\epsilon$  is the precision parameter defined by the desired output error in the trace norm. Unlike the HHL algorithm, the SVE-based method does not require performing Hamiltonian simulation on A, making it advantageous particularly when A is dense. Therefore, we refer to it as a quantum dense linear system (QDLS) algorithm. An important component of the QDLS algorithm is the quantum singular value estimation (QSVE), introduced in [34]. It makes use of a memory model that supports efficient preparation of states which correspond to the row vectors of A and the vector of the row Euclidean norms of A. We will start by introducing this memory model, followed by an outline of the quantum SVE algorithm. Finally, we put the components together and present the QDLS algorithm.

## 3.1 Memory model

In order to keep our description of the memory model general, we consider a rectangular matrix  $A \in \mathbb{R}^{m \times n}$ . Instead of using a model that allows for black-box access to the matrix elements, here we work in a model that realises a data structure which satisfies the following properties:

• Given access to the data structure, a quantum computer can perform the following mappings in  $\mathcal{O}(\operatorname{polylog}(mn))$  time.

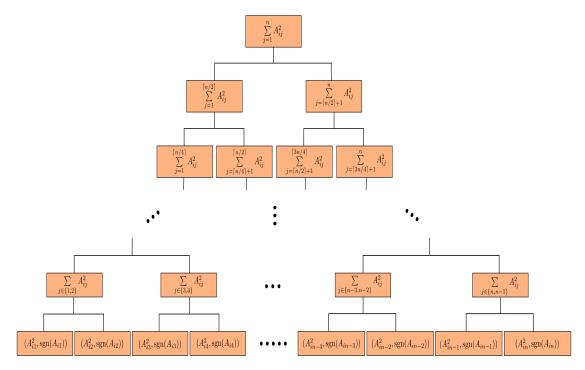
$$U_{\mathcal{M}}: |i\rangle |0\rangle \to |i, \vec{A}_i\rangle = \frac{1}{\|\vec{A}_i\|} \sum_{j=1}^n A_{ij} |i, j\rangle,$$

$$U_{\mathcal{N}}: |0\rangle |j\rangle \to |\vec{A}_F, j\rangle = \frac{1}{\|A\|_F} \sum_{i=1}^m \|\vec{A}_i\| |i, j\rangle,$$
(3.1)

where  $\vec{A}_i \in \mathbb{R}^n$  are the row vectors of the matrix A and  $\vec{A}_F \in \mathbb{R}^m$  is a vector of the Euclidean norms of the rows, i.e.  $(\vec{A}_F)_i = \|A_i\|$ .

• The time needed to store a new entry  $A_{ij}$  is in  $\mathcal{O}(\log^2(mn))$ . The data structure has size  $\mathcal{O}(w\log(mn))$  with w denoting the number of non-zero entries in A.

One way to construct the data structure as described with the above-desired properties is based on a size m array of binary trees, where each tree contains no more than n leaves. The leaves store the squared values of the corresponding matrix element  $|A_{ij}|^2$ , together with the sign,  $sgn(A_{ij})$ . Each internal node of a binary tree stores the summation of the values in the subtree rooted at it, as such the root of the  $i^{th}$  tree stores  $\|\vec{A}_i\|^2$ ,  $i \in [m]$ . In order to access row Frobenius norm vectors, one merely need to construct an additional binary tree, in which the  $i^{th}$  leaf contains  $\|\vec{A}_i\|^2$ . We show the schematic diagram to demonstrate one of the trees in Figure 3.1. More details about the realisation of this data structure can be found in Ref. [34] and [35].



**Figure 3.1:** The schematic diagram for one out of the (m+1) binary trees in the data structure used to store the matrix A. The depth of the tree is in  $\mathcal{O}(\log n)$ .

#### 3.2 Quantum singular value estimation

Having stated the memory model, we are now in the position to outline the quantum singular value estimation (QSVE) subroutine. The QSVE can be seen as an extension of phase estimation to non-unitary matrices. Let the matrix  $A \in \mathbb{R}^{m \times n}$  have the singular value decomposition

$$A = \sum_{i} \sigma_{i} \mathbf{u_{i}} \mathbf{v_{i}}^{\dagger}, \tag{3.2}$$

where  $\mathbf{u_i}$  and  $\mathbf{v_i}$  are the left and right singular vectors respectively, and  $\sigma_i$  are the corresponding singular values. Since the left and the right singular vectors each form a complete set of orthonormal bases, an arbitrary input state can be written as the superposition,  $\sum_i \alpha_i |\mathbf{v}_i\rangle$ , where  $|\mathbf{v}_i\rangle$  is a quantum state vector which encodes  $\mathbf{v}_i$ . The

quantum SVE subroutine performs the following mapping,

$$\sum_{i} \alpha_{i} |\mathbf{v}_{i}\rangle \to \sum_{i} \alpha_{i} |\mathbf{v}_{i}\rangle |\overline{\sigma}_{i}\rangle, \qquad (3.3)$$

where  $|\overline{\sigma_i}\rangle$  is a state vector encoding the estimates for the singular values of A with a precision  $\delta$ , so that  $|\overline{\sigma_i} - \sigma_i| \leq \delta$  for all i.

An algorithm for QSVE with a runtime of  $\tilde{\mathcal{O}}(\|A\|_F/\delta)$  was introduced in Ref. [34], and applied to quantum recommendation systems. It is the main tool required for the quantum dense linear system (QDLS) algorithm [33] to be presented in this chapter. In this section, we first give an overview of QSVE with essential mathematical background and high-level intuition. Then we outline the procedures of QSVE. Finally, we provide by a brief analysis of the algorithm, while a more thorough analysis can be found in Ref. [34, 35].

#### 3.2.1 Overview

The QSVE algorithm is based on the idea of quantum walks. It makes use of the connection between the singular values  $\sigma_i$  of the matrix  $A = \sum_i \sigma_i \mathbf{u_i} \mathbf{v_i}^{\dagger}$  and the principal angles,  $\theta_i$  between certain associated subspaces. There exist a factorisation,

$$\frac{A}{\|A\|_F} = \mathcal{M}^{\dagger} \mathcal{N}, \tag{3.4}$$

where  $\mathcal{M} \in \mathbb{R}^{mn \times m}$  and  $\mathcal{N} \in \mathbb{R}^{mn \times n}$  are isometries with column spaces denoted  $\mathcal{C}_{\mathcal{M}}$  and  $\mathcal{C}_{\mathcal{N}}$  respectively. The isometries  $\mathcal{M}$  act on an arbitrary input state vector  $|\alpha\rangle$  as a mapping that appends a register which stores the row vectors  $\vec{A}_i$ , such that

$$\mathcal{M}: |\alpha\rangle = \sum_{i=1}^{m} \alpha_i |i\rangle \to \sum_{i=1}^{m} \alpha_i |i, \vec{A}_i\rangle = |\mathcal{M}\alpha\rangle.$$
 (3.5)

similarly, the isometries  $\mathcal{N}$  act on an arbitrary input state vector  $|\alpha\rangle$  as a mapping that appends a register which stores the vector  $\vec{A_F}$ , in which the entries are the row vector Euclidean norms  $||\vec{A_i}||$ , such that

$$\mathcal{N}: |\alpha\rangle = \sum_{j=1}^{n} \alpha_j |j\rangle \to \sum_{j=1}^{n} \alpha_j |\vec{A_F}, j\rangle = |\mathcal{N}\alpha\rangle.$$
 (3.6)

The above maps can be efficiently implemented given the memory model as described in Section 3.1. The factorisation Eq. 3.4 then follows directly from the amplitude encodings of  $\vec{A}_i$  and  $\vec{A}_F$ , as we have

$$|i, \vec{A}_i\rangle = \frac{1}{\|\vec{A}_i\|} \sum_{j=1}^n A_{ij} |i, j\rangle,$$
  
 $|\vec{A}_F, j\rangle = \frac{1}{\|A\|_F} \sum_{i=1}^m \|\vec{A}_i\| |i, j\rangle.$  (3.7)

Taking the inner product of the above equations leads to

$$(\mathcal{M}^{\dagger}\mathcal{N})_{ij} = \langle i, \vec{A}_i | \vec{A}_F, j \rangle = \frac{A_{ij}}{\|A\|_F}.$$
 (3.8)

A similar calculation shows that  $\mathcal{M}$  and  $\mathcal{N}$  have orthonormal columns and thus  $\mathcal{M}^{\dagger}\mathcal{M} = I_m$  and  $\mathcal{N}^{\dagger}\mathcal{N} = I_n$ . The singular values of the normalised matrix  $\frac{A}{\|A\|_F}$  have a one-to-one correspondence to the principal angles between the subspaces  $\mathcal{C}_{\mathcal{M}}$  and  $\mathcal{C}_{\mathcal{N}}$ . The efficiency of QSVE relies on the fact that given the matrix A stored in a data structure as described in Section 3.1, the following unitary operator W can be implemented efficiently,

$$W = (2\mathcal{M}\mathcal{M}^{\dagger} - I_{mn})(2\mathcal{N}\mathcal{N}^{\dagger} - I_{mn}), \tag{3.9}$$

where  $I_{mn}$  denotes the  $(mn) \times (mn)$  identity matrix. Note the fact that W acts on  $|\mathcal{N}\mathbf{v}_i\rangle$  as a rotation in the plane of  $\{\mathcal{M}\mathbf{u}_i, \mathcal{N}\mathbf{v}_i\}$  by  $\theta_i$ . Hence the two dimensional

sub-space spanned by  $\{\mathcal{M}\mathbf{u}_i, \mathcal{N}\mathbf{v}_i\}$  is invariant under W. The eigenvectors of W,  $|w_i^{\pm}\rangle$ , therefore have corresponding eigenvalues  $\exp(\pm i\theta_i)$ . We can write in the eigenbasis of W,  $|\mathcal{N}\mathbf{v}_i\rangle = \omega_i^+|w_i^+\rangle + \omega_i^-|w_i^-\rangle$ , with  $|\omega_i^-|^2 + |\omega_i^+|^2 = 1$ , and phase estimation can be performed to estimate  $\pm \theta_i$ . Finally the singular values of A,  $\{\sigma_i\}$  are computed via  $\sigma_i = \cos(\theta_i/2) \|A\|_F$ , a relation which will be shown in Section 3.2.3.

Intuitively the operators  $2\mathcal{M}\mathcal{M}^{\dagger} - I_{mn}$  and  $2\mathcal{N}\mathcal{N}^{\dagger} - I_{mn}$  can be seen as a generalisation of the Grover diffusion operator [2]. They act on the subspaces  $\mathcal{C}_{\mathcal{M}}$  and  $\mathcal{C}_{\mathcal{N}}$  respectively as reflection operators. Thus applying W represents two sequential reflections, on the  $\mathcal{C}_{\mathcal{N}}$  and then the  $\mathcal{C}_{\mathcal{M}}$  subspaces. As such W has the interpretation of taking a step in the bipartite quantum walk as formulated in Ref. [16] with the discriminant matrix given by our normalised target matrix  $\frac{A}{\|A\|_F}$ . The connections between quantum walks and the eigenvalues of the discriminant matrix are well-known in the literature and have been used in numerous previous works [16, 36, 37].

#### 3.2.2 Procedures

Having introduced the mathematical background, we are now in the position to outline the procedures of QSVE.

- 1. Create an arbitrary input state  $|\alpha\rangle = \sum_i \alpha_i |\mathbf{v}_i\rangle$ , where  $|\mathbf{v}_i\rangle$  encodes the  $i^{th}$  normalised left singular vector of A.
- 2. Append a register with size  $\log m$ ,  $|0^{\lceil \log m \rceil}\rangle$ . Query the data structure to apply  $U_N$ , and create the state

$$|\mathcal{N}\alpha\rangle = \sum_{i} \alpha_{i} |\mathcal{N}v_{i}\rangle = \sum_{i} \alpha_{i} (\omega_{i}^{+} |w_{i}^{+}\rangle + \omega_{i}^{-} |w_{i}^{-}\rangle). \tag{3.10}$$

- 3. Perform phase estimation [14] with precision  $2\delta > 0$  on input  $|\mathcal{N}\alpha\rangle$  for  $W = (2\mathcal{M}\mathcal{M}^{\dagger} I_{mn})(2\mathcal{N}\mathcal{N}^{\dagger} I_{mn})$  and obtain  $\sum_{i} \alpha_{i}(\omega_{i}^{+} | w_{i}^{+}, \overline{\theta}_{i}\rangle + \omega_{i}^{-} | w_{i}^{-}, -\overline{\theta}_{i}\rangle)$ , where  $\overline{\theta_{i}}$  is the estimated phase  $\theta_{i}$ , such that  $|\overline{\theta}_{i} \theta_{i}| \leq 2\delta$ .
- 4. On the output register of phase estimation compute  $\overline{\sigma}_i = \cos(\pm \overline{\theta_i}/2)||A||_F$  to obtain  $\sum_i \alpha_i(\omega_i^+ |w_i^+\rangle + \omega_i^- |w_i^-\rangle) |\overline{\sigma}_i\rangle$ .
- 5. Apply the reversed computation of Step 2 to obtain

$$\sum_{i} \alpha_{i} |\mathbf{v}_{i}\rangle |\overline{\sigma}_{i}\rangle. \tag{3.11}$$

#### 3.2.3 Brief analysis

Stated in a compact manner, the correctness and efficiency of the QSVE algorithm rely on the following:

- The mappings,  $|\alpha\rangle \to |\mathcal{M}\alpha\rangle$  and  $|\alpha\rangle \to |\mathcal{N}\alpha\rangle$  can be performed in time  $\mathcal{O}(\operatorname{polylog}(mn))$ , where the isometries  $\mathcal{M}$  and  $\mathcal{N}$  satisfy  $\mathcal{M}^{\dagger}\mathcal{M} = I_m$  and  $\mathcal{N}^{\dagger}\mathcal{N} = I_n$  and the factorisation  $A/\|A\|_F = \mathcal{M}^{\dagger}\mathcal{N}$ .
- The reflection operators  $(2\mathcal{M}\mathcal{M}^{\dagger} I_{mn})$  and  $(2\mathcal{N}\mathcal{N}^{\dagger} I_{mn})$ , hence the unitary  $W = (2\mathcal{M}\mathcal{M}^{\dagger} I_{mn})(2\mathcal{N}\mathcal{N}^{\dagger} I_{mn})$  can be implemented in time  $\mathcal{O}$  (polylog(mn)). The unitary W acts on  $|\mathcal{N}v_i\rangle$  as a rotation in the plane of  $\{\mathcal{M}\mathbf{u}_i, \mathcal{N}\mathbf{v}_i\}$  by  $\theta_i$ , such that  $\sigma_i = \cos\frac{\theta_i}{2}\|A\|_F$ , where  $\sigma_i$  is the  $i^{th}$  singular value for A.

As previously shown, the first two items in the above listing are guaranteed by applying the appropriate data structure in Section 3.1. It remains to show the relationship between the eigenvalues of W and the singular values of A. We start by considering the action of

W as follows,

$$W |\mathcal{N}v_{i}\rangle = (2\mathcal{M}\mathcal{M}^{\dagger} - I_{mn})(2\mathcal{N}\mathcal{N}^{\dagger} - I_{mn}) |\mathcal{N}v_{i}\rangle$$

$$= (2\mathcal{M}\mathcal{M}^{\dagger} - I_{mn}) |\mathcal{N}v_{i}\rangle$$

$$= 2\mathcal{M}\frac{A}{\|A\|_{F}} |\mathbf{v}_{i}\rangle - |\mathcal{N}v_{i}\rangle.$$
(3.12)

Using the singular value decomposition  $A = \sum_i \sigma_i |\mathbf{u}_i\rangle \langle \mathbf{v}_i|$ , and the fact that the right singular vectors  $\{\mathbf{v}_i\}$  are mutually orthonormal, we have

$$W |\mathcal{N}v_i\rangle = \frac{2\sigma_i}{\|A\|_F} |\mathcal{M}u_i\rangle - |\mathcal{N}v_i\rangle.$$
 (3.13)

It is now visible that W has rotated  $|\mathcal{N}v_i\rangle$  in the plane of  $\{\mathcal{M}\mathbf{u}_i, \mathcal{N}\mathbf{v}_i\}$  by an angle  $\theta_i$ , such that

$$\cos \theta_{i} = \langle \mathcal{N}v_{i} | W | \mathcal{N}v_{i} \rangle$$

$$= \frac{2\sigma_{i}}{\|A\|_{F}^{2}} \langle v_{i} | A^{\dagger} | \mathbf{u}_{i} \rangle - 1$$

$$= \frac{2\sigma_{i}^{2}}{\|A\|_{F}^{2}} - 1. \tag{3.14}$$

Note that we have used the fact  $(2\mathcal{M}\mathcal{M}^{\dagger} - I_{mn})$  is a reflection in  $|\mathcal{M}u_i\rangle$  and that  $A^{\dagger} = \mathcal{N}^{\dagger}\mathcal{M} = \sum_{i} \sigma_i |\mathbf{v}_i\rangle \langle u_i|$ . Hence we have established the angle between  $|\mathcal{N}v_i\rangle$  and  $|\mathcal{M}u_i\rangle$  is  $\frac{\theta_i}{2}$ , which amounts to half of the total rotation angle. Comparing the last line of Eq. 3.14 with the half-angle formula for cosine functions, leads to the desired relation,

$$\cos\left(\frac{\theta_i}{2}\right) = \frac{\sigma_i}{\|A\|_F}.\tag{3.15}$$

The run time of QSVE is dominated by the phase estimation which returns an  $\delta$ -error estimated eigenvalue  $\overline{\theta}_i$ , such that  $|\overline{\theta}_i - \theta_i| \leq 2\delta$ . This error propagates to the

estimated singular value as  $\overline{\sigma}_i = \cos{(\overline{\theta}_i/2)} \|A\|_F$ . The error in the estimated singular values can hence be bounded from above by  $|\overline{\sigma}_i - \sigma_i| \leq \delta \|A\|_F$ . The unitary W can be implemented in time  $\mathcal{O}$  (polylog(mn)) by using the suitable data structure. In summary, the total runtime for quantum singular value estimation with additive error  $\delta \|A\|_F$  is in  $\mathcal{O}$  (polylog $(mn)/\delta$ ).

#### 3.3 The QDLS algorithm

The application of the QSVE algorithm is particularly interesting for solving linear systems with a dense matrix since the QSVE runtime depends on the Frobenius norm  $\|A\|_F$ , instead of the sparsity s(A). We now show that applying the QSVE algorithm leads to an efficient quantum algorithm for solving dense linear systems. Recall the fact that given a symmetric matrix  $A \in \mathbb{R}^{n \times n}$  with spectral decomposition

$$A = \sum_{i \in [n]} \lambda_i \mathbf{s}_i \mathbf{s}_i^{\dagger}, \tag{3.16}$$

the singular value decomposition of A has the form of

$$A = \sum_{i \in [n]} |\lambda_i| \mathbf{u}_i \mathbf{v}_i^{\dagger}, \tag{3.17}$$

where the left and right singular vectors  $\mathbf{u}_i$  and  $\mathbf{v}_i$  are equivalent to the eigenvectors  $\mathbf{s}_i$  up to an ambiguity of sign, such that  $\mathbf{s}_i = \mathbf{u}_i = \pm \mathbf{v}_i$ . Applying the QSVE algorithm to a positive definite matrix immediately yields the solution of the linear system as the estimated singular values and eigenvalues are equal,  $\overline{\sigma}_i = |\overline{\lambda}_i|$ . For a general symmetric matrix, QSVE leads to the estimation of  $|\lambda_i|$  but not its sign,  $sign(\lambda_i)$ . Therefore in order to solve general linear systems, we need a to recover  $sign(\lambda_i)$ .

In Section 3.3.1, we will present a linear system algorithm with a simple technique to recovers the signs using the QSVE procedure as an oracle incurring only a constant multiplicative overhead. We assume that A has been rescaled so that its eigenvalues lie within the interval  $[-1,-1/\kappa] \cup [1/\kappa,1]$ , where  $\kappa$  denotes the condition number of A. This is the same assumption made in [27] and also indicated in the review [38]. We will show in Section 3.3.2 the algorithm runs in  $\tilde{\mathcal{O}}(\sqrt{n})$  time for arbitrary matrices with a bounded spectral norm, and hence has no explicit dependence on the sparsity.

#### 3.3.1 Procedures

We are now in the position to outline the procedures of the quantum dense linear system algorithm.

1. Prepare a quantum state  $|\mathbf{b}\rangle$  which encodes the vector  $\mathbf{b} \in \mathbb{R}^n$  as

$$|\mathbf{b}\rangle = (\mathbf{b}^T \mathbf{b})^{-1/2} \sum_i \beta_i |\mathbf{v}_i\rangle,$$
 (3.18)

where  $|\mathbf{v}_i\rangle$  stores the  $i^{th}$  singular vectors of A.

2. Perform the QSVE algorithm for matrices A and for  $A' = A + \mu I$  with  $\delta < 1/2\kappa$  and  $\mu = 1/\kappa$  to obtain

$$(\mathbf{b}^T \mathbf{b})^{-1/2} \sum_{i} \beta_i |\mathbf{v}_i\rangle_{\mathbf{A}} ||\overline{\lambda}_i|\rangle_{\mathbf{B}} ||\overline{\lambda}_i + \mu|\rangle_{\mathbf{C}}.$$
 (3.19)

3. Append an ancillary register and set its value to 1 if the value in register B is greater than that in register C and apply a conditional phase gate, which leads to

$$(\mathbf{b}^{T}\mathbf{b})^{-1/2} \sum_{i} (-1)^{f_{i}} \beta_{i} |\mathbf{v}_{i}\rangle_{A} ||\overline{\lambda}_{i}|\rangle_{B} ||\overline{\lambda}_{i} + \mu|\rangle_{C} |f_{i}\rangle_{D}.$$
 (3.20)

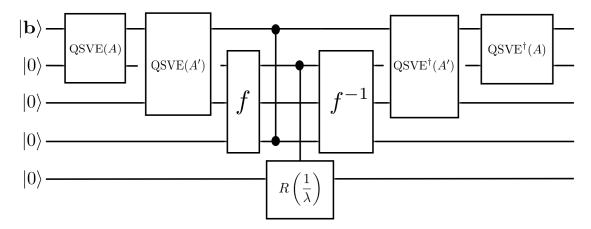
4. Append another ancillary register and apply a rotation conditioned on register B with  $\gamma = \mathcal{O}(1/\kappa)$ . Then uncompute the registers B, C and D to obtain

$$(\mathbf{b}^T \mathbf{b})^{-1/2} \sum_{i} (-1)^{f_i} \beta_i | \mathbf{v_i} \rangle \left( \frac{\gamma}{|\overline{\lambda_i}|} | 0 \rangle + \sqrt{1 - \left(\frac{\gamma}{|\overline{\lambda_i}|}\right)^2} | 1 \rangle \right). \tag{3.21}$$

A high-level circuit diagram that describes the QDLS algorithm until this step is shown in Figure 3.2.

5. Measure the last register in the computational basis. Conditioned on obtaining  $|0\rangle$ , the system is in the desired state,

$$(\mathbf{b}^T \mathbf{b})^{-1/2} \sum_{i} (-1)^{f_i} \frac{\beta_i}{|\overline{\lambda_i}|} |\mathbf{v_i}\rangle.$$
 (3.22)



**Figure 3.2:** The circuit diagram for the QDLS algorithm until Step. 4. The QSVE subroutine is applied for A and  $A' = A + \mu I$ , obtaining the respective singular values in superposition stored in two quantum registers. Then the function f compares the value of the second and third registers, and stores the outcome in the fourth register. A phase gate is then applied conditioned on the value of the fourth register, which successfully recovers the sign of the desired eigenvalues for inversion. The subsequent controlled rotation and uncomputation proceed similarly as the original linear system algorithm of Ref. [27].

#### 3.3.2 Analysis

**Sign recovery** We first argue that above QDLS algorithm correctly recovers the sign of the  $\lambda_i$ . The algorithm compares the estimates obtained by performing QSVE for A and for  $A' = A + \mu I_n$ , where  $\mu$  is a positive scalar chosen to be the inverse of the condition number,  $\kappa$ . The matrix A' has eigenvalues  $\lambda_i + \mu$  while the corresponding eigenvectors are the same as those of A. Not that if  $\lambda_i \geq 0$ , we have

$$|\lambda_i + \mu| = |\lambda_i| + |\mu| \ge |\lambda_i|. \tag{3.23}$$

However if  $\lambda_i < -\mu/2$ , then we instead have

$$|\lambda_i + \mu| < |\lambda_i|. \tag{3.24}$$

Thus if we had perfect estimation for the singular values, then choosing  $\mu < 2/\kappa$  would recover the sign correctly as the eigenvalues of A lie in the interval  $[-1, -1/\kappa] \cup [1/\kappa, 1]$ . In an imperfect setting of QSVE, with the choice of  $\mu = 1/\kappa$  and  $\delta < 1/2\kappa$  the signs are correctly recovered for all  $\lambda_i$ .

Runtime and errors As we will show later in this section, the additive error  $\epsilon$  for the linear system solver is related to the QSVE precision parameter  $\delta$  via  $\delta = \mathcal{O}\left(\frac{\epsilon}{\kappa \|A\|_F}\right)$ . The success probability of the final post-selection step requires on average  $\mathcal{O}\left(\kappa^2\right)$  repetitions of the coherent computation. However, applying amplitude amplification [30,39] can reduce this cost to  $\mathcal{O}\left(\kappa\right)$ . Hence an upper-bound of the runtime of our algorithm is given by  $\mathcal{O}\left(\kappa^2 \cdot \operatorname{polylog}(n)\|A\|_F/\epsilon\right)$ . The error dependence on the Frobenius norm suggests that our algorithm is most accurate when the  $\|A\|_F$  is bounded by some constant, in which case the algorithm returns the output state with a constant  $\epsilon$ -error in polylogarithmic time even if the matrix is non-sparse. More generally, as in the

HHL algorithm [27], we can assume that the spectral norm  $\|A\|_*$  is bounded by a constant, although the Frobenius norm may scale with the dimensionality of the matrix. In such cases we have  $\|A\|_F = \mathcal{O}(\sqrt{n})$ , and the QDLS algorithm runs in time  $\mathcal{O}(\kappa^2\sqrt{n}\cdot\mathrm{polylog}(n)/\epsilon)$  and returns the output with a constant  $\epsilon$  additive error. Furthermore, since  $\|A\|_F \leq \sqrt{r}\|A\|_*$ , where r denotes the rank of A, the runtime may also be written as  $\mathcal{O}(\kappa^2\sqrt{r}\cdot\mathrm{polylog}(n)/\epsilon)$ . Hence an exponentially more advantageous runtime is achievable if the rank of A is polylogarithmic in n.

**Error bound details** We now establish error bounds on the final state. In a similar fashion to the analysis of Ref. [27], we use the filter functions f and g [40], which allow us to invert only the well-conditioned part of the matrix, that is, the space which is spanned by the eigenvectors with eigenvalues,  $\lambda_i \geq 1/\kappa$ . We define the functions,

$$f(\lambda) = \begin{cases} \frac{1}{\kappa \gamma \lambda}, & |\lambda| \ge 1/\kappa; \\ \eta_1(\lambda), & \frac{1}{\kappa} > |\lambda| > \frac{1}{2\kappa}; \\ 0, & \frac{1}{2\kappa} \ge |\lambda|; \end{cases}$$
(3.25)

and

$$g(\lambda) = \begin{cases} 0, & |\lambda| \ge 1/\kappa; \\ \eta_2(\lambda), & \frac{1}{\kappa} > |\lambda| > \frac{1}{2\kappa}; \\ \frac{1}{2}, & \frac{1}{2\kappa} > |\lambda|, \end{cases}$$
(3.26)

where  $\gamma = \mathcal{O}(1/\kappa)$  is the parameter chosen in Step 4 of the algorithm in Section 3.3.1 to ensure that the probability amplitudes are bounded by unity after the controlled rotation by any eigenvalues. The functions  $\eta_1$  and  $\eta_2$  are interpolating functions chosen such that

 $f^2(\lambda)+g^2(\lambda)\leq 1$  for all  $\lambda\in\mathbb{R}.$  A possible (non-unique) choice of  $\eta_1$  and  $\eta_2$  can be

$$\eta_1(\lambda) = \frac{1}{2} \sin\left(\frac{\pi}{2} \cdot \frac{\lambda - \frac{1}{\kappa'}}{\frac{1}{\kappa} - \frac{1}{\kappa'}}\right),\tag{3.27}$$

and

$$\eta_2(\lambda) = \frac{1}{2} \cos\left(\frac{\pi}{2} \cdot \frac{\lambda - \frac{1}{\kappa'}}{\frac{1}{\kappa} - \frac{1}{\kappa'}}\right),\tag{3.28}$$

Note that the presented QDLS algorithm corresponds to the choice  $g(\lambda) = 0$ . We then define the map

$$|h(\lambda)\rangle := \sqrt{1 - f(\lambda)^2 - g(\lambda)^2} |NO\rangle + f(\lambda) |WC\rangle + g(\lambda) |IC\rangle,$$
 (3.29)

where  $|{\rm NO}\rangle$  indicates that no matrix inversion has taken place,  $|{\rm IC}\rangle$  means that part of  $|{\rm b}\rangle$  is in the ill-conditioned subspace of A, and  $|{\rm WC}\rangle$  means that the matrix inversion has taken place and is in the well-conditioned subspace of A. This allows us to invert only the well-conditioned part of the matrix while it flags the ill-conditioned ones and interpolates between those two behaviours when  $1/(2\kappa) < |\lambda| < 1/\kappa$ . We therefore only invert eigenvalues which are larger than  $1/(2\kappa)$ . This subtlety is the motivation behind our choice of  $\mu$  in the algorithm.

Let Q denote the error-free operation corresponding to the QSVE subroutine followed by the controlled rotation without post-selection, such that

$$|\psi\rangle := \mathcal{Q} |\mathbf{b}\rangle |0\rangle \to \sum_{i} \beta_{i} |\mathbf{v}_{i}\rangle |\mathbf{h}(\lambda_{i})\rangle.$$
 (3.30)

 $\overline{\mathcal{Q}}$  in contrast describes the same procedure but the phase estimation step is erroneous, such that

$$|\overline{\psi}\rangle := \overline{\mathcal{Q}} |\mathbf{b}\rangle |0\rangle \to \sum_{i} \beta_{i} |\mathbf{v}_{i}\rangle |\mathbf{h}(\overline{\lambda}_{i})\rangle.$$
 (3.31)

In order to bound the error,  $\|\overline{Q} - Q\|$ , we choose a general state  $|\mathbf{b}\rangle$ , and find the equivalent error bound  $\|Q|\mathbf{b}\rangle - \overline{Q}|\mathbf{b}\rangle\| := \||\overline{\psi}\rangle - |\psi\rangle\|$ . We need to make use of the fact that the map  $\lambda \to |h(\lambda)\rangle$  is  $\mathcal{O}(\kappa)$ -Lipschitz [27]. That is to say  $\forall \lambda_i \neq \lambda_j$  for some  $c \leq \pi/2 = \mathcal{O}(1)$ , we have

$$\||\mathbf{h}(\lambda_i)\rangle - |\mathbf{h}(\lambda_i)\rangle\| \le c\kappa |\lambda_i - \lambda_i|.$$
 (3.32)

Note that it suffices to lower-bound  $\operatorname{Re}(\langle \overline{\psi} | \psi \rangle)$  since we have

$$\||\overline{\psi}\rangle - |\psi\rangle\| = \sqrt{2\left(1 - \operatorname{Re}(\langle \overline{\psi}|\psi\rangle)\right)},$$
 (3.33)

Now we take the inner product between Eq. 3.30 and Eq. 3.31 to obtain

$$\operatorname{Re}(\langle \overline{\psi} | \psi \rangle) = \sum_{i} |\beta_{i}|^{2} \operatorname{Re}(\langle \operatorname{h}(\overline{\lambda}_{i}) | \operatorname{h}(\lambda_{i}) \rangle). \tag{3.34}$$

Next we use the error bounds of the QSVE subroutine for the eigenvalue distance, i.e.  $|\lambda_i - \overline{\lambda}_i| \leq \delta ||A||_F$ , which leads to

$$\operatorname{Re}(\langle \overline{\psi} | \psi \rangle) \ge \sum_{i} |\beta_{i}|^{2} \left( 1 - \frac{c^{2} \kappa^{2} \delta^{2} ||A||_{F}^{2}}{2} \right). \tag{3.35}$$

This is a consequence of the finite accuracy phase estimation, and the  $\mathcal{O}(\kappa)$ -Lipschitz property of Eq. 3.32. Since  $0 \leq \text{Re}(\langle \overline{\psi} | \psi \rangle) \leq 1$ , it follows that

$$1 - \operatorname{Re}(\langle \overline{\psi} | \psi \rangle) \le \sum_{i} |\beta_{i}|^{2} \left( \frac{c^{2} \kappa^{2} \delta^{2} ||A||_{F}^{2}}{2} \right). \tag{3.36}$$

Finally we use the fact that  $\sum_i |\beta_i|^2 = 1$ , the distance can be bounded as

$$\||\overline{\psi}\rangle - |\psi\rangle\| \le \mathcal{O}\left(\kappa\delta\|A\|_F\right).$$
 (3.37)

If this additive error on the output state is needed to be on the order of  $\epsilon$ , we need to take the phase estimation accuracy to be  $\delta = \mathcal{O}\left(\frac{\epsilon}{\kappa \|A\|_F}\right)$ . This results in a runtime that scales as  $\mathcal{O}\left(\kappa \|A\|_F \cdot \operatorname{polylog}(n)/\epsilon\right)$ . In order to successfully perform the final post-selection step, we need to repeat the algorithm on average  $\kappa^2$  times. This additional multiplicative factor of  $\kappa^2$  can be reduced to  $\kappa$  using amplitude amplification [30,39]. Putting everything together, we have an overall runtime that scales as  $\mathcal{O}\left(\kappa^2 \|A\|_F \cdot \operatorname{polylog}(n)/\epsilon\right)$ .

#### 3.4 Summary and discussions

We have shown in this chapter that given  $\mathbf{b} \in \mathbb{R}^n$  and a Hermitian matrix  $A \in \mathbb{R}^{n \times n}$  with spectral decomposition  $A = \sum_i \lambda_i \mathbf{s}_i \mathbf{s}_i^{\dagger}$  stored in a suitable data structure, the QDLS algorithm returns the state  $|\overline{A^{-1}\mathbf{b}}\rangle$  such that  $\left\||\overline{A^{-1}\mathbf{b}}\rangle - |A^{-1}\mathbf{b}\rangle\right\| \leq \epsilon$ . The runtime of the algorithm scales as  $\mathcal{O}\left(\kappa^2 \cdot \operatorname{polylog}(n) \cdot \|A\|_F/\epsilon\right)$ , where  $\kappa$  is the condition number and  $\|A\|_F$  is the Frobenius norm of A.

**Bounded spectral norm** Assuming the spectral norm,  $||A||_*$ , is bounded by a constant or grows no faster than polylogorithmically in n, the overall runtime scaling reduces to

 $\mathcal{O}(\kappa^2 \sqrt{n} \cdot \operatorname{polylog}(n)/\epsilon)$ , since we have

$$||A||_F = \sqrt{\sum_{i=1}^{n} \lambda_i^2} \le \sqrt{n|\lambda|_{max}^2} \le \sqrt{n}||A||_*.$$
 (3.38)

This amounts to a polynomial speed-up over the runtime scaling achieved in Ref. [27] when applied to dense matrices with black-box Hamiltonian simulation. The bounded spectral norm is a realistic assumption if classical normalisation preprocessing can be applied so that the maximum absolute values of  $\lambda_i$  is bounded. As the same bounded spectral norm assumption is also required in the error analysis of Ref. [27], the algorithm presented in this chapter represents a new state-of-the-art for solving dense linear systems on a quantum computer.

**Low-rank** In special cases, the matrix A has a low-rank structure, such that the number of non-zero eigenvalues grows no faster than polylogarithmically in n. In such scenarios, the runtime of the presented QDLS algorithm scales as  $\mathcal{O}\left(\kappa^2 \cdot \operatorname{polylog}(n)/\epsilon\right)$ , which amounts to an exponential improvement over previously existing algorithms for solving dense linear system problems.

**Distinction in memory models** Note that the memory model described in Section 3.1 is distinct from the black-box model. This QSVE-based linear system algorithm achieves a  $\tilde{\mathcal{O}}(\sqrt{n})$ -scaling for dense matrices in this augmented quantum memory model, and it is an interesting question whether a similar scaling is achievable in the black-box matrix element access model.

**Non-invertible matrix** The SVE-based algorithm also applies to more general scenarios where the matrix A is not invertible. Then the algorithm will instead compute the Moore-Penrose pseudo-inverse. The runtime in these cases will be bounded by  $1/|\lambda_{min}|$ 

instead of  $\kappa$ , where  $\lambda_{min}$  is the non-zero eigenvalue for A with the smallest absolute value.

**Outlooks** From a practical point of view, the constant runtime overhead for a given set of elementary fault-tolerant quantum gates is an important consideration. Scherer *et al.* [41] showed that implementations of the HHL algorithm [27] potentially suffer from a large constant overhead with currently available technology, which may hinder the prospects of near-term applications. Whether or not the SVE-based QDLS algorithm has considerably smaller constant overhead, due to the absence of Hamiltonian simulation, remains an interesting open question.

## Part II

# Gaussian processes with quantum algorithms

## **Chapter 4**

## Gaussian processes in classical machine learning

In the previous chapters, we have introduced the basics of quantum computation and have seen some examples of quantum algorithms. Particularly, in Section 2.2.2 of Chapter 2 and in Chapter 3, we have seen a quantum variant of the linear systems problem can be efficiently solved by a quantum computer with the access to suitable memory models. In this part of the thesis, we apply some of these quantum ideas to a powerful model in supervised machine learning, Gaussian processes (GP). To start with, in this chapter, we will follow the notation of Ref. [42] and introduce the basics of GPs and review the typical classical implementations of inference with GP models as well as GP model selection. In Chapters 5 and 6, we will follow closely Ref. [43] and [44] and present quantum algorithms for computing GP regression models and training GP regression models respectively. In Chapter 7, we will make use of the quantum GP algorithms to present a quantum approach to Bayesian deep learning.

#### 4.1 Introduction

Supervised machine learning amounts to inferring a function from labelled training data [45]. The GPs represent an approach to supervised learning that models the latent functions associated with the outputs of an inference problem as an infinite-dimensional generalisation of a Gaussian distribution [42]. The GP approach offers numerous desirable properties such as being capable of capturing a wide range of behaviours with only a simple set of parameters, the ability to easily express uncertainty, and admitting a natural Bayesian interpretation. As such GP models have been widely used across a broad spectrum of applications, ranging from robotics, data mining, geophysics (where GP approaches are also known as kriging), climate modelling, and predicting price behaviour of commodities in financial markets.

Although GP models are becoming increasingly popular in the classical community of machine learning, they are known to be computationally expensive, which hinders their widespread adoption. A practical implementation of Gaussian process regression (GPR) model with n training points typically requires  $\Omega(n^3)$  basic operations [42]. This has lead to significant amount of effort aimed at reducing the computational cost of working with such models, with investigations into low-rank approximations of GPs [46], variational approximations [47] and Bayesian model combination for distributed GPs [48]. A thorough discussion of these approximation methods is beyond the scope of this thesis. Instead, we will argue that quantum computation offers efficient exact implementation of GPR even when the size of the input data is classically infeasible.

The contents of this chapter are organised as follows: In Section 4.1.1, we will introduce some preliminary definitions and concepts necessary for describing GPs as regression models. In Section 4.2, we will present the basics of GPR as well as its typical classical implementation. In Section 4.3 we will review the classical GP model selection procedures, with an emphasis on the figure of merit for a given model's performance.

In Section 4.4, we will discuss the connection between GPs and deep neural networks, mainly following the results of [49]. This chapter provides only a basic level introduction to GPs. Readers are referred to Ref. [42, 49–51] for further details.

#### 4.1.1 Preliminaries

**Multivariate Gaussian distributions** If a vector of random variables  $\mathbf{x} \in \mathbb{R}^k$  follows a multivariate Gaussian distribution with a mean vector,  $\boldsymbol{\mu}$  and a covariance matrix,  $\Sigma$ , its probability density function is given by,

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})\right), \tag{4.1}$$

where  $|\Sigma|$  denotes the determinant of the covariance matrix. We denote this distribution as  $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ .

Gaussian processes A Gaussian process (GP) is defined as a set of random variables, any finite subset of which follows a joint multivariate Gaussian distribution [42]. A GP model is entirely specified by a prior mean function,  $\mu(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ , and a covariance function (kernel),  $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}'))]$  of some underlying actual process  $f(\mathbf{x})$ . We write

$$f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$
 (4.2)

to denote a Gaussian process. For simplicity, we will assume the prior mean to be zero without loss of generality.

**Marginalisation property** As a requirement for consistency, models for statistical inference need to satisfy the following marginalisation property: Given a set of random

variables S and a statistical model that specifies a probability distribution P, for any subsets  $S' \subset S$ , the corresponding probability distribution is given by the marginal distribution for S' in P. Intuitively, this means that the distribution of a larger set of variables needs to be consistent with the distribution of its subsets. This requirement is automatically satisfied by the GP definition.

#### 4.2 Gaussian process regression

In this section, we introduce Gaussian processes as a regression model, following closely Chapter 2 of Ref. [42]. We will consider a supervised learning problem with a training dataset  $\mathcal{T}$  with n d-dimensional input points,  $\{\mathbf{x}_i\}_{i=0}^{n-1}$ , and their corresponding output points,  $\{y_i\}_{i=0}^{n-1}$ , such that  $\mathcal{T} = \{\mathbf{x}_i, y_i\}_{i=0}^{n-1}$ . The goals is to infer an underlying function  $f(\mathbf{x})$  from the observed input-output pairs subject to Gaussian random noise,

$$y = f(\mathbf{x}) + \varepsilon_{\text{noise}},\tag{4.3}$$

where  $\varepsilon_{\text{noise}} \sim \mathcal{N}(0, \sigma_n^2)$  is independent, identically distributed (i.i.d.) noise that follows a Gaussian distribution with 0 mean and  $\sigma_n^2$  variance. Since the underlying  $f(\mathbf{x})$  is not directly observed, it is known as the "latent function". When given a new input "test point",  $\mathbf{x}_*$ , our model aims at generating a predictive distribution for  $f_* = f(\mathbf{x}_*)$ . The Gaussian process regression approach models the latent function  $\{f(\mathbf{x_i})\}_{i=0}^{n-1}$  as a joint multivariate Gaussian distribution [42].

#### 4.2.1 Linear model with Gaussian noise

We start by considering the standard model of linear regression so that the underlying function  $f(\mathbf{x})$  of an input vector  $\mathbf{x}$  is given by

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{w},\tag{4.4}$$

where the weight vector  $\mathbf{w}$  contains the parameters of the linear model. Under our Gaussian noise assumption, the actual observed values y are given by  $y = \mathbf{x}^T \mathbf{w} + \varepsilon_{\text{noise}}$ .

**Likelihood** The likelihood is defined by the probability density of the observed values conditioned on the given parameters. The independence assumption allows us to factor over the points in the whole training set, and write the likelihood as

$$p(\mathbf{y}|X, \mathbf{w}) = \prod_{i=1}^{n} p(y_i|\mathbf{x}_i, \mathbf{w})$$

$$= \prod_{i=1}^{n} \frac{1}{\sigma_n \sqrt{2\pi}} \exp\left(-\frac{(y_i - \mathbf{x}^T \mathbf{w})^2)}{2\sigma_n^2}\right)$$

$$= \left(\frac{1}{\sigma_n \sqrt{2\pi}}\right)^n \exp\left(-\frac{|\mathbf{y} - X^T \mathbf{w}|^2}{2\sigma_n^2}\right)$$

$$= \mathcal{N}\left(X^T \mathbf{w}, \sigma_n^2 I\right), \tag{4.5}$$

where  $X \in \mathbb{R}^{d \times n}$  denotes the matrix containing the entire set of input data points, and  $\mathbf{y} \in \mathbb{R}^n$  denotes the vector containing the entire set of output data points. We have shown the likelihood of a linear model with Gaussian noise follows a multivariate Gaussian distribution with mean vector  $X^T\mathbf{w}$  and covariance matrix  $\sigma_n^2 I$ .

**Bayesian inference** In a Bayesian approach, we assume a prior distribution over  $\mathbf{w}$  which expresses a belief about the parameters before observing the outputs. We choose a Gaussian prior with  $\mathbf{0}$  mean and covariance matrix  $\Sigma_p$ :  $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \Sigma_p)$ . Bayesian

inference of the linear model follows the posterior distribution over w, which can be evaluated using the Bayes' theorem,

$$p(\mathbf{w}|\mathbf{y}, X) = \frac{p(\mathbf{y}|X, \mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|X)}.$$
(4.6)

The factor  $p(\mathbf{y}|X)$ , known as the marginal likelihood, is independent of  $\mathbf{w}$  and acts as a normalisation constant. We have posterior distribution,  $p(\mathbf{w}|\mathbf{y}, X)$  proportional to

$$p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w}) = \exp\left(-\frac{1}{2\sigma_n^2}(\mathbf{y} - X^T\mathbf{w})^T(\mathbf{y} - X^T\mathbf{w})\right) \exp\left(-\frac{1}{2}\mathbf{w}^T\Sigma_p^{-1}\mathbf{w}\right)$$
$$= \exp\left(-\frac{1}{2}(\mathbf{w} - \bar{\mathbf{w}})^T\left(\frac{X^TX}{\sigma_n^2} + \Sigma_p^{-1}\right)(\mathbf{w} - \bar{\mathbf{w}})\right), \tag{4.7}$$

where we have used the shorthand  $\bar{\mathbf{w}} = \sigma_n^{-2} C^{-1} X \mathbf{y}$  with  $C = \frac{X X^T}{\sigma_n^2} + \Sigma_p^{-1}$ . Thus the posterior follows a Gaussian distribution,  $p(\mathbf{w}|\mathbf{y}, X) \sim \mathcal{N}\left(\sigma_n^{-2} C^{-1} X \mathbf{y}, C^{-1}\right)$ .

**Predictive distribution** Given a new input test point  $\mathbf{x}_*$ , the predictive distribution for  $f_* = f(\mathbf{x}_*)$  can then by evaluated with following Bayesian integral,

$$p(f_*|\mathbf{x}_*, X, \mathbf{y}) = \int p(f_*|\mathbf{x}_*, \mathbf{w}) p(\mathbf{w}|X, \mathbf{y}) d\mathbf{w}$$
$$= \mathcal{N}\left(\frac{1}{\sigma_n^2} \mathbf{x}_*^T A^{-1} X \mathbf{y}, \mathbf{x}_*^T A^{-1} \mathbf{x}_*\right). \tag{4.8}$$

Therefore the predictive distribution  $f_*$  conditioned on the training set with X and y is a Gaussian with mean  $\frac{1}{\sigma_n^2} \mathbf{x}_*^T A^{-1} X \mathbf{y}$  and variance  $\mathbf{x}_*^T A^{-1} \mathbf{x}_*$ .

#### 4.2.2 Feature space projection

The standard linear model described above often suffers from limited expressiveness and fails to capture interesting higher order features. A simple enhancement known as feature space projection overcomes this issue. The idea is to have the inputs mapped into certain

chosen space with higher dimensions, which is specified by a set of basis functions. That is, we make use of a function  $\phi(\mathbf{x})$  to project an input vector  $\mathbf{x} \in \mathbb{R}^D$  into a feature space with dimension N. Instead of being applied directly on the inputs, the linear model is instead applied in this projected feature space. For instance, for a scalar input x, possible choices of the feature space basis functions include  $\phi(x) = (1, x, x^2, ...)$ ,  $\phi(x) = (1, \sin(x), \cos(x), ...)$ , etc. The problem of choosing the appropriate basis functions is related to the model selection of GP, which we will address in Section 4.3.

**Prediction** After the feature space projection, the regression model is augmented into  $f(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{w}$ , with  $\mathbf{w} \in \mathbb{R}^N$ . Following an analogous Bayesian analysis as before, we arrive at the following formula for the predictive distribution,

$$p(f_*|\mathbf{x}_*, X, \mathbf{y}) = \mathcal{N}\left(\frac{1}{\sigma_n^2}\phi(\mathbf{x}_*)^T C^{-1}\phi(X)\mathbf{y}, \phi(\mathbf{x}_*)^T C^{-1}\phi(\mathbf{x}_*)\right), \tag{4.9}$$

with  $C = \sigma_n^{-2} \phi(X) \phi(X)^T + \Sigma_p^{-1}$ . Alternatively, this can be written as

$$\mathcal{N}\left(\phi_*^T \Sigma_p \Phi(K + \sigma_n^2 I)^{-1} \mathbf{y}, \phi_*^T \Sigma_p \phi_* - \phi_*^T \Sigma_p \Phi(K + \sigma_n^2 I)^{-1} \Phi^T \Sigma_p \phi_*\right), \tag{4.10}$$

with the shorthand notations,  $\Phi = \phi(X)$ ,  $\phi_* = \phi(\mathbf{x}_*)$  and  $K = \Phi^T \Sigma_p \Phi$ .

**Kernel trick** We now replace the inner products in the feature space by functions in the input space by defining the covariance function,  $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \Sigma_p \phi(\mathbf{x}')$ , and the associated vector  $\mathbf{k} = \Phi^T \Sigma_p \phi_*$ . This leads to

$$p(f_*|\mathbf{x}_*, X, \mathbf{y}) = \mathcal{N}\left(\mathbf{k}^T (K + \sigma_n^2 I)^{-1} \mathbf{y}, k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}^T (K + \sigma_n^2 I)^{-1} \mathbf{k}\right). \tag{4.11}$$

The predictive distribution of  $f_*$  is therefore a Gaussian distribution specified by  $p(f_*|\mathbf{x}_*, \mathcal{T}) \sim \mathcal{N}(\bar{f}_*, \mathbb{V}[f_*])$ , where

$$\bar{f}_* = \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{y} \tag{4.12}$$

$$\mathbb{V}[f_*] = k\left(\mathbf{x}_*, \mathbf{x}_*\right) - \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{k}_*. \tag{4.13}$$

Hence the mean predictor Eq. 4.12 and the variance Eq. 4.13 are the central quantities of interest and the main goals of computation in Gaussian process regression.

#### 4.2.3 Classical computation and complexity

The typical classical implementation of GPR is based on computing the Cholesky decomposition of  $(K + \sigma_n^2 I)$ . This amounts to finding the Cholesky factor, the lower-triangular matrix L that satisfies  $(K + \sigma_n^2 I) = LL^T$ . Computing the Cholesky factor has a cost proportional to  $n^3$ , and it is numerically stable. The mean predictor can be expressed as  $\bar{f}_* = \mathbf{k}_*^T \alpha$  by defining  $\alpha = (K + \sigma_n^2 I)^{-1} \mathbf{y}$ . The vector  $\alpha$  is then obtained by solving  $LL^T \alpha = \mathbf{y}$ . Let  $\mathbf{y}' = L \setminus \mathbf{y}$  denote the solution to the triangular linear system  $L\mathbf{y}' = \mathbf{y}$ . The vector  $\alpha$  can then be rewritten as  $\alpha = L^T \setminus L \setminus \mathbf{y}$ , hence computing  $\alpha$  simply amounts to solving two triangular systems. This has a runtime which scales as  $\mathcal{O}(n^2)$ . Similarly, the variance  $\mathbb{V}[f_*]$  can be expressed in terms of the Cholesky factor as  $\mathbb{V}[f_*] = k(\mathbf{x}_*, \mathbf{x}_*) - (L \setminus \mathbf{k}_*)^T (L \setminus \mathbf{k}_*)$ . Hence it also has a  $\mathcal{O}(n^2)$  runtime. Thus the overall runtime of classically computing the mean predictor and the associated variance for a GP model scales as  $\mathcal{O}(n^3)$ . When dealing with large-scale problems with greater than  $10^3$  input points, exact inference with GPR is practically intractable.

#### 4.3 Training Gaussian processes

Model selection refers to the process of choosing the preferred variations of the model used in a supervised learning task, to achieve better predictive performance. In the context of GPs, this amounts to selecting a covariance function. In practice, a family of functions is usually considered. The parameters of the family of kernels are referred to as the kernel hyperparameters, and a range of optimisers are used in order to tune these hyperparameters based on the observed data. This model selection process is commonly known as the training of a Gaussian process. Since training typically involves repeated evaluation of certain cost functions that characterise how well a given model is performing on the problem, it generally carries a runtime overhead that scales polynomially with the input size. In this section, we will follow the conventions of Chapter 5 of [42] and review the basics of training a GP model. Our emphasis is on introducing the log marginal likelihood (LML) as a measure for the model's suitability and the classical computation of the LML function.

#### 4.3.1 Log marginal likelihood

The natural figure or merit that measures the performance of a supervised machine learning model is the marginal likelihood. In the context of GPR, it is the probability density of the observed output vector conditioned on the model's covariance matrix and the Gaussian noise variance,  $p(\mathbf{y}|K+\sigma_n^2I)$ . As such, training the GP model amounts to optimising the conditioned probability of the observed data given the GP prior by choosing the covariance function and tuning the respective hyperparameters.

For simplicity, we will keep the assumption that the model has a zero prior mean. Since the prior distribution of the observed vector of outputs  $\mathbf{y}$  only differs from that of the latent function  $f(\mathbf{x})$  by a Gaussian noise with variance  $\sigma_n^2$ , it is clear that we can write down the distribution as  $\mathbf{y} \sim \mathcal{N}(0, K + \sigma_n^2 I)$ . The logarithm of marginal

likelihood LML =  $\log[p(\mathbf{y}|K+\sigma_n^2I)]$  then follows straight-forwardly from the definition of Gaussian distribution, and we have

$$LML = -\frac{1}{2}\mathbf{y}^{T}(K + \sigma_{n}^{2}I)^{-1}\mathbf{y} - \frac{1}{2}\log\det[K + \sigma_{n}^{2}I] - \frac{n}{2}\log 2\pi.$$
 (4.14)

Since the logarithm is monotonic, maximising LML is equivalent to directly maximising  $p(\mathbf{y}|K + \sigma_n^2 I)$ .

Interpretations Note that only the first term of Eq. 4.14 involves the observed outputs y. This is the contribution to LML that actually measures how the model is performing at fitting the training data. The second term depends only on the covariance matrix with the identity noise entry and can be interpreted as a penalty on the model's complexity. It generally disfavours models that happen to overfit the training set. The last term is normalisation constant ensuring the probability is bounded by one. It is easily computable. Hence only the first two terms in LML involve extensive matrix computations, and could potentially present bottlenecks in the efficiency of training.

**Hyperparameter optimisation** Training requires tuning the model's hyperparameters in order to maximise the LML. A standard approach is based on gradient descent methods. This requires evaluating the variation of LML with respect to a change in each hyperparameter  $\theta_j$ . We will come back to this point in the context of applying quantum algorithms in Chapter 6.

#### 4.3.2 Implementations and complexity

The runtime of classically computing LML is dominated by the matrix inversion and determinant computation. In standard implementations based on Cholesky decompositions, the runtime scales with the input size as  $\mathcal{O}(n^3)$ . Because of the high computation

cost in exact implementations, numerous compromising approaches have been proposed in the machine learning community. For examples, GPs are sometimes chosen to have covariance matrices with fixed ranks to make them computational trackable. In such scenarios, the cost of training can be reduced to  $\mathcal{O}(nr^2)$ , with r denoting the rank of the covariance matrix in the model [46]. This, however, significantly limits the range and the complexity of the functions accessible to the GP model, which could translate into sub-optimal performance. In low-dimensional cases, approaches such as hierarchical matrix factorisation [52] provide good options for implementing GP training, but they do not generalise well to problems with high dimensional datasets, which are often essential to consider in machine learning.

#### 4.3.3 Stochastic trace estimation

As an alternative approach, stochastic trace estimation has gained popularity in recent years [53,54]. These methods make use of the fact that given a matrix  $A \in \mathbb{R}^{n \times n}$ , the logarithm of its determinant is equal to the trace of the  $\log(A)$ , as we have

$$\operatorname{Tr}[\log(A)] = \sum_{i=1}^{n} \log \lambda_i = \log[\det(A)], \tag{4.15}$$

where  $\{\lambda_i\}$  are the eigenvalues of A.

The matrix logarithm in Eq. 4.15 can then approximated by truncating the Taylor series of the logarithmic function,

$$\log(A) \approx \sum_{a=1}^{d} \frac{(I-A)^a}{a}.$$
(4.16)

Alternatively it can be approximated with a Chebyshev polynomial of a specified degree d. Using a trace estimation approach will still require matrix-vector multiplication when raising factors such as A, or (I - A). However the advantage arises as the inner

product form  $\mathbf{z}^{\dagger} \log(A)\mathbf{z}$  can be evaluated in  $\mathcal{O}(n^2)$  for some  $\mathbf{z} \in \mathbb{R}^n$ , where the vector  $\mathbf{z}$  is a so called 'probing vector'. These vectors can be chosen such in a number of ways [55, 56], and they should satisfy  $\mathbb{E}[\mathbf{z}^{\dagger} \log(A)\mathbf{z}] = \mathrm{Tr}(\log(A))$ . Note that there are two major sources of error that can occur in such an approach, namely the errors due approximating  $\log(A)$  with a finite expansion, and the errors directly related to the stochastic trace estimation. We draw special interests to these stochastic trace estimation methods as the approach based on quantum algorithms to be presented in Chapter 6 can be understood as an extension of this class of trace estimation algorithms. As we will show, besides offering a reduction in computational time, the quantum algorithms also use an exact representation of  $\log(K + \sigma_n^2 I)$  to machine precision, which implies a significant suppression in approximation error.

#### 4.4 Connection with deep learning

In this section, we briefly review the connection between Gaussian processes and deep neural network models based on the results of Ref. [49], which provides a Bayesian approach to deep learning. We will leverage this connection to construct a quantum algorithm for Bayesian deep learning in Chapter 7.

Single hidden layer The correspondence between Gaussian processes and a neural network with only a single hidden layer is well-known and discussed [50]. Let  $\mathbf{z}(\mathbf{x}) \in \mathbb{R}^{d_{out}}$  denote the output vector of a neural network with an input vector  $\mathbf{x} \in \mathbb{R}^{d_{in}}$ , with  $z_i(\mathbf{x})$  denoting the  $i^{th}$  component of the output layer. If we assume the weight and bias parameters of the neural network are independent and identically distributed (i.i.d.), each  $z_i$  will be a sum of i.i.d terms. As such, if the hidden layer has an infinite width, the Central Limit Theorem implies that  $z_i$  follows a Gaussian distribution. Now consider a set of n input points, with corresponding outputs  $\{z_i(\mathbf{x}_1), z_i(\mathbf{x}_2), \dots z_i(\mathbf{x}_n)\}$ . Any finite

collection of this output set will follow a joint multivariate Gaussian distribution. By definition,  $z_i$  corresponds to a GP,  $z_i \sim \mathcal{GP}(\mu, K)$ , with a covariance matrix  $K(\mathbf{x}, \mathbf{x}') = \mathbb{E}[z_i(\mathbf{x})z_i(\mathbf{x}')]$ . Conventionally, the weight and bias parameters are chosen to have zero mean so that  $\mu = 0$ .

Deep networks The above correspondence between the GP and the single hidden layer network is generalised to a deep neural network architecture in a recursive manner [49,51]. Let  $z_i^l$  denote the  $i^{th}$  component of the output of the  $l^{th}$  layer. By induction, it follows that  $z_i^l \sim \mathcal{GP}(0,K^l)$ . The covariance matrix on the  $l^{th}$  layer is given by  $K^l(\mathbf{x},\mathbf{x}') = \mathbb{E}[z_i^l(\mathbf{x})z_i^l(\mathbf{x}')]$ . In order to explicitly compute  $K^l(\mathbf{x},\mathbf{x}')$ , we need to specify the Gaussian variance on the weight and bias parameters,  $\sigma_w^2$  and  $\sigma_b^2$ , as well as the non-linear activation functions,  $\phi$  at each layer. We have the following recursive formula for the  $l^{th}$  layer covariance function,

$$K^{l}(\mathbf{x}, \mathbf{x}') = \sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(z_i^{l-1}(\mathbf{x}))\phi(z_i^{l-1}(\mathbf{x}'))], \tag{4.17}$$

where  $z_i^{l-1} \sim \mathcal{GP}(0, K^{l-1})$ . The base case of the induction is given by the layer zero covariance function,

$$K^{0}(\mathbf{x}, \mathbf{x}') = \sigma_b^2 + \sigma_w^2 \left(\frac{\mathbf{x}.\mathbf{x}'}{d_{in}}\right). \tag{4.18}$$

The Bayesian training of the neural network amounts to computing the mean and variance of the predictive distribution, while selecting the GP covariance function and tuning the hyper-parameters is related to choosing the neural network model class, depth, nonlinearity and parameter initialisations. Numerical experiments suggest that neural networks with infinite-width hidden layers trained with Gaussian priors outperform finite-width neural networks trained with stochastic gradient descent in many cases [49].

## Chapter 5

## Quantum enhanced Gaussian

### processes

Having reviewed the basics of Gaussian processes as classical regression models in the previous chapter, now we move on to present a quantum algorithms for enhancing the efficiency of computing GPR. We will start by describing a quantum state preparation procedure that encodes a classical input vector into a quantum state. Quantum state preparation would be important not only for the quantum Gaussian processes algorithm but more generally for all machine learning applications where one desires to use a quantum computer to analyse classical datasets. We will then describe the procedure for the quantum Gaussian process algorithm, followed by a discussion of practicality, and potential caveats in applying the proposed quantum algorithm. The material of this chapter is based on the work of Ref. [57] and [43].

#### 5.1 State preparation

When applying quantum computation to problems with classical input, it is almost always necessary to prepare quantum states that encode the classical input vectors. For instance, in Chapter 2 and 3 we have seen that the quantum linear system problem requires an input quantum state that encodes the known vector in the corresponding classical linear system.

#### 5.1.1 Quantum random access memory

We are specifically concerned with the task of state preparation which involves creating

$$|\mathbf{v}\rangle = \|\mathbf{v}\|_2^{-1} \sum_{i=1}^n v_i |i\rangle, \qquad (5.1)$$

given some vector  $\mathbf{v} \in \mathbb{R}^n$  stored in quantum random access memory (QRAM) [58]. Such a memory structure allows the quantum computer to access data stored in multiple memory locations in a quantum superposition. That is, it allows for operations of the following type,

$$\sum_{i,j} \alpha_{ij} |i\rangle |j\rangle \xrightarrow{\text{QRAM}} \sum_{i,j} \alpha_{ij} |i\rangle |j+m_i\rangle, \qquad (5.2)$$

where  $m_i$  denotes the *i*th entry stored in memory. As such, QRAM enables probabilistically producing  $|\mathbf{v}\rangle$  for any  $\mathbf{v}$  stored in memory.

As a general procedure, to create  $|\mathbf{v}\rangle$  for any  $\mathbf{v}$ , we start with an initial query state,  $n^{-\frac{1}{2}}\sum_i|i\rangle\,|0\rangle$ , and then use the QRAM to map the query state into  $n^{-\frac{1}{2}}\sum_i|i\rangle\,|v_i\rangle$ . We then append a register with ancillary qubits prepared in state  $|0\rangle$  and rotated conditioned on the value in second register, which leads to the state

$$n^{-\frac{1}{2}} \sum_{i} |i\rangle |v_{i}\rangle \left(\sqrt{1 - |v_{i}|^{2}} |0\rangle + v_{i} |1\rangle\right), \tag{5.3}$$

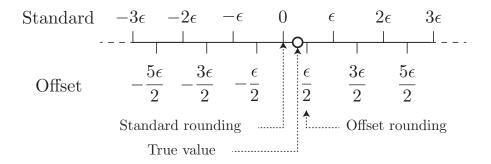
where we have assumed for simplicity that the vector  $\mathbf{v}$  is normalised such that  $|v_i| \leq 1$ . Next, perform a second QRAM call to reverse the computation of  $|v_i\rangle$ . Finally, post-selecting on the state  $|1\rangle$  leads to the desired state that encodes the classical vector,  $|\mathbf{v}\rangle$ .

Success probability The probability of projecting onto the desirable subspace in the final step is given by  $n^{-1}\sum_i |v_i|^2$ . In the case where the entries of  $\mathbf{v}$  are of similar magnitude,  $|\mathbf{v}\rangle$  can be prepared using only a constant number of queries. However, a potential caveat arises when a small number of entries in the vector are significantly larger than the others, in such cases, projecting on the correct state requires  $\Omega\left(\sqrt{n}\right)$  QRAM queries [59], this can be seen as a consequence of the lower bounds on unordered search [60]. The same issue would persist when it is only required to prepare an approximate vector  $|\mathbf{v}'\rangle$  that satisfies  $|\mathbf{v}'-\mathbf{v}|_2 \leq \epsilon$ , where  $\epsilon$  is a sufficiently small constant error.

#### 5.1.2 Robustness and rounding conventions

Fortunately, data processing tasks in practical machine learning almost always implicitly assume a certain level of robustness against small perturbation in the  $\infty$ -norm which measures only the largest entry-wise error. In particular, any digital data processing based on fixed or floating-point arithmetic only makes sense if the outcome of the analysis remains valid if the features in the input vector deviate from the original values below the machine precision. Due to the sheer nature of measurements in the real world, it is practically reasonable to assume the data points are specified with finite precision. Hence the appropriate error constraint which reflects the realistic analytic scenarios is only that  $|\mathbf{v}' - \mathbf{v}|_{\infty} \le \epsilon$ , instead of requiring a close approximation in the 2-norm.

Alternative rounding Assuming the data processing inherently have tolerance against an  $\epsilon$  perturbations in  $\infty$ -norm allows us to work with the vector with entries  $v_i'$  which are half-integer multiples of the base precision  $\epsilon$ . In this alternative numerical rounding convention (as shown in Figure 5.1),  $\mathbf{v}'$  is chosen to be the closest representable vector to  $\mathbf{v}$ , which satisfies  $|\mathbf{v}' - \mathbf{v}|_{\infty} \leq \frac{\epsilon}{2}$ , and the distance from the original value of the data is less than  $\epsilon$ . Note that this offset rounding does not contain an exact representation of 0. This new convention can be either directly realised in the loading stage of the QRAM, or equivalently implemented at the controlled rotation stage, as shown in Eq. 5.3. In



**Figure 5.1:** Numerical rounding conventions. In the standard rounding convention, scalar values are rounded to the nearest integer multiple of precision  $\epsilon$ . Alternatively, we can consider an offset rounding convention, where the rounding is to the nearest half-integer multiple of  $\epsilon$ . In either scheme, the rounded value is always within  $\frac{\epsilon}{2}$  of the true value.

some cases, always using a positive sign offset  $(+\epsilon/2)$  to data-points will introduce an undesirable systematic error in the loaded vector. To overcome this potential issue, one can choose to implement a nearly white noise offset. This can be achieved by either utilising a suitable pseudo-random number generator which is seeded by the corresponding memory location, or by including random data stored in other locations of the QRAM.

The robustness requirement against small perturbation in the  $\infty$ - norm guarantees the overall analysis is insensitive to using the above-described offset rounding convention. Furthermore, note that the success probability of the final projection step is lower

bounded by  $\frac{\epsilon^2}{4}$ . Hence preparing the quantum state that encodes v can succeed independent of the dimensionality, n. This is due to the absence of an exactly representable of 0 in the offset rounding convention. This success probability in state preparation can further be enhanced to  $\Omega(\epsilon)$  with the technique of fixed-point quantum amplitude amplification described in Ref. [61]. Note that the base precision parameter  $\epsilon$  need not be on the order of machine precision. Any values of  $\epsilon$  which is small compared with the known accuracy level of the input data will ensure that the final error is negligible. Generally speaking, low precision data will have a constantly more efficient loading procedure then high precision data. Most importantly, the number of necessary QRAM queries for successful state preparation procedure will always be upper bounded by the inverse of a constant precision parameter which is independent of the size of the database.

In summary, efficient quantum state preparation to encode a classical input vector is possible in any data processing application which is robust under small ∞-norm perturbations. As a consequence, the caveat related to state preparation highlighted by Aaronson in Ref. [26] can generally be overcome in the context of machine learning, due to the inherent robustness assumption. However, this robustness feature not necessarily shared by other application such as computational physics or numerical mathematics where exact vector entries representations could potentially be hard requirements of any meaningful analysis. As one important example of robust applications of quantum machine learning, Gaussian processes are the main topic of this part of the thesis. We will explicitly introduce a state preparation procedure in the next section.

#### 5.1.3 State preparation for GPR

In order to adapt the QRAM based state preparation scheme to Gaussian processes applications, we need to modify it to prepare a state corresponding to the  $s_v$ -sparse

vector v with entries  $v_i$ . We start with a register prepared in a superposition

$$s_{\mathbf{v}}^{-1/2} \sum_{i:v_i \neq 0} |i\rangle \otimes |0\rangle. \tag{5.4}$$

Then we use the index stored in the first register, to conditionally rotate the ancillary register, so that the rotation is based on the ith non-zero entry of  $\mathbf{v}$ . The resultant state of the system is

$$|\tilde{\mathbf{v}}\rangle = \frac{1}{\sqrt{s_{\mathbf{v}}}} \sum_{i:v_i \neq 0} |i\rangle \otimes \left(\sqrt{1 - c_{\mathbf{v}}^2 v_i^2} |0\rangle + c_{\mathbf{v}} v_i |1\rangle\right),$$
 (5.5)

where  $c_{\mathbf{v}} \leq \min_i |v_i|^{-1}$  is the chosen constant to normalise the unitary rotation. Finally, post-selecting on the ancillary register being in state  $|1\rangle$  projects the first register to the required state  $|\mathbf{v}\rangle = \frac{\mathbf{v}}{||\mathbf{v}||}$ . In rare cases, the vector could be vastly dominated by a handful of large value entries, the previously described offset rounding convention can then be applied to ensure a constant success probability in preparing the quantum state for Gaussian processes.

#### 5.2 Quantum Gaussian process algorithm

The essential idea of applying quantum algorithms to GPR comes from the observation that the computation of the central quantities of interest in GPR,  $f_*$  and  $\mathbb{V}[f_*]$ , as written in Eq. 4.12 and Eq. 4.13, involves solving linear systems of the forms  $(K + \sigma_n^2 I)\alpha = \mathbf{y}$  and  $(K + \sigma_n^2 I)\eta = \mathbf{k}_*$  respectively, where  $\mathbf{k}_*^T \alpha = \bar{f}_*$  and  $k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T \eta = \mathbb{V}[f_*]$ . The common linear structure suggests that we can apply the quantum linear system algorithm to extract useful information.

#### **5.2.1** Inner product estimation

As a prerequisite component to the quantum Gaussian process algorithm we here introduce a mechanism to estimate the inner product  $\langle \mathbf{u}|\mathbf{v}\rangle$  for a given pair of real vectors  $\mathbf{u}$  and  $\mathbf{v}$ . Although the squared version,  $|\langle \mathbf{u}|\mathbf{v}\rangle|^2$ , can be easily computed using a controlled-swap test, as presented in Ref. [62], for the purpose of GPs we need to compute both the magnitude and sign of this inner product. Since the controlled-swap test gives the result estimate in terms of a probability, the sign of  $\langle \mathbf{u}|\mathbf{v}\rangle$  is not directly accessible. Thus in order to estimate the inner product, we instead use an augmented version of the state preparation technique, in which an additional ancillary qubit is introduced to determine whether the target state is  $|\mathbf{u}\rangle$  or  $|\mathbf{v}\rangle$ . Specifically we initialise the ancillary qubit in the state

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle),\tag{5.6}$$

which results in a joint state,

$$|\Phi_{\mathbf{u},\mathbf{v}}\rangle = \frac{1}{\sqrt{2s_{\mathbf{u}}}} \sum_{i:u_{i}\neq0} |0\rangle |i\rangle \left(\sqrt{1 - c_{\mathbf{u}}^{2} u_{i}^{2}} |0\rangle + c_{\mathbf{u}} u_{i} |1\rangle\right) + \frac{1}{\sqrt{2s_{\mathbf{v}}}} \sum_{i:v_{i}\neq0} |1\rangle |i\rangle \left(\sqrt{1 - c_{\mathbf{v}}^{2} v_{i}^{2}} |0\rangle + c_{\mathbf{v}} v_{i} |1\rangle\right).$$
(5.7)

Then measuring the operator  $M=X\otimes I\otimes |1\rangle \langle 1|$  results in an expectation value

$$\langle M \rangle = s_{\mathbf{u}}^{-1/2} s_{\mathbf{v}}^{-1/2} c_{\mathbf{u}} c_{\mathbf{v}} \mathbf{u}^{T} \mathbf{v}. \tag{5.8}$$

#### 5.2.2 Procedures

Now we are in a position to introduce a quantum algorithm for computing the quantities of the form  $\mathbf{u}^T A^{-1} \mathbf{v}$ , which can, in turn, be applied to compute the central quantities of

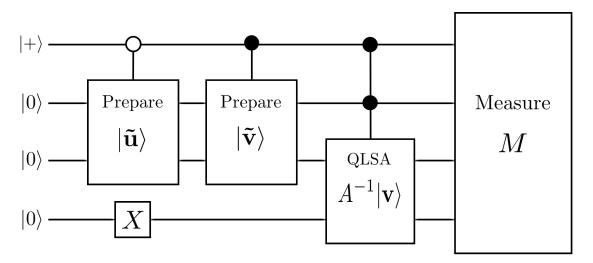
GP regression. To do so, we combine the techniques of state preparation, inner product estimation together with the quantum linear system algorithm (QLSA) described in Chapters 2 and 3. The general procedure is as follows:

- 1. Initialise the system in the state  $|+\rangle_A |0\rangle_B |0\rangle_C |0\rangle_D$ , where the subscripts A, B, C and D label different registers.
- 2. Conditioned on register A being in state  $|0\rangle$ , query the QRAM and prepare registers B and C in the state  $|\tilde{\mathbf{u}}\rangle$ , such that the ancillary qubit is placed in register C with the rest of the state in register B, and apply an X gate to register D.
- 3. Conditioned on register A being in state  $|1\rangle$ , query the QRAM and prepare registers B and C in the state  $|\tilde{\mathbf{v}}\rangle$  such that the ancillary qubit is placed in register C with the rest of the state in register B.
- 4. Conditioned on both registers A and C being in state |1⟩, apply QLSA using B as the input register and using D as the ancillary register. A fifth register E is introduced for the phase estimation subroutine in the QLSA, but since it is eventually uncomputed and returned to the zero state, we will omit it in the description of the states after each step for simplicity.
- 5. Measure the system with the observable  $M=X_AI_B\ket{1}\bra{1}_C\ket{1}\bra{1}_D$ .

The measurement result will be a random variable with an expectation value,

$$\langle M \rangle = c s_{\mathbf{u}}^{-1/2} s_{\mathbf{v}}^{-1/2} c_{\mathbf{u}} c_{\mathbf{v}} \mathbf{u}^{T} A^{-1} \mathbf{v}.$$
 (5.9)

A circuit diagram describing the above procedures for computing  $\mathbf{u}^T A^{-1} \mathbf{v}$  is shown in Figure 5.2.



**Figure 5.2:** Circuit diagram for computing the form  $\mathbf{u}^T A^{-1} \mathbf{v}$ , where  $M = X \otimes I \otimes |1\rangle \langle 1| \otimes |1\rangle \langle 1|$ .

**Derivations** To see the validity of the above algorithm, note that the state of the system after Step 4 is given by

$$\frac{1}{\sqrt{2s_{\mathbf{u}}}} |0\rangle_{A} \sum_{i:u_{i}\neq0} |i\rangle_{B} \left(\sqrt{1-c_{\mathbf{u}}^{2}u_{i}^{2}} |0\rangle_{C} + c_{\mathbf{u}}u_{i} |1\rangle_{C}\right) |1\rangle_{D}$$

$$+ \frac{1}{\sqrt{2s_{\mathbf{v}}}} |1\rangle_{A} \sum_{i:v_{i}\neq0} c_{\mathbf{v}}\beta_{i} |\mu_{i}\rangle_{B} |1\rangle_{C} \left(\sqrt{1-\frac{c^{2}}{\lambda_{i}^{2}}} |0\rangle_{D} + \frac{c}{\lambda_{i}} |1\rangle_{D}\right)$$

$$+ \frac{1}{\sqrt{2s_{\mathbf{v}}}} |1\rangle_{A} \sum_{i:v_{i}\neq0} \sqrt{1-c_{\mathbf{v}}^{2}v_{i}^{2}} |i\rangle_{B} |0\rangle_{C} |0\rangle_{D}, \qquad (5.10)$$

where  $|\mu_i\rangle$  denotes the *i*th eigenvector of A with corresponding eigenvalue  $\lambda_i$ , and  $\{\beta_i\}$  denotes the coordinates of  $\mathbf{v}$  in the basis of  $\{|\mu_i\rangle\}$ . The subsequent projection of this state onto  $|1\rangle$  for registers C and D results the sub-normalised state

$$\frac{1}{\sqrt{2s_{\mathbf{u}}}} |0\rangle_{A} \sum_{i:u_{i}\neq 0} c_{\mathbf{u}} \gamma_{i} |\mu_{i}\rangle_{B} + \frac{1}{\sqrt{2s_{\mathbf{v}}}} |1\rangle_{A} \sum_{i:v_{i}\neq 0}^{n} \frac{c}{\lambda_{i}} c_{\mathbf{v}} \beta_{i} |\mu_{i}\rangle_{B}, \qquad (5.11)$$

where  $\{\gamma_i\}$  are the coordinates of **u** in the basis of  $\{|\mu_i\rangle\}$ . As a result, the expectation value of the final measurement is given by

$$\sum_{i} \frac{1}{4} \left( \left( \frac{c_{\mathbf{u}}}{\sqrt{s_{\mathbf{u}}}} \gamma_{i} + \frac{c_{\mathbf{v}}c}{\sqrt{s_{\mathbf{v}}}} \frac{\beta_{i}}{\lambda_{i}} \right)^{2} - \left( \frac{c_{\mathbf{u}}}{\sqrt{s_{\mathbf{u}}}} \gamma_{i} - \frac{c_{\mathbf{v}}c}{\sqrt{s_{\mathbf{v}}}} \frac{\beta_{i}}{\lambda_{i}} \right)^{2} \right) \\
= \frac{c_{\mathbf{u}}c_{\mathbf{v}}c}{\sqrt{s_{\mathbf{u}}s_{\mathbf{v}}}} \mathbf{u}^{T} A^{-1} \mathbf{v}. \tag{5.12}$$

The expectation value for the measurement in the final step,  $\langle M \rangle$ , must match the above, thus we have

$$\langle M \rangle = \frac{c_{\mathbf{u}} c_{\mathbf{v}} c}{\sqrt{s_{\mathbf{u}} s_{\mathbf{v}}}} \mathbf{u}^T A^{-1} \mathbf{v}. \tag{5.13}$$

It should be noted the estimation of  $\langle M \rangle$  in involves sampling m on repeated runs of the algorithm, which results in a sampling variance that scales as  $m^{-1}$ .

The above-outlined algorithm for estimating the inner product form  $\mathbf{u}^T A^{-1} \mathbf{v}$  can be used to construct a quantum algorithm for approximating both the mean predictor and variance predictor in computing GP regression, which we will illustrate in the following.

### 5.2.3 Mean predictor

In order to approximate the mean predictor,  $\mathbf{k}_*^T(K + \sigma_n^2 I)^{-1}\mathbf{y} = \mathbf{y}^T(K + \sigma_n^2 I)^{-1}\mathbf{k}_*$ , we set  $\mathbf{u} = \mathbf{y}$ ,  $A = K + \sigma_n^2 I$  and  $\mathbf{v} = \mathbf{k}_*$ . Since K is positive semi-definite, the minimum eigenvalue of A is lower bounded by  $\sigma_n^2$ , and hence we take the normalisation constant  $c = \sigma_n^2$  in each run of the QLSA. This leads to

$$\langle M \rangle = \frac{\sigma_n^2 c_{\mathbf{k}_*} c_{\mathbf{y}}}{\sqrt{s_{\mathbf{k}_*} s_{\mathbf{y}}}} \mathbf{y}^T (K + \sigma_n^2 I)^{-1} \mathbf{k}_*, \tag{5.14}$$

and therefore

$$\bar{f}_* = \frac{\sqrt{s_{\mathbf{k}_*} s_{\mathbf{y}}}}{\sigma_n^2 c_{\mathbf{k}_*} c_{\mathbf{y}}} \langle M \rangle. \tag{5.15}$$

Here  $c_{\mathbf{k}_*}$  and  $c_{\mathbf{y}}$  are taken to be the inverted maximum absolute values of the entries in  $\mathbf{k}_*$  and  $\mathbf{y}$  respectively, which we treat as constants. Hence the variance in estimating the value of  $\bar{f}_*$  will scale as  $s_{\mathbf{k}_*}s_{\mathbf{y}}m^{-1}$ . In the case of K being s-sparse, we have  $s_{\mathbf{k}_*} \leq s$  since  $\mathbf{k}_*$  reflects the same dependencies as K. While  $\mathbf{y}$  will not, in general, be sparse, we can instead replace it in the estimation procedure with a vector  $\mathbf{y}'$  with a small number of non-zero entries and still obtain a good approximation to  $\bar{f}_*$ , whenever the spectral norm of  $K + \sigma_n^2 I$  is bounded, which will virtually always be the case for GP regression. This is because of the fact that

$$(K + \sigma_n^2 I)^{-1} = \sum_{d} (-1)^d (K + (\sigma_n^2 - 1)I)^d,$$
 (5.16)

and hence that  $(K + \sigma_n^2 I)^{-1}$  can be approximated by a polynomial in  $(K + (\sigma_n^2 - 1)I)$  of some fixed degree, which will result in a matrix of constant sparsity. Hence  $(K + \sigma_n^2 I)^{-1} \mathbf{k}_*$  will be an approximately sparse vector, and its inner product with  $\mathbf{y}$  can be well approximated by the inner product with a vector  $\mathbf{y}'$  where the only non-zero entries correspond to the location of non-negligible entries of  $(K + \sigma_n^2 I)^{-1} \mathbf{k}_*$ . In conclusion, only a constant number of repetitions of the algorithm is needed to achieve a fixed variance of estimation.

### **5.2.4** Variance predictor

In order to approximate the variance  $V[f_*]$ , the same procedure is followed as for the mean predictor, except that u is now taken to be  $k_*$  instead of y. This yields

$$\langle M \rangle = \frac{\sigma_n^2 c_{\mathbf{k}_*}^2}{s_{\mathbf{k}_*}} \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{k}_*, \tag{5.17}$$

and therefore we have

$$\mathbb{V}[f_*] = k(\mathbf{x}_*, \mathbf{x}_*) - \frac{s_{\mathbf{k}_*}}{\sigma_n^2 c_{\mathbf{k}_*}^2} \langle M \rangle.$$
 (5.18)

As with the mean predictor in Section 5.2.3,  $\langle M \rangle$  needs to be measured on a constant number of independent runs of the algorithm in order to yield a desired fixed variance on the estimate.

#### 5.3 Discussions

We have shown that the QLSA introduced in Section 2.2.2 can be applied to evaluating the two central objective quantities in GPR problems, the mean predictor, and the variance predictor. Inherited from the computational time of QLSA, this quantum GPR procedure achieves an exponential speed-up over classical implementations under two assumptions about the covariance matrix,  $(K + \sigma_n^2 I)$ , namely, the matrix is sparse and well-conditioned. We discuss the practicalities of these assumptions.

Sparsely constructed GP GPs with sparse covariance matrices are of significant interests in many real-world applications, particularly when the problem involves inference from large datasets [63]. For example, these sparsely constructed Gaussian processes are used to make a unified framework for robotic mapping [64]. In the field of pattern recognition, sparsely constructed Gaussian processes have been used to solve realistic action recognition problems [65]. A widely used technique to construct a sparse covariance matrix is setting the covariance function to zero beyond a certain distance between any two data points with a compactly supported function. This is known as covariance tapering and has been proven to approximate the Matérn family of covariance functions with a small squared error [66]. An explicit example in geostatistics kriging where the dataset gives rise to a highly sparse covariance matrix is presented in Ref. [67].

In the above cases where the GPR computation only involves sparse covariance matrices, our proposed algorithm circumvents the major potential caveats of QLSA, and an exponential advantage over its classical counter-part is attainable. For other applications where s scales linearly with n, our algorithm provides a polynomial speed-up over the best-known classical GPR algorithm, even though an exponential speed-up is not always guaranteed.

Conditioning To implement quantum GPR efficiently, the matrix  $(K + \sigma_n^2 I)$  needs to be well-conditioned. The ratio of largest and smallest eigenvalue  $\kappa$  needs to stay low as n increases for the matrix to be robustly invertible. In classical GPR, conditioning is already a well-recognised issue. A general strategy to cope with the problem is to increase the noise variance  $\sigma_n^2 I$  manually by a certain amount to dilute the ratio  $\kappa$  without severely affecting the statistical properties of the model. This increase in  $\sigma_n^2 I$  can be seen as a small amount of noise (jitter) in the input signal. This technique is not new to quantum GPR and may be seen throughout the classical GP literature and mainstream implementations [68]. Therefore, for almost all practical purposes, we can assume the matrix is well-conditioned before applying the quantum algorithm. Moreover, when we apply our algorithm on a sparse kernel, the preconditioning method presented in Ref. [32] can be applied to suppress the growth of  $\kappa$  further. In fact, under the realistic assumption that the maximum entry of a sparse K is bounded by a constant, the maximum eigenvalue of  $(K + \sigma_n^2 I)$  must be bounded by a constant. This is a consequence of the Gershgorin circle theorem [69] which can be expressed in terms of the following inequality,

$$|\lambda - A_{ii}| \le \sum_{j \ne i} |A_{ij}|. \tag{5.19}$$

Note that since  $A=(K+\sigma_n^2I)$ , the minimum eigenvalue of A is lower bounded by  $\sigma_n^2$ . Likewise, we have the diagonal elements bounded by  $A_{ii} \geq \sigma_n^2$  and the off-diagonal sum

 $\sum_{j\neq i} |A_{ij}|$  upper bounded by the sparsity of A scaled by the magnitude of its maximum entry. Hence from Eq. 5.19 we deduce the maximum eigenvalue of A is upper bounded by a constant that is independent of n. As a result, under the sparse and bounded element kernel matrix assumption, conditioning does not provide a barrier to our proposed quantum GPR algorithm. In summary, we have argued that conditioning does not hinder the application of quantum GPR, and the algorithm is most advantageous when one is concerned with a sparse kernel. Under such circumstances, an exponential speed-up is achievable. Hence having addressed all the major potential caveats of QLSA [26], the quantum GPR algorithm is shown to be a robust application with practical significance.

## Chapter 6

## Training quantum Gaussian processes

As presented in the previous chapter, the quantum Gaussian process algorithm provides a speed-up in computing predictions and the associated variances given a fixed kernel. It is desirable to also have a correspondingly efficient quantum routine for kernel and hyperparameter selection. In particular, it would be desirable to evaluate a measure of the model's performance with a quantum routine that supplements the main learning algorithm. With this motivation, we propose a quantum approach to improve the efficiency of GP training based on evaluating the logarithm of marginal likelihood (LML) of the Gaussian distribution of the observed data. The material of this chapter is based on the work of Ref. [44].

## **6.1 Quantum** LML algorithm

Here we introduce a quantum algorithm for estimating the LML given the kernel matrix of a Gaussian process, which serves as the standard metric for a kernel's performance on the given data set. The complete estimation of LML is obtained by combining the "penalty" and the "data fit" terms. For the purpose of GP training, we are concerned with estimating the variation,  $\delta$ LML, with respect to a training step, where the prefix  $\delta$  denotes the variation in a quantity between training steps.

#### 6.1.1 Augmented linear algorithm

The data fit term of the LML Eq. 4.14,  $\frac{1}{2}\mathbf{y}^T(K+\sigma_n^2)^{-1}\mathbf{y}$  relates the outputs  $\mathbf{y}$  to the covariance matrix K. Here we demonstrate a modified version of the QLSA [27], and show that it can be used to calculate the data fit term. As discussed in Chapter 2, The QLSA makes use of the quantum phase estimation to obtain the superposition of the eigenvalues,  $\lambda_i$  of  $A \in \mathbb{R}^{n \times n}$  encoded in the form of binary bit-strings, where A is the matrix in the linear system  $A | \mathbf{x} \rangle = | \mathbf{b} \rangle$ . An ancillary qubit is then rotated conditioned on the values of  $f(\lambda_i)$ . In the case of the original linear system algorithm, the function f is simply chosen to be  $f(\lambda) = 1/\lambda$ . Post-selecting this ancillary qubit followed by the reversal of the phase estimation step results in finding  $A^{-1} | \mathbf{b} \rangle$  with success probability  $\langle \mathbf{b} | (A^{-1})^{\dagger} A^{-1} | \mathbf{b} \rangle$ . As noted in Ref. [27], the same method can be extended to obtain  $f(A) | \mathbf{b} \rangle$  for any computable function f.

Here we apply an augmented version of the QLSA by choosing  $f(\lambda) = \frac{1}{\sqrt{\lambda}}$  instead of the original inversion. The procedure for estimating the data fit term is given as follows:

- 1. Use QRAM queries to prepare  $|\mathbf{y}\rangle = \frac{\mathbf{y}}{\|\mathbf{y}\|}$  with the state preparation technique described in Section 5.1.
- 2. Set  $|\mathbf{b}\rangle = |\mathbf{y}\rangle$  and  $A = K + \sigma_n^2 I$ , and run the augmented QLSA with  $f(\lambda) = \frac{1}{\sqrt{\lambda}}$ , which leads to  $A^{-\frac{1}{2}} |\mathbf{y}\rangle$  with success probability  $\langle \mathbf{y} | A^{-1} | \mathbf{y} \rangle$
- 3. Sampling on multiple runs of the augmented QLSA thus gives a Monte Carlo estimate of the data fit term with mean  $\mathbf{y}^T A^{-1} \mathbf{y}$  and variance bounded by  $\frac{1}{4} ||\mathbf{y}||^2 \sigma_n^{-2}$ .

Note that on top of leading to the desired estimation for the data fit term, this choice of  $f(\lambda)$  also reduces the inconvenient effect of poor conditioning by a square-root as the success probability of the measurement step is increased as  $\sqrt{\lambda} \ge \lambda$  for all  $0 \le \lambda \le 1$ . When A is well-conditioned and sparse, the runtime of sampling from such a distribution

is logarithmic in the dimension of y, inherited from the computational cost of QLSA in Ref. [27].

#### **6.1.2** Log determinant algorithm

The second term of the LML in Eq. 4.14,  $-\frac{1}{2} \log \det[K + \sigma_n^2 I]$  can be estimated via a quantum algorithm that samples the eigenvalues of a Hermitian matrix  $A \in \mathbb{R}^{n \times n}$  uniformly at random. The algorithm proceeds as follows:

1. Prepare  $\log_2 n$  qubits in maximally mixed state,  $\frac{1}{n}\sum_{i=1}^n |i\rangle\langle i|$ , and store this in a first register. This can be achieved simply by preparing the register in a random computational basis state. Note that a maximally mixed state is maximally mixed in any basis, hence we can choose to represent the density matrix for the system in the eigenbasis  $\{|e_i\rangle\}$  of a matrix  $A=K+\sigma_n^2I$ :

$$\frac{1}{n} \sum_{i=1}^{n} |e_i\rangle \langle e_i|. \tag{6.1}$$

2. Append a second register in a superposition state given by  $\frac{1}{\sqrt{T}}\sum_{\tau=1}^{T}|\tau\rangle$ , so that the composite system is in the state

$$\frac{1}{nT} \sum_{\tau,\tau'=1}^{T} \sum_{i=1}^{n} |e_i\rangle \langle e_i| \otimes |\tau\rangle \langle \tau'|, \qquad (6.2)$$

where the time period parameter T is chosen to be a sufficiently large value in the same way as in Eq. 2.10.

3. Perform a Hamiltonian simulation and evolve the first register with the Hermitian matrix (-A) for time specified by the second register. This is achieved by applying the conditional unitary evolution  $\sum_{\tau=1}^{T} \mathrm{e}^{iAt_0\tau/T} \otimes |\tau\rangle \langle \tau|$ , where  $t_0 = O(1/\epsilon)$  is

chosen with respect to the  $\epsilon$ -bounded error required in the algorithm. We thus obtain

$$\frac{1}{nT} \sum_{\tau,\tau'=1}^{T} \sum_{i=1}^{n} e^{i\lambda_i t_0(\tau-\tau')/T} |e_i\rangle \langle e_i| \otimes |\tau\rangle \langle \tau'|.$$
(6.3)

4. Complete the phase estimation by performing a quantum Fourier transform of the second register. The resulting estimated eigenvalues of A,  $\{\lambda_i\}$ , are then stored in the second register as a binary bit-string up to a finite precision. This results in the system being in state,

$$\frac{1}{n} \sum_{i=1}^{n} |e_i\rangle \langle e_i| \otimes |\lambda_i\rangle \langle \lambda_i|. \tag{6.4}$$

5. Measure the second register in computational basis to obtain a random  $\lambda_i$ . By using the identity,

$$\langle \log \lambda_i \rangle = \frac{1}{n} \sum_{i=1}^n \log \lambda_i = \frac{1}{n} \operatorname{Tr}[\log(A)] = \frac{1}{n} \log[\det(A)],$$
 (6.5)

The desired quantity  $\log[\det(A)]$  is given then by  $n\langle \log \lambda_i \rangle$ , which will needs to be estimated by sampling eigenvalues of A on repeated runs of the procedure.

Hence the "penalty" term of the LML can be estimated using the above eigenvalue sampling procedure, by setting  $A = K + \sigma_n^2 I$ . This procedure can be seen as a finite dimensional analogue of the continuous variable model proposed in Ref. [70].

**Runtime** The optimised phase estimation procedure [28,71] comes with an error,  $\epsilon_{\lambda_i}$ , which scales as  $\mathcal{O}(1/t_0)$  in estimating each  $\lambda_i$ . This implies the error associated with the logarithm of a single eigenvalue scales as  $\epsilon = \left|\frac{d \log \lambda_i}{d \lambda_i} \epsilon_{\lambda_i}\right| = \mathcal{O}\left(\frac{1}{\lambda_i t_0}\right)$ . Furthermore, in the context of GP training, there generally exists a  $\sigma_n^2 I$  noise contribution to the

covariance matrix, due to uncertainty in the observed data. Thus, in general, we have the minimum eigenvalue,  $\lambda_{\min} \geq \sigma_n^2$ . Hence, the total bounded-error single-run of the algorithm takes time scaling logarithmically in n as  $t = \tilde{\mathcal{O}}\left(\frac{s\log n}{\sigma_n^2\epsilon}\right)$ .

Due to the linear sparsity dependence from the Hamiltonian simulation step, the proposed quantum algorithm performs best when the covariance matrix is some constant *s*-sparse, in which case our algorithm provides an exponential speed-up over the classical GP training procedure. Such sparsely constructed GPs have found applications in a range of interesting problems, especially when large datasets are involved [63], as discussed in Chapter 5.

When dealing with non-sparse but low-rank matrices, another technique of Hamiltonian simulation involving density matrix exponentiation [21] can potentially be applied. Note that the covariance matrices are by definition symmetric, real and positive semidefinite, and therefore have a very similar mathematical structure to the density matrix representation of quantum states. Hence this seminal technique of density matrix exponentiation potentially allows us to implement  $e^{-iAt}$  in  $\tilde{\mathcal{O}}(\log n)$  time, even if the matrix is not sparse. However, the covariance matrix needs to be normalised to have a unit trace for the application of density matrix exponentiation. This pre-processing can be done efficiently if one can exploit the analytical structure of the covariance matrix. Also note that if the eigenvalues of the covariance matrix are relatively uniform, the time required to implement the unitary for a complete cycle will scale as  $\mathcal{O}(n)$ . Hence applying density matrix exponentiation is most effective when the covariance matrix is approximately low-rank [21].

**Stochastic trace estimation** We briefly compare the quantum log determinant algorithm with classical stochastic trace estimation methods. It is clear that the quantum algorithm offers a precise method to compute  $\log(A)$  rather than either the truncated Taylor series or Chebyshev polynomial approximations. When measurements of the

second register are taken, a single  $\log(\lambda_i)$  is computed, and hence our proposed approach can be seen as quantum stochastic trace estimation. The main advantage, however, comes from the reduction in computation time from polynomial to sub-linear. A natural question which arises is whether the complete GP training can scale sub-linearly in n, since if not, an exponential improvement in computing the LML in each step would yield only a polynomial improvement in precision.

#### **6.2** Variation estimation

The figure of merit for the estimation error is the relative variance, as it quantifies the amount of dispersion between the estimated and the actual value of LML. In order to demonstrate the quantum advantage in the training process, it is therefore necessary to show that the relative variance with respect to a change in hyperparemeter,  $\delta\theta$ , does not scale up with n. We consider the following,

$$\frac{\operatorname{Var}\left[\delta \operatorname{LML}\right]}{\left[\delta \operatorname{LML}\right]^{2}} = \frac{\delta \left[\log[\det(A)]\right] + \delta \left[\mathbf{y}^{T} A^{-1} \mathbf{y}\right]}{\left[\frac{\partial}{\partial \theta} \left(\log[\det(A)] + \mathbf{y}^{T} A^{-1} \mathbf{y}\right) \delta \theta\right]^{2}}.$$
(6.6)

Now we write the  $\mathbf{y}$  as a linear combination of the eigenvectors,  $\mathbf{e}_i$  of A, such that  $\mathbf{y} = \sum_i \gamma_i \mathbf{e}_i$ , and  $\mathbf{y}^T A^{-1} \mathbf{y} = \sum_i |\gamma_i|^2 \lambda_i^{-1}$ , we have

$$\frac{\delta \left[\log[\det(A)]\right] + \delta \left[\mathbf{y}^{T} A^{-1} \mathbf{y}\right]}{\left[\frac{\partial}{\partial \theta} \left(\log[\det(A)] + \mathbf{y}^{T} A^{-1} \mathbf{y}\right) \delta \theta\right]^{2}} \leq \frac{n^{2} \left(\delta \left[\log \lambda_{i}\right] + \frac{1}{4} \left\langle y_{i}^{2} \right\rangle \sigma_{n}^{-2}\right)}{\left[\frac{\partial}{\partial \theta} \left(\sum_{i} \log \lambda_{i} + \sum_{i} |\gamma_{i}|^{2} \lambda_{i}^{-1}\right) \delta \theta\right]^{2}} \\
\leq \frac{\left\langle (\log \lambda_{i})^{2} \right\rangle + \frac{1}{4} \left\langle y_{i}^{2} \right\rangle \sigma_{n}^{-2}}{\left\langle \delta \lambda_{i} / \lambda_{i} + \delta \left(|\gamma_{i}|^{2} / \lambda_{i}\right)\right\rangle^{2}}, \tag{6.7}$$

where the expectation value notation is used to denote the average over all choices of i. Hence the relative variance in estimating the variation of LML with respect to a training step has no explicit dependence on n.

Note that the number of hyperparameters is dependent only on the kernel, and thus potentially independent of the number of data points. Provided we are working to constant precision, the number of optimisation steps which require LML computation is upper bounded by a constant.

### **6.3** Summary

We have shown a quantum procedure for calculating LML which improves the efficiency from a classical  $\mathcal{O}(n^3)$  scaling to a logarithmic scaling with respect to the size of input under certain assumptions. Specifically, if either the structure of the covariance matrix is constant s-sparse or approximately low-rank, the quantum approach provides an exponential speed-up. Even in the cases when the Hamiltonian simulation step inevitably consumes a  $\tilde{O}(n\log n)$  time overhead, this quantum algorithm still achieves a polynomial speed-up over the best known classical approach to training full-rank GPs. When applied to a non-sparse covariance matrix that has a low-rank structure, the density matrix exponentiation procedure [21] can still lead to a logarithmic time algorithm. In other cases, the singular value estimation based linear system algorithm presented in Chapter 3 can be applied to achieve a runtime that scales as  $\mathcal{O}(\sqrt{n}\log n)$ , which provides a polynomial speed-up over its best known classical counterpart, provided that the spectral norm of A is bounded by a constant with respect to the growth of n.

The quantum GP training procedure presented in this chapter provides an efficient way to evaluate the performance of a given kernel matrix, which is a crucial component of the model selection problem in supervised learning. This procedure applied in conjunction with the quantum GP algorithm in Chapter 5 provides a complete quantum approach for statistical inference with GP models, which can lead to an exponential or polynomial speedup over its best-known classical counterpart, depending on the specific kernel matrix structures.

# **Chapter 7**

# **Quantum Bayesian Deep Learning**

We have presented a complete quantum approach to supervised learning with Gaussian processes in Chapters 5 and 6. By now we have seen the quantum algorithms for computing the predictive mean and variance of a GP posterior as well as the LML which is the core component of training a GP model. In this chapter, we exploit the connection between GPs and neural networks as discussed in Section 4.4, and apply the quantum enhanced GPs to design a quantum algorithm for deep learning. We will also experimentally demonstrate the algorithm on contemporary quantum computers and analyse its robustness with respect to realistic noise models. Specifically, we will make use of both the Rigetti Forest [72] and the IBM QISKit [73] software stacks to implement the quantum algorithm and provide an analysis of the performance of simulators under a realistic noise model. When using real quantum processing units, we implement a simplified, shallow-circuit version of the algorithm, and compare the outcome with the simulations. The results presented in this chapter are based on Ref. [74].

## 7.1 Quantum Bayesian training of neural networks

Bayesian methods provide great advantages compared to traditional techniques in machine learning, which include automated ways of learning structure and avoiding overfitting, robustness to adversarial attacks [75,76] and the ability to estimate uncertainties associated with predictions as previously discussed. The Bayesian framework has novelly been extended to various deep architectures [77,78]. Recent advances in this direction have established a connection between deep feedforward neural networks and Gaussian processes. This connection novelly allows for Bayesian training of deep neural networks with a Gaussian prior, circumventing the more traditional backpropagation procedure [49,51]. We have briefly reviewed this correspondence between GP and deep neural networks in Section 4.4. Recall that the base case covariance matrix  $K^0$  has elements

$$K^{0}(\mathbf{x}, \mathbf{x}') = \sigma_{b}^{2} + \sigma_{w}^{2} \left( \frac{\mathbf{x} \cdot \mathbf{x}'}{d_{in}} \right). \tag{7.1}$$

To compute the covariance matrix corresponding to the  $l^{th}$  layer of the network, we use the following recursive formula to forward propagate the kernel,

$$K^{l}(\mathbf{x}, \mathbf{x}') = \sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(z_i^{l-1}(\mathbf{x}))\phi(z_i^{l-1}(\mathbf{x}'))]. \tag{7.2}$$

For a general non-linear activation function  $\phi$ , this can only be evaluated with numerical integration. Therefore a complete quantum algorithm for general activation functions is likely to be untraceable. Fortunately, there is a useful special case in which only the ReLU activation function,  $f(x) = \max(0, x)$ , is used on each layer. In this case, the  $l^{th}$ 

layer covariance function has the following analytical form [49]:

$$K^{l}(\mathbf{x}, \mathbf{x}') = \sigma_{b}^{2} + \frac{\sigma_{w}^{2}}{2\pi} \sqrt{K^{l-1}(\mathbf{x}', \mathbf{x}')K^{l-1}(\mathbf{x}, \mathbf{x})} \left(\arcsin\left(\theta_{\mathbf{x}, \mathbf{x}'}^{l-1}\right) - (\pi - \theta_{\mathbf{x}, \mathbf{x}'}^{l-1}) \arccos\left(\theta_{\mathbf{x}, \mathbf{x}'}^{l-1}\right)\right),$$

$$(7.3)$$

where

$$\theta_{\mathbf{x},\mathbf{x}'}^l = \arccos\left(\frac{K^l(\mathbf{x},\mathbf{x}')}{\sqrt{K^l(\mathbf{x},\mathbf{x})K^l(\mathbf{x}'\mathbf{x}')}}\right).$$
 (7.4)

Note that the non-linear functions featured in Eq. 7.3 can be approximated by polynomial series with certain convergence conditions. The factor  $K^l(x,x)K^l(x',x')$  represents outer products between the two identical vectors of diagonal entries in  $K^l$ . As such, the computation of Eq. 7.3 can be decomposed into such outer product operations combined with element-wise matrix multiplication. For a L-layer infinite width neural network, the formula Eq. 7.3 needs to be evaluated for all positive integer values of  $l \leq L$ .

Applying quantum GP Recall that the quantum GP algorithm in Chapter 5 computes the mean predictor,  $\bar{f}_* = \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{y}$  and the variance predictor,  $\mathbb{V}[f_*] = k (\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{k}_*$  of a GP posterior, where  $(K + \sigma_n^2 I)$  is the covariance matrix with Gaussian noise entries of variance  $\sigma_n^2$ , and  $\mathbf{k}_*$  is the row in the covariance matrix that corresponds to the target point for prediction. Assuming the oracular access to the matrix elements of K, the quantum GP algorithm simulates  $(K + \sigma_n^2 I)$  as a Hamiltonian acting on an input state,  $|\mathbf{b}\rangle$ , and performs phase estimation to extract the eigenvalues of  $(K + \sigma_n^2 I)$ . By inverting the eigenvalues in a superposition and performing a controlled-rotation on an ancillary system base on the inverted eigenvalues, the algorithm probabilistically completes a computation of  $(K + \sigma_n^2 I)^{-1} |\mathbf{b}\rangle$ . We then use a quantum

inner product estimation procedure to obtain a good estimate for  $\mathbf{k}_*^T(K+\sigma_n^2I)^{-1}\mathbf{b}$ . The encoding state  $|\mathbf{b}\rangle$  is chosen to be  $|\mathbf{b}\rangle = |\mathbf{y}\rangle$  or  $|\mathbf{b}\rangle = |\mathbf{k}_*\rangle$  for computing the mean or variance predictor respectively. To apply the quantum GP algorithm for the Bayesian training of a L-layer infinite width neural network, we simply use  $\{\mathbf x_i\}_{i=1}^n$  and  $\mathbf y$  to represent the input and output points of the training set of the neural network, and choose the elements of K by evaluating the covariance function  $K^L(\mathbf{x}, \mathbf{x}')$ . The non-trivial extension to the quantum GP algorithm needed is for coherently evaluating  $K^{L}(\mathbf{x}, \mathbf{x}')$ , which we will address in the following Sections 7.1.1, 7.1.2 and 7.1.3. It is important to clearly state the assumptions about how the matrix elements of  $K^0$  can be accessed. We consider the following two different (but related) models: Firstly, we can assume black-box access to the elements of  $K^0$ . In this model, the Hamiltonian simulation subroutine discussed in Section 2.1.3 can be directly used in the quantum GP algorithm. Secondly, we can assume that  $K^0$  is presented as the quantum density matrix of a qubit system. Multiple copies of such a density matrix allow for a technique inspired by the quantum principle component analysis algorithm [79]. We will use the first model for the simplest case of a single-layer network and the second model for the multiple-layer deep architecture.

#### 7.1.1 Single-layer case

For the simplest single-layer case, we assume black-box access to the matrix elements of the base case such that we have the oracle  $O_{K^0}$  to perform the following mapping,

$$O_{K^0} |j,k\rangle |z\rangle \to |j,k\rangle |z \oplus K^0_{ik}\rangle,$$
 (7.5)

where the matrix elements are denoted as  $K_{jk}^0 = K^0(\mathbf{x}_j, \mathbf{x}_k)$ . The desired kernel function of Eq. 7.3 can be implemented by direct classical computation on oracle queries. The

desired covariance matrix,  $K^1$  is then simulated as a Hamiltonian, as discussed in Section 2.1.3, in order to construct the controlled unitary operation needed for the quantum GP algorithm.

#### 7.1.2 Multi-layer case

In the case of multi-layer network architectures, we describe a method to simulate the  $l^{th}$  layer kernel matrix as a Hamiltonian. Our approach is inspired by the quantum principle component analysis algorithm [79] where the density matrix  $\rho$  of a quantum state is treated as a Hamiltonian and used to construct the desired controlled unitary  $e^{it\rho}$  acting on a target quantum state for a time period t. A thorough description of this density matrix-based Hamiltonian simulation procedure is presented in Ref. [20]. Here we will first give an overview of the quantum method, while the detailed analysis is presented later in Section 7.1.3.

To apply density matrix-based Hamiltonian simulation using the  $l^{th}$  layer covariance matrix, we need to incorporate techniques to compute certain element-wise matrix operations between two density matrices. It is convenient to define the following:

$$S_1 = \sum_{j,k} |j\rangle\langle k| \otimes |j\rangle\langle k| \otimes |k\rangle\langle j|, \tag{7.6}$$

$$S_2 = \sum_{j,k} |j\rangle\langle j| \otimes |k\rangle\langle k| \otimes |k\rangle\langle j|. \tag{7.7}$$

With an augmented version of the density matrix exponentiation scheme of Ref. [79],  $S_1$  computes the exponential of the Hadamard product of two density matrices, while  $S_2$  computes the exponential of the outer product between the diagonal entries of two

density matrices. Specifically, we have

$$\operatorname{tr}_{1,2}\left\{e^{-iS_1\delta}(\rho_1\otimes\rho_2\otimes\sigma)e^{iS_1\delta}\right\} = \exp[-i(\rho_1\odot\rho_2)\delta]\sigma\exp[i(\rho_1\odot\rho_2)\delta] + \mathcal{O}(\delta^2),$$
(7.8)

where  $\rho_1 \odot \rho_2$  denotes the Hadamard product between  $\rho_1$  and  $\rho_2$ , and  $\mathrm{tr}_{1,2}$  denotes the partial trace over the first and second subsystems. The factor  $\delta$  represents a small evolution time. We also have

$$\operatorname{tr}_{1,2}\left\{e^{-iS_2\delta}(\rho_1\otimes\rho_2\otimes\sigma)e^{iS_2\delta}\right\} = \exp\left[-i(\rho_1\otimes\rho_2)\delta\right]\sigma\exp\left[i(\rho_1\otimes\rho_2)\delta\right] + \mathcal{O}(\delta^2),$$
(7.9)

where  $\rho_1 \oslash \rho_2$  denotes the outer product between the diagonal entries of  $\rho_1$  and  $\rho_2$ . The derivation of Eq. 7.8 and Eq. 7.9 are presented in Section 7.1.3. Both  $S_1$  and  $S_2$  are sparse and hence can be efficiently simulated as Hamiltonians with quantum walk based algorithms [15,17]. We then need to make use of some polynomial series in  $K^0(x,x')$  to approximately compute  $K^l(x,x')$ . Note that the products involved in this polynomial are the Hadamard product denoted by  $\odot$ , and the diagonal outer product denoted by  $\odot$ . We will denote the polynomial in  $K^0$  to the order N(l) which approximates the  $l^{th}$  layer kernel function as  $P^N_{\odot,\odot}(K^0)$ . By using a generalised  $\tilde{S}$  operator which combines the components in  $S_1$  and  $S_2$ , one can implement a total number N of  $\odot$  and  $\odot$  operations in arbitrary orders. In Section 7.1.3, we will show this simply amounts to summing over the tensor product of the projectors  $|j\rangle\langle j|$ ,  $|j\rangle\langle k|$ , and  $|k\rangle\langle k|$ . Similar polynomial series simulation problems were addressed in Refs. [20, 80], but the type of product considered in these works was standard matrix multiplication instead of element-wise operations.

The method described above allows for approximately implementing the operation  $e^{itK^l}\sigma e^{-itK^l}$ , where  $\sigma$  is an arbitrary input state which in our case is taken to be  $\sigma = |\mathbf{b}\rangle\langle\mathbf{b}|$ . Thus given multiple copies of a density matrix which encodes the initial layer

covariance matrix,  $K^0$ , the unitary operator,  $\exp(-itK^l)$  can be constructed to act on an arbitrary input state, as required by applying the quantum GP algorithm.

#### 7.1.3 Coherent element-wise operations

In this section, we give a more formal description of the quantum method to compute the polynomial  $P_{\odot,\oslash}^N(K^0)$ . The main results needed are summarised by the following Lemmas 1 and 2, and Theorem 1.

**Lemma 1** (Hadamard product simulation [74]). Given  $O(t^2/\epsilon)$  copies of d-dimensional qubit density matrices,  $\rho_1$  and  $\rho_2$ , let  $\rho_1 \odot \rho_2$  denote the Hadamard product between  $\rho_1$  and  $\rho_2$ . There exists a quantum algorithm to implement the unitary  $e^{-i\rho_1 \odot \rho_2 t}$  on a d-dimensional qubit input state  $\sigma$ , for a time t to accuracy  $\epsilon$  in operator norm.

*Proof.* The usual SWAP matrix for quantum principal component analysis [79] is given by  $S = \sum_{j,k} |j\rangle\langle k| \otimes |k\rangle\langle j|$ . Here we take the modified SWAP operator  $S_1 = \sum_{j,k} |j\rangle\langle k| \otimes |j\rangle\langle k| \otimes |k\rangle\langle j|$ . With an arbitrary input state  $\sigma$ , the following operation can be efficiently approximated for small  $\delta$ :

$$\operatorname{tr}_{1,2}\left\{e^{-iS_1\delta}(\rho_1\otimes\rho_2\otimes\sigma)e^{iS_1\delta}\right\},\tag{7.10}$$

The trace is over the subspaces of  $\rho_1$  and  $\rho_2$ . Expanding to  $\mathcal{O}(\delta^2)$  leads to

$$\operatorname{tr}_{1,2}\left\{e^{-iS_1\delta}(\rho_1\otimes\rho_2\otimes\sigma)e^{iS_1\delta}\right\}$$

$$=1-i\operatorname{tr}_{1,2}\left\{S_1(\rho_1\otimes\rho_2\otimes\sigma)\right\}\delta+i\operatorname{tr}_{1,2}\left\{(\rho_1\otimes\rho_2\otimes\sigma)S_1\right\}\delta+\mathcal{O}(\delta^2).$$
(7.11)

Examining the first  $\mathcal{O}(\delta)$  reveals

$$\operatorname{tr}_{1,2}\{S_{1}(\rho_{1}\otimes\rho_{2}\otimes\sigma)\} = \operatorname{tr}_{1,2}\{\sum_{j,k}|j\rangle\langle k|\otimes|j\rangle\langle k|\otimes|k\rangle\langle j|(\rho_{1}\otimes\rho_{2}\otimes\sigma)\}$$

$$= \sum_{n,m,j,k}\langle n|j\rangle\langle k|\rho_{1}|n\rangle\langle m|j\rangle\langle k|\rho_{2}|m\rangle|k\rangle\langle j|\sigma$$

$$= \sum_{j,k}\langle k|\rho_{1}|j\rangle\langle k|\rho_{2}|j\rangle|k\rangle\langle j|\sigma$$

$$= (\rho_{1}\odot\rho_{2})\sigma. \tag{7.12}$$

In the same manner we have

$$\operatorname{tr}_{1,2}\{(\rho_1 \otimes \rho_2 \otimes \sigma)S_1\} = \sigma(\rho_1 \odot \rho_2). \tag{7.13}$$

Thus in summary, we have shown that

$$\operatorname{tr}_{1,2}\left\{e^{-iS_1\delta}(\rho_1\otimes\rho_2\otimes\sigma)e^{iS_1\delta}\right\} = \sigma - i[(\rho_1\odot\rho_2),\sigma]\delta + \mathcal{O}(\delta^2). \tag{7.14}$$

The above is equivalent to applying the unitary  $\exp[-i(\rho_1 \odot \rho_2)\delta]$  to  $\sigma$  up to  $\mathcal{O}(\delta^2)$ :

$$\exp[-i(\rho_1 \odot \rho_2)\delta]\sigma \exp[i(\rho_1 \odot \rho_2)\delta]$$

$$=[I - i(\rho_1 \odot \rho_2)\delta + \mathcal{O}(\delta^2)]\sigma[I + i(\rho_1 \odot \rho_2)\delta + \mathcal{O}(\delta^2)]$$

$$=\sigma - i[(\rho_1 \odot \rho_2), \sigma]\delta + \mathcal{O}(\delta^2). \tag{7.15}$$

Comparing the above two equations validates Eq. 7.8. Note that if the small time parameter is taken to be  $\delta = \epsilon/t$ , and the above procedure is implemented  $\mathcal{O}(t^2/\epsilon)$  times, the overall effect amounts to implementing the desired operation,  $e^{-i\rho t}\sigma e^{i\rho t}$  up to an error  $\mathcal{O}(\delta^2 t^2/\epsilon) = \mathcal{O}(\epsilon)$ , while consuming  $\mathcal{O}(t^2/\epsilon)$  copies of  $\rho_1$  and  $\rho_2$ . This concludes the proof of Lemma 1.

Note that an alternative approach for Hadamard product simulation is described in [81], where the input are given as Hamiltonian on the exponents of unitary operators, rather than density matrices as discussed here.

**Lemma 2** (Diagonal outer product simulation [74]). Given  $O(t^2/\epsilon)$  copies of d-dimensional qubit density matrices,  $\rho_1$  and  $\rho_2$ , let  $\rho_1 \oslash \rho_2$  denote the outer product between the diagonal entries of  $\rho_1$  and  $\rho_2$ . There exists a quantum algorithm to implement the unitary  $e^{-i\rho_1 \oslash \rho_2 t}$  on a d-dimensional qubit input state,  $\sigma$ , for a time t to accuracy  $\epsilon$  in operator norm.

*Proof.* By simply re-indexing the  $S_1$  operator, one obtains  $S_2 = \sum_{j,k} |j\rangle\langle j| \otimes |k\rangle\langle k| \otimes |k\rangle\langle j|$ . Analogously with the proof of Lemma 1, we have

$$\operatorname{tr}_{1,2}\left\{e^{-iS_2\delta}(\rho_1\otimes\rho_2\otimes\sigma)e^{iS_1\delta}\right\} = \sigma - i[(\rho_1\otimes\rho_2),\sigma]\delta + \mathcal{O}(\delta^2). \tag{7.16}$$

The above equation can be compared with

$$\exp[-i(\rho_1 \otimes \rho_2)\delta]\sigma \exp[i(\rho_1 \otimes \rho_2)\delta] = \sigma - i[(\rho_1 \otimes \rho_2), \sigma]\delta + \mathcal{O}(\delta^2). \tag{7.17}$$

The equivalence up to the linear term in  $\delta$  validates of Eq. 7.9. As with Lemma 1, with  $\mathcal{O}(t^2/\epsilon)$  repetitions consuming  $\mathcal{O}(t^2/\epsilon)$  copies of  $\rho_1$  and  $\rho_2$ , the desired  $e^{-i\rho t}\sigma e^{i\rho t}$  can be implemented up to error  $\epsilon$ .

Given the density matrix  $\rho=K^0$  which encodes the base case covariance matrix, we approximate the non-linear kernel function at  $l^{th}$  layer with the order N polynomial,  $P^N_{(\odot,\oslash)}(\rho)=\sum_r^N c_r \rho^{(\odot,\oslash)r}$ . Here the label  $(\odot,\oslash)$  indicates that we work in the setting where the types of product operation involved for taking the  $r^{th}$  power of  $\rho$  are arbitrary combinations of Hadamard products and diagonal outer products. Now we are in the

position of presenting the main theorem required to implement the kernel function at the  $l^{th}$  layer.

**Theorem 1** (Element-wise polynomial simulation [74]). Given  $\mathcal{O}(N^2t^2/\epsilon)$  copies of the d-dimensional qubit density matrix  $\rho$ , and the order-N polynomial of Hadamard and diagonal outer products,  $P_{\odot,\odot}^N(\rho) = \sum_r^N c_r \rho^{(\odot,\odot)r}$ , there exists a quantum algorithm to implement the unitary  $e^{-iP_{(\odot,\odot)}^N(\rho)t}$  on a d-dimensional qubit input state  $\sigma$  for a time t to accuracy  $\epsilon$  in operator norm.

*Proof.* We first address how to implement the unitary  $e^{-i\rho^{(\odot,\odot)}r_t}$ . Intuitively, this can be achieved by constructing a generalized  $\tilde{S}$  operator with tensor product components of  $|j\rangle\langle j|$ ,  $|j\rangle\langle k|$ ,  $|k\rangle\langle k|$  and  $|k\rangle\langle j|$ , corresponding to the contributing elements in the matrices in each term. We give a recursive procedure to determine  $\tilde{S}$ :

In the case of r=2, we have already shown in Lemma 1 and Lemma 2 the desired operation can be achieved using  $S_1$  and  $S_2$  corresponding to the  $\odot$  and  $\oslash$  cases respectively. Thus we can write the base case of the recursive procedure as

$$\tilde{S}^{(r=2)} = \sum_{j,k} T^{(2)}(j,k) \otimes |k\rangle\langle j|, \tag{7.18}$$

where  $T^{(2)}(j,k)$  denotes the possible combinations of tensor products,  $|j\rangle\langle k|\otimes|j\rangle\langle k|$  or  $|j\rangle\langle j|\otimes|k\rangle\langle k|$ . Now consider the r=3 case, the additional factor of  $\rho$  will come in two possible cases. If it comes as a  $\odot$  product, the updated operator  $\tilde{S}^{(r=3)}_{\odot}$  is simply given by

$$\tilde{S}_{\odot}^{(r=3)} = \sum_{j,k} T^{(2)}(j,k) \otimes |j\rangle\langle k| \otimes |k\rangle\langle j|. \tag{7.19}$$

If the additional  $\rho$  comes in as a  $\oslash$  product, the updated operator  $\tilde{S}^{(r=3)}_{\oslash}$  is instead given by

$$\tilde{S}_{\oslash}^{(r=3)} = \sum_{j,k} |j\rangle\langle j| \otimes |j\rangle\langle j| \otimes |k\rangle\langle k| \otimes |k\rangle\langle j|. \tag{7.20}$$

This can be seen by observing that the contributing elements to a  $\oslash$  product are exclusively diagonal, which we use  $|j\rangle\langle j|$  to pick up. Any off-diagonal information about the previous element-wise product operations is irrelevant. In general, if we have the  $r^{th}$  order  $\tilde{S}$  operator given by

$$\tilde{S}^{(r)} = \sum_{j,k} T^{(r)}(j,k) \otimes |k\rangle\langle j|, \tag{7.21}$$

the operators  $\tilde{S}_{\odot}^{(r+1)}$  and  $\tilde{S}_{\oslash}^{(r+1)}$  can be generated as follows:

$$\tilde{S}_{\odot}^{(r+1)} = \sum_{j,k} T^{(r)}(j,k) \otimes |j\rangle\langle k| \otimes |k\rangle\langle j|, \tag{7.22}$$

$$\tilde{S}_{\oslash}^{(r+1)} = \sum_{j,k} (|j\rangle\langle j|)^{\otimes r} \otimes |k\rangle\langle k| \otimes |k\rangle\langle j|. \tag{7.23}$$

We have shown a recursive procedure to construct  $\tilde{S}^{(r)}$  up to r=N such that

$$\operatorname{tr}_{1...r}\left\{e^{-i\tilde{S}^{(r)}\delta}(\rho^{\otimes r}\otimes\sigma)e^{i\tilde{S}^{(r)}\delta}\right\} = \exp\left[-i\rho^{(\odot,\oslash)r}\delta\right]\sigma\exp\left[i\rho^{(\odot,\oslash)r}\delta\right] + \mathcal{O}(\delta^2), \quad (7.24)$$

for a small evolution  $\delta$ . Analogously with Lemma 1 and Lemma 2, with  $\mathcal{O}(t^2/\epsilon)$  repetitions consuming  $\mathcal{O}(rt^2/\epsilon)$  copies of  $\rho$ , the desired

$$\exp\left[-i\rho^{(\odot,\oslash)r}t\right]\sigma\exp\left[i\rho^{(\odot,\oslash)r}t\right] \tag{7.25}$$

can be implemented up to an  $\epsilon$  error. Finally one makes use of the Lie product formula for summing the terms in the polynomial [82–84]:

$$e^{i\delta(A+B)+\mathcal{O}(\delta^2/m)} = (e^{i\delta A/m}e^{i\delta B/m})^m, \tag{7.26}$$

where A and B are taken to different terms in  $P^N_{\odot,\oslash}(\rho) = \sum_r^N c_r \rho^{(\odot,\oslash)r}$ , and the factors  $c_r$  simply amount to multiplying the  $S^{(r)}$  matrices with the respective coefficients. The parameter m can be chosen to further suppress the error by repeating the entire procedure. However, for the purpose of implementing  $\mathrm{e}^{-iP^N_{(\odot,\oslash)}(\rho)t}\sigma\mathrm{e}^{iP^N_{(\odot,\oslash)}(\rho)t}$  to our desired accuracy  $\epsilon$ ,  $\mathcal{O}(N^2t^2/\epsilon)$  copies of  $\rho$  are required. The quadratic dependency in the order of the polynomial,  $N^2$  stems from implementing the unitary  $\exp\left[-i\rho^{(\odot,\oslash)r}t\right]$  up to r=N, each consuming  $\mathcal{O}(Nt^2/\epsilon)$  copies as previously argued.

### 7.2 Experiments

We have performed the following two sets of experiments to demonstrate the Hermitian matrix inversion component of the quantum GP algorithm:

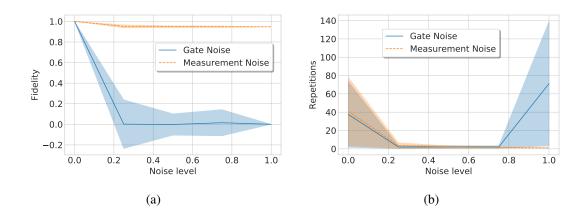
- Simulations of the quantum matrix inversion on quantum virtual machines, the classical simulators of Rigetti's Forest API [72] with analysis of varying noise models' impacts on the outputs.
- 2. A small-scale (2 × 2) implementation of quantum matrix inversion in both PyQuil, run on Rigetti's Quantum Processing Unit (QPU), and in IBM's QISKit software stack, run on IBM's Quantum Experience [73].

The PyQuil framework provides advanced gate decomposition features that allow for arbitrary unitary operations on a multi-qubit quantum state. The simulated noise models of the Rigetti's quantum virtual machine allows for an analysis of the expected accuracy and computational overhead of actual quantum implementations. QISKit also provides a noisy classical simulator, which we use to compare the performance of the quantum matrix inversion algorithm on the real QPU against simulations with realistic noise models. The quantum processing units we use for actual implementations are IBM's 16-qubit Rueschlikon (IBMQX5) [85] and Rigetti's 8-qubit 8Q-Agave. While the numbers of available qubits in both cases are higher than the number required for the implementation (a total of six for the  $2 \times 2$  matrix inversion), the depth requirement of the circuit grows significantly for larger matrices.

### 7.2.1 Simulations on a quantum virtual machine

Here we present the results from the simulations conducted with Rigetti's quantum virtual machine. We have performed two sets of experiments to analyse the effect of different types of noise on the algorithm. Firstly, we restrict to the simplest non-trivial case of inverting a  $2\times 2$  matrix which is chosen to be  $A=\frac{1}{2}\begin{pmatrix}3&1\\1&3\end{pmatrix}$  with the problem-specific circuit in Ref. [86]. The circuit involved is significantly shallower than the one required by the full algorithm, which is described in Ref. [87], making it more practically viable to implement on current and near-term quantum computers due to its reduced depth. Secondly, we simulate the full quantum matrix inversion algorithm [27, 87]. This requires a large number of ancillary qubits for the computation of the reciprocals of the eigenvalues. We will simulate the inversion of a  $4\times 4$  matrix with four bits of precision.

We work with two noise models: The first one, known as the "gate noise", applies a Pauli X operator with a certain probability on each qubit after every gate application. The second one, known as the "measurement noise", applies a Pauli X operator with certain probability only on every qubit that is measured before the measurement takes place. As such, the measurement noise can also be interpreted as a readout error.

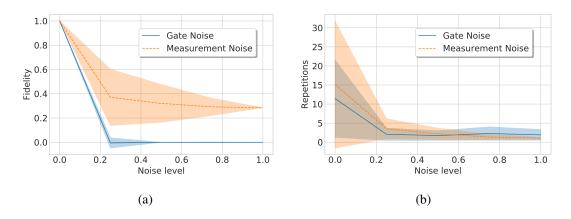


**Figure 7.1:** Simulated gate and measurement noise on a specialised circuit for inverting A. (a) The fidelity shows the overlap with the expected correct state after the computation. A zero fidelity means the output state is orthogonal to the correct solution, while a unit fidelity means the output state is the correct result. (b) The number of repetitions indicates the average number required to execute the probabilistic program before it succeeds.

The simulation results of the quantum inversion of the  $2 \times 2$  matrix A is presented in Figure 7.1. We analyse the following two critical factors, namely the fidelity between the expected result and the simulated output, given that inversion has succeeded, and the average repetition of the coherent part of the algorithm needed to obtain a successful run. Note that in our noisy setting, success in the post-selection does not guarantee the correctness of the output. Our results show that measurement noise has a smaller impact on the result than gate noise which for reasonably low noise levels already renders the output state orthogonal to the expected result. Interestingly, as the noise level increases, the average number of repetitions decreases.

The simulation results of general quantum matrix inversion algorithm on a random  $4 \times 4$  matrix is presented in Figure 7.2. We see that the output's sensitive to noise has increased as the circuit involved became deeper. However, the noise level for which the output reaches zero fidelity is approximately the same in both the  $2 \times 2$  and  $4 \times 4$  cases, and it would be interesting to see whether it remains constant for larger instances.

The simulation still shows better robustness to measurement noise, but with its effect appearing to be stronger compared with the problem-specific algorithm of Figure 7.1. As before in the  $2 \times 2$  case, measurement noise introduces bit flips to registers storing measurement results, which eventually leads to an apparent low number of repetitions, but at the expense of lower fidelities with the expected output.



**Figure 7.2:** Simulated gate and measurement noise on the generic circuit for inverting a  $4 \times 4$  matrix with four bits of precision on eigenvalues.

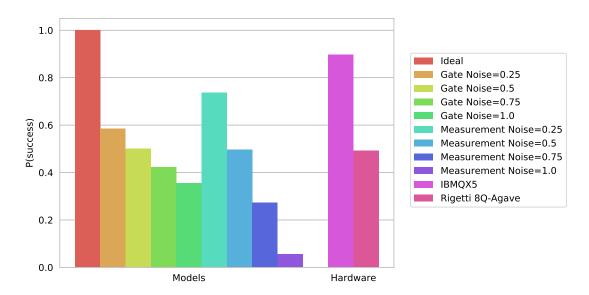
#### 7.2.2 Implementations on quantum processing units

In this section, we implement the restricted  $2 \times 2$ -matrix inversion algorithm with two real quantum processors. We have chosen to implement a restricted version of the algorithm due to the limitations of the currently available hardware with respect to qubit numbers, qubit-qubit connectivity, and coherence times. Note that one does not have direct access to the complete information of the output state, but only samples of measurement results. To gauge the correctness of the output, we will perform a SWAP test [62, 88] with the expected output encoded in auxiliary qubits, and use a flag qubit to indicate a successful run of the test. With multiple runs, the figure of merit is the probability of success, P(success), which can then be related to the fidelity by  $\mathcal{F} = |2P(\text{success}) - 1|$ .

We have implemented the restricted matrix inversion algorithm on both the Rigetti's 8Q-Agave and the IBM's IBMQX5 quantum processing units. The IBM QISKit software [73] also provides a classical simulator to run noisy experiments, and we use these to benchmark the performance of the runs on the real chips. As with simulations in Rigetti's software stack, we expect the measurement noise to have a smaller effect than the gate noise. Note that the flag qubit of the swap test is also subject to readout error under the simulated measurement noise. Therefore an apparent low P(success) in the high measurement error regime could have included many instances of successful runs, falsely reported by the flag qubit. Gate noise on the other hand directly affects the computations in the circuit. Therefore the lower success probabilities now reflect a real discrepancy between the actual output and desired states. In this case, the success probabilities lie in the range of [0.35, 0.6], which translates into fidelities in the range of [0, 0.3]. The probability of success is 89%, which translates into a fidelity with the expected outcome of 0.78. This is a very encouraging result, despite the small size of the matrix inverted. The results are shown in Figure 7.3.

### 7.3 Summary

We have developed a quantum algorithm for a Bayesian approach to deep learning, which makes use of the quantum Gaussian processes algorithm with a kernel matrix corresponding to ReLU activation functions on each layer of the deep network with infinite width. In the simplest case of a single layer architecture, we assume the kernel matrix can be classically evaluated and efficiently simulated as a Hamiltonian to be used in the quantum GP algorithm. In the case of multi-layer, deep architectures, we worked with a model where the kernel matrix corresponding to the layer zero (the base case) can be encoded as a density matrix. We then designed a recursive procedure to simulate the Hamiltonian corresponding to the kernel matrix at an arbitrary depth, which given a fixed



**Figure 7.3:** Success probability of the SWAP test for different noisy simulations and executions on the IBM's and Rigetti's QPUs (rightmost bars). The noise models include gate noise and measurement noise, with different probabilities of failure. The algorithm is run 8192 times for each instance, after which P(success) is evaluated.

accuracy requirement and only consumes a quadratic number of copies of the density matrix. In order to analyse the practical feasibility of the algorithm, we implemented its core subroutine, quantum matrix inversion, on both quantum simulators and real state-of-the-art quantum processors. We observed that the accuracy drops sharply with noise, but even with current, small-scale quantum computers, reasonably high success rates can still be achieved.

Although these experimental results are promising, we should note that they do not constitute sufficient evidence that the full quantum algorithm for Bayesian deep learning can be efficiently implemented in near-term quantum technologies. A fully quantum implementation, including recursively simulating the required Hamiltonian corresponding to the covariance matrix at deep layers, will be an interesting avenue for future research.

# **Part III**

**Quantum correlations and causality** 

# **Chapter 8**

## Geometry of quantum correlations

In the previous parts of the thesis, we have seen that quantum computation can be applied to statistical inference in classical datasets. Particularly, we have focused on the statistical model of Gaussian processes, and shown that phase estimation based methods can provide provable quantum advantages. In this chapter, we take a different approach and look at another aspect of statistical inference in the quantum era, where the data itself is inherently quantum. We consider the problem of inferring quantum correlations from measurement events. The material of this chapter follows closely from Ref. [89].

#### 8.1 Introduction

The study of quantum correlations has long held an important role in fundamental physics [90,91], and more recently given rise to promising prospects of quantum technologies [92,93]. In the usual formulation of non-relativisite quantum theory, the state of a system can extend across space but is only defined at a particular instant in time. The distinction between the roles of space and time contrasts with relativity [94] where they are treated in an even-handed fashion, and has led to a general preference to study temporal quantum correlations in a rather separated manner from their spatial counter-parts [95–101]. Here we aim at taking a unifying approach to study quantum correlations for observables defined across space-time in a general formalism. In order to do so, we make use of

the pseudo-density matrix (PDM) formalism introduced in Ref. [102] as an extended framework of quantum correlations, which generalises the notion of a quantum state to the temporal domain, treating space and time on an equal footing.

We will focus on the simplest and most fundamental case, that of two-point correlation functions. In the spatial setting, this would correspond to bipartite quantum correlations, which can exhibit entanglement. In the temporal setting, we consider the correlations between two sequential measurements separated by an arbitrary quantum channel evolution on a single qubit quantum state. Our study presents the geometry of bipartite correlations in both the spatial and temporal cases and establishes a symmetric structure between them. We observe that this symmetry is broken in the presence of certain non-unital channels. As such these non-unital channels produce a novel set of temporal correlations that are statistically identical to bipartite quantum entanglement.

#### 8.1.1 Density matrices and spatial correlations

**Density matrices** As introduced previously in Section 1.3.1, a density matrix is defined as a probability mixtures of pure quantum states. However, there is also another way of interpreting the density matrices, as the mixture of the expectation values of every possible Pauli measurements resulting in a linear combination of different Pauli components. Particularly for an *n*-qubit system, we have

$$\rho = \frac{1}{2^n} \sum_{i_1=0}^3 \dots \sum_{i_n=0}^3 \left\langle \bigotimes_{j=1}^n \sigma_{i_j} \right\rangle \bigotimes_{j=1}^n \sigma_{i_j}, \tag{8.1}$$

where the indices i label different Pauli operators and the identity operator with  $\sigma_0 = \mathbb{I}$ ,  $\sigma_1 = X$ ,  $\sigma_2 = Y$ , and  $\sigma_3 = Z$ , while the sub-indices j of each i labels different qubits in the system. In order to have a valid density matrix, we need to further require  $\rho$  to be positive semi-definite.

Geometry of spatial correlations Consider the matrix  $\mathcal{C}$  whose elements  $\mathcal{C}_{kl}$  are given by the Pauli correlation functions  $\langle \sigma_k \sigma_l \rangle = \mathrm{Tr}[(\sigma_k \sigma_l)\rho], k, l = 1, 2, 3$  of a two-qubit bipartite state  $\rho$ . It is clear that under local unitary transformations,  $\mathcal{C}$  can be brought into a diagonalised form  $\mathcal{C}'$ . It is known that  $\mathcal{C}'$  can always be written as a convex combination of  $\mathcal{C}_1 = diag[1, -1, 1], \mathcal{C}_2 = diag[-1, 1, 1], \mathcal{C}_3 = diag[1, 1, -1]$  and  $\mathcal{C}_4 = diag[-1, -1, -1]$ , which corresponds to the correlation matrices of the four maximally-entangled Bell states respectively [103]. Geometrically, one can visualise this convex set of correlation functions in three-dimensional real space as a tetrahedron whose four vertices in the  $\langle XX \rangle_- \langle YY \rangle_- \langle ZZ \rangle$  coordinate system are given by the diagonal entries of  $\mathcal{C}_1$ ,  $\mathcal{C}_2$ ,  $\mathcal{C}_3$ , and  $\mathcal{C}_4$  [104]. We shall name such a tetrahedron the spatial tetrahedron, denoted as  $\mathcal{T}_s$ . The geometry of spatial quantum correlations has been a fruitful area of research, interested readers are referred to [105] for a comprehensive text on this subject.

#### 8.1.2 The pseudo-density matrix formalism

The density matrix of a quantum state can be naturally extended into the temporal domain and used to define the PDM [102] as

$$R = \frac{1}{2^n} \sum_{i_1=0}^{3} \dots \sum_{i_n=0}^{3} \langle \{\sigma_{i_j}\}_{j=1}^n \rangle \bigotimes_{j=1}^n \sigma_{i_j},$$
 (8.2)

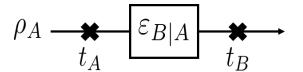
where sub-indices j of each i now label different measurement events in the system. The factor  $\langle \{\sigma_{i_j}\}_{j=1}^n \rangle$  denotes the expectation value of the product of the n Pauli observables. Physically, it corresponds to a correlation function of a size-n sequence of Pauli measurements  $\sigma_{i_j} \in \{\sigma_0, ..., \sigma_3\}$ . Note that R is a Hermitian matrix with unit trace, as it is with conventional density matrices. Furthermore, if the measurement events are space-like separated, R is positive semi-definite and hence resembles a valid density

matrix. However, the mathematical structure of Eq. 8.2 does not exclude the possibility of having negative eigenvalues. When negative eigenvalues are present, the Pauli observables can no longer be interpreted as measurements events on distinct sub-systems of a common quantum state. In such cases, the PDM novelly captures local measurement events happening at arbitrary time instances, in contrast to the case for conventional density matrices.

Measure of causality Since the presence of negative eigenvalues is a witness to causal relationships, it is natural to quantify temporal correlations with some measure based on the trace norm. A causality measure was thus introduced in Ref. [102] as  $f_{tr}(R) = \|R\|_{tr} - 1$ , which possesses desirable properties in close analogy with entanglement monotones for spatial correlations, namely,  $f_{tr}(R) \ge 0$  and  $f_{tr}(R_2) = 1$  for any  $R_2$  generated by two consecutive measurements of a closed system with a single qubit  $(R_2)$  is maximally causal);  $f_{tr}(R)$  is invariant under unitary transformations;  $f_{tr}(R)$  is non-increasing under local operations (c.f. entanglement is non-increasing under LOCC);  $f_{tr}$  is a convex function. We will revisit these properties in Chapter 9, where a logarithmic variant of the trace norm measure will play a significant role.

## 8.2 General two-time quantum correlations

Here we describe the quantum correlations between Pauli measurements at two time instances. The corresponding physical scenario is depicted in Figure 8.1, where a single-qubit system  $\rho_A$  subject to a quantum channel between two measurement events at times  $t_A$  and  $t_B$ . The channel is described by a completely positive trace-preserving (CPTP) map  $\varepsilon_{B|A}$ , which maps the family of operators from the state space  $\mathcal{H}_A$  at  $t_A$  to the state space  $\mathcal{H}_B$  at  $t_B$ .



**Figure 8.1:** The physical scenario of general two-time quantum correlations: A single-qubit system  $\rho_A$  is measured at  $t_A$  and  $t_B$  with a quantum channel in between described by the CPTP map,  $\varepsilon_{B|A}$ .

## 8.2.1 The two-point temporal PDM

It it clear from the definition of PDM, Eq. 8.2 that the expectation value of the product of n Pauli observables is given by

$$\langle \{\sigma_{i_j}\}_{j=1}^n \rangle = \operatorname{Tr}\left[\left(\bigotimes_{j=1}^n \sigma_{i_j}\right) R\right].$$
 (8.3)

In the case of two sequential events, n=2. Supposing the evolution between  $t_A$  and  $t_B$  is the identity, the only non-zero Pauli correlation functions are

$$\langle \{\sigma_{1}, \sigma_{1}\} \rangle = \langle \{\sigma_{2}, \sigma_{2}\} \rangle = \langle \{\sigma_{3}, \sigma_{3}\} \rangle = \langle \{\sigma_{0}, \sigma_{0}\} \rangle = 1,$$

$$\langle \{\sigma_{0}, \sigma_{1}\} \rangle = \langle \{\sigma_{1}, \sigma_{0}\} \rangle = \langle \sigma_{1} \rangle,$$

$$\langle \{\sigma_{0}, \sigma_{2}\} \rangle = \langle \{\sigma_{2}, \sigma_{0}\} \rangle = \langle \sigma_{2} \rangle,$$

$$\langle \{\sigma_{0}, \sigma_{3}\} \rangle = \langle \{\sigma_{3}, \sigma_{0}\} \rangle = \langle \sigma_{3} \rangle.$$
(8.4)

Here  $\{...\}$  denotes sets of operators, which should not be confused with a similar notation for anti-commutators. On the other hand, we can write a single-qubit density operator  $\rho_A$  as

$$\rho_A = \frac{1}{2} \left( \sigma_0 + \langle \sigma_1 \rangle \, \sigma_1 + \langle \sigma_2 \rangle \, \sigma_2 + \langle \sigma_3 \rangle \, \sigma_3 \right). \tag{8.5}$$

We now compare the coefficients of Pauli components and obtain  $R = \{\rho_A \otimes \frac{1}{2}, SWAP\}$ , where  $SWAP = \frac{1}{2} \sum_{i=0}^3 \sigma_i \otimes \sigma_i$ , and here  $\{...\}$  denotes the anti-commutator, such that  $\{A,B\} = AB + BA$ . In a general setting, a channel that acts on the system in between the time instances  $t_A$  and  $t_B$  as a CPTP map  $\varepsilon_{B|A}$  is included. Note that the map does not affect any observables at  $t_A$ , but introduces a transformation according to its adjoint map on the observables at  $t_B$ . Therefore the two-time PDM across such a channel can be written as

$$R_{AB} = (\mathcal{I}_A \otimes \varepsilon_{B|A}) \left( \{ \rho_A \otimes \frac{\mathrm{I}}{2}, SWAP \} \right), \tag{8.6}$$

where  $\mathcal{I}_A$  denotes the identity super-operator acting on A. The above expression is in agreement with the Jordan product representation given in Ref. [106]:

$$R_{AB} = \{ \rho_A \otimes \frac{I}{2}, E_{AB} \}, \tag{8.7}$$

where  $E_{AB} = \sum_{ij} \left( \mathcal{I}_A \otimes \varepsilon_{B|A} \right) \left( |i\rangle \left\langle j|_A \otimes |j\rangle \left\langle i|_B \right) \right)$  is an operator acting on  $\mathcal{H}_A \otimes \mathcal{H}_B$  that is Jamiołkowski-isomorphic to  $\varepsilon_{B|A}$ . The correlations described by  $R_{AB}$  are "purely" temporal in the sense that the underlying dynamics are defined by a CPTP map on a single qubit.

### 8.2.2 Single-qubit quantum channels

To proceed further, we need to exploit the structures of the quantum channel  $\varepsilon_{B|A}$ . It was established in Ref. [107] that the complete positivity requirement leads to a particularly useful trigonometric parameterisation of the set of possible  $\varepsilon_{B|A}$  in the Pauli basis [108]. Concretely, this set corresponds to the convex closure of the maps defined

by the following Kraus operators up to permutations among  $\{\sigma_1, \sigma_2, \sigma_3\}$ :

$$K_{+} = \left[\cos\frac{v}{2}\cos\frac{u}{2}\right]\sigma_{0} + \left[\sin\frac{v}{2}\sin\frac{u}{2}\right]\sigma_{3},$$

$$K_{-} = \left[\sin\frac{v}{2}\cos\frac{u}{2}\right]\sigma_{1} - i\left[\cos\frac{v}{2}\sin\frac{u}{2}\right]\sigma_{2},$$
(8.8)

where  $v \in [0, \pi], u \in [0, 2\pi]$ . The above Kraus operators act on  $\sigma_i$  as the following:

$$K_{+}\sigma_{0}K_{+}^{\dagger} + K_{-}\sigma_{0}K_{-}^{\dagger} = \sigma_{0} + \sin(u)\sin(v)\sigma_{3},$$

$$K_{+}\sigma_{1}K_{+}^{\dagger} + K_{-}\sigma_{1}K_{-}^{\dagger} = \cos(u)\sigma_{1},$$

$$K_{+}\sigma_{2}K_{+}^{\dagger} + K_{-}\sigma_{2}K_{-}^{\dagger} = \cos(v)\sigma_{2},$$

$$K_{+}\sigma_{3}K_{+}^{\dagger} + K_{-}\sigma_{3}K_{-}^{\dagger} = \cos(u)\cos(v)\sigma_{3}.$$
(8.9)

#### 8.2.3 Convex closure

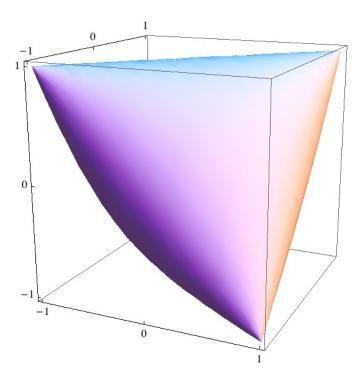
We now expand Eq. 8.6 into its Pauli components and substitute into Eq. 8.3, and obtain

$$\langle \sigma_k \sigma_k \rangle = \text{Tr} \left[ \langle \sigma_k \rangle_{\rho_A} \varepsilon_{B|A}(\sigma_0) \sigma_k + \varepsilon_{B|A}(\sigma_k) \sigma_k \right],$$
 (8.10)

where  $\langle \sigma_k \rangle_{\rho_A}$  denotes the expectation value of the  $\sigma_k$  observable on the initial state  $\rho_A$ . By setting  $\rho_A = |0\rangle \langle 0|$  and applying the Kraus operators in Eq. 8.8, we obtain the parametric equations which characterise the convex set of possible correlation functions as followed:

$$\langle \sigma_1 \sigma_1 \rangle = \cos(u),$$
  
 $\langle \sigma_2 \sigma_2 \rangle = \cos(v),$   
 $\langle \sigma_3 \sigma_3 \rangle = \cos(u - v).$  (8.11)

The set of three Pauli correlations  $\langle \sigma_k \sigma_k \rangle = \operatorname{Tr} \left[ (\sigma_k \otimes \sigma_k) R_{AB} \right]$  fully characterises any two-point correlations  $\langle \sigma_k \sigma_l \rangle$  up to local unitary transformations, for k, l = 1, 2, 3. Note that the choice of permutation among  $\{\sigma_1, \sigma_2, \sigma_3\}$  is arbitrary and hence does not affect the resultant convex set enclosed by the parametric surface. We illustrate the set of attainable  $\langle \sigma_k \sigma_k \rangle$  as points in the real coordinator space  $\{\langle \sigma_1 \sigma_1 \rangle, \langle \sigma_2 \sigma_2 \rangle, \langle \sigma_3 \sigma_3 \rangle\}$  in FIG. 8.2, which depicts the geometry of two-time Pauli correlations. The figure shows a parametric plot of the equations  $\langle \sigma_1 \sigma_1 \rangle = \cos(u), \langle \sigma_2 \sigma_2 \rangle = \cos(v)$  and  $\langle \sigma_3 \sigma_3 \rangle = \cos(u-v)$ , where  $v \in [0,\pi], u \in [0,2\pi]$ . Note that a similar structure was found when three sequential observables were considered in the context of Leggett-Garg inequalities [97].



**Figure 8.2:** The surface enclosing the set of possible values of two-point temporal correlations in the real space of  $\{\langle \sigma_1 \sigma_1 \rangle, \langle \sigma_2 \sigma_2 \rangle, \langle \sigma_3 \sigma_3 \rangle\}$ .

## **8.3** Two-point correlations in space-time

In this section, we focus on the cases where the initial system  $\rho_A$  is maximally-mixed. As mentioned in Section 8.1.1, the set of spatial correlations described by two-qubit density matrices can be depicted in the space of  $\{\langle \sigma_1 \sigma_1 \rangle, \langle \sigma_2 \sigma_2 \rangle, \langle \sigma_3 \sigma_3 \rangle\}$  as the convex hull enclosed by the tetrahedron  $\mathcal{T}_s$  with vertices of odd parity (1,1,-1), (1,-1,1), (-1,1,1) and (-1,-1,-1). These vertices correspond to the four Bell states. The set of temporal correlations described by  $R_{AB}$  with  $\rho_A = \frac{1}{2}$  is simply the reflection of  $\mathcal{T}_s$  in the  $\langle \sigma_1 \sigma_1 \rangle - \langle \sigma_3 \sigma_3 \rangle$  plane. The resulting tetrahedron  $\mathcal{T}_t$  has vertices of even parity (1,-1,-1), (1,1,1), (-1,-1,1) and (-1,1,-1). This follows from the relation Eq. 8.7,  $R_{AB} = \frac{1}{2} E_{AB}$ , when setting  $\rho_A = \frac{1}{2}$ . A partial transpose over sub-system A, which geometrically corresponds to the reflection, yields

$$R_{AB}^{\mathcal{PT}} = \left(\mathcal{I}_A \otimes \frac{\varepsilon_{B|A}}{2}\right) \sum_{ij} |ii\rangle \langle jj|_{AB} = \rho_{AB}^{\text{Choi}}(\varepsilon_{B|A}), \tag{8.12}$$

where  $\rho_{AB}^{\text{Choi}}(\varepsilon_{B|A})$  is the Choi matrix of  $\varepsilon_{B|A}$  [109]. For arbitrary choices of  $\varepsilon_{B|A}$ , the Choi matrices describe the same set of correlations,  $\mathcal{T}_s$  as two-qubit density matrices. As the partial transpose over sub-system A generates a reflection in the  $\langle \sigma_1 \sigma_1 \rangle - \langle \sigma_3 \sigma_3 \rangle$  plane, the set  $\mathcal{T}_t$  is simply an inverted copy of  $\mathcal{T}_s$ .

Distance from separability The Peres-Horodecki criterion [104] implies that the octahedron region formed by the overlap between the two tetrahedra  $\mathcal{T}_t$  and  $\mathcal{T}_s$  corresponds to the set of separable states. With this insight, we can make a natural connection between the entanglement measure, negativity [110],  $f_{\mathcal{N}}(\rho_{AB}) = \frac{1}{2}(\|\rho_{AB}^{\mathcal{P}\mathcal{T}}\|_{tr} - 1)$  and the causality measure  $f_{tr}$ . Consider a two-qubit state  $\rho_{AB}^{\text{Choi}}$  as the Choi matrix of  $\varepsilon_{B|A}$  in Eq. 8.6, leading to  $f_{tr}(R_{AB}) = 2f_{\mathcal{N}}(\rho_{AB}^{\text{Choi}})$ . It was shown in Ref. [111] that the entanglement measure  $f_{\mathcal{N}}$  can be visualised as the Euclidean distance  $D_s$  between a point in  $\mathcal{T}_s$  and the

nearest point in the octahedron, such that  $D_s = \frac{4f_N}{\sqrt{3}}$ . Hence, by analogy we can establish a geometric interpretation for  $f_{tr}$  as the Euclidean distance  $D_t$  between a point in  $T_t$  and the nearest point on the face of the octahedron, such that  $D_t = \frac{2f_{tr}}{\sqrt{3}}$ .

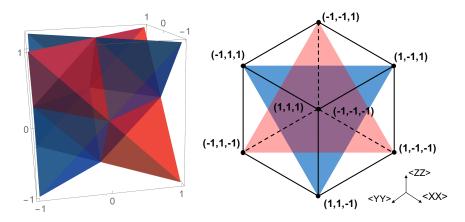
Mixed space-time correlations Beyond the geometry of the purely temporal and spatial correlations, a two-point PDM generally describes an arbitrary mixture of spatial and temporal correlations. Consider sequential Pauli measurements,  $\sigma_A$  and  $\sigma_B$  on one sub-system of a maximally-entangled pair. If the sub-system evolves through a CP-map,  $\langle \sigma_A \sigma_B \rangle$  lies in the  $\mathcal{T}_t$  as shown. However, if a SWAP operation is applied before the second measurement, then the reduced dynamics on sub-system A will no longer be described by a CP-map. Under these conditions the correlations  $\langle \sigma_A \sigma_B \rangle$  will span  $\mathcal{T}_s$ . Furthermore, if SWAP is applied probabilistically, the possible correlations span the entire volume of the cube formed by the vertices of  $\mathcal{T}_t$  and  $\mathcal{T}_s$ , fully inscribing the spatial and temporal tetrahedra. It is clear that the cube is the largest possible set of space-time quantum correlations, since  $-1 \leq \langle \sigma_A \sigma_A \rangle \leq 1$ , and the set of possible correlation functions forms a convex set. We depict the geometry of different types of two-point correlations in space-time in Figure 8.3.

Unital channels The results of Figure 8.3 assumes the initial state  $\rho_A$  is maximally-mixed. Interestingly,  $T_t$  also describes temporal correlations for an arbitrary input state  $\rho_A$  but with the channel restricted to be unital which means  $\varepsilon_{B|A}(\sigma_0) = \sigma_0$ . This is because only non-unital maps act non-trivially on the local components  $\sigma_k \otimes \sigma_0$  of the PDM, which leads to an augmented set of correlations. Specifically, note that the first term in the trace of Eq. 8.10 vanishes whenever either  $\rho_A$  is maximally-mixed or  $\varepsilon_{B|A}$  is

a unital map, in which case the parametric equations reduce to

$$\langle \sigma_1 \sigma_1 \rangle = \cos(u),$$
  
 $\langle \sigma_2 \sigma_2 \rangle = \cos(v),$   
 $\langle \sigma_3 \sigma_3 \rangle = \cos(u) \cos(v).$  (8.13)

The above equations give a parametric surface with the extremal points (1, 1, 1), (1, -1, -1), (-1, 1, -1) and (-1, -1, 1). The convex enclosure of these points gives exactly the temporal tetrahedron,  $\mathcal{T}_t$ . Hence we can see there exists a conditional reflective symmetry between the sets of temporal and spatial correlations. This symmetry is shown to be broken in the presence of certain non-unital channels, which give rise to the set of attainable temporal correlation shown in Figure 8.2

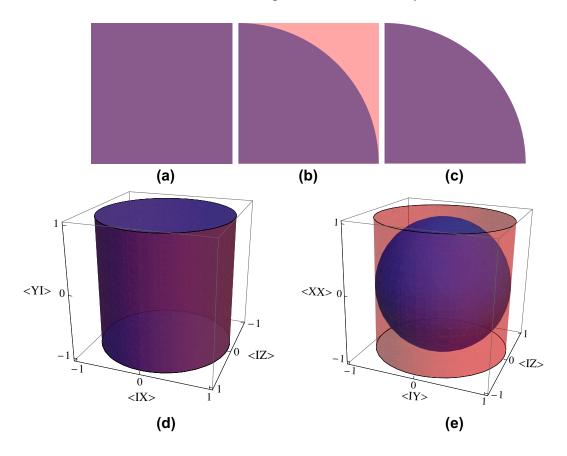


**Figure 8.3:** The spatial and temporal tetrahedrons with the blue region representing  $\mathcal{T}_s$ , and the red region representing  $\mathcal{T}_t$  (Left). A perspective plot viewing from the (-1, -1, -1) direction, where the purple hexagon is a projection of the octahedron overlap, the blue and red triangles are projections of  $\mathcal{T}_s$  and  $\mathcal{T}_t$  respectively (Right).

### 8.3.1 General PDM Pauli components

The remaining components of the two-point PDM includes all possible combinations of  $\sigma_{A_1}, \sigma_{A_2}, \sigma_{B_1}, \sigma_{B_2} \in \{I, X, Y, Z\}$ . The geometry of correlations is illustrated in Figure

8.4. The figure presents the types of correlations in two-point PDMs as 2-D projections onto the planes of  $\{\langle \sigma_{A_1}\sigma_{B_1}\rangle, \langle \sigma_{A_2}\sigma_{B_2}\rangle\}$  in Figures 8.4a, 8.4b and 8.4c. The sets of correlations are shown for the first quadrant. The full 2-D projection is generated in a symmetric manner about the origin. In Figures 8.4d and 8.4e, we give instances in the 3-D spaces corresponding to this 2-D projections. The red region highlights extra correlations attainable in a valid PDM compared to a valid density matrix.



**Figure 8.4:** Here we have (a) Type a:  $[\sigma_{A_1} \otimes \sigma_{B_1}, \sigma_{A_2} \otimes \sigma_{B_2}] = 0$ . Temporal and spatial correlations both lie in the purple unit square; (b) Type b:  $\{\sigma_{A_1} \otimes \sigma_{B_1}, \sigma_{A_2} \otimes \sigma_{B_2}\} = 0$ , and one out of the four operators is  $\sigma_0$ . Spatial correlations lie in the purple quarter unit circle, while temporal correlations lie in the unit square. The red region is allowed by valid PDMs but not density matrices; (c) Type c: In all other cases, correlations are bounded by the purple quarter circle; (d) An example of 3-D spaces corresponding to a combination of type a and type c 2-D projections; (e) An example of 3-D spaces corresponding to a combination of type b and type c 2-D projections.

The above completely characterises the two-point spatial and temporal correlations for qubit systems. The space of possible temporal correlations is strictly larger than the space of possible spatial correlations. These extra correlations cannot originate solely from spatially separated events, and hence are a signature of causal influence between measurement events.

#### 8.4 Discussions

**Quantum causal inference** The geometric structure of spatial and temporal correlations presented in this chapter has potential application to quantum causal inference (see Ref. [112] for an introduction). Given the outcomes of two sets of measurements, one can estimate the expectation values of two-point correlations, and identify the corresponding coordinates in the provided geometric structure, and infer whether there exists a causal relationship between the measurement events.

Sequentially mimicked entanglement Note that the "inflated tetrahedron" in Figure 8.2 inscribes a larger volume than  $\mathcal{T}_t$ . Therefore it partially overlaps with the non-separable regions in  $\mathcal{T}_s$ . Hence there exist temporal correlations that are statistically identical to entangled correlations in space. Physically, this implies entanglement can be partially mimicked by sequential correlation described by a single-qubit PDM, and that it is impossible to distinguish between the two cases by only examining the correlation statistics. An instance of this result is reflected in the violation of the temporal CHSH inequality [96], which can be expressed entirely in terms of  $\langle \sigma_1 \sigma_1 \rangle$  and  $\langle \sigma_2 \sigma_2 \rangle$  correlations. The "inflated tetrahedron" imposes constraints in the space of all three Pauli correlations, hence serves as a stronger geometric criterion for classifying quantum correlations and can act as a causal witness. Here we should emphasize the vertices of  $\mathcal{T}_s$  that correspond to maximally-entangled states do not overlap with the

temporally attainable set. The inability to simulate correlations generated by Bell states with sequential measurements is related to the impossibility of constructing a quantum universal-NOT gate [113].

# **Chapter 9**

# Causality in quantum communication

We have introduced the PDM formalism in Section 8.1.2 of the previous chapter and used it as a framework to demonstrate the geometric structure of quantum correlation in both the spatial and the temporal domains. One particularly novel aspect of the PDM formalism is the ability to quantify causal relations between sequential measurement events with a causality measure which is computed by the trace norm of a given PDM. In this chapter, we show that quantum causality plays an operational role in quantum communication. Since realistic channels for quantum communication task are noisy, it is of practical interests to quantify the capacity of the channel being used. Existing results have successfully connected quantum channel capacities with spatial correlations [114, 115]. For instance, the quantum capacity of a channel is known to be equivalent to the highest rate at which it can be used to generate entanglement [116]. The notion of quantum causality characterises the temporal aspect of quantum correlations analogously with entanglement in the spatial case. Here we take the intuitive step to uncover a connection between quantum causality and channel capacity. Concretely, we prove the amount of temporal correlations between two ends of the noisy quantum channel, as quantified by a logarithmic variant of the causality measure, implies a general upper bound on its channel capacity, which we will call the causal bound of quantum channel capacities. Conveniently, the mathematical expression of the causal bound is more

straightforward to evaluate than most previously known bounds for quantum capacities. We will further demonstrate the utility of the causal bound by applying it to a class of shifted depolarising channels, which shows improvement over previously known results of Ref. [117] and [118]. The material presented in this chapter is closely based on Ref. [119].

## 9.1 Bounding quantum channel capacities

One of the central objectives of information theory is to determine the maximum rate of reliable transmission of information using a given communication channel. In classical information theory, the early work of Shannon proved that a simple expression governs the capacity of discrete memoryless channels [120].

When considering the capacity of a quantum channel,  $\mathcal{N}$ , one has to take into account the possible necessity of encoding information in states entangled across multiple copies of the channels, to obtain the maximal capacity per use. Hence an exact computation of the capacity of a quantum channel amounts to taking the supremum over tensor products of an arbitrary number of copies of the same channel. As such, an exact characterisation of a channels' capability to transmit quantum information has proved to be a much more challenging task. In the absence of formulae for the exact capacities, one is often forced to rely on bounds for the quantum capacity that are tractable to evaluate [121–127]. Nevertheless, a significant amount of progress in the context of quantum communication has been made in determining the achievable rates for transmitting quantum information over noisy channels [114,115,118,128]. However, the existing formulae for quantum capacities often involve inherent optimisation problems, leading to significant computational difficulties. The reader is referred to Ref. [116] for a review of related results.

Here we take a new approach and present a general upper bound on the quantum capacities of quantum channels that are based on causality considerations. Apart from the theoretical novelty of connecting between quantum causality and concrete communication problems, these new causal bounds further allow direct computation without requiring optimisation.

#### 9.1.1 Logarithmic Causality

Recall in Section 8.1.2, we reviewed a measure of causality based on the trace norm of the pseudo-density matrix. Here we introduce a useful logarithmic variant of this trace norm measure,  $F(R) = \log_2 ||R||_1$ . The logarithmic causality measure is similar to causality monotones introduced in Ref. [102] (reviewed in Section 8.1.2), but it sacrifices convexity in favour of additivity when applied to tensor products of PDMs. Being in close analogy to the logarithmic negativity in entanglement measures, the logarithmic causality also satisfies the following important properties:

- 1.  $F(R) \ge 0$ , with F(R) = 0 if R is positive semi-definite, and  $F(R_2) = 1$  for  $R_2$  generated by two consecutive measurements of a closed system with a single qubit,
- 2. F(R) is invariant under unitary transformations,
- 3. F(R) is non-increasing under local operations,
- 4.  $F(\sum_{i} p_i R_i) \leq \max_{i} F(R_i)$ , for any probability distribution  $\{p_i\}$ .
- 5.  $F(R \otimes S) = F(R) + F(S)$ .

Since  $F(R) = \log_2(f_{tr}(R) + 1)$  and the logarithm function is monotonic, Properties 1-3 in the above follow straight-forwardly from the corresponding properties of the causality monotone  $f_{tr}(R) = ||R||_1 - 1$  which were proved in Ref. [102]. Property 4 also

follows from the monotonicity of the logarithm function which implies  $F(\sum_i p_i R_i) \le \max_i F(R_i \sum_j p_j)$ , and hence  $F(\sum_i p_i R_i) \le \max_i F(R_i)$ . As for Property 5, we note the fact that

$$\log_2 \|R \otimes S\|_1 = \log_2 \|R\|_1 \|S\|_1 = \log_2 \|R\|_1 + \log_2 \|S\|_1, \tag{9.1}$$

which is equivalent to the desired property,  $F(R \otimes S) = F(R) + F(S)$ .

### 9.1.2 PDM representation of quantum channels

We consider a qubit-to-qubit channel, denoted as  $\mathcal{N}_1$  acting on a single qubit quantum state specified by an initial density matrix  $\rho$ . The PDM associated to such a process, denoted by  $R_{\mathcal{N}_1}$  involves a single use of the channel  $\mathcal{N}_1$  and two measurements, one before and one after  $\mathcal{N}_1$ . By Eq. 8.6 in Section 8.2, we have

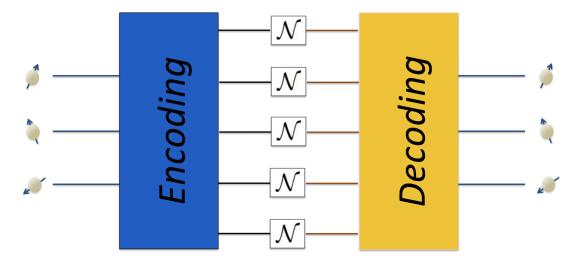
$$R_{\mathcal{N}_1} = (\mathcal{I} \otimes \mathcal{N}_1) \left( \{ \rho \otimes \frac{\mathrm{I}}{2}, \mathrm{SWAP} \} \right).$$
 (9.2)

For the purpose of this chapter, we fix the input to be a maximally mixed state, so that  $\rho = \frac{1}{2}$ . As a result, we are able to generalise Eq. 9.2 to describe an arbitrary quantum channel  $\mathcal{N}$  acting on a collection of l qubits, which leads to

$$R_{\mathcal{N}} = (\mathcal{I} \otimes \mathcal{N}) \left( \frac{\text{SWAP}^{\otimes l}}{2^{l}} \right). \tag{9.3}$$

We further focus our attention to one-way quantum communications. Thus we ought to consider the most general procedure for approximating the ideal (identity) channel with multiple copies of available resource channels. This amounts to combining n parallel uses of the resource channel preceded by some encoding operations and followed by some decoding operations. The schematic diagram of the communication process being

considered is shown below in Figure 9.1.



**Figure 9.1:** The state of a system of k qubits is encoded into a larger Hilbert space. The encoded quantum information is then passed forward using n parallel copies of the resource channel  $\mathcal{N}$ . The sent information is then decoded back into a system of k qubits. In the most general setting, the input and the output of channel  $\mathcal{N}$  need not have the same dimension. The encoding and decoding operations are both described by CPTP maps.

#### 9.1.3 Causal bound

**High-level outline** The logarithmic causality,  $F(R_N)$  can be used to bound the number of uses of the resource  $\mathcal N$  needed to approximate the ideal channel  $\mathcal I^{\otimes k}$ . To do so, we compare the causality across the collection of channels with the causality across the identity channel. As a result of the Property 4. in Section 9.1.1,  $F(\sum_i p_i R_i) \leq \max_i F(R_i)$ , and the fact that for quantum channel capacity consideration it suffices to consider isometric encodings [129], the causality across the combined channels does not increase under encoding and decoding. Further exploiting the additivity of causality to relate k to the number of uses of the channel leads to the result that quantum capacity Q of channel  $\mathcal N$  is upper bounded by  $F(R_{\mathcal N})$ ,

$$Q(\mathcal{N}) \le F(R_{\mathcal{N}}). \tag{9.4}$$

#### Remarks

- Computing  $F(R_N)$  is efficient for channels acting on relatively small Hilbert spaces, as it only requires finding the logarithm of the trace norm of a matrix. Importantly evaluating the causal bound does not involve any optimisation.
- Note that the relation Eq. 9.4 implies that any channel with  $F(R_N) = 0$  has quantum capacity equal to zero. This reflects the fact that such a channel exhibits correlations that could have been produced by measurements on distinct subsystems of a quantum state, and so the system is necessarily constrained by the no-signalling theorem.
- When  $F(R_N)$  is strictly positive, the correlations between the two ends of the channel cannot be captured by bipartite density matrices, thus signifying information being passed forward in time.
- Although the causal bound was presented for channels acting on the collection of
  qubits, this result applies to channels with arbitrary input and output dimensions.
   In such cases, it suffices to restrict the channel to act only on a subspace of the 2<sup>k</sup>
  dimensional Hilbert space.

#### **9.1.4** Proof

In order to prove Eq. 9.4, we start with constructing the PDM that corresponds to a channel obtained by using n copies of the resource channel  $\mathcal{N}$  preceded by the encoding operation E and followed the decoding D. Let  $\mathcal{M} = D \circ \mathcal{N}^{\otimes n} \circ E$ . The PDM to consider,  $R_{\mathcal{M}}$  is related to that of the ideal channel via

$$R_{\mathcal{M}} = (\mathcal{I}^{\otimes k} \otimes \mathcal{M})(R_{\mathcal{I}^{\otimes k}}). \tag{9.5}$$

We add and subtract  $R_{\mathcal{I}^{\otimes k}}$  on the left hand side of Eq. 9.5, and apply the reverse triangle inequality to obtain

$$||R_{\mathcal{M}}||_1 \ge ||R_{\mathcal{I}^{\otimes k}}||_1 - ||R_{\mathcal{M}} - R_{\mathcal{I}^{\otimes k}}||_1.$$
 (9.6)

The trace distance between two pseudo-density matrices can be related to distance in the diamond norm [116] as follows,

$$||R_{\mathcal{M}} - R_{\mathcal{I}^{\otimes k}}||_{1} = ||(\mathcal{I}^{\otimes k} \otimes (\mathcal{M} - \mathcal{I}^{\otimes k}))(R_{\mathcal{I}^{\otimes k}})||_{1}$$

$$\leq ||\mathcal{M} - \mathcal{I}^{\otimes k}||_{\diamond} ||R_{\mathcal{I}^{\otimes k}}||_{1}, \tag{9.7}$$

where  $\|\cdot\|_{\diamond}$  denotes the diamond norm. We define  $\epsilon = \|\mathcal{M} - \mathcal{I}^{\otimes k}\|_{\diamond}$  and use the upper bound of Eq. 9.7 as well as the positivity of  $\|R_{\mathcal{I}^{\otimes k}}\|_1$  to obtain

$$\frac{\|R_{\mathcal{M}}\|_1}{\|R_{\mathcal{I}^{\otimes k}}\|_1} \ge 1 - \epsilon. \tag{9.8}$$

Taking the logarithm on both sides of the above inequality leads to

$$F(R_{\mathcal{M}}) - F(R_{\mathcal{T} \otimes k}) \ge \log_2(1 - \epsilon). \tag{9.9}$$

The connection between the PDM and SWAP matrix and the non-increasing property of the trace norm under the partial trace together leads to the fact that the causality does not increase under decoding and encoding operations. This gives

$$F(R_{\mathcal{M}}) \le F(R_{\mathcal{N}}^{\otimes n}). \tag{9.10}$$

A detailed proof of the above inequality Eq. 9.10 is presented in 9.2.1. Furthermore, this same property of F guarantees that even if we had allowed the encoding and decoding

operations to operate on entangled ancillary registers, Eq. 9.10 is still valid [11,130,131]. Hence the resultant bounds based on Eq. 9.10 are also bounds on the entanglement-assisted capacities. It is important to note this non-increasing property does not hold for any other Schatten norm but the trace norm.

The additivity property of F under tensor products (Property 5. in Section 9.1.1) implies  $F(R_N^{\otimes n}) = nF(R_N)$  and  $F(R_{\mathcal{I}^{\otimes k}}) = kF(R_{\mathcal{I}})$ , which leads to

$$nF(R_{\mathcal{N}}) - kF(R_{\mathcal{I}}) \ge \log_2(1 - \epsilon). \tag{9.11}$$

Finally, using  $F(R_{\mathcal{I}})=1$  for a quantum capacity with respect to a single qubit system, we obtain

$$\frac{k}{n} \le F(R_{\mathcal{N}}) - \frac{\log_2(1-\epsilon)}{n}.\tag{9.12}$$

The relation between the  $\epsilon$  distance in diamond norm and the distance in the completely bounded infinity norm in turn guarantees  $\epsilon$  goes to zero as n approaches infinity. More details on this fact is presented in Section 9.2.2. This concludes the proof of the bound,  $Q(\mathcal{N}) \leq F(R_{\mathcal{N}})$ .

## 9.2 Mathematical details

### 9.2.1 Non-increasing property

An important property used in proving the causal bound was that the decoding and encoding operations do not increase causality, such that  $F(R_{\mathcal{M}}) \leq F(R_{\mathcal{N}}^{\otimes n})$ . To prove this property, we need to make use of the following lemma,

**Lemma 3.** Let K be a linear map from k qubits to m qubits. Then

$$(I \otimes K)SWAP^{\otimes k}(I \otimes K^{\dagger}) = (K^{\dagger} \otimes I)SWAP^{\otimes m}(K \otimes I), \tag{9.13}$$

where  $(A \otimes B)$  means that A and B are applied to the first and second subsystems of each of the SWAPs respectively.

*Proof.* Let  $K = \sum_{i=0}^{2^k-1} \sum_{j=0}^{2^m-1} e_{ij} |j\rangle \langle i|$ . The tensor product of k-qubit SWAPs can be written as

$$SWAP^{\otimes k} = \sum_{u,v=0}^{2^{k}-1} (|u\rangle \otimes |v\rangle)(\langle v| \otimes \langle u|).$$
(9.14)

Substituting Eq. 9.14 into the left hand side of Eq. 9.13 leads to

$$(I^{\otimes k} \otimes K) \operatorname{SWAP}^{\otimes k} (I^{\otimes k} \otimes K^{\dagger})$$

$$= \sum_{i,j,i',j',u,v} (I \otimes |j\rangle \langle i|) |u\rangle |v\rangle \langle v| \langle u| (I \otimes |i'\rangle \langle j'|) e_{ij} e_{i'j'}^{*}$$

$$= \sum_{j,j'=0}^{2^{m}-1} \sum_{u,v=0}^{2^{k}-1} |u\rangle |j\rangle \langle v| \langle j'| e_{vj} e_{uj'}^{*}.$$
(9.15)

Similarly evaluating the right hand side of Eq. 9.13 we get

$$(K^{\dagger} \otimes I^{\otimes m}) \operatorname{SWAP}^{\otimes m} (K \otimes I^{\otimes m})$$

$$= \sum_{i,j,i',j',u,v} (|i\rangle \langle j| \otimes I) |u\rangle |v\rangle \langle v| \langle u| (|j'\rangle \langle i'| \otimes I) e_{ij}^* e_{i'j'}$$

$$= \sum_{i,i'=0}^{2^k-1} \sum_{u,v=0}^{2^n-1} |i\rangle |v\rangle \langle i'| \langle u| e_{i'v} e_{iu}^*$$

$$= \sum_{j,j'=0}^{2^n-1} \sum_{u,v=0}^{2^k-1} |u\rangle |j\rangle \langle v| \langle j'| e_{vj} e_{uj'}^*, \tag{9.16}$$

where in the last step we have relabelled the indices.

We are now in the position to prove the desired non-increasing property of logarithmic causality, which is summarised in the lemma below.

**Lemma 4.** Let  $\mathcal E$  and  $\mathcal D$  be encoding and decoding operations and  $\mathcal M=\mathcal D\circ\mathcal N^{\otimes n}\circ\mathcal E$ . Then

$$\log_2 \|R_{\mathcal{M}}\|_1 \le \log_2 \|R_{\mathcal{N}}^{\otimes n}\|_1. \tag{9.17}$$

*Proof.* The decoding procedure is a local operation and therefore from Property 4. of F(R) in Section 9.1.1, we have

$$||R_{\mathcal{M}}||_1 \le ||(\mathcal{I} \otimes (\mathcal{N}^{\otimes n} \circ \mathcal{E}))(R_{\mathcal{I}^{\otimes k}})||_1. \tag{9.18}$$

Let  $\mathcal{E}$  encode k qubits into m qubits. Using Lemma 1, we have

$$\|(\mathcal{I} \otimes (\mathcal{N}^{\otimes n} \circ \mathcal{E}))(R_{\mathcal{I}^{\otimes k}})\|_{1} = \|(\mathcal{E}^{\dagger} \otimes \mathcal{N}^{\otimes n})(R_{\mathcal{I}^{\otimes m}})\|_{1}$$
$$= \|(\mathcal{E}^{\dagger} \otimes \mathcal{I})(R_{N}^{\otimes n})\|_{1}. \tag{9.19}$$

The PDM  $R_{\mathcal{N}}^{\otimes n}$  can be decomposed into its positive and negative part, and rewritten as

$$R_{\mathcal{N}}^{\otimes n} = R_{+} - R_{-},$$
 (9.20)

where both  $R_+$  and  $R_-$  are positive semi-definite. Applying the triangle inequality gives

$$\|(\mathcal{E}^{\dagger} \otimes \mathcal{I})(R_{\mathcal{N}}^{\otimes n})\|_{1} \leq \|(\mathcal{E}^{\dagger} \otimes \mathcal{I})(R_{+})\|_{1} + \|(\mathcal{E}^{\dagger} \otimes \mathcal{I})(R_{-})\|_{1}$$

$$= \operatorname{tr}((\mathcal{E}^{\dagger} \otimes \mathcal{I})(R_{+})) + \operatorname{tr}((\mathcal{E}^{\dagger} \otimes \mathcal{I})(R_{-}))$$

$$= \operatorname{tr}((\mathcal{E}^{\dagger} \otimes \mathcal{I})(R_{+} + R_{-})). \tag{9.21}$$

It is well-known that in bounding quantum channel capacity, one can restrict  $\mathcal E$  to be an

isometry with only one non-zero Kraus operator, which we denote by K [129]. This allows us to write

$$\operatorname{tr}((\mathcal{E}^{\dagger} \otimes \mathcal{I})(R_{+} + R_{-})) = \operatorname{tr}((K^{\dagger} \otimes I)(R_{+} + R_{-})(K \otimes I))$$
$$= \operatorname{tr}((KK^{\dagger} \otimes I)(R_{+} + R_{-})), \tag{9.22}$$

where the second equality follows from the cyclic property of the trace. Since  $P = KK^{\dagger} \otimes \mathcal{I}^{\otimes n}$  is a projector, so that PP = P, we have

$$\operatorname{tr}(P(R_{+} + R_{-})) = \operatorname{tr}(P(R_{+} + R_{-})P) = ||P(R_{+} + R_{-})P||_{1}. \tag{9.23}$$

Next we applying the Hölder's inequality twice and make use of the fact the infinity norm of a projector equals one, and obtain

$$||P(R_{+} + R_{-})P||_{1} \le ||P||_{\infty} ||R_{+} + R_{-}||_{1} ||P||_{\infty}$$

$$= ||R_{+} + R_{-}||_{1}, \qquad (9.24)$$

where  $\|\cdot\|_{\infty}$  denotes the infinity norm and is defined by the largest singular value of a matrix. Furthermore,  $R_+$  and  $R_-$  are by definition orthogonal. Hence

$$||R_{+} + R_{-}||_{1} = ||R_{+} - R_{-}||_{1} = ||R_{\mathcal{N}}^{\otimes n}||_{1},$$
 (9.25)

which leads to  $||R_{\mathcal{M}}||_1 \leq ||R_{\mathcal{N}}^{\otimes n}||_1$ . Finally, use the fact that logarithm is a monotonic function, the desired property Eq. 9.17 follows.

#### 9.2.2 Large-n limit

Here we prove that the error parameter  $\epsilon$  in Eq. 9.12 goes to zero in the limit of large n. By the definition of the distance in diamond norm, we have

$$\epsilon = \|\mathcal{I}^{\otimes k} \otimes (\mathcal{M} - \mathcal{I}^{\otimes k})\|_{1}$$

$$= \sup_{\|X\|_{1}=1} \|(\mathcal{I}^{\otimes k} \otimes (\mathcal{M} - \mathcal{I}^{\otimes k}))(X)\|_{1}.$$
(9.26)

Consider the spectral decomposition of Hermitian  $X = \sum_i \lambda_i |\psi_i\rangle \langle \psi_i|$ , where  $\{|\psi_i\rangle\}$  denotes an orthonormal basis, and  $\{\lambda_i\}$  are the corresponding eigenvalues. Define  $\mathcal{A} = (\mathcal{I}^{\otimes k} \otimes (\mathcal{M} - \mathcal{I}^{\otimes k}))$ , and we have

$$\epsilon = \sup_{\{|\psi_{i}\rangle\}_{i}, \sum_{i} |\lambda_{i}| = 1} \left\| \mathcal{A}\left(\sum_{i} \lambda_{i} |\psi_{i}\rangle\langle\psi_{i}|\right) \right\|_{1}$$

$$\leq \sup_{\{|\psi_{i}\rangle\}_{i}, \sum_{i} |\lambda_{i}| = 1} \left(\sum_{i} |\lambda_{i}| \|\mathcal{A}(|\psi_{i}\rangle\langle\psi_{i}|) \|_{1}\right)$$

$$\leq \sup_{|\psi\rangle} \|\mathcal{A}(|\psi\rangle\langle\psi|) \|_{1}.$$
(9.27)

Note that  $\mathcal{A}$  represents the difference of two linear maps  $\mathcal{I}^{\otimes k} \otimes \mathcal{I}^{\otimes k}$  and  $\mathcal{I}^{\otimes k} \otimes \mathcal{M}$ , by linearity we have

$$\sup_{|\psi\rangle} \|\mathcal{A}(|\psi\rangle\langle\psi|)\|_{1} = \sup_{|\psi\rangle} \|(\mathcal{I}^{\otimes k} \otimes \mathcal{I}^{\otimes k})(|\psi\rangle\langle\psi|) - (\mathcal{I}^{\otimes k} \otimes \mathcal{M})(|\psi\rangle\langle\psi|)\|_{1}. \quad (9.28)$$

In the above we have inside a supremum the trace distance between two quantum states. Now we need to relate the distance between quantum states measured by the 1-norm to that measured in terms of the fidelity. Let

$$f(\rho, \sigma) = \operatorname{tr} \sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}$$
 (9.29)

denote the fidelity between two positive semi-definite matrices. If  $\rho=|\psi\rangle\langle\psi|$ , then  $f(\rho,\sigma)=\sqrt{\langle\psi|\sigma|\psi\rangle}$ . The Fuchs-van de Graaf inequalities [132] imply

$$1 - f(\rho, \sigma) \le \frac{1}{2} \|\rho - \sigma\|_1 \le \sqrt{1 - f(\rho, \sigma)^2}.$$
 (9.30)

Hence we have

$$\frac{1}{2} \sup_{|\psi\rangle} \|\mathcal{A}(|\psi\rangle\langle\psi|)\|_{1} \leq \sqrt{1 - \inf_{|\psi\rangle} f((\mathcal{I}^{\otimes k} \otimes \mathcal{I}^{\otimes k})(|\psi\rangle\langle\psi|), (\mathcal{I}^{\otimes k} \otimes \mathcal{M})(|\psi\rangle\langle\psi|))^{2}}.$$
(9.31)

The above inequality is related to entanglement fidelity  $F_e(\rho, \Phi)$  of a state  $\rho$  with respect to the channel  $\Phi$  which has a set of Kraus operators,  $\mathcal{K}$ , and acts on the state as  $\Phi(\rho) = \sum_{A \in K} A \rho A^{\dagger}$ . From Schumacher's formula [133], we have

$$F_{e}(\rho, \Phi) = \langle \phi | (\Phi \otimes \mathcal{I})(|\phi\rangle\langle\phi|) | \phi \rangle$$

$$= f((\Phi \otimes \mathcal{I})(|\phi\rangle\langle\phi|), |\phi\rangle\langle\phi|)$$

$$= \sum_{A \in K} |\operatorname{tr} \rho A|^{2}, \tag{9.32}$$

where  $|\phi\rangle$  is introduced as a purification of  $\rho$ . We denote  $F_e(\Phi) = \inf_{\rho} F_e(\rho, \Phi)$ , and have

$$F_e(\Phi) = \inf_{|\phi\rangle} \langle \phi | (\Phi \otimes \mathcal{I})(|\phi\rangle \langle \phi|) | \phi\rangle = \inf_{|\phi\rangle} f(|\phi\rangle \langle \phi|, (\Phi \otimes \mathcal{I})(|\phi\rangle \langle \phi|))^2. \tag{9.33}$$

Hence using the notation for the entanglement fidelity, we can write

$$\frac{1}{2} \sup_{|\psi\rangle} \|\mathcal{A}(|\psi\rangle\langle\psi|)\|_1 \le \sqrt{1 - F_e(\mathcal{M})}. \tag{9.34}$$

Thus  $\epsilon \leq 2\sqrt{1 - F_e(\mathcal{M})}$ . Now use the following relation proved by Kretschmann and Werner in Proposition 4.3 of Ref. [134]

$$1 - F_e(\Phi) \le 4\sqrt{\|\Phi - \mathcal{I}\|_{cb}} \le 8\left(1 - F_e(\Phi)\right)^{1/4},\tag{9.35}$$

where  $\|\cdot\|_{cb}$  denotes the completely bounded norm induced on the operator infinity norm [135]. We obtain

$$\epsilon \le 2\sqrt{4\sqrt{\|\mathcal{M} - \mathcal{I}\|_{cb}}}$$

$$= 4\|\mathcal{M} - \mathcal{I}\|_{cb}^{1/4}.$$
(9.36)

Since  $\|\mathcal{M} - \mathcal{I}\|_{cb}$  is guaranteed to approach zero as n approaches infinity in the channel capacity theorems,  $\epsilon$  here also approaches zero. This concludes the proof.

## 9.3 Application of causal bound

## 9.3.1 Comparison with Holevo and Werner bound

Here we compare the causal bound with a simple well-known bound on quantum capacities of Holevo and Werner (HW) which is general, and has a similar form, but requires optimisation [118]. Given a quantum channel  $\mathcal{N}$ , and a transpose map  $\mathcal{T}$ , the Holevo-Werner upper bound on the quantum capacity is

$$Q_{\mathcal{T}}(\mathcal{N}) = \log_2 \|\mathcal{N}\mathcal{T}\|_{\diamond} = \log_2 \|\mathcal{I} \otimes \mathcal{N}\mathcal{T}\|_1. \tag{9.37}$$

By the definition of the induced norm this can be rewritten as

$$Q_{\mathcal{T}}(\mathcal{N}) = \sup_{\rho} \left( \log_2 \| (\mathcal{I} \otimes \mathcal{N}\mathcal{T})(\rho) \|_1 \right). \tag{9.38}$$

Now we compare the above to the causal bound. In the case of the maximally mixed input, the pseudo-density matrix becomes

$$R_{\mathcal{N}} = (\mathcal{I} \otimes \mathcal{N}) \left( \frac{\text{SWAP}^{\otimes k}}{2^k} \right) = (\mathcal{I} \otimes \mathcal{N} \mathcal{T}) (|\Phi^+\rangle \langle \Phi^+|)^{\otimes k}. \tag{9.39}$$

The causal bound then reads

$$F(R_{\mathcal{N}}) = \log_2 \| (\mathcal{I} \otimes \mathcal{N}\mathcal{T}) (|\Phi^+\rangle \langle \Phi^+|)^{\otimes k} \|_1.$$
 (9.40)

Comparing this to the HW bound in Eq. 9.38, it is clear that  $F(R_N) \leq Q_T(N)$ , and the two are equal when the supremum is achieved at the maximally entangled state  $(|\Phi^+\rangle \langle \Phi^+|)^{\otimes k}$ . Hence we have shown the causal bound is better or equal to the HW bound.

## 9.3.2 Shifted depolarising channel

As an illustration of applying the causal bound, we consider the class of shifted depolarising channels. A shifted depolarising channel generalises the well-studied quantum depolarising channel [136, 137]. It outputs either the input state or the state  $\frac{I+\gamma Z}{2}$  shifted from the maximally mixed state with probability 4p. For a single qubit the shifted depolarising channel can be defined by

$$\mathcal{N}_{\gamma}(\rho) = (1 - 4p)\rho + 4p\left(\frac{I + \gamma Z}{2}\right),\tag{9.41}$$

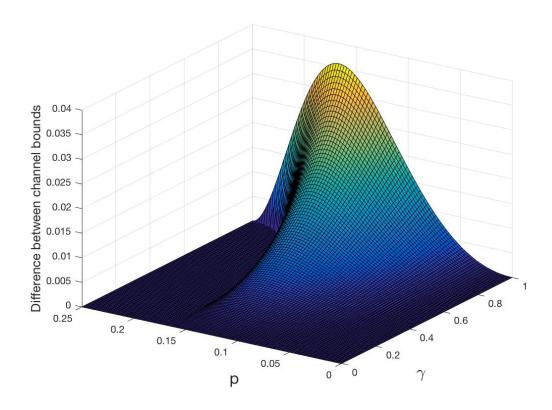
where the parameter  $\gamma \in [0,1]$  parametrises the shift, with a zero  $\gamma$  corresponding to the standard depolarising channel. The PDM representation of the single qubit shifted depolarising channel,  $R_{\mathcal{N}_{\gamma}}$  is derived using Eq. 9.3, from which we obtain an analytic expression for the value of  $F(R_{\mathcal{N}_{\gamma}})$ , and hence an upper bound on its quantum capacity

of the channel,

$$Q(\mathcal{N}_{\gamma}) \leq F(R_{\mathcal{N}_{\gamma}})$$

$$= \log_{2} \left( 1 - p + \frac{1}{2} \sqrt{1 - 8p + 16p^{2} + 4\gamma^{2}p^{2}} + \frac{1}{2} \left| 2p - \sqrt{1 - 8p + 16p^{2} + 4\gamma^{2}p^{2}} \right| \right). \tag{9.42}$$

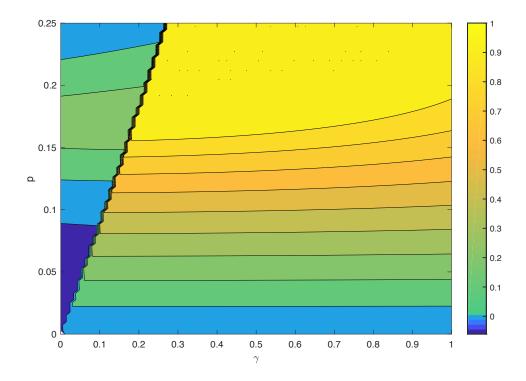
We show in Figure 9.2 the difference between the HW bound and the causal bound on quantum channel capacity of a shifted depolarising channel. Note that the two bounds are identical for standard depolarising channel where there is no shift. However, the causal bound is tighter when the shift  $\gamma$  increases.



**Figure 9.2:** Difference between the HW and causal bound s on quantum channel capacity of a shifted depolarising channel.

Hence the shifted depolarising channel constitutes a class of examples for which the

causal bound is strictly tighter than the HW bound. Furthermore, we found that the causal bound  $F(R_{N_{\gamma}})$  also shows improvement upon the best known bound from Ref. [117]. In Figure 9.3, we show the difference between the previously known bound from Ref. [117] and the causal bound on the quantum channel capacity of a shifted depolarizing channel.



**Figure 9.3:** Difference between the previously known bound from Ref. [117] and the causal bound on quantum channel capacity of a shifted depolarising channel.

The causal bound is tighter for almost all values of  $\gamma$  and p. Only in the region of small shift  $\gamma$  and small probability p, which corresponds to the bottom left corner of the diagram, the causal bound is less tight. Note that the shifted depolarising channel reduces to the standard depolarising channel when  $\gamma=0$ , and the identity channel when p=0. The causal bound is not the tightest known bound for the standard depolarising channel, while it evaluates exactly to the quantum channel capacity for the identity channel.

## 9.4 Summary and discussions

In this chapter, we have presented a general upper bound on the quantum capacity of noisy quantum channels based on fundamental causality considerations. Contrary to most other existing bounds, the computation of the causal bound does not involve an explicit optimisation problem. The logarithmic causality measure used here is in close analogy with the entanglement logarithmic negativity and possesses desired properties which make it useful for studying channel capacities.

Our approach based on quantum causality is generally applicable to arbitrary quantum channels and can produce non-trivial upper bounds for any given channel. Therefore, this result could further help the understanding of the communication rate of complex systems for which optimisation methods are computationally too costly, including quantum networks and quantum communication between many parties [138, 139].

Research on the spatial quantum correlations has lead to the formulation of various entanglement monotones with various corresponding operational meanings and applications, e.g., distillable entanglement, entanglement cost, squashed entanglement [103,140]. As a temporal counterpart of quantum correlations, the result presented in this chapter initiates research on the operational significance of causality measures that might prove useful in a broader range of applications.

# **Chapter 10**

## **Conclusion**

In this thesis, I started by describing the useful quantum algorithms for linear algebra, then moved on to illustrate how the quantum algorithm machinery can be applied to enhance classical supervised learning. In the last part of the thesis, we studied the notion of causality in an ensemble of quantum states. I presented results on inferring causal corrections, and the connection between quantum causality and the limit of transmitting quantum information over a noisy channel. Here we provide a summary of new research progress discussed in this thesis and give a brief outlook for avenues of future research.

## 10.1 Summary

In Chapter 3, I have shown a new quantum algorithm for solving the quantum linear system problem. This approach is based on a quantum singular value decomposition technique which in turn makes use of a data structure that provides oracle access to the row vectors of a matrix and the vector of row norms. Since our approach does not involve explicitly simulating the system's defining matrix as a Hamiltonian, the resultant runtime does not depend on sparsity, which gives the new linear systems algorithm an advantage over the existing approach for dense matrices. As a result of the error dependence in singular value decomposition, the fixed-error runtime of our linear system algorithm has a linear dependence on the Frobenius norm of the matrix. Nevertheless, we have proved

our algorithm has a  $\tilde{\mathcal{O}}(\sqrt{n})$  runtime in the general case, providing a polynomial speedup over the previous state-of-the-art. In the special case of the matrix having a low-rank structure, our algorithm exhibits an even more advantageous  $\tilde{\mathcal{O}}(\log n)$  runtime scaling.

In Chapters 5 and 6, we applied quantum algorithms to supervised machine learning using Gaussian processes. For computing the mean and variance predictor of a given GP model, we have shown the quantum linear systems approach can be applied to achieve exponential or polynomial speedups over classical implementations depending on whether the covariance matrix is sparse or not. For the purpose of training GPs, we have presented a quantum approach for evaluating the logarithm of the marginal likelihood of the model on a given dataset. The quantum GP training approach has two main components, the augmented quantum linear system algorithm for quantifying the model's performance on the training data, and the quantum log determinant algorithm for quantifying the complexity of the model. We have shown the quantum GP training approach allows for efficiently evaluating the variation of marginal likelihood on each training step, which is the main computation bottleneck for model selections for GPs. The quantum GP prediction and training procedures together provide a concrete use-case in supervised learning for which quantum computation has a provable advantage over the best-known classical implementation.

In Chapter 7, we built upon the previously discussed quantum GP algorithm and leveraged a connection between deep neural network models and Gaussian processes to develop a quantum algorithm for deep learning. The presented quantum approach to deep learning is Bayesian as the training of the parameters in the neural network amounts to evaluating a Gaussian posterior distribution instead of the more conventional methods, such as backpropagation with stochastic gradient descent. To simulate the Hamiltonian that represents the multi-layer kernel matrix, we designed a quantum method based on density matrix exponentiation and proved the computational overhead in terms of the

required number of resource density matrix which encodes the base case kernel matrix. Furthermore, we have demonstrated the matrix inversion component of quantum GP regression by performing experiments on quantum simulators as well as the state-of-the-art quantum processing units, which have shown encouraging results, despite the implementation being a small-scale variant of the full algorithm.

In Chapter 8 and 9, we looked into the concept of causality in the quantum domain. Specifically, we have made use of the pseudo-density matrix formalism to derive the geometric structure of spatial and temporal two-point quantum correlations, which serves as an analytical toolkit for inferring causal relations in quantum datasets. Furthermore, the geometric structure can be seen as a strong witness of quantum entanglement, distinguishing it from possible sequentially generated statistics. We then further apply quantum causality in the pseudo-density matrix formalism to quantum communication and derived a general upper bound for the quantum channel capacity of a given a noisy channel.

### 10.2 Outlook

The results presented in this thesis provide numerous potential avenues for further research. As discussed earlier in Chapter 3, it would be useful to conduct a detailed resource analysis for the QDLS algorithm. Since it circumvents the costly Hamiltonian simulation subroutine, as required by the previous quantum linear system algorithms, implementing the QDLS algorithm may require significantly less elementary gate operations compared to the analysis presented in Ref. [41]. Given the close analogy between the quantum walk based approach of QDLS algorithm and the quantum search algorithm [2], it is also interesting to ask whether the runtime  $\tilde{\mathcal{O}}(\sqrt{n}\log n)$  is optimal given the required memory model following a similar logic of the optimality of quantum search [141].

The main direction of interest for quantum enhanced GPs and Bayesian deep learning presented in Chapter 5, 6 and 7 is experimental. Although the development of practical hardware for quantum computing is still at its infancy, early quantum computers have already become available and will continue to grow in scale and noise tolerance. It is an exciting time to ask whether near-term quantum computing can truly enhance machine learning, either on a qualitative or a quantitative level. We hope that before too long the quantum GP algorithms, its corresponding training algorithms, and the quantum GP induced Bayesian deep learning approach can be fully implemented with real quantum devices on large-scale datasets, and ultimately produce analytical power beyond what is classically achievable.

The geometric structure presented in Chapter 8 identifies a class of quantum operations which can generate sequential statistics that mimics entanglement. It would be interesting to observe these correlations experimentally. Furthermore, as quantum entanglement is famously given a significant role in quantum cryptography [92], it is interesting to ask whether its temporal counter-part, causality would provide similar applicational prospects. The results presented in Chapter 9 are a concrete example of the operational meaning of causality, where it is shown to be significant to the field of quantum communication. Thus the presented work initiates a thread of research on the practical applications of quantum causality.

# **Bibliography**

- [1] Richard P Feynman. Simulating physics with computers. *International journal of theoretical physics*, 21(6/7):467–488, 1982.
- [2] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219. ACM, 1996.
- [3] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.
- [4] Esma Aïmeur, Gilles Brassard, and Sébastien Gambs. Machine learning in a quantum world. In *Advances in Artificial Intelligence*, pages 431–442. Springer, 2006.
- [5] Kristen L Pudenz and Daniel A Lidar. Quantum adiabatic machine learning. *Quantum information processing*, 12(5):2027–2070, 2013.
- [6] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum algorithms for supervised and unsupervised machine learning. arXiv preprint arXiv:1307.0411, 2013.
- [7] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical Review Letters*, 113(13):130503, 2014.

BIBLIOGRAPHY 128

[8] Vedran Dunjko and Hans J Briegel. Machine learning\& artificial intelligence in the quantum domain. *arXiv preprint arXiv:1709.02779*, 2017.

- [9] Alejandro Perdomo-Ortiz, Marcello Benedetti, John Realpe-Gómez, and Rupak Biswas. Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers. *arXiv preprint arXiv:1708.09757*, 2017.
- [10] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.
- [11] Peter W Shor. The classical capacity achievable by a quantum channel assisted by limited entanglement. *arXiv preprint quant-ph/0402129*, 2004.
- [12] Alexei Yu Kitaev, Alexander Shen, and Mikhail N Vyalyi. *Classical and quantum computation*. Number 47. American Mathematical Soc., 2002.
- [13] Lisa Hales and Sean Hallgren. An improved quantum fourier transform algorithm and applications. In *Foundations of Computer Science*, 2000. *Proceedings*. 41st Annual Symposium on, pages 515–525. IEEE, 2000.
- [14] A. Yu. Kitaev. Quantum measurements and the Abelian stabilizer problem. 1995.
- [15] Dominic W. Berry and Andrew M. Childs. Black-box Hamiltonian simulation and unitary implementation. *Quantum Information & Computation*, 12(1–2):29–62, 2009.
- [16] Mario Szegedy. Quantum speed-up of markov chain based algorithms. In Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on, pages 32–41. IEEE, 2004.
- [17] Dominic W Berry, Andrew M Childs, and Robin Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *Foundations of Computer*

BIBLIOGRAPHY 129

Science (FOCS), 2015 IEEE 56th Annual Symposium on, pages 792–809. IEEE, 2015.

- [18] Dominic W Berry, Andrew M Childs, Richard Cleve, Robin Kothari, and Rolando D Somma. Exponential improvement in precision for simulating sparse hamiltonians. In *Forum of Mathematics, Sigma*, volume 5. Cambridge University Press, 2017.
- [19] Guang Hao Low and Isaac L Chuang. Hamiltonian simulation by qubitization. arXiv preprint arXiv:1610.06546, 2016.
- [20] Shelby Kimmel, Cedric Yen-Yu Lin, Guang Hao Low, Maris Ozols, and Theodore J Yoder. Hamiltonian simulation with optimal sample complexity. npj Quantum Information, 3(1):13, 2017.
- [21] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631, 2014.
- [22] Andrew M Childs. Lecture notes on quantum algorithms. *Lecture notes at University of Maryland*, 2017.
- [23] Jonathan Richard Shewchuk et al. An introduction to the conjugate gradient method without the agonizing pain, 1994.
- [24] Don Coppersmith and S Winograd. Matrix multiplication via arithmetic progressions. *Journal of symbolic computation*, 1990.
- [25] François Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings* of the 39th international symposium on symbolic and algebraic computation, pages 296–303. ACM, 2014.
- [26] Scott Aaronson. Read the fine print. *Nature Physics*, 11(4), 2015.

[27] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103, 2009.

- [28] Vladimír Bužek, Radoslav Derka, and Serge Massar. Optimal quantum clocks. *Physical Review Letters*, 82(10):2207, 1999.
- [29] Dominic W Berry, Graeme Ahokas, Richard Cleve, and Barry C Sanders. Efficient quantum algorithms for simulating sparse hamiltonians. *Communications in Mathematical Physics*, 270(2):359–371, 2007.
- [30] Andris Ambainis. Variable time amplitude amplification and quantum algorithms for linear algebra problems. In 29th International Symposium on Theoretical Aspects of Computer Science, STACS 2012, February 29th March 3rd, 2012, Paris, France, 2012.
- [31] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. Quantum linear systems algorithm with exponentially improved dependence on precision. 2015.
- [32] B. D. Clader, B. C. Jacobs, and C. R. Sprouse. Preconditioned quantum linear system algorithm. *Physical Review Letters*, 110(25), 2013.
- [33] Leonard Wossnig, Zhikuan Zhao, and Anupam Prakash. Quantum linear system algorithm for dense matrices. *Physical review letters*, 120(5):050502, 2018.
- [34] Iordanis Kerenidis and Anupam Prakash. Quantum recommendation systems. In *Innovations in Theoretical Computer Science*, 2017.
- [35] Anupam Prakash. *Quantum Algorithms for Linear Algebra and Machine Learning*. PhD thesis, University of California, Berkeley, 2014.
- [36] Andrew M Childs. On the relationship between continuous-and discrete-time quantum walk. *Communications in Mathematical Physics*, 294(2):581–603, 2010.

[37] Miklos Santha. Quantum walk based search algorithms. *Theory and Applications of Models of Computation*, pages 31–46, 2008.

- [38] Aram W Harrow. Review of quantum algorithms for systems of linear equations. arXiv preprint arXiv:1501.00008, 2014.
- [39] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Quantum Computation and Information*, 305, 2002.
- [40] P. C. Hansen. Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion. 1998.
- [41] Artur Scherer, Benoît Valiron, Siun-Chuon Mau, Scott Alexander, Eric van den Berg, and Thomas E. Chapuran. Concrete resource analysis of the quantum linear-system algorithm used to compute the electromagnetic scattering cross section of a 2d target. *Quantum Information Processing*, 16, 2017.
- [42] Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*, volume 14. 2004.
- [43] Zhikuan Zhao, Jack K Fitzsimons, and Joseph F Fitzsimons. Quantum assisted gaussian process regression. *arXiv preprint arXiv:1512.03929*, 2015.
- [44] Zhikuan Zhao, Jack K Fitzsimons, Michael A Osborne, Stephen J Roberts, and Joseph F Fitzsimons. Quantum algorithms for training gaussian processes. *arXiv* preprint arXiv:1803.10520, 2018.
- [45] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

[46] Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005.

- [47] James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. In *Uncertainty in Artificial Intelligence*, 2013.
- [48] Marc Peter Deisenroth and Jun Wei Ng. Distributed gaussian processes. *arXiv* preprint arXiv:1502.02843, 2015.
- [49] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as Gaussian processes. 2017.
- [50] Radford M. Neal. Priors for infinite networks. Technical Report crg-tr-94-1, University of Toronto, 1994.
- [51] Alexander G. de G. Matthews, Jiri Hron, Mark Rowland, Richard E. Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. In *International Conference on Learning Representations*, 2018.
- [52] Victor Minden, Anil Damle, Kenneth L Ho, and Lexing Ying. Fast spatial gaussian process maximum likelihood estimation via skeletonization factorizations. *arXiv* preprint arXiv:1603.08057, 2016.
- [53] R Kelley Pace and James P LeSage. Chebyshev approximation of log-determinants of spatial weight matrices. *Computational Statistics & Data Analysis*, 45(2):179–196, 2004.
- [54] Christos Boutsidis, Petros Drineas, Prabhanjan Kambadur, and Anastasios Zouzias. A randomized algorithm for approximating the log determinant of a symmetric positive definite matrix. *arXiv preprint arXiv:1503.00374*, 2015.

[55] Haim Avron and Sivan Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM (JACM)*, 58(2):8, 2011.

- [56] J. K. Fitzsimons, M. A Osborne, S. J. Roberts, and J. F. Fitzsimons. Improved stochastic trace estimation using mutually unbiased bases. arXiv preprint arXiv:1608.00117, 2016.
- [57] Zhikuan Zhao, Vedran Dunjko, Jack K Fitzsimons, Patrick Rebentrost, and Joseph F Fitzsimons. A note on state preparation for quantum machine learning. arXiv preprint arXiv:1804.00281, 2018.
- [58] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Architectures for a quantum random access memory. *Physical Review A*, 78(5):052310, 2008.
- [59] Andrei N Soklakov and Rüdiger Schack. Efficient state preparation for a register of quantum bits. *Physical Review A*, 73(1):012307, 2006.
- [60] Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *arXiv preprint quant-ph/9605034*, 1996.
- [61] Theodore J Yoder, Guang Hao Low, and Isaac L Chuang. Fixed-point quantum search with an optimal number of queries. *Physical review letters*, 113(21):210501, 2014.
- [62] Liming Zhao, Carlos A Pérez-Delgado, and Joseph F Fitzsimons. Fast graph operations in quantum computation. *arXiv preprint arXiv:1510.03742*, 2015.
- [63] Arman Melkumyan and Fabio Ramos. A sparse covariance function for exact gaussian process inference in large datasets. In *IJCAI*, volume 9, pages 1936–1942, 2009.

[64] Soohwan Kim and Jonghyuk Kim. Gpmap: A unified framework for robotic mapping based on sparse gaussian processes. In *Field and Service Robotics*, pages 319–332. Springer, 2015.

- [65] Li Liu, Ling Shao, Feng Zheng, and Xuelong Li. Realistic action recognition via sparsely-constructed gaussian processes. *Pattern Recognition*, 47(12):3819–3827, 2014.
- [66] Reinhard Furrer, Marc G Genton, and Douglas Nychka. Covariance tapering for interpolation of large spatial datasets. *Journal of Computational and Graphical Statistics*, 15(3):502–523, 2006.
- [67] Ronald P Barry and R Kelley Pace. Kriging with large data sets using sparse matrix techniques. *Communications in Statistics-Simulation and Computation*, 26(2):619–629, 1997.
- [68] J Bernardo, J Berger, A Dawid, A Smith, et al. Regression and classification using gaussian process priors. *Bayesian statistics*, 6:475, 1998.
- [69] Richard S Varga. *Geršgorin and his circles*, volume 36. Springer Science & Business Media, 2010.
- [70] Nana Liu, Jayne Thompson, Christian Weedbrook, Seth Lloyd, Vlatko Vedral, Mile Gu, and Kavan Modi. The power of one qumode. *arXiv preprint* arXiv:1510.04758, 2015.
- [71] A Luis and J Peřina. Optimum phase-shift estimation and the quantum description of the phase difference. *Physical review A*, 54(5):4564, 1996.
- [72] Robert S. Smith, Michael J. Curtis, and William J. Zeng. A practical quantum instruction set architecture. 2016.

[73] Andrew W. Cross, Lev S. Bishop, John A. Smolin, and Jay M. Gambetta. Open quantum assembly language. 2017.

- [74] Zhikuan Zhao, Alejandro Pozas-Kerstjens, Patrick Rebentrost, and Peter Wittek. Bayesian deep learning on a quantum computer. arXiv preprint arXiv:1806.11463, 2018.
- [75] John Bradshaw, Alexander G. de G. Matthews, and Zoubin Ghahramani. Adversarial examples, uncertainty, and transfer testing robustness in Gaussian process hybrid deep networks. 2017.
- [76] Kathrin Grosse, David Pfaff, Michael Thomas Smith, and Michael Backes. How wrong am I? studying adversarial examples and their impact on uncertainty in Gaussian process machine learning models. 2017.
- [77] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. 2015.
- [78] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of ICML-26, 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR.
- [79] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical Review Letters*, 113:130503, September 2014.
- [80] Patrick Rebentrost, Maria Schuld, Francesco Petruccione, and Seth Lloyd. Quantum gradient descent and newton's method for constrained polynomial optimization. *arXiv preprint arXiv:1612.01789*, 2016.

- [81] Liming Zhao. Quantum algorithms and data structures. 2018.
- [82] Masuo Suzuki. General theory of higher-order decomposition of exponential operators and symplectic integrators. *Physics Letters A*, 165(5-6):387–395, 1992.
- [83] Andrew M Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 59–68. ACM, 2003.
- [84] Nathan Wiebe, Dominic Berry, Peter Høyer, and Barry C Sanders. Higher order decompositions of ordered operator exponentials. *Journal of Physics A: Mathematical and Theoretical*, 43(6):065203, 2010.
- [85] Yuanhao Wang, Ying Li, Zhang-qi Yin, and Bei Zeng. 16-qubit IBM universal quantum computer can be fully entangled. 2018.
- [86] Yudong Cao, Anmer Daskin, Steven Frankel, and Sabre Kais. Quantum circuit design for solving linear systems of equations. *Molecular Physics*, 110(15-16):1675–1680, 2012.
- [87] Yudong Cao, Anargyros Papageorgiou, Iasonas Petras, Joseph Traub, and Sabre Kais. Quantum algorithm and circuit design solving the Poisson equation. *New Journal of Physics*, 15(1):013021, 2013.
- [88] Daniel Gottesman and Isaac Chuang. Quantum digital signatures. 2001.
- [89] Zhikuan Zhao, Robert Pisarczyk, Jayne Thompson, Mile Gu, Vlatko Vedral, and Joseph F Fitzsimons. Geometry of quantum correlations in space-time. *arXiv* preprint arXiv:1711.05955, 2017.

[90] Albert Einstein, Boris Podolsky, and Nathan Rosen. Can quantum-mechanical description of physical reality be considered complete? *Physical review*, 47(10):777, 1935.

- [91] John S Bell. On the einstein podolsky rosen paradox. *Physics*, 1(195), 1964.
- [92] Artur K Ekert. Quantum cryptography based on bell's theorem. *Physical Review Letters*, 67(6):661, 1991.
- [93] Aram Harrow, Patrick Hayden, and Debbie Leung. Superdense coding of quantum states. *Physical Review Letters*, 92(18):187901, 2004.
- [94] Chris J Isham. Canonical quantum gravity and the problem of time. In *Integrable* systems, quantum groups, and quantum field theories, pages 157–287. Springer, 1993.
- [95] A. J. Leggett and A. Garg. Quantum mechanics versus macroscopic realism: Is the flux there when nobody looks? *Physical Review Letters*, 54(9):857–860, 1985.
- [96] Caslav Brukner, Samuel Taylor, Sancho Cheung, and Vlatko Vedral. Quantum entanglement in time. *arXiv preprint quant-ph/0402127*, 2004.
- [97] Costantino Budroni, Tobias Moroder, Matthias Kleinmann, and Otfried Gühne. Bounding temporal quantum correlations. *Physical Review Letters*, 111(2):020403, 2013.
- [98] Simon Milz, Felix A Pollock, and Kavan Modi. An introduction to operational quantum dynamics. *arXiv preprint arXiv:1708.00769*, 2017.
- [99] Jordan Cotler, Chao-Ming Jian, Xiao-Liang Qi, and Frank Wilczek. Superdensity operators for spacetime quantum mechanics. *arXiv* preprint arXiv:1711.03119, 2017.

[100] Kavan Modi. Operational approach to open dynamics and quantifying initial correlations. *Scientific reports*, 2:srep00581, 2012.

- [101] Clive Emary, Neill Lambert, and Franco Nori. Leggett–garg inequalities. *Reports on Progress in Physics*, 77(1):016001, 2013.
- [102] Joseph F Fitzsimons, Jonathan A Jones, and Vlatko Vedral. Quantum correlations which imply causation. *Scientific Reports*, 5, 2015.
- [103] Ryszard Horodecki, Paweł Horodecki, Michał Horodecki, and Karol Horodecki. Quantum entanglement. *Reviews of modern physics*, 81(2):865, 2009.
- [104] Michał Horodecki, Paweł Horodecki, and Ryszard Horodecki. Separability of mixed states: necessary and sufficient conditions. *Physics Letters A*, 223(1):1–8, 1996.
- [105] Karol Zyczkowski and Ingemar Bengtsson. Geometry of quantum states, 2006.
- [106] Dominic Horsman, Chris Heunen, Matthew F Pusey, Jonathan Barrett, and Robert W Spekkens. Can a quantum state over time resemble a quantum state at a single time? In *Proc. R. Soc. A*, volume 473, 2017.
- [107] Mary Beth Ruskai, Stanislaw Szarek, and Elisabeth Werner. An analysis of completely-positive trace-preserving maps on m2. *Linear Algebra and its Applications*, 347(1-3):159–187, 2002.
- [108] Christopher King and Mary Beth Ruskai. Minimal entropy of states emerging from noisy quantum channels. *IEEE Transactions on information theory*, 47(1):192– 209, 2001.
- [109] Man-Duen Choi. Completely positive linear maps on complex matrices. *Linear Algebra and its Applications*, 10(3):285–290, 1975.

[110] Guifré Vidal and Reinhard F Werner. Computable measure of entanglement. *Physical Review A*, 65(3):032314, 2002.

- [111] D Mundarain and J Stephany. Concurrence and negativity as distances. *arXiv* preprint arXiv:0712.1015, 2007.
- [112] Katja Ried, Megan Agnew, Lydia Vermeyden, Dominik Janzing, Robert W Spekkens, and Kevin J Resch. A quantum advantage for inferring causal structure. *Nature Physics*, 11(5):414–420, 2015.
- [113] V Bužek, M Hillery, and RF Werner. Optimal manipulations with qubits: Universal-not gate. *Physical Review A*, 60(4):R2626, 1999.
- [114] Peter W Shor. The quantum channel capacity and coherent information. In *Lecture* notes, MSRI Workshop on Quantum Computation, 2002.
- [115] Igor Devetak. The private classical capacity and quantum capacity of a quantum channel. *Information Theory, IEEE Transactions on*, 51(1):44–55, 2005.
- [116] Mark M Wilde. Quantum information theory. Cambridge University Press, 2013.
- [117] Yingkai Ouyang. Channel covariance, twirling, contraction, and some upper bounds on the quantum capacity. *Quantum Information and Computation*, 14(11):0917–0936, 2014.
- [118] Alexander S Holevo and Reinhard F Werner. Evaluating capacities of bosonic gaussian channels. *Physical Review A*, 63(3):032312, 2001.
- [119] Robert Pisarczyk, Zhikuan Zhao, Yingkai Ouyang, Vlatko Vedral, and Joseph F Fitzsimons. Causal limit on quantum communication. *arXiv preprint* arXiv:1804.02594, 2018.

[120] C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:379–423 and 623–656, 1948.

- [121] Masahiro Takeoka, Saikat Guha, and Mark M Wilde. The squashed entanglement of a quantum channel. *IEEE Transactions on Information Theory*, 60(8):4987–4998, 2014.
- [122] Alexander Müller-Hermes, David Reeb, and Michael M Wolf. Positivity of linear maps under tensor powers. *Journal of Mathematical Physics*, 57(1):015202, 2016.
- [123] Xin Wang and Runyao Duan. A semidefinite programming upper bound of quantum capacity. In *Information Theory (ISIT), 2016 IEEE International Symposium* on, pages 1690–1694. IEEE, 2016.
- [124] David Sutter, Volkher B Scholz, and Renato Renner. Approximate degradable quantum channels. In *Information Theory (ISIT)*, 2015 IEEE International Symposium on, pages 2767–2771. IEEE, 2015.
- [125] Xin Wang, Kun Fang, and Runyao Duan. Semidefinite programming converse bounds for quantum communication. *arXiv preprint arXiv:1709.00200*, 2017.
- [126] Marco Tomamichel, Mark M Wilde, and Andreas Winter. Strong converse rates for quantum communication. *IEEE Transactions on Information Theory*, 63(1):715–727, 2017.
- [127] Mario Berta and Mark M Wilde. Amortization does not enhance the max-rains information of a quantum channel. *arXiv preprint arXiv:1709.04907*, 2017.
- [128] Seth Lloyd. Capacity of the noisy quantum channel. *Physical Review A*, 55(3):1613, 1997.

[129] Howard Barnum, Emanuel Knill, and Michael A Nielsen. On quantum fidelities and channel capacities. *IEEE Transactions on Information Theory*, 46(4):1317–1329, 2000.

- [130] Charles H Bennett, Peter W Shor, John A Smolin, and Ashish V Thapliyal. Entanglement-assisted classical capacity of noisy quantum channels. *Physical Review Letters*, 83(15):3081–3084, 1999.
- [131] Charles H. Bennett, Peter W. Shor, John A. Smolin, and Ashish V. Thapliyal. Entanglement-assisted capacity of a quantum channel and the reverse shannon theorem. *Information Theory, IEEE Transactions on*, 48(10):2637–2655, 2002.
- [132] Christopher A Fuchs and Jeroen Van De Graaf. Cryptographic distinguishability measures for quantum-mechanical states. *IEEE Transactions on Information Theory*, 45(4):1216–1227, 1999.
- [133] Benjamin Schumacher. Sending entanglement through noisy quantum channels. *Phys. Rev. A*, 54(4):2614–2628, 1996.
- [134] Dennis Kretschmann and Reinhard F Werner. Tema con variazioni: quantum channel capacity. *New Journal of Physics*, 6(1):26, 2004.
- [135] Vern Paulsen. *Completely bounded maps and dilations*, volume 146. Longman Scientific & Technical Harlow, 1986.
- [136] Christopher King. The capacity of the quantum depolarizing channel. *IEEE Transactions on Information Theory*, 49(1):221–229, 2003.
- [137] Graeme Smith and John A Smolin. Additive extensions of a quantum channel. In *Information Theory Workshop*, 2008. ITW'08. IEEE, pages 368–372. IEEE, 2008.

[138] Debbie Leung, Jonathan Oppenheim, and Andreas Winter. Quantum network communication - the butterfly and beyond. *IEEE Transactions on Information Theory*, 56(7):3478–3490, 2010.

- [139] Masahito Hayashi, Kazuo Iwama, Harumichi Nishimura, Rudy Raymond, and Shigeru Yamashita. Quantum network coding. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 610–621. Springer, 2007.
- [140] Matthias Christandl and Andreas Winter. "Squashed entanglement": an additive entanglement measure. *Journal of mathematical physics*, 45(3):829–840, 2004.
- [141] Christof Zalka. Grover's quantum searching algorithm is optimal. *Physical Review A*, 60(4):2746, 1999.