UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Resource estimation and routing of two qubit entanglement for distributed quantum computing

by Hudson Thomas Leone

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Dr. Simon Devitt, Dr. Peter Rhode, Associate Prof. Nathan Langford, and Prof. Michael Bremner

University of Technology Sydney
Faculty of Engineering and I.T.

September 2024

Certificate of Original Authorship

I, **Hudson Thomas Leone** declare that this thesis, is submitted in fulfillment of the requirements for the award of Doctor of Philosophy (Software Engineering), in the Faculty of Engineering and I.T. at the University of Technology Sydney. This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution. This research is supported by the Australian Government Research Training Program.

Signature:

Production Note:
Signature removed prior to publication.

Abstract

Two qubit entanglement is a precious commodity for quantum computing, as it enables lossless transmission of arbitrary quantum states. My first result is a framework for studying entanglement routing algorithms that I call the cost-vector formalism. Here, the edges of a multi-graph represent entanglement links that may exist between users. Entanglement routing is treated as a sequence of edge-contractions, which extend and refine entanglement throughout the network. My second work investigates lattice surgery rates between distant surface codes using entanglement mediated by quantum satellites. Our results suggest quantum satellites are unsuitable for continental-scale distributed quantum computation. My third work analyzes lattice surgery rates between trapped-ion surface codes in separate ion traps. Additionally, I estimate the number of ions that will be required to perform lattice surgery at various rates. Our prohibitive estimates indicate an urgent need for improved optical coupling for trapped-ion quantum computers to remain viable in the long term. In an unrelated final chapter, I present software that assembles quantum circuits to simulate arbitrary linear interferometers in the second quantization picture and I discuss various other simulation approaches.

To Jesus,
through Mary

Acknowledgements

With great joy and gratitude, I wish to acknowledge the following people for helping me to finish this doctoral thesis in one way or another. Beginning with those in my professional life, I first thank Dr. Peter Rhode for taking me on as his student when I had nothing to my name but a dubious undergraduate record. Peter, I will always appreciate your compassion and good humor. I thank Dr. Simon Devitt for graciously taking over as my lead supervisor when Peter left in 2022, this despite the fact that my research at the time had nothing to do with him. I thank my co-supervisors Professor Michael "Mick" Bremmner and Associate Professor Nathan Langford for their support. In particular, I thank Nathan for his generous show of time in teaching me how to write technically (and for correcting me when I've stepped out of line). I extend my gratitude to all my colleagues and collaborators, especially to Dr. Peter Turner, Dr. Alan Robertson, Dr. Ryan Mann, Dr. Thinh Le, Dr. Sam "Patrick" Elman, Srikara Shankara, Chris Howell, Jannis Ruh, Gabe White, Karl Rombauts, Gozde Ustun, Anthony O'Rourke, and Karl Lin. Dr. Yuval Sanders was especially influential in my academic formation, and I thank him not only for being a great scientist, but also a dear friend.

In my personal life, I would first like to thank my family. To my father Mark, my mother Heather, my brother Austin and all my extended family – I love you very much. To Fr. Dominic Murphy (O.P.), Fr. Paul Rousse (O.P.) and Fr. Richard Wallace (F.S.S.P.) for their spiritual direction, particularly during the pandemic. I thank the Catholic society of UTS, especially Shaughn Gilroy and Rian Galliot for their friendship and (dare I say) heroic virtue. Finally I thank all my friends from the Maternal Heart of Mary church, and the broader Catholic community here in Sydney.

Thank you all for your support!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> And which of you by taking thought,
> can add to his stature one cubit?
>
> Matthew 6:27

1.1 Thesis overview

This thesis covers a range of research topics that, although somewhat disparate, are broadly unified by a focus on the resource analysis of two qubit entanglement. The chapters (with the exception of the introduction) are arranged in chronological order. In this first chapter, I introduce the requisite concepts that are needed to understand and contextualise my work. In the second chapter, I present a new framework for studying entanglement routing algorithms. In my third chapter, I investigate the feasibility of using quantum satellites to distribute entanglement resources for distributed quantum computation. In the fourth chapter, I estimate the number of ions that a trapped-ion quantum computer needs *per module* in order to perform reliable two qubit operations between modules. The fifth and final chapter is entirely distinct from the others and has nothing to do with two-qubit entanglement. There, I present several approaches for simulating linear interferometers using digital quantum computers. I also showcase my software, which transpiles a linear interferometer into a corresponding quantum circuit. A list of my first author publications and pre-prints is presented at the end of this document.

1.2 Fundamentals of quantum computing

Much like in any other natural science, the postulates of quantum information theory may vary in resolution depending on the scope of one's research objectives. For clarity then, I begin my thesis by enumerating the postulates of quantum computing that are sufficient for understanding my research. I credit section 2.2 of Nielsen and Chuang for inspiration in helping me to succinctly formulate the premises laid out in this introduction [46].

Pure quantum states

In the context of quantum mechanics, a *particle* is a thing that can be manipulated and observed. The information content of a particle is called its *state*. This state can include *continuous variables* like position or momentum, but for quantum computer science we typically only care about a *finite set* of properties we can affect to do computations. Speaking abstractly then, a *pure state* is a unit-length vector in a finite dimensional *Hilbert space* $\mathcal{H}$. This is a complex Euclidean space equipped with an inner product $\langle \cdot, \cdot \rangle$. A pure state of a Hilbert space is notated with a Dirac *ket* (e.g. $|\psi\rangle \in \mathcal{H}$) and is also called a *statevector* or a *wavefunction*. The *amplitudes* of a state are the complex-valued coordinates of the vector with respect to a basis $\beta = \{|\psi_i\rangle\}$ of $\mathcal{H}$. When a state is a linear combination of two or more elements of β, we say the wavefunction is in a *coherent superposition*. If two states are related to each other by a *global phase*, they can be treated as the same state. [1] The conventional units of quantum information are called *qubits*, and are states of the two dimensional Hilbert space $\mathcal{H}_2$. The basis states of a qubit are denoted $|0\rangle$ and $|1\rangle$ and are sometimes called the *computational states*.

Multi-particle states

A system of multiple particles also has a state. If the particles are all *distinguishable* as we typically assume, [2] its state is a unit vector in a Hilbert space that is a *Kronecker product* of the spaces for each particle. For example, the state space of three particles with states in Hilbert spaces of dimension two, three and five respectively is,

[1] This is because the global phase of a state is a *non-observable quantity*; No measurement can resolve overall phase. Although this sounds bad, we generally don't concern ourselves since global phase doesn't affect quantum computations.

[2] Describing the state of indistinguishable particles is more nuanced since these sorts of ensembles are invariant under particle permutations. I treat this in greater detail in the last chapter of my thesis when it becomes relevant for simulating quantum optics.

$$\mathcal{H}_2 \otimes \mathcal{H}_3 \otimes \mathcal{H}_5 \tag{1.1}$$

Measurements of pure states

A *measurement* is a process involving one or more particles where an observer learns a property of the system called an *observable*. The measured value of the observable cannot typically be predicted in advance, though (as we will soon see) the likelihood of measuring any particular value can be determined from the amplitudes of the wavefunction. Initially, Einstein et. al. speculated that non-determinism in quantum measurements was an illusion caused by imperfect information; Their reasoning was that a system of particles could have a *hidden variable* (somehow inaccessible to scientists) that fixed its measurement outcome in advance [20]. The seminal result of John Bell however empirically demonstrated the contrary; Quantum mechanics as we understand it today is incompatible with a hidden-variable model [5]. This indicates that probabilistic measurements are a fact of nature and not necessarily a result of imperfect information.

In general, measurements change the wavefunction in a way that depends on what was observed. This is called *wavefunction collapse* and is irreversible without having prior information about the state. The simplest type of quantum measurement is the *Projection Valued Measure* (PVM) which, broadly speaking, describes the possible ways in which a wavefunction collapse may occur. [3] A PVM is a collection of linear operators $\Pi = \{M_i\}$ over a Hilbert space $\mathcal{H}$ that satisfies the following properties:

1. The operators of Π are all orthogonal:

$$M_i M_j = \delta_{i,j} M_i \tag{1.2}$$

   Here, $\delta_{i,j}$ is the discrete Dirac-delta function; It is equal to 1 if $i = j$ and else is equal to 0. A consequence of this property is that a measured state is projected onto one of $||\Pi||$ orthogonal subspaces of $\mathcal{H}$ where $||\cdot||$ denotes set cardinality.

[3] On the other hand, the *most* general description of a quantum measurement is the *Positive Operator Valued Measure* (POVM). One main conceptual difference between this and the PVM is that a POVM allows for the possibility of a *null measurement result* in which no useful information was learned about the state. POVMs are essential in circumstances where one wishes to discriminate between two non-orthogonal states which, although generally useful, is not relevant for the work in this thesis

2. Each $M_i \in \Pi$ is a projection:

$$M_i^2 = M_i \tag{1.3}$$

Meaning that repeated measurements don't affect the initial result since $M_i^2|\psi\rangle = M_i|\psi\rangle$.

3. The operators of Π satisfy the completness relation:

$$\sum_i M_i = I_{\mathcal{H}} \tag{1.4}$$

This property stems from the fact that a measurement has to return *something* when implemented. (A null result is not possible for a PVM!)

Given some initial state $|\psi\rangle$, the probability p_i of measuring the outcome corresponding to the projection M_i is the inner product of the projected vector $M_i|\psi\rangle$ with itself

$$p_i = \langle\psi|M_i^\dagger M_i|\psi\rangle \tag{1.5}$$

Where $M_i^\dagger$ is the conjugate transpose of M_i. Although the projected vector $M_i|\psi\rangle$ points in the same direction as the collapsed state, it is not necessarily equal to the state itself since it may not be unit length. We therefore *renormalise* the vector by dividing it by the square root of the measurement probability. The resulting state is then

$$|\psi_i\rangle := \frac{M_i|\psi\rangle}{\sqrt{\langle\psi|M_i^\dagger M_i|\psi\rangle}} \tag{1.6}$$

Measurement basis

If $\beta = \{|\psi_i\rangle\}$ is a basis for a Hilbert space $\mathcal{H}$, it is easily verified that the operators $\Pi_\beta = \{|\psi_i\rangle\langle\psi_i|\}$ form a PVM over $\mathcal{H}$ where $|\psi_i\rangle\langle\psi_i|$ is the *outer product* of $|\psi_i\rangle$ with itself. Here, we say that Π_β is a measurement uniquely defined with respect to the basis β. A device that implements Π_β is able to distinguish between all states in β with certainty. More generally, if a state $|\phi\rangle \in \mathcal{H}$ is in a *coherent superposition* of β elements the probability of measuring $|\psi_i\rangle$ with Π_β is easily seen from equation 1.5 to be

$$p_i = |\langle\psi_i|\phi\rangle|^2 \tag{1.7}$$

Density operators and mixed states

In the previous section, we saw that measurements of pure states are inherently non-deterministic even when we have perfect knowledge of the underlying statevector. In practical situations moreover, we will almost certainly have additional *classical uncertainty* about the states we posses. When a system of particles has a chance of being one of *several possible* states, the ensemble is said to be *mixed*. These states are represented not with statevectors, but with probabilistic mixtures of pure state *density operators*. For context, if $|\psi\rangle$ is a pure state, its corresponding density operator is the outer product of $|\psi\rangle$ with itself ($\rho_\psi := |\psi\rangle\langle\psi|$). Let the set of density operators defined with respect to the Hilbert space $\mathcal{H}$ be denoted $\mathcal{D}(\mathcal{H})$. Every density operator $\rho \in \mathcal{D}(\mathcal{H})$ satisfies the following properties:

1. ρ is a *Hermitian* matrix, meaning that

$$\rho = \rho^\dagger \tag{1.8}$$

2. The *trace* of ρ (the sum of the diagonal elements) is equal to one

$$\mathrm{Tr}(\rho) = 1 \tag{1.9}$$

3. ρ is *positive semi-definite* meaning that for all $|x\rangle \in \mathcal{H}$, we have that

$$\langle x|\rho|x\rangle \geq 0 \tag{1.10}$$

   Consequently, the eigenvalues of ρ are all non-negative. This property is necessary in order to make sure that every measurement probability is greater than zero.

The general form of a mixed state is

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i| \tag{1.11}$$

Where each p_i is the probability that the underlying state is $|\psi_i\rangle$. An important fact worth noting is that the set of pure states $\{|\psi_i\rangle\}$ that make up a mixed state is *not unique*. As a trivial example of this non-uniqueness, observe that

$$\frac{1}{2}\Big(|0\rangle\langle 0| + |1\rangle\langle 1|\Big) = \frac{1}{2}\Big(|+\rangle\langle +| + |-\rangle\langle -|\Big) \tag{1.12}$$

Where $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$.

Similar to vectors, density operators can be written with respect to *different coordinate bases*. Specifically, if $\beta = \{|\beta_i\rangle\}$ forms a basis of $\mathcal{H}$, then ρ can be written as follows:

$$\rho = \sum_{i,j} c_{i,j} |\beta_i\rangle\langle\beta_j| \tag{1.13}$$

The matrix $\rho_\beta := [c_{i,j}] \in \mathcal{D}(\mathcal{H})$ is the density operator ρ expressed in the β basis. The $c_{i,j}$ coordinates where $i = j$ are called *diagonal entries* since they populate the diagonal elements of ρ_β. Likewise, the terms where $i \neq j$ are called *off-diagonal* entries.

A short treatment on the spectral decomposition

In the previous section, we left our summations *unbounded* since a density operator can (in theory) be represented with a mixture of *arbitrarily many* pure states. For the sake of practicality however, it is preferable to describe the ensemble using *as few pure states as possible*. We therefore ask: What is the minimum number of pure states that are required to describe any state ρ in $\mathcal{D}(\mathcal{H}_n)$? This is answered by the *spectral decomposition theorem*, which as an aside is a rather important result for quantum information theory. For the sake of brevity however, I will only state the basic content of this theorem, and explain how it resolves the aforementioned question. A more general instance of this theorem is the *singular value decomposition* which is treated in most linear algebra textbooks (for example in [4]).

The spectral decomposition theorem states that any *normal operator* M (i.e. an operator M such that $M^\dagger M = MM^\dagger$) acting over a vector space V is a *diagonal* matrix when expressed in its *eigenbasis* Λ. Moreover, the diagonal entries of this matrix are the *eigenvalues* for each corresponding eignvector of M. Since density operators *are instances* of normal operators, it follows from this theorem that every $\rho \in \mathcal{D}(\mathcal{H}_d)$ can be written as a linear combination of *at most* d pure (and orthogonal) states $\{|\gamma_i\rangle\}$ which happen to be the eigenvectors of ρ:

$$\rho = \sum_{i=1}^{d} \lambda_i |\gamma_i\rangle\langle\gamma_i| \tag{1.14}$$

Here, each λ_i is the eigenvalue associated with $|\gamma_i\rangle$ (i.e. $\rho|\gamma_i\rangle = \lambda_i|\gamma_i\rangle$). To avoid confusion with the other representations discussed so far, I will refer to equation 1.14 as the *canonical form* of the density matrix.

Measurements on mixed states

Let ρ be a mixed state in $\mathcal{D}(\mathcal{H})$, and let $\Pi = \{M_i\}$ be a PVM defined with respect to $\mathcal{H}$. The probability of measuring the outcome corresponding to the projection M_i is

$$p_i = \mathrm{Tr}(M_i\rho) \tag{1.15}$$

and the resulting state is

$$\rho_i = \frac{M_i\rho M_i^\dagger}{\mathrm{Tr}(M_i\rho)} \tag{1.16}$$

An important fact about mixed state measurement is that if Π_β is a PVM defined over the basis β, the probability of measuring each basis state $|\beta_i\rangle$ is given by $c_{i,i}$, which is the corresponding diagonal entry of $\rho_{[\beta]}$. Consequently, we note that the off-diagonal entries of $\rho_{[\beta]}$ *do not influence* the Π_β measurement, and so they are said to be *non-observables* for this PVM.

The dilation postulate

Earlier, we established that a density operator is a quantum state where there is some classical uncertainty about the underlying wave-function. Unlike the *genuine* non determinism attributed to a coherent superposition of states, this classical uncertainty indicates that we have imperfect information about our system. In other words, if we knew more about the state of our surroundings, we could (in theory) recover a complete description of the original system. Formally, the *dilation postulate* [4] proposes that every mixed state is *actually* a pure state in a larger (but inaccessible) Hilbert space. As an

[4]Note that *"dilation"* is a non-standard term for this concept. Historically, mapping mixed states to larger pure states was always known as *purification* [39] which unfortunately conflicts with the similarly named (unrelated) topic of *entanglement purification*. John Smolin has charmingly referred to the dilation premise as the "Church of the Higher Hilbert Space" [26], which (although amusing), I will avoid out of reverence for Holy Mother Church.

example, let $\rho \in \mathcal{D}(\mathcal{H}_d)$ be a mixed state which, when expressed in its canonical form (see eq. 1.14), is:

$$\rho = \sum_{i=1}^{d} \lambda_i |\gamma_i\rangle\langle\gamma_i| \tag{1.17}$$

To represent this mixture as a pure state, we suppose that each $|\gamma_i\rangle$ is associated with an environmental state $|e_i\rangle \in \mathcal{H}_{\text{env}}$. In other words, if it were possible to measure the environment in the state $|e_i\rangle$, we would know with certainty that the system of interest is in the state $|\gamma_i\rangle$. The pure state $|\Phi_\rho\rangle \in \mathcal{H} \otimes \mathcal{H}_{\text{env}}$ corresponding to the mixed state ρ is therefore

$$|\Phi_\rho\rangle := e^{i\phi} \sum_{i=1}^{d} \sqrt{\lambda_i}\, |\psi_i\rangle \otimes |e_i\rangle \tag{1.18}$$

Where $e^{i\phi}$ is some (irrelevant) global phase.

Transformations of quantum states

Transformations of quantum states are specified by *unitary operations* over the corresponding Hilbert space. An operator U is unitary if

$$U^\dagger U = U U^\dagger = I_{\mathcal{H}} \tag{1.19}$$

Where $U^\dagger$ is the conjugate transpose of U and $I_{\mathcal{H}}$ is the identity operator over the Hilbert space. For a pure state $|\psi\rangle$, the action of a unitary transformation is given by the matrix-vector multiplication:

$$|\psi'\rangle = U|\psi\rangle \tag{1.20}$$

Likewise, for a mixed state ρ, the action of a unitary U is given by the linear conjugation:

$$\rho' = U \rho\, U^\dagger \tag{1.21}$$

Geometrically, a unitary transformation on a pure state $|\psi\rangle$ corresponds to a rotation of $|\psi\rangle$ about some fixed axis of the Hilbert space.

1.3 Quantum circuits and basic gates

An unavoidable challenge (or opportunity depending on your perspective!) of quantum computing is that the Hilbert space of an n particle ensemble grows *exponentially* with respect to n. Consequently, we programmers are limited to the *tiny* subset of unitary transformations that can be efficiently implemented on $\mathcal{H}$. Even still, describing these operations in a comprehensible way remains an important design consideration due to the sheer size of the Hilbert space. Several approaches exist for representing such operations diagrammatically, but the most general (and popular) method is the *quantum circuit model*. This is a graphical representation of a unitary transformation that consists of *wires* and *gates*. Each wire stands in for a particle (which is always a qubit unless otherwise specified) and each gate is an operation over a subset of wires. Gates are executed in order from left to right, as indicated by the example presented in fig. 1.1. Some common gates are inherited from classical logic such as the negation X which sends $|0\rangle$ to $|1\rangle$ and vice versa. Other gates, like the Hadamard or Phase (not pictured) are more exotic and better exemplify the unusual ways in which quantum information may be processed. Some gates may be *parameterised*, which means they take variable arguments (an archetypal example is the *rotation gate* which rotates a state by θ degrees about a fixed axis). A *gateset* is a collection of distinct quantum gates, and a gateset is called *universal* if any unitary can (theoretically) be implemented as a circuit built exclusively using elements found in the gateset. Table 1.1 presents a selection of basic gates that will feature throughout this thesis.

1.3.1 The Pauli operators as a matrix basis

The single qubit *Pauli operators* $\mathcal{S}_1 = \{I, X, Y, Z\}$ (See table 1.1) are an important set of single qubit unitary operators with many interesting properties. Again, for the sake of brevity, it suffices to say that they form a basis for the 2×2 complex matrices. In other words, for all $M \in \mathbb{C}^{2\times 2}$, there exists a unique set of numbers $\alpha_I, \alpha_X, \alpha_Y, \alpha_Z \in \mathbb{C}$ such that

$$M := \alpha_I I + \alpha_X X + \alpha_Y Y + \alpha_Z Z \tag{1.22}$$

A *Pauli string* is a Kronecker product of single qubit Pauli operators. Let $\mathcal{S}_n$ denote

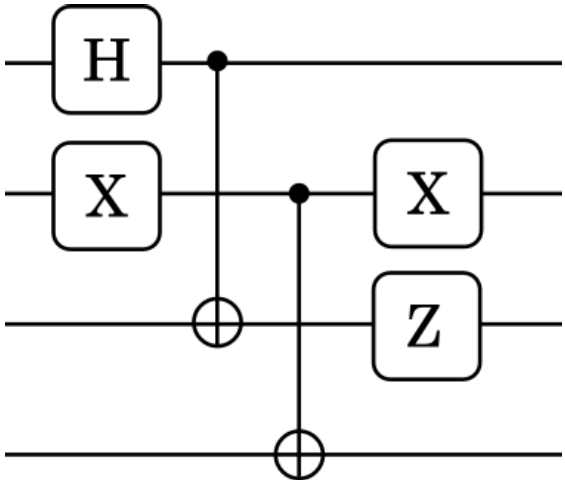

Figure 1.1: An arbitrary example of a quantum circuit. From left to right, the order of operations is $(H \otimes X \otimes I \otimes I), (CNOT_{13} \otimes CNOT_{24}), (I \otimes X \otimes Z \otimes Z)$

Gate	Matrix	Comments
X	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	The *bitflip operator* (Sends $\|0\rangle$ to $\|1\rangle$ and vice versa)
Z	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	The *phaseflip operator* (Sends $\|+\rangle$ to $\|-\rangle$ and vice versa)
Y	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	Equal to the product XZ up to a phase.
H	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	The *Hadamard operator*. Useful for putting states in an equal superposition or converting between the Z-basis $\{\|0\rangle, \|1\rangle\}$ and the X-basis $\{\|+\rangle, \|-\rangle\}$
$CNOT$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	The controlled not operation. For an input state $\|x\rangle \otimes \|y\rangle$, this operator does nothing if $\|x\rangle = \|0\rangle$ but implements a bitflip on the latter qubit if $\|x\rangle = 1$.

Table 1.1: A selection of some quantum gates that are commonly used throughout this thesis

the set of all Pauli strings of length n and observe that $|\mathcal{S}_n| = 4^n$. Since $\mathcal{S}_1$ forms a basis for $\mathbb{C}^{2\times 2}$., it follows that $\mathcal{S}_n$ forms a basis for the $\mathbb{C}^{2^n\times 2^n}$ matrices. This means that for every $M \in \mathbb{C}^{2^n\times 2^n}$, there exists complex values α_i such that

$$\mathcal{M} = \sum_{i=1}^{4^n} \alpha_i \tilde{\sigma}_i \tag{1.23}$$

Where $\tilde{\sigma}_i \in \mathcal{S}_n$.

1.4 Subsystems and Partial trace

Given a collection of particles that make up a state, it is often desirable to learn the state associated with a *subset* of those particles. In quantum communications for example, a multi-particle state may be held by several independent agents who want to characterise their subsystems in a way that is independent of what the other agents do.

Let $\rho_{A,B}$ be a multi-particle density operator supported by the Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B$ and suppose that Alice is a scientist who holds all the particles of $\rho_{A,B}$ corresponding to the subspace $\mathcal{H}_A$. The state $\rho_A \in \mathcal{D}(\mathcal{H}_A)$ is the state Alice *believes* she holds assuming she knows nothing of the other subsystem. The *unique* operation that calculates ρ_A from ρ_{AB} is called the *partial trace* with respect to B. If $\{|\beta_i\rangle\}$ is a basis of $\mathcal{H}_B$, then the partial trace is,

$$\mathrm{Tr}_B(\rho_{A,B}) := \sum_i \left(I_A \otimes \langle\beta_i|\right) \rho_{A,B} \left(I_A \otimes |\beta_i\rangle\right) \tag{1.24}$$

In other words, the partial trace Tr_B is the projection of ρ_{AB} onto the subspace $\mathcal{H}_A \subseteq \mathcal{H}_{AB}$.

1.5 Quantum channels

In an ideal computation, an isolated state evolves according to a pre-specified unitary operation. In reality however, the particles of that state also interact undesirably with the environment which introduces deviant effects called *noise*. Characterising and correcting noise is a top priority since it is impossible to perform meaningful calculations if there is too much noise present. Any process in which a state ρ interacts with the environment

is called a *quantum channel* $\mathcal{E}(\rho)$. [5]

Formally, a quantum channel $\mathcal{E}$ is a *completely positive* and *trace preserving* super-operator. [6] A *super-operator* is a linear operator that acts over other linear operators. The positivity and trace-preserving conditions are imposed to ensure that a quantum channel always maps density operators to density operators. A super-operator is *positive* if it maps positive elements (i.e. operators with positive eigenvalues) to other positive elements:

$$\mathcal{E}(\rho) \geq 0 \tag{1.25}$$

Complete positivity means this is also true for states with an ancillary system that is not acted on.

$$\mathcal{E}_A(\rho_{A,B}) \geq 0 \tag{1.26}$$

Where $\mathcal{E}_A$ is a channel supported by the Hilbert space $\mathcal{H}_A$ and $\rho_{A,B}$ is a state in $\mathcal{H}_A \otimes \mathcal{H}_B$. It is possible (and sometimes preferable) to define a quantum process $\tilde{\mathcal{E}}$ that does *not* preserve trace. Such a process may be interpreted as a quantum channel with some probability p of *losing the state*. This is especially useful for quantum processes in which a *measurement* takes place, in which case there some chance of the desired behavior going unobserved. As such, these types of lossy processes will also be referred to as quantum channels, even though they are not (strictly speaking).

1.5.1 The operator-sum representation

Suppose we have a Hilbert space $\mathcal{H}_s$ that represents our system of interest. Everything that is *not* a part of this system is called the *environment* and belongs to the much bigger Hilbert space $\mathcal{H}_e$. Our initial state is a density operator in $\mathcal{D}(\mathcal{H}_s \otimes \mathcal{H}_e)$. For convenience, we make the simplifying assumption that our system is completely isolated from the environment[7]. In the language of density operators, this means our initial state

[5] Although the name "channel" evokes the idea of putting a state in transit, it really refers to any kind of noisy process.

[6] Most standard texts on quantum information will refer to a quantum channel as a 'map,' which is perhaps a bit vague. I prefer the term *super-operator* (advanced by John Preskill [49] among others) since it better delineates the fact that a channel maps operators to operators.

[7] In general, this is not true. Interactions between the system and environment can cause the two parts to become *entangled* (See sec. 1.6). If $\rho_{s,e}$ is an state where the system and environment are initially entangled, then $\mathrm{Tr}_e(U\rho_{s,e}U^\dagger)$ can fail to be a completely positive (or even linear!) map [55].

is of the form:

$$\rho_s \otimes \rho_e \tag{1.27}$$

Where $\rho_s \in \mathcal{H}_s$ and $\rho_e \in \mathcal{H}_e$. By our *dilation postulate* however, we can further simplify by adding degrees of freedom to the environment in order to replace ρ_e with a pure state $|e_0\rangle$ in a larger space $\mathcal{H}'_e$.

$$\rho \otimes |e_0\rangle\langle e_0| \tag{1.28}$$

A noisy computation on ρ is therefore described by a unitary evolution on this system, followed by a partial trace over the environment.

$$\mathcal{E}(\rho) = \mathrm{Tr}_{\mathrm{env}}\Big(U_{s,\mathrm{env}}\left(\rho \otimes |e_0\rangle\langle e_0|\right) U^\dagger_{s,\mathrm{env}}\Big) \tag{1.29}$$

Although this definition is true in theory, there is no way of calculating $\mathcal{E}(\rho)$ from this description alone due to the sheer size of the environment. To simplify, we first recall the definition of the partial trace from equation 1.24 and substitute to obtain:

$$\mathcal{E}(\rho) = \sum_{i=1}^{d}\Big(I_s \otimes \langle e_i|\Big)\Big(U_{s,\mathrm{env}}\left(\rho \otimes |e_0\rangle\langle e_0|\right) U^\dagger_{s,\mathrm{env}}\Big)\Big(I_s \otimes |e_i\rangle\Big) \tag{1.30}$$

We now define the following *operation elements*

$$E_i := \Big(I_s \otimes \langle e_i|\Big)\, U_{s,\mathrm{env}}\, \Big(I_s \otimes |e_0\rangle\Big) \tag{1.31}$$

Where each E_i is an $n \times n$ complex valued operator. Finally, we see the action of a quantum operator $\mathcal{E}$ over a state ρ can be described as a *finite sum* of linear conjugations:

$$\mathcal{E}(\rho) = \sum_{i=1}^{d} E_i \rho E_i^\dagger \tag{1.32}$$

For the quantum channel to be *trace preserving*, we impose the constraint that these operation elements must satisfy the *completion relation*. This means that:

$$\sum_{k=1}^{d} E_k^\dagger E_k = I \tag{1.33}$$

Consequently, the initial state of the system must be seperated from the environment for a quantum channel to be well defined.

1.5.2 The quantum process matrix

In the previous section, we saw that a quantum channel acting over density operators in a d-dimensional Hilbert space could be described with up to d operation elements $\{E_i\}$, which are $d \times d$ complex-valued matrices. We can *further* simplify this description in the special case where $\mathcal{E}$ is a channel acting over an n-qubit ensemble, in which case $d = 2^n$. From section 1.3.1, we recall that the set of length-n Pauli strings $\mathcal{S}_n$ forms a basis for the $\mathbb{C}^{2^n \times 2^n}$ matrices. Consequently, each E_i can be written as a linear combination of the 4^n possible Pauli strings:

$$E_i = \sum_{i=1}^{4^n} \alpha_i \tilde{\sigma}_i \tag{1.34}$$

The overall multi-qubit channel can therefore be written in the form

$$\mathcal{E}(\rho) = \sum_{i,j=1}^{4^n} \chi_{i,j}\ \tilde{\sigma}_i\ \rho\ \tilde{\sigma}_j \tag{1.35}$$

The matrix $[\chi_{i,j}]$ is known as the *quantum process matrix*, and is common way to specify the action of a particular channel.

1.6 Entanglement

Entanglement is a fundamental phenomenon of quantum mechanics and remains an active topic of research even a century after its conceptualization [33]. A multi-particle pure state $|\psi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_n$ is said to be *entangled* if it cannot be written as a product of single particle states. Specifically:

$$|\psi\rangle \neq |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle \tag{1.36}$$

Where $|\psi_i\rangle \in \mathcal{H}_i$. If a state is not entangled, it is called *separable*. Similarly, a multi-particle mixed state ρ is entangled if it cannot be written as a linear combination of separable-state density operators. This definition, although sufficient, is admittedly somewhat thin. A more practical understanding is that a state is entangled if a measurement of one particle *can influence a subsequent measurement elsewhere* . A classic example of an entangled state is the two qubit *Bell pair*

$$|\phi^+\rangle = \frac{1}{\sqrt{2}}\big(|00\rangle + |11\rangle\big) \tag{1.37}$$

Here we see that by measuring the first qubit of $|\phi^+\rangle$ in the computational basis, the second qubit immediately collapses to the same state as the first (irrespective of what was measured). Moreover, I stress that correlated measurement outcomes do not *necessarily* indicate the existence of entanglement. As an example, consider the state $|00\rangle$. Although these particles have correlated outcomes when measured in the computational basis, the state is clearly separable. Consequently, it is evident that a measurement on one qubit (in any basis) does not affect a measurement on the other.

An important fact about entanglement is that it *cannot be remotely generated*; [8] If two parties Alice and Bob each have a collection of particles in their possession, there is no way for them to entangle any part of their ensembles if all they can perform are local operations and classical communications (*LOCC* [15]). The only way then to entangle particles is to either interact them directly, or to have them interact with other entangled particles.

1.6.1 Quantifying general entanglement

In the previous section, we stipulated that a state is entangled if a measurement on one particle influences a subsequent measurement elsewhere. Although every entangled state satisfies this property, a state may be *more* or *less* entangled depending on how significant this influence is. A key challenge then is finding a sensible way to *quantify* this interdependence and therefore quantify the *amount of entanglement*. Curiously, this turns out to be a hard problem in general. Part of the difficulty is that for systems of more than two particles, there are *non-equivalent types* of entanglement; Two categories of entangled states are said to be non-equivalent if there is no sequence of *stochastic local operations and classical communications* (SLOCC) [9] that maps a state from one category to a state in another. All two-qubit states are known to be equivalent SLOCC [43], while for *three* and *four* qubit states there are two [19] and nine [62] types of entanglement respectively. Stranger properties are known to emerge for still larger ensembles [37].

[8] Neither can the strength of a partially entangled state be increased *on average*, though surprisingly it is possible to *gamble* entanglement; Specifically, there are local strategies which give you the chance to gain more entanglement at the risk of losing what you started with.

[9] These are like LOCC operations, except the overall success probability is allowed to be less than 100%.

Based on this circumstantial evidence, it would seem that a complete characterisation of entanglement is intractable for anything beyond five or six particles.

Happily for us however, the entanglement use-cases considered in this thesis (see section 1.6.6) only require the use of *bipartite entanglement*, which are entangled states distributed between two parties. At face value, this may not seem any better since bipartite states *may also* consist of arbitrarily many particles. As we will see in the following sections however, there is a *unique* entanglement measure for *pure* bipartite states called the *entropy of entanglement* which describes the rate at which such states may be converted *to* and *from* maximally entangled pairs of qubits. This allows us to quantify the entanglement of *any* pure bipartite state in terms of a *standard entanglement resource* (though we will see this is somewhat more difficult for mixed states).

1.6.2 Quantifying pure bipartite entanglement

Let $\mathcal{H}_A \otimes \mathcal{H}_B$ be a joint Hilbert space between two parties Alice and Bob where $\dim(\mathcal{H}_A) = \dim(\mathcal{H}_B) = d$, and let $|\psi\rangle_{A,B} \in \mathcal{H}_A \otimes \mathcal{H}_B$ be an *arbitrary* pure state between them. By the Schmidt decomposition (See section 2.5 of Nielsen and Chuang, [46]), we can express $|\psi\rangle_{A,B}$ as

$$|\psi\rangle_{A,B} = \sum_{i=1}^{d} c_i \, |\alpha_i\rangle_A |\beta_i\rangle_B \tag{1.38}$$

Where $|\alpha_i\rangle$ and $|\beta_i\rangle$ are basis states of $\mathcal{H}_A$ and $\mathcal{H}_B$, and where $c_i \in \mathbb{R}$. Taking the partial trace of this state with respect to subsystem B gives us the mixture

$$\mathrm{Tr}_B\Big(|\psi\rangle\langle\psi|_{A,B}\Big) = \sum_i c_i^2 |\alpha_i\rangle\langle\alpha_i| \tag{1.39}$$

Now suppose we had traced out subsystem A *instead* of B. Interestingly, we obtain the *same* mixture of states (up to the relabeling $|\alpha_i\rangle \to |\beta_i\rangle$)

$$\mathrm{Tr}_A\Big(|\psi\rangle\langle\psi|_{AB}\Big) = \sum_i c_i^2 |\beta_i\rangle\langle\beta_i| \tag{1.40}$$

The key insight made by Popescu and Rohrlich [48] is that the *entropy* (or *spread*) of these $\{c_i^2\}$ probability terms can be used to quantify the amount of entanglement in a pure bipartite state; This metric is called the *entropy of entanglement* and is defined as:

$$E\Big(|\psi\rangle_{A,B}\Big) = -\sum_{i=1}^{d} c_i^2 \ln c_i^2 \tag{1.41}$$

In other words, the more entangled a pure state is, the more *mixed* its subsystem is. Equivalently, the entropy of entanglement is the *Von Neumann entropy* of the reduced density matrix $\rho_A := \mathrm{Tr}_A\Big(|\psi\rangle\langle\psi|_{A,B}\Big)$

$$E\Big(|\psi\rangle_{A,B}\Big) = -\mathrm{Tr}\Big(\rho_A \ln_{\mathrm{mat}} \rho_A\Big) \tag{1.42}$$

Where $\ln_{\mathrm{mat}}$ is the *matrix logarithm* in the natural basis. When ρ_A is expressed in its *cannonical form* (see eq. 1.14):

$$\rho_A = \sum_{i=1}^{d} \lambda_i |\gamma_i\rangle\langle\gamma_i| \tag{1.43}$$

The Von Neumann entropy simplifies to the form we saw in equation 1.41:

$$E\Big(|\psi\rangle_{A,B}\Big) = -\sum_{i=1}^{d} \lambda_i^2 \ln \lambda_i^2 \tag{1.44}$$

Importantly, the entropy of entanglement is the *unique* measure of entanglement for pure bipartite states [48] [10] . It is also the *asymptotic rate* at which maximally entangled two-qubit states (for example $|\phi^+\rangle$) can be *distilled* from many copies of the initial state [7]. In other words, if Alice and Bob have *infinitely many* copies of $\rho_{A,B}$ shared between themselves, there exists an $LOCC$ protocol they can perform which, on average, yields $E(\rho_{A,B})$ copies of $|\phi^+\rangle$ for every one copy of $\rho_{A,B}$ supplied.

The Bell basis

Due to the significance of the maximally entangled two qubit pairs as *units of bipartite entanglement*, I briefly pause this discourse to introduce the *Bell states*. These are a collection of four maximally entangled two qubit states that altogether form a basis for the two qubit Hilbert space $\mathcal{H}_4$.

[10] Note that other entanglement monotones *do exist* that are well-defined for two-qubit states and are *numerically distinct* from each other [65], [22]. However, they are equivalent in the sense that they preserve *partial orderings* on two-qubit states. For example let μ_1 and μ_2 be distinct entanglement measures and let ρ and σ be two qubit states. If $\mu_1(\rho) < \mu_1(\sigma)$ then it necessarily follows that $\mu_2(\rho) < \mu_2(\sigma)$.

$$|\phi^{\pm}\rangle = \tfrac{1}{\sqrt{2}}\Big(|00\rangle \pm |11\rangle\Big)$$
$$|\psi^{\pm}\rangle = \tfrac{1}{\sqrt{2}}\Big(|01\rangle \pm |10\rangle\Big) \tag{1.45}$$

1.6.3 Quantifying mixed bipartite entanglement

Entanglement of formation

Although the entropy of entanglement is a suitable measure for bipartite pure states, it fails to meaningfully quantify the amount of entanglement present in a *mixed state*. This is because classical uncertainty *also* contributes to a higher entropy in the reduced density operator. To illustrate, observe that the two states $|\phi^+\rangle\langle\phi^+|$ and $\frac{1}{2}\Big(|00\rangle\langle 00| + |11\rangle\langle 11|\Big)$ have the same *entropy of entanglement* despite the fact that the latter state is evidently *separable*.

One intuitive idea for measuring the entanglement of a mixed state is to sum up the entropy of entanglement for *each* pure state in the mixture (weighted by their respective probabilities). We recall from earlier however that the set of pure states making up a mixed state is *not unique*. Consequently, this "measure" of entanglement changes depending on how the state is represented. To solve this problem, let $\mathcal{P}$ be the set of *all* pure state ensembles that can be used to *represent* the mixed state $\rho_{A,B}$. For example, if $\{|\psi_i\rangle\} \in \mathcal{P}$, then there exists probabilities $\{p_i\}$ such that

$$\rho_{A,B} = \sum_i p_i |\psi_i\rangle\langle\psi_i| \tag{1.46}$$

The *entanglement of formation* (EoF) is defined as the *smallest possible* entropy of entanglement for a state $\rho_{A,B}$ [69]

$$E_f\Big(\rho_{A,B}\Big) := \inf_{\mathcal{P}} \Big\{ \sum_i p_i E\Big(|\psi_i\rangle_{A,B}\Big) \Big\} \tag{1.47}$$

This quantity is *closely related* to the *entanglement cost* [28], which is the minimum rate of maximally entangled two qubit pairs that are required (asymptotically) to create single instances of $\rho_{A,B}$.

$$E_c\Big(\rho_{A,B}\Big) := \lim_{n\to\infty} \frac{E_f\Big(\rho_{A,B}^{\otimes n}\Big)}{n} \tag{1.48}$$

The entanglement cost is easily seen to be *equal* to the EoF if $E_f(\rho_{A,B}^{\otimes n}) = nE_f(\rho_{A,B})$, which is a property called *additivity*. Presently, it is an *open question* whether or not all bipartite states are additive, though various categories of states are definitively known to satisfy this property [72].

An *explicit formula* for the EoF is known for two-qubit states [68], though more generally this is a difficult quantity to calculate exactly. Nevertheless, there is a *tight* lower bound that is efficiently computed [14]. Surprisingly, it is a hard problem to even determine whether a given $\rho_{A,B}$ is entangled *at all* [38] (Though there are sufficient criteria known for the qubit and *qutrit* cases [31]).

Entanglement of distillation

Another related quantity to the entanglement cost is the *entanglement of distillation* (EoD). While the entanglement cost is the number of pairs it takes to form a given $\rho_{A,B}$, the EoD indicates how many pairs can be asymptotically *distilled* from $\rho_{A,B}$. Interestingly, these two quantities are *not necessarily equal* for mixed states. There are two features that contribute to this gap. The first is the existence of so-called *bound states* [30], which are *non-separable* bipartite states that *cannot* be distilled to any number of maximally entangled pairs with an SLOCC protocol. The second factor is that the distillations of some states are asymptotically *irreversible* [63]. Amazingly, this is true even for states with *no bound entanglement* [64]. A semi-definite program by Rains [52] (later improved by Wang et. al. [66]) can bound the *attainable fidelity* of a given $\rho_{A,B}$ with respect to a *maximally entangled* bipartite state. In general however, it appears that *much less* is known about the EoD than the entanglement cost.

1.6.4 A primer on purification

An indispensable subroutine for distributed quantum computing and quantum networking is *entanglement purification*. Here, we imagine that maximally entangled pairs of qubits have been distributed across an *error channel* that causes the entanglement to partially *decohere*. A *purification protocol* [17] takes n of these imperfect pairs and uses a sequence of LOCC operations to non-deterministically refine these resources into $m < n$ maximally entangled pairs.

There is some good news and bad news when it comes to the study of purification.

The good news is that there are no two-qubit states that are *bound-entangled* [32]. Consequently, *all* two-qubit entangled states are distillable [32]. The *bad news* is that there are two-qubit states for which distillation is asymptotically irreversible [63]. This means that (even for this simple case of two-qubit entanglement) the EoD is not generally equal to the entanglement cost. Moreover, quantifying the EoD for an arbitrary mixed pair of qubits remains an open problem [8]. As purification is closely related to *quantum error correction*, I will return to this topic after introducing the requisite material.

1.6.5 State fidelity as a proxy for two qubit entanglement

In the previous section, we arrived at the (somewhat disappointing) conclusion that the EoD of a two-qubit mixed state is not generally known. We nevertheless require some means of gauging the *entanglement quality* of a two-qubit state. One common idea across the purification literature (see [9], [18], [17] for examples) is to use the *state fidelity* with respect to the maximally entangled pair $|\phi^+\rangle$. In general, the fidelity between two states ρ and σ is defined

$$F(\rho, \sigma) = \mathrm{Tr}\left(\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}\right)^2 \tag{1.49}$$

However since our comparison state $|\phi^+\rangle$ is pure, the fidelity simplifies to,

$$F\Big(\rho, |\phi^+\rangle\langle\phi^+|\Big) = \langle\phi^+|\rho|\phi^+\rangle \tag{1.50}$$

In this special case, we see that the fidelity is the *probability* that the state ρ *collapses* to the state $|\phi^+\rangle\langle\phi^+|$ when measured in the *Bell basis*.

1.6.6 Applications of entanglement

Besides its theoretical significance, entanglement also has various technological applications which I now briefly outline. Arguably the most significant of these use-cases is *quantum state teleportation*, where distributed entanglement is used as a resource to noiselessly transport quantum states [6] [47]. This is especially important for communicating large quantum states that would otherwise be impossible to specify within a classical message. Consequently, state teleportation enables *distributed quantum computing* which is when multiple quantum computers work together on the same computational

problem. Outside of computation, state-teleportation can also be used for *quantum cryptography*. In the well-established E91 protocol, distributed entanglement enables a qubit to be teleported which is then measured to establish a shared one-time pad between the entanglement holders [21]. I will revisit teleportation in greater detail during chapter 2 when it becomes relevant for my research on quantum networking.

The *Quantum graph states* are an important class of highly entangled states that are used as resources in *measurement-based quantum computing* (MBQC). Unlike in the circuit model, MBQC works by preparing a large graph state and measuring the qubits in various rotated bases to affect computations [12]. In *quantum metrology* [36], distributed entanglement can be used to capture images at higher resolutions [71] [70] and synchronise clocks more accurately [44] than would ordinarily be possible. Finally I note *superdense coding*, which is a communication protocol that utilizes distributed entanglement to effectively *double* the capacity of a classical communication channel. As interesting as this is however, it is unlikely to be of practical importance in the near term future due to the present difficulty of distributing entanglement.

1.7 Quantum error correction

In classical computer science, electrical components (or signals) encode data as a sequence of *bits*. Noise from the environment can cause these bits to randomly flip their values, but we can protect the information (at least to some extent) with an *encoding* that adds a certain amount of *redundancy*. A classic example is the *three-bit repetition code* which maps each '0' of a message to '000' and each '1' to '111'. In this way, if we see a string like '011', we have high confidence that the intended bit was '1' since it is less probable for two errors to have occurred than one. In general, the goal of error correction is to develop *error correcting codes* that *maximize* the number of errors the data can tolerate while simultaneously *minimizing* the amount of redundant information required. This same essential principle is also true for *Quantum Error Correction* (QEC).

In quantum computer science however, data is encoded as a vector in a Hilbert space and *quantum errors* are channels that both deviate the state from its intended orientation and also introduce classical uncertainty about the underlying state. Unlike with classical error correction, encoding a state into multiple redundant copies of itself is not possible

because of the '*no-cloning theorem*' which prohibits quantum operations for duplicating unknown quantum information [67]. This together with the limitless variety of possible errors and the fact that states cannot be directly observed makes QEC appear utterly impossible at first glance. [11] Luckily, this turns out not to be the case. At a bird's-eye-view, a state can be encoded into a highly entangled ensemble and then corrected by measuring a fraction of the particles. The measurements cause the encoded state to be projected onto a known subspace of the ensemble, which is the mechanism by which quantum errors are corrected. This process can be repeated *ad infinitum* by re-entangling and re-measuring the ensemble in a two-step cycle.

There are various types of quantum error correction, but the most common by far is the *stabilizer code*, which we treat in section 1.7.2. For introductory reviews on quantum error correction more broadly, see Devitt et. al. [16] and Roffe [54].

1.7.1 Digitization of quantum errors

A key feature of quantum information that greatly simplifies our understanding of QEC is that any kind of quantum error can be accounted for by only correcting a *finite* subset of errors [56]. This is known as the *digitization of quantum error*. To illustrate, suppose we have a single qubit pure state $|\psi\rangle$ that is subject to a *coherent error* U (In other words, no mixing with the environment takes place). Owing to the fact that the Pauli operators $\{I, X, Y, Z\}$ are a *basis* for the 2×2 matrices, the state $U|\psi\rangle$ can be written as a linear combination of the (non-orthogonal) states $\{|\psi\rangle, X|\psi\rangle, Y|\psi\rangle, Z|\psi\rangle\}$:

$$\begin{aligned} U|\psi\rangle &= \Big(\alpha_I I + \alpha_X X + \alpha_Z Z + \alpha_Y Y\Big)|\psi\rangle \\ &= \alpha_I|\psi\rangle + \alpha_X X|\psi\rangle + \alpha_Z Z|\psi\rangle + \alpha_Y Y|\psi\rangle \end{aligned} \tag{1.51}$$

Where $\alpha_I, \alpha_X, \alpha_Z, \alpha_Y \in \mathbb{C}$. We can further simplify by noting that $Y = iXZ$ and absorbing the complex phase into the α_Y amplitude to obtain:

$$U|\psi\rangle = \alpha_I|\psi\rangle + \alpha_X X|\psi\rangle + \alpha_Z Z|\psi\rangle + \alpha_{XZ} XZ|\psi\rangle \tag{1.52}$$

[11] As an historical aside, the apparent impossibility of QEC caused a good deal of justifiable skepticism about the utility of quantum computers when interest spiked after the discovery of Shor's algorithm. (See [60] and [41] for examples)

We can now interpret the corrupted state $U|\psi\rangle$ as a coherent superposition of the *intended state* $|\psi\rangle$ with three *error terms*. The X contribution is called a *bitflip error* while the Z contribution is called a *phaseflip error*. The (formerly) Y term is evidently seen to be a combination of *both* types of error.

This digitization principle also applies for *mixed states*. Suppose that our intended state $|\psi\rangle$ is scrambled to a mixed state ρ by a single qubit channel $\mathcal{E}$. From section 1.5.2, we recall that any such ρ can be written in the form

$$\rho = \sum_{i,j=1}^{4} \chi_{i,j}\, \sigma_i|\psi\rangle\langle\psi|\sigma_j \tag{1.53}$$

Where $\sigma_i \in \{I, X, Z, XZ\}$ and where $\chi_{i,j} \in \mathbb{C}$. Although it is evident that this noisy state ρ is a mixture of various X and Z error contributions, some care has to be taken when interpreting this decomposition since the off-diagonal terms (i.e. $X|\psi\rangle\langle\psi|Z$) *are not* quantum states [12] , and consequently cannot be observed [13] . To improve our understanding of eq. 1.53, we can separate the diagonal (observable) terms from the off-diagonal (non-observable) terms to obtain the expression:

$$\rho = \sum_{i}^{4} \chi_{i,i}\, \sigma_i|\psi\rangle\langle\psi|\sigma_i + (\text{non-observables}) \tag{1.54}$$

Now it is clear that ρ may be interpreted as a *probabilistic mixture* of the intended state $|\psi\rangle$ together with the error terms $X|\psi\rangle, Z|\psi\rangle, XZ|\psi\rangle$.

Finally, it is straightforward to see how the digitization of quantum errors extends to *multi-qubit systems* since $\mathcal{S}_n$ (the set of length n Pauli strings) forms a basis for the linear operators of $\mathcal{H}_2^{\otimes n}$ (see section 1.3.1). Consequently, any multi-qubit state $|\Psi\rangle \in \mathcal{H}_2^{\otimes n}$ can be written as a linear combination over *all possible* X and Z errors over the particle ensemble:

$$|\Psi\rangle = \sum_{\tilde{\sigma}\in\mathcal{S}_n} c_{\tilde{\sigma}}\, \tilde{\sigma}|\Psi\rangle \tag{1.55}$$

In summary, every quantum error over an ensemble of qubits can be described as a combination of single qubit X and Z errors. In theory then, we can protect a multi-

[12] This is because any operator $|\psi\rangle\langle\phi|$ where $|\psi\rangle \neq |\phi\rangle$ fails to be a density operator, as $\text{Tr}(|\psi\rangle\langle\phi|) \neq 1$

[13] Though the values of each $\chi_{i,j}$ can be inferred through *quantum state tomography*

qubit state $|\Psi\rangle$ from *any* quantum error provided we have a measurement device that distinguishes between every possible string of single qubit errors on $|\Psi\rangle$. Although such a POVM exists in principle, there is little practical sense in trying to implement it since we would have to know our target state $|\Psi\rangle$ *in advance*. For quantum error correction to be *useful*, we need to correct errors on *arbitrary unknown states*. In the next few sections, we demonstrate how the *quantum stabilizer codes* resolve this difficulty by encoding states as a superposition of so-called *stabilizer states* that are both easy to specify and correct.

1.7.2 The stabilizer formalism

Prior to presenting the stabilizer code, it is essential to review the fundamentals of the *stabilizer formalism*. A state $|\psi\rangle$ is said to be *stabilized* by an operator U if $|\psi\rangle$ is an eigenstate of U with a corresponding eigenvalue of one:

$$|\psi\rangle = U|\psi\rangle \tag{1.56}$$

A *Pauli string* $\tilde{\sigma}$ is a product of the single qubit Pauli gates $\{I, X, Y, Z\}$. An example of a three qubit Pauli string is $X \otimes I \otimes Z$. The n-qubit *Pauli group* $\mathcal{P}_n$ is the set of all n-qubit Pauli strings with phases $\{\pm 1, \pm i\}$ together with matrix multiplication as the group action. For example, the one-qubit Pauli group is

$$\mathcal{P}_1 = \{*, \pm I, \pm iI, \pm X, \pm iX, \pm Z, \pm iZ, \pm Y, \pm iZ\} \tag{1.57}$$

Let $\mathcal{S}_n$ be a subgroup of $\mathcal{P}_n$ and define $\mathcal{H}_\mathcal{S} \subseteq \mathcal{H}_n$ to be the Hilbert space *stabilized by* $\mathcal{S}_n$. This means there exists a basis β of $\mathcal{H}_\mathcal{S}$ such that every basis element $|\beta_i\rangle$ is stabilized by every element of $\mathcal{S}_n$. In other words:

$$|\beta_i\rangle \in \beta \;\Leftrightarrow\; \tilde{\sigma}|\beta\rangle = |\beta\rangle \qquad \forall \tilde{\sigma} \in \mathcal{S}_n \tag{1.58}$$

In practice, subgroups of $\mathcal{P}_n$ tend to be quite large. Fortunately, there is a concise way of representing Pauli subgroups using *group generators*. Let $G_\mathcal{S}$ be a subset of $\mathcal{S}_n$ and let $\langle G_\mathcal{S}\rangle$ denote the set generated by $G_\mathcal{S}$. This $G_\mathcal{S}$ is said to generate $\mathcal{S}_n$ if $\mathcal{S}_n = \langle G_\mathcal{S}\rangle$. In other words, $\mathcal{S}_n$ is generated by $G_\mathcal{S}$ if every element of $\mathcal{S}_n$ can be expressed as a product of elements from $G_\mathcal{S}$. A group generator is called *dependent* if there are *redundant* terms in the generating set. For example, if $G_{\mathcal{S}1} = \{\tilde{\sigma}_1, \tilde{\sigma}_2, \cdots, \tilde{\sigma}_n\}$

and $G_{\mathcal{S}_2} = \{\tilde{\sigma}_1, \tilde{\sigma}_2, \cdots, \tilde{\sigma}_n, \tilde{\sigma}_{n+1}\}$ are both generators that generate $\mathcal{S}_n$, it is clear the latter generator is dependent since the $\tilde{\sigma}_{n+1}$ term contributes nothing. Naturally, a generating set is called *independent* if it is not dependent.

Proposition. ***(Nielsen and Chuang, Prop 10.5 [46])***
If $G_{\mathcal{S}}$ is an independent generator of $\mathcal{S}_n$ with $n-k$ elements, the Hilbert space $\mathcal{H}_{\mathcal{S}}$ which is stabilized by $\mathcal{S}$ has a dimension of 2^k.

In other words, the dimension of $\mathcal{H}_{\mathcal{S}}$ is *halved* every time a new independent element is added to the generator $G_{\mathcal{S}}$. This halving can continue until there are n independent stabilizers, at which point $\mathcal{H}_{\mathcal{S}}$ contains just a single state. Any state in $\mathcal{H}_2^{\otimes n}$ that can be uniquely identified with a generating set of n independent Pauli strings is called a *stabilizer state*.

Introducing stabilizer checks

The elements of a group generator $G_{\mathcal{S}}$ are informally known as *stabilizer checks*. This is because Pauli strings, in addition to being linear operators, are also *observables* [14] Formally, an observable is a Hermitian operator M whose unit-length *eigenvectors* are the possible states that can be measured (with the appropriate PVM). The corresponding *eigenvalues* are taken to be *labels* that (potentially) distinguish the different measurement outcomes [15].

An important property of the Pauli stabilizer checks is that every eigenvalue of a Pauli string is either $+1$ or -1. When a stabilizer check is *enforced* (i.e. measured), the state is projected either onto the subspace spanned by the $+1$ eigenvectors or the -1 eigenvectors respectively. The practical meaning of these measurements will be clarified in the following subsections.

X and Z checks

As a trivial example of a stabilizer check, let us investigate the Pauli observables X and Z to see how they affect a single qubit state $|\psi\rangle$. From table 1.2, we see that all the operators each have a $+1$ and a -1 eigenvector that together span $\mathcal{H}_2$. When a Z check

[14] Which we recall are the measurable quantities of a state

[15] I say *potentially* because if two states have the same eigenvalue, there is no way to distinguish them using that observable.

Z	
Eigenvalue	Eigenvectors
$+1$	$\{\|0\rangle\}$
-1	$\{\|1\rangle\}$

X	
Eigenvalue	Eigenvectors
$+1$	$\{\|+\rangle\}$
-1	$\{\|-\rangle\}$

Table 1.2: The eigensystems of the Z, X and Y observables respectively

is performed on $|\psi\rangle$, the state either collapses to $|0\rangle$ or $|1\rangle$ depending on whether the measurement device registered a $+1$ or -1 outcome respectively. An X check does the same thing, but for the states $|+\rangle$ and $|-\rangle$.

In either case, the Z and X observables measure the *Boolean parity* of the state with respect to the $\{|0\rangle, |1\rangle\}$, and $\{|+\rangle, |-\rangle\}$ bases accordingly. When a $+1$ state is measured, we say the state has *even* parity with respect to the measurement basis whereas when -1 is measured we say the state has *odd* parity.

Special case: I

The single qubit identity operator I is a *special case* of observable since *every* $|\psi\rangle \in \mathcal{H}_2$ is stabilized by I. Consequently, when a stabilizer check of I is "performed," no information about the qubit is learned.

Multi-qubit stabilizer checks

A multi-qubit stabilizer check measures the *overall* eigen-parity of a state. Specifically, for a Pauli string $\tilde{\sigma} := \sigma_1 \otimes \sigma_2 \otimes \cdots \otimes \sigma_n$ where $\sigma_i \in \{I, X, Y, Z\}$, a measurement of the $\tilde{\sigma}$ observable tells you the value of the product:

$$\lambda_{\text{prod}} = \lambda_1(\sigma_1) \times \lambda_2(\sigma_2) \times \cdots \times \lambda_n(\sigma_n) \tag{1.59}$$

Where each $\lambda_i(\sigma_i) \in \{+1, -1\}$ is the eigenvalue of ith qubit with respect to the σ_i observable. As a quick example, suppose we implement the $X \otimes I \otimes Z$ check on an arbitrary three qubit state $|\psi\rangle$ and obtain a measurement outcome of -1. From this, we know that the single-qubit eigenvalues of our state must satisfy the equation:

$$-1 = \lambda_1(X) \times \lambda_2(I) \times \lambda_3(Z) \tag{1.60}$$

We can simplify by observing that $\lambda_2(I) = 1$ since *every* single qubit state is stabilized

by the identity operator.

$$-1 = \lambda_1(X) \times \lambda_3(Z) \tag{1.61}$$

With this information, we can easily deduce that there are two possible forms for the three qubit state $|\psi\rangle$ which are presented in the table below: Either the -1 eigenvalue comes from the first qubit in which case its state is $|-\rangle$ and the third qubit is $|0\rangle$, or else the -1 eigenvalue comes from the second qubit in which case the first and third qubits are $|+\rangle$ and $|1\rangle$ respectively.

A measurement of $X \otimes I \otimes Z$ yields -1:	
Possible Eigenvalues	Possible state
$-1, 1, 1$	$\|-\rangle \otimes \|?\rangle \otimes \|0\rangle$
$1, 1, -1$	$\|+\rangle \otimes \|?\rangle \otimes \|1\rangle$

Table 1.3: The possible eigenvalues and states when a -1 outcome is measured with the XIZ stabilizer check. Here, I use $|?\rangle$ to denote an unknown single qubit state.

Two-qubit example

ZZ	
Eigenvalue	Eigenvectors
$+1$	$\{\|00\rangle, \|11\rangle\}$
-1	$\{\|01\rangle, \|10\rangle\}$

XX	
Eigenvalue	Eigenvectors
$+1$	$\left\{\frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle), \frac{1}{\sqrt{2}}(\|01\rangle + \|10\rangle)\right\}$
-1	$\left\{\frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle), \frac{1}{\sqrt{2}}(\|01\rangle - \|10\rangle)\right\}$

Table 1.4: The eigensystems of the two qubit ZZ and XX observables.

Let us now consider a more sophisticated example where we apply the *two-qubit* stabilizer checks XX and ZZ to an arbitrary two-qubit state $|\psi\rangle \in \mathcal{H}_4$. As before, the eigensystems of these observables are presented in table 1.4. Suppose for the sake of argument that we measure the ZZ check first and observe a value of $+1$. Based on this information, we know that $|\psi\rangle$ has been projected onto the subspace of $\mathcal{H}_4$ spanned by $|00\rangle$ and $|11\rangle$. In other words, the resulting state is guaranteed to be of the form:

$$\alpha|00\rangle + \beta|11\rangle \tag{1.62}$$

We continue now with a measurement of the XX check. If we happen to measure an outcome of -1, we know that we've projected the state onto the space spanned by

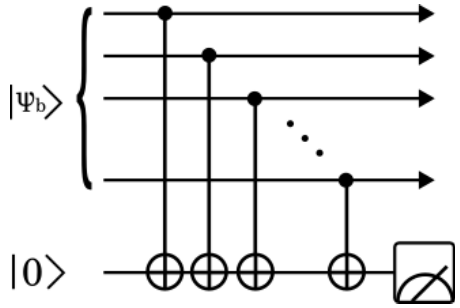

Figure 1.2: A quantum circuit that measures the parity of the encoded bit-string $|\psi_b\rangle = |b_1\rangle \otimes |b_2\rangle \otimes \cdots \otimes |b_n\rangle$. Here, an ancillary qubit is initialised to the $|0\rangle$ state and targeted n consecutive CNOTs that are controlled by each qubit of $|\psi_b\rangle$. These operations leave $|\psi_b\rangle$ unchanged, but increment the ancilla qubit by one (modulo 2) for each $|1\rangle$ state present in the bitstring.

$\frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$ and $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$. Ordinarily, this would suggest that our new state is of the form

$$\frac{1}{\sqrt{2}}\Big((\gamma(|00\rangle - |11\rangle) + \delta(|01\rangle - |10\rangle)\Big) \tag{1.63}$$

However, based on our prior information from eq. 1.62, we know that $\delta = 0$ since no part of our state includes $|01\rangle$ or $|10\rangle$ components. We therefore have sure knowledge that our state is now:

$$|\phi^-\rangle := \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \tag{1.64}$$

It is worth emphasising that this final state is *not* a stabilizer state since it isn't a simultaneous $+1$ eigenstate of the XX and ZZ stabilizers. It is however *closely related* to the stabilizer state $|\phi^+\rangle$, which is what we would have collapsed to if we had measured $+1$ on ZZ instead of -1. If our intention was to *create* $|\phi^+\rangle$ by enforcing its generating stabilizer checks, we can nevertheless do so by applying a *local correction* to $|\phi^-\rangle$. Specifically, we can apply a Z gate to either the first or second qubit of $|\phi^-\rangle$:

$$(I \otimes Z)|\phi^-\rangle \to |\phi^+\rangle \tag{1.65}$$

Stabilizer check circuits

So far, we have explored the theoretical aspects of the stabilizer formalism, but we have not yet addressed how stabilizer checks are implemented in practice. As a quick primer,

suppose we have a bitstring $(b_1, b_2, \cdots, b_n)$ where each $b_i \in \{0, 1\}$. The overall *parity* of the string is easily calculated by taking the bit-wise sum

$$b_1 \oplus b_2 \oplus \cdots \oplus b_n \tag{1.66}$$

The value of this sum is 0 if there are an even number of ones in the bitstring, and is 1 otherwise. Suppose now that the bitstring is encoded as an n-qubit state $|\psi_b\rangle$:

$$|\psi_b\rangle := |b_1\rangle \otimes |b_2\rangle \otimes \cdots \otimes |b_n\rangle \tag{1.67}$$

Where $|b_i\rangle = |0\rangle$ if $b_i = 0$ and $|b_i\rangle = |1\rangle$ otherwise. To implement the same bitwise sum as we did before, we can make use of the CNOT gate defined in section 1.3. It is easily verified that if $|x\rangle, |y\rangle \in \{|0\rangle, |1\rangle\}$, the effect of the CNOT on $|x\rangle \otimes |y\rangle$ is:

$$|x\rangle \otimes |y\rangle \xrightarrow{\text{CNOT}} |x\rangle \otimes |x \oplus y\rangle \tag{1.68}$$

In other words, the *target qubit* is replaced with the sum of x and y *modulo 2*. We can measure the parity of $|\psi_b\rangle$ using the circuit illustrated in fig. 1.2. Here, an *ancillary* qubit is targeted by n CNOT operations that are controlled by each of the n qubits respectively. From the initial state $|\psi_b\rangle \otimes |0\rangle_{\text{anc}}$, the final state (before measurement) is easily calculated to be:

$$|\psi_b\rangle \otimes |b_1 \oplus b_2 \oplus \cdots \oplus b_n\rangle_{\text{anc}} \tag{1.69}$$

Evidently, a measurement of the ancilla qubit in the computational basis will now reveal the parity of the encoded bitstring.

Stabilizer checks *generalise* the idea of bitwise parity by allowing individual qubits to be measured with respect to *different* Pauli observables. We recall from section 1.7.2 that a qubit has *even* parity with respect to a Pauli observable σ_i if it is a $+1$ eigenstate or else it has *odd* parity. From equation 1.59, the overall parity of a state with respect to a stabilizer $\tilde{\sigma}$ is the product of the single qubit eigenvalues

$$\lambda_1(\sigma_1) \times \lambda_2(\sigma_2) \times \cdots \times \lambda_n(\sigma_n) \tag{1.70}$$

In the previous example, we calculated the bitwise parity of a state by counting the number of $|1\rangle$ terms modulo 2. Since $|1\rangle$ is a -1 eigenstate of Z, we can equivalently say that the circuit in figure 1.2 is an implementation of the *all-Z* check:

$$Z_1 \otimes Z_2 \otimes \cdots \otimes Z_n \tag{1.71}$$

An *arbitrary* stabilizer check is performed in a similar way by incrementing the state of an ancillary qubit for each -1 eigenstate before measuring it in the computational basis.

We have seen how the CNOT can be used to implement bitwise addition in the Z basis. Let us now treat the problem of doing bitwise addition in the X basis. We recall that the $+1$ eigenstate of X is $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ while the -1 eigenstate is $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Let $y \in \{0, 1\}$ and observe that the $|+\rangle$ and $|-\rangle$ states can be parameterised in the following way:

$$\Big|f(y)\Big\rangle := \frac{1}{\sqrt{2}}\Big(|0\rangle + (-1)^y|1\rangle\Big) \tag{1.72}$$

Clearly $\Big|f(y)\Big\rangle = |+\rangle$ when $y = 0$ and $\Big|f(y)\Big\rangle = |-\rangle$ when $y = 1$. To do bitwise addition in the X basis, we require a two qubit operation that implements the following transformation:

$$\Big|f(y)\Big\rangle|x\rangle \rightarrow \Big|f(y)\Big\rangle|x \oplus y\rangle \tag{1.73}$$

Where $x \in \{0, 1\}$. In other words, we increment the ancilla state by one if $y = 1$. To accomplish this, we first recall that a Hadamard operation translates the Z basis states $\{|0\rangle, |1\rangle\}$ to the X basis states $\{|+\rangle, |-\rangle\}$ and vice versa. Consequently,

$$H|f(y)\rangle = |y\rangle \tag{1.74}$$

After this first Hadamard transformation, we can apply a CNOT to perform the bitwise addition in the computational basis

$$CNOT\Big(|y\rangle \otimes |x\rangle\Big) = |y\rangle \otimes |y \oplus x\rangle \tag{1.75}$$

We complete the addition by applying another Hadamard to rotate the first qubit back to its original orientation. The overall addition operation is therefore described by the operator:

$$(H \otimes I)\, CNOT\, (H \otimes I) \tag{1.76}$$

Constructing a single bit adder in the Y basis is trivial since a Y error can effectively be treated as a combination of X and Z. We demonstrate each of these three adders working together in figure 1.3 to implement the $Z \otimes X \otimes XZ$ stabilizer check.

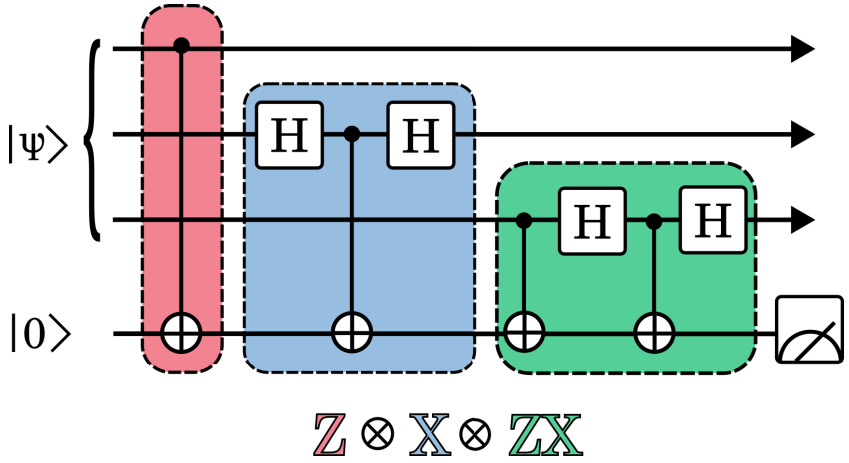

Figure 1.3: A circuit implementing an $Z \otimes X \otimes (ZX)$ stabilizer check. The bottom-most qubit of this circuit is an ancillary qubit that is first coupled with the three qubits of the state $|\psi\rangle$ and then measured to determine the overall parity. Each component of the stabilizer check can be thought of as a single bit adder in the X, Z and XZ bases respectively that increments the ancilla state if the corresponding qubit is a -1 eigenvalue of that basis.

1.7.3 Stabilizer codes

In the previous sections, we saw how an n qubit *stabilizer state* $|S\rangle$ could be constructed by enforcing the n stabilizer generators (checks) of its generating group. Suppose now that we enforce *all but one* of these checks. By the proposition made in section 1.7.2, the Hilbert space spanned by this reduced group has dimension two. Consequently, this new space $\mathcal{H}_L \subseteq \mathcal{H}_2^{\otimes n}$ is spanned by $|S\rangle$ together with an *orthogonal* stabilizer state $|S_\perp\rangle$. The *key insight* behind the stabilizer code is that these states can be put in coherent superposition to *encode a qubit of information*. Using the relabeling $|S\rangle \to |0\rangle_L$ and $|S_\perp\rangle \to |1\rangle_L$, it is clear that any state in $\mathcal{H}_L$ takes the form of a qubit:

$$\alpha|0\rangle_L + \beta|1\rangle_L \tag{1.77}$$

Since $|0\rangle_L$ and $|1\rangle_L$ are both stabilized by the same generators, this *logical* (short for *logically encoded*) qubit can be protected from noise by periodically enforcing the $n-1$ stabilizer checks. [16]

Ideally, each check is measured to be in its $+1$ eigenspace, which almost certainly guarantees that no error has occurred on the logical qubit. More likely however, noise on the encoded state will cause some checks to *erroneously* project the state onto their -1 eigenspace instead. When this happens, it is necessary to identify what kind of errors occurred and on which qubits so they can be later corrected. The problem of (quickly) determining the most likely configuration of errors is called *decoding* and is an active topic of research within quantum error correction. [29] [13] [45]

Characterising stabilizer codes

In a classical error correcting code, the bitstring that encodes a given character is called a *codeword*. The *robustness* of a code is determined by how *dissimilar* any pair of words are, since greater differences mean that more errors are needed for one word to be mistaken for another. The dissimilarity between codewords c_1 and c_2 is quantified by the *Hamming distance* $H(c_1, c_2)$, which is the minimum number of bits that need to be flipped to change c_1 into c_2 or vice-versa. If C is the set of all words in a code, then the *code distance* δ is defined to be the minimum distance between any two codewords in C.

$$\delta = \min_{c,c' \in C} H(c, c') \tag{1.78}$$

On the other hand, the words of a *stabilizer code* are *orthogonal stabilizer states* which are put in coherent superposition to encode another state. If $|s_1\rangle$ and $|s_2\rangle$ are codewords, then their dissimilarity is measured by the *quantum Hamming distance* $H_q(|s_1\rangle, |s_2\rangle)$. From section 1.7.1 we recall that quantum errors can be *digitized* as single qubit bitflips

[16] There's a good deal of nuance hiding in this statement. First, I note that this logical state is not *completely impervious* to noise; It is possible in theory for a *correlated error* to occur over the ensemble that modifies the values of α and β. We usually say this is highly unlikely to happen because we implicitly assume that our noise is *independently and identically distributed* over the qubits. Consequently, we suppose that *correlated behavior* (i.e. qubits spontaneously rotating in the same direction) is exponentially suppressed with the number of qubits in the state. This *might not* be the case in practice, so a knowledge of the underlying architecture is a necessity when designing QEC codes for deployment.

or phaseflips. The quantum Hamming distance is therefore the minimum number of *single qubit errors* [17] needed to convert $|s_1\rangle$ into $|s_2\rangle$ or vice versa. If $\langle G_{\mathcal{S}}\rangle$ is the set of stabilizer generators (i.e. codewords), the *code distance* d of a stabilizer code is the minimum quantum Hamming distance between any pair of generators in $\langle G_{\mathcal{S}}\rangle$

$$d = \min_{|s\rangle,|s'\rangle} H_q(|s\rangle, |s'\rangle) \tag{1.79}$$

The *overall performance* of a stabilizer code is characterised by three parameters that are conventionally labeled $[[n, k, d]]$. Here, n is the number of physical qubits that are required to implement the code, while k is the number of logical qubits that can be encoded, and d is the code distance defined by equation 1.79.

Code threshold

Theoretically, these $[[n, k, d]]$ parameters are all that are required to quantify the performance of a code relative to another. For implementing error codes in practice however, it is necessary to learn an additional quantity called the *error threshold*; This is the *maximum rate* of single qubit errors that the code can *exponentially suppress* as its distance is increased [1] [40]. To illustrate, imagine a code with n qubits that each have some probability p of suffering an error. The likelihood $\mathcal{P}_{\text{err}}$ of *at least one error occurring* is easily seen to be

$$\mathcal{P}_{\text{err}} = \sum_{k=1}^{n} \binom{n}{k} p^k (1-p)^{n-k} \tag{1.80}$$

A well-established fact is that a distance d code (either classical or quantum) can *correct* up to $c = (d-1)/2$ errors. Provided that p is below the threshold of the code, what error correction does to $\mathcal{P}_{\text{err}}$ is eliminate the *leading terms* of the polynomial up to and including p^c.

$$\mathcal{P}_{\text{err}} \xrightarrow[\text{QEC}]{} \sum_{k=\mathbf{c+1}}^{n} \binom{n}{k} p^k (1-p)^{n-k} \tag{1.81}$$

What this means is that every time the code distance d is increased, the polynomial $\mathcal{P}_{\text{err}}$ describing the overall error probability *loses its next lowest order term.* Hence, we

[17] i.e. X, Z, or XZ errors

say that errors are *exponentially suppressed* when the single qubit error rate p is below threshold.

This model however presumes that physical errors only occur *before* the error correction begins. In reality, physical errors are *more likely* to be introduced during correction since many imperfect gates and measurements are needed to implement the stabilizer checks. Typically then, the threshold of the code cannot be determined analytically, but must be estimated by *simulating* the device noise [27].

It is therefore difficult to anticipate whether or not a given code will have a high threshold. As a rule of thumb however, codes with higher thresholds usually have stabilizer checks that are *sparsely populated*, meaning there are relatively few X, Z and Y terms. This is because each *non-trivial* Pauli check requires a two qubit operation, which introduces more opportunities for errors to occur and propagate throughout the code.

A final detail worth pointing out is that the error threshold of a quantum code depends *significantly* on the underlying *bias* of the noise model, which is the ratio of X errors to Z errors. High bias is a desirable quality for qubit errors since this asymmetry can be exploited to correct more faults than would otherwise be possible [3] [58] [59]. As most theorists do not typically work with a *specific* quantum architecture in mind, the convention is to assume the worst-case scenario and work under the assumption that individual qubits suffer *unbiased noise*. Unless otherwise indicated, I will follow this same convention.

Quantum computation with stabilizer codes

While safeguarding quantum information is an important application in its own right, our main goal with stabilizer codes (and QEC more broadly) is to use encoded qubits for *quantum computation*. To achieve this, we require an implementation of a *universal gateset*. This means engineering circuits that act on the *physical qubits* of one or more codes in such a way as to implement a *universal set* of logical gates. Designing such circuits is a complex challenge, not only because the codes may contain a large number of physical qubits, but also because the circuits must operate below the *threshold of the code*. If a circuit is *long* or *highly connected*, it will take longer to implement since physical errors must be corrected more frequently than for a more *fault-tolerant* operation. Some circuits, like the *transveral operations* [53] are easy to implement since errors on

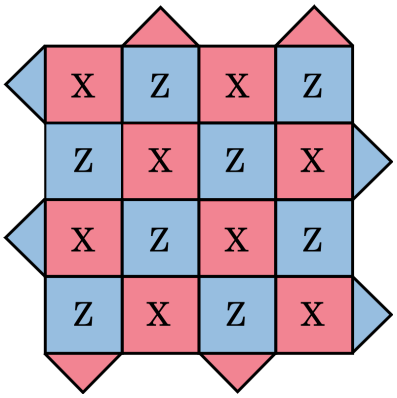

Figure 1.4: An illustration of the distance 5 *rotated surface code*. The vertices (line intersections) of this diagram represent *qubits* while the faces (closed shapes) represent *parity checks* on those qubits.

physical qubits remain local. Other operations, (especially the *non-Clifford* gates) are more complex and tend to require orders of magnitude more time (or resources [11]) to implement.

1.8 The surface code

The surface code [24] is a popular choice of stabilizer code that will feature extensively throughout this thesis. It is well known for its high error threshold of between 0.1% [23] and 1% for unbiased noise [57] and up to 43.7% for biased noise [59]. All these values fall within the range of experimental feasibility [50] [25]. In fact, a recent result has demonstrated two surface codes comprised of superconducting qubits that each operate below their respective error thresholds [2]. This groundbreaking research has shown, for the first time, a device capable of exponentially suppressing *arbitrary* quantum errors.

In addition to its high threshold, the surface code has a number of other attractive features. Efficient, universal quantum computation is known to be possible for surface code qubits using *only* nearest neighbor interactions [42]. This makes the surface code amenable to a *wide range* of quantum architectures. A fast *decoding algorithm* is also known for the surface code [29], which may otherwise be a rate limiting factor.

In its original implementation, the surface code is a $[[d^2 + (d-1)^2, 1, d]]$ stabilizer code. In this thesis however, every mention of the surface code will refer to a similar (but slightly more efficient) variant called the *rotated surface code* which has the parameters $[[d^2, 1, d]]$. A stabilizer diagram of the surface code is presented in figure 1.4.

In the long run, alternative encoding strategies might prevail over surface codes. The *Low Density Parity Check* (LDPC) codes for example are a topical category of stabilizer code today since the number of physical qubits scales *linearly* with respect to the code distance as opposed to the *quadratic* scaling for surface codes. Recently, Bravyi et. al. proposed an LDPC code with a *comparable* error threshold to the surface code [10]. Currently however, no universal gateset is known for LDPC codes.

1.8.1 Transversal operations on Surface codes

One of the necessary conditions for universality is the existence of an *entangling gate* which, for the sake of simplicity, we suppose is a two qubit gate like a CNOT. There are two main options for implementing entangling gates between surface codes [18] . A *transversal* two-qubit gate is performed by coupling every physical qubit in one code to a counterpart in an adjacent code. These transversal operations are challenging to implement in two-dimensional architectures due to the numerous *non-local* interactions required [61]. An easier alternative is *lattice surgery* which, unlike transversal gates, can be implemented with only nearest neighbor interactions [34]. In brief, lattice surgery is executed by performing the stabilizer checks over two code patches as if they were *one elongated patch*. The disadvantage of lattice surgery is that, unlike a transversal gate, it requires *measurements* on a subset of the physical qubits which can introduce new *undetected* errors into the ensemble. To correct for this, it is necessary to perform at least d rounds of lattice surgery to build confidence that the operation was done correctly [34]. This makes lattice surgery slow compared with the transversal option but is still the favored option on account of its easier implementation.

An important, though unrelated, fact is that transversal gates and lattice surgery can both be performed on surface codes that are not *directly adjacent* in space by using *shared entanglement*. Specifically, maximally entangled qubit pairs can be used to teleport the required two-qubit gates in either case. For surface codes of distance d, transversal gates require a total of d^2 pairs since each qubit in a $d \times d$ lattice must be matched up with its counterpart in the other lattice. Coincidentally, lattice surgery also requires d^2 pairs since d pairs are needed for each of the d rounds that are necessary to account for measurement errors. Although these two operations require the same total amount of entanglement,

[18] Other options like *braiding* exist for when qubits are encoded as surface code *defects* [35]. We do not consider anything like this.

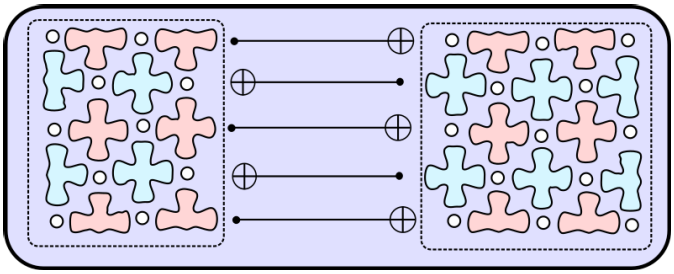

Figure 1.5: A cartoon of a lattice surgery operation. On the left and right are two surface code encoded qubits The floral pattern tiles depict stabilizer checks over the *white circles* which represent the *data qubits* encoding the logical qubits on either side. To implement a lattice surgery operation, d non-local CNOT operations are required between the two surface codes which can each be implemented with a maximally entangled two-qubit pair.

the *rate* of required entanglement is much smaller for lattice surgery since only d pairs are required at each time-step.

1.9 Purification in the context of error correction

1.9.1 Two-qubit errors and the Bell basis

We recall from section 1.7.1 that all quantum errors over an ensemble of qubits can be *digitized* as a combination X and Z errors. Naively then, we expect the need to accommodate for *15 possible error syndromes* when *purifying* entangled pairs:

$$\Big\{IX,\, IY,\, IZ,\, XI,\, XX,\, XY,\, \cdots,\, ZY,\, ZZ\Big\} \tag{1.82}$$

This however turns out to be redundant. By working in the *Bell basis* (see section 1.6.2), we can restrict our attention to the X and Z errors on *just one* qubit. To see this, let us recall that the Bell states are a collection of four maximally entangled pairs that together form a basis for the two qubit Hilbert space.

$$\begin{aligned} |\phi^{\pm}\rangle &= \tfrac{1}{\sqrt{2}}\Big(|00\rangle \pm |11\rangle\Big) \\ |\psi^{\pm}\rangle &= \tfrac{1}{\sqrt{2}}\Big(|01\rangle \pm |10\rangle\Big) \end{aligned} \tag{1.83}$$

Significantly, each Bell pair is related to $|\phi^{+}\rangle$ by *single qubit Pauli operations*.

$$\Big(I \otimes X\Big)|\phi^+\rangle = |\psi^+\rangle$$

$$\Big(I \otimes Z\Big)|\phi^+\rangle = |\phi^-\rangle \tag{1.84}$$

$$\Big(I \otimes XZ\Big)|\phi^+\rangle = |\psi^-\rangle$$

If $|\phi^+\rangle$ is our target state, this leads to the interpretation that $|\phi^-\rangle, |\psi^+\rangle$, and $|\psi^-\rangle$ are pairs that have suffered a *phase-flip* error, a *bit-flip* error or *both* respectively on one of their qubits. Note that because of the states' symmetry, it *does not matter* which qubit has suffered the error. For example:

$$\Big(I \otimes X\Big)|\phi^+\rangle = \Big(X \otimes I\Big)|\phi^+\rangle = |\psi^+\rangle \tag{1.85}$$

From this, we see that all two qubit errors $(\sigma_i \otimes \sigma_j)$ can be interpreted as a single qubit error $(I \otimes \sigma_k)$ up to some phase, where $\sigma_i, \sigma_j, \sigma_k \in \{X, Y, Z\}$.

1.9.2 Challenges of error correcting entangled states

A complication of correcting single qubit errors in the context of entanglement purification is that single qubit measurements *collapse* the entanglement that we would like to refine. Suppose for example we have the mixed state

$$\rho_{A,B} := 0.9\,|\phi^+\rangle\langle\phi^+|_{A,B} + 0.1\,|\psi^+\rangle\langle\psi^+|_{A,B} \tag{1.86}$$

Alice and Bob could easily *learn* what state they possess by performing computational measurements on both of their qubits. In this scenario they will observe either $|00\rangle$ or $|11\rangle$ ninety percent of the time, in which case the state *was* $|\phi^+\rangle$. Otherwise, they will observe $|01\rangle$ or $|10\rangle$ in which case their state had suffered an X error. This information is useless to them however since every state they project onto is *separable*!

As a naive solution to this problem, let us recall from example 1.7.2 that our target pair $|\phi^+\rangle$ is a *stabilizer state* that is enforced by the generating set $\langle XX, ZZ\rangle$. Any two qubit state $|\psi\rangle \in \mathcal{H}_4$ can therefore be mapped to $|\phi^+\rangle$ (up to local corrections) by enforcing the XX and ZZ parity checks. The issue with this approach however is that these two stabilizers act on *both* qubits of $|\psi\rangle$ and are therefore *non-local* operations.

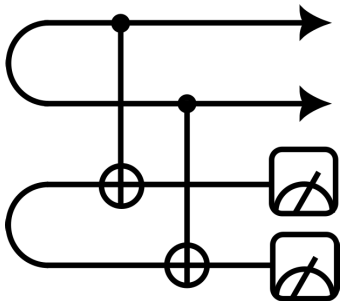

Figure 1.6: The *"parity check"* purification of Bennett et. al. [7]. The two semicircles at the start of the circuit denote the preparation of *partially entangled pairs* between the corresponding qubits they connect. Here we imagine that qubits 1 and 3 belong to one party (Alice) while qubits 2 and 4 belong to a distant party (Bob). The protocol consists of what Bennett et. al. describe as a *bilateral CNOT*. Here, Alice and Bob perform a CNOT between their respective qubits. Following this, Alice and Bob each measure the qubit they had targeted. The protocol *succeeds* if they measured either $|00\rangle$ or $|11\rangle$, else the purification is said to have *failed*.

As I will show in the following sections however, the XX and ZZ checks can each be enforced on a state $|\psi\rangle$ by *using up* a maximally entangled pair as a resource. Clearly then, it is not advantageous to distill $|\phi^+\rangle$ from $|\psi\rangle$ by enforcing $\langle XX, ZZ\rangle$, since this requires *two* entangled pairs but only yields *one*.

A better idea is to begin with an ensemble of partially entangled pairs $\rho_{A,B}^{\otimes n}$ and then perform a sequence of *non-local* stabilizer checks that (with high probability) correctly diagnose which pairs have suffered what kind of single qubit error. This is essentially *identical* to the objective of QEC where we wish to determine the most likely configuration of errors that have occurred over an ensemble of qubits. Bennett et. al. were the first to point out this equivalence between QEC and purification [9], though not in the language of stabilizers as I do here.

1.9.3 Example: Bennett purification protocol

The purification protocol that will feature the *most* throughout this thesis is the $2 \to 1$[19] algorithm by Bennett et. al. [7]. I refer to this as the *parity check protocol*, since its effect is to measure the ZZ observable on *one* entangled pair using *another* as a resource. The circuit for this purification is presented in figure 1.6. To begin, let us assume that distant parties Alice and Bob share two partially entangled pairs. We name one of these pairs

[19] i.e. where two pairs are used to produce one higher quality pair

the *control* and the other the *target*. Let us ignore Z errors for the time being. The objective of Alice and Bob in this experiment is to determine whether the control pair has suffered an X error. To this end, they implement what Bennett et. al. describe as a *bilateral controlled-not* (BCNOT) by each performing a CNOT between their halves of the control and target pairs respectively. Let p denote the probability of an X error occurring on a given pair.

The first possibility is that neither of the two pairs have suffered an X error. This occurs with probability $(1-p)^2$. In this scenario, the BCNOT does nothing to the input state, and the subsequent computational measurements will result in either $|00\rangle$ or $|11\rangle$. These *even* parity outcomes indicate that (very probably) no X error has taken place. The second possibility is that a single X error has occurred on the control or the target (which happens with probability $2p(1-p)$). In either scenario, the computational measurements will now have *odd* parity ($|01\rangle$, $|10\rangle$). This signals that an error has occurred, however as we do not know which pair contributed the error, we discard the output pair. The third and final possibility (which occurs with probability p^2) is that both pairs have suffered an X error. When this happens, the X error from the source propagates onto and therefore *eliminates* the error on the target. This causes the measurements to mistakenly report that *no error* has occurred.

Let us now ignore the X errors to discuss the effect of Z errors in this circuit. We first recall that a Z error $|\phi^+\rangle \rightarrow |\phi^-\rangle$ introduces a *relative phase* onto an entangled pair. Evidently, this type of error cannot be detected through computational measurements and therefore, the circuit of 1.6 does not identify any Z error. What is unfortunate however is that a Z error on the *target* pair *propagates* upward through the BCNOT and causes the control pair to suffer an *undetectable* Z error. In this way, we see that although this *parity check* protocol reduces the overall likelihood of an X error, the probability of a Z error increases.

Despite this trade-off, it is nevertheless possible to use *recursive* instances of this protocol to distill entangled pairs to an arbitrarily high fidelity (albeit with a *vanishing* yield [7] [18]) [20] . Nevertheless, the *short-depth* of this circuit together with its effectiveness at correcting *biased noise* [7] make it a natural first choice for entanglement purification on NISQ devices [51].

[20]The essential idea is to run multiple instances of the ZZ purification, and then perform XX parity checks (See fig. 1.7) on *those* surviving pairs, and so on repeatedly until a certain desired fidelity is met.

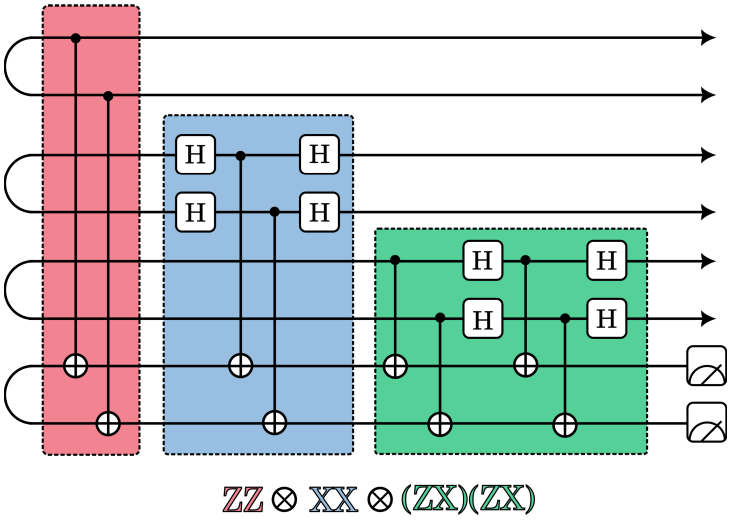

Figure 1.7: An entanglement purification circuit that *generalizes* the stabilizer check from figure 1.3. The semicircles at the start of the circuit indicate that the corresponding qubits are *partially entangled* pairs. The bottom-most pair of this circuit is an *ancillary pair* that is expended to measure the stabilizer. Observe that the ZZ check is identical to the bilateral CNOT of the parity check protocol described in figure 1.6. As with figure 1.3, each coloured component of this stabilizer check can thought of as a single bit adder, that increments the ancilla state if the pairs have suffered an X, Z or XZ error respectively.

1.9.4 Purification protocols as stabilizer codes

In the previous example, we showed that the ZZ parity check protocol of Bennett et. al. could be used to *detect* (but not correct) a single X error between two pairs. With a slight modification to this circuit (See figure 1.7), we could have *instead* performed an XX parity check to detect a single Z error among two pairs. By *combining* these basic circuits (in a similar way as we did for the stabilizers in figure 1.3), we can create *arbitrary* checks over an *ensemble* of distributed pairs. To illustrate, consider figure 1.7, which is a *purification circuit* [21] that implements the stabilizer check $ZZ \otimes XX \otimes (ZX)(ZX)$. Consequently, we see there is a *one-to-one* correspondence between stabilizer codes and purification protocols – an observation that was first made by Bennett et. al. [9] [22] .

[21] What makes this a *purification circuit* as opposed a QEC circuit more broadly is that we gain information about an ensemble of distributed pairs (without collapsing them) using *only* LOCC operations.

[22] Author's note: This insight is included only for completeness. It will not feature again except briefly in chapter two.

Bibliography

[1] Dorit Aharonov and Michael Ben-Or. Fault-tolerant quantum computation with constant error rate. *SIAM Journal on Computing*, 38(4):1207–1282, 2008.

[2] Google Quantum AI and Collaborators. Quantum error correction below the surface code threshold, 2024.

[3] Panos Aliferis and John Preskill. Fault-tolerant quantum computation against biased noise. *Phys. Rev. A*, 78:052331, Nov 2008.

[4] Sheldon Axler. *Linear Algebra Done Right*. Springer International Publishing, 2024.

[5] J. S. Bell. On the einstein podolsky rosen paradox. *Physics Physique Fizika*, 1:195–200, Nov 1964.

[6] Charles H. Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K. Wootters. Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. *Physical Review Letters*, 70:1895, 1993.

[7] Charles H. Bennett, Gilles Brassard, Sandu Popescu, Benjamin Schumacher, John A. Smolin, and William K. Wootters. Purification of noisy entanglement and faithful teleportation via noisy channels. *Phys. Rev. Lett.*, 76:722–725, Jan 1996.

[8] Charles H. Bennett, David P. DiVincenzo, John A. Smolin, and William K. Wootters. Mixed-state entanglement and quantum error correction. *Phys. Rev. A*, 54:3824–3851, Nov 1996.

[9] Charles H. Bennett, David P. DiVincenzo, John A. Smolin, and William K. Wootters. Mixed-state entanglement and quantum error correction. *Physical Review A*, 54(5):3824–3851, November 1996.

[10] Sergey Bravyi, Andrew W. Cross, Jay M. Gambetta, Dmitri Maslov, Patrick Rall, and Theodore J. Yoder. High-threshold and low-overhead fault-tolerant quantum memory. *Nature*, 627(8005):778–782, March 2024.

[11] Sergey Bravyi and Jeongwan Haah. Magic-state distillation with low overhead. *Phys. Rev. A*, 86:052329, Nov 2012.

[12] H. J. Briegel, D. E. Browne, W. Dür, R. Raussendorf, and M. Van den Nest. Measurement-based quantum computation. *Nature Physics*, 5(1):19–26, January 2009.

[13] Edward H. Chen, Theodore J. Yoder, Youngseok Kim, Neereja Sundaresan, Srikanth Srinivasan, Muyuan Li, Antonio D. Córcoles, Andrew W. Cross, and Maika Takita. Calibrated decoders for experimental quantum error correction. *Phys. Rev. Lett.*, 128:110504, Mar 2022.

[14] Kai Chen, Sergio Albeverio, and Shao-Ming Fei. Entanglement of formation of bipartite quantum states. *Phys. Rev. Lett.*, 95:210501, Nov 2005.

[15] Eric Chitambar, Debbie Leung, Laura Mančinska, Maris Ozols, and Andreas Winter. Everything you always wanted to know about locc (but were afraid to ask). *Communications in Mathematical Physics*, 328(1):303–326, March 2014.

[16] Simon J Devitt, William J Munro, and Kae Nemoto. Quantum error correction for beginners. *Reports on Progress in Physics*, 76(7):076001, June 2013.

[17] W Dur and H J Briegel. Entanglement purification and quantum error correction. *Reports on Progress in Physics*, 70(8):1381–1424, jul 2007.

[18] W. Dür, H.-J. Briegel, J. I. Cirac, and P. Zoller. Quantum repeaters based on entanglement purification. *Physical Review A*, 59:169, 1999.

[19] W. Dür, G. Vidal, and J. I. Cirac. Three qubits can be entangled in two inequivalent ways. *Phys. Rev. A*, 62:062314, Nov 2000.

[20] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 47:777–780, May 1935.

[21] Artur K. Ekert. Quantum cryptography based on bell's theorem. *Phys. Rev. Lett.*, 67:661–663, Aug 1991.

[22] Volkan Erol. Entanglement monotones and measures: an overview, 2017.

[23] Austin G. Fowler. Proof of finite surface code threshold for matching. *Phys. Rev. Lett.*, 109:180502, Nov 2012.

[24] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86(3), Sep 2012.

[25] J. P. Gaebler, A. M. Meier, T. R. Tan, R. Bowler, Y. Lin, D. Hanneke, J. D. Jost, J. P. Home, E. Knill, D. Leibfried, and D. J. Wineland. Randomized benchmarking of multiqubit gates. *Phys. Rev. Lett.*, 108:260503, Jun 2012.

[26] Daniel Gottesman and Hoi-Kwong Lo. From Quantum Cheating to Quantum Security. *Physics Today*, 53(11):22–27, 11 2000.

[27] Mauricio Gutiérrez and Kenneth R. Brown. Comparison of a quantum error-correction threshold for exact and approximate errors. *Phys. Rev. A*, 91:022335, Feb 2015.

[28] Patrick M Hayden, Michal Horodecki, and Barbara M Terhal. The asymptotic entanglement cost of preparing a quantum state. *Journal of Physics A: Mathematical and General*, 34(35):6891–6898, August 2001.

[29] Oscar Higgott and Craig Gidney. Sparse blossom: correcting a million errors per core second with minimum-weight matching, 2023.

[30] Michał Horodecki, Paweł Horodecki, and Ryszard Horodecki. Mixed-state entanglement and distillation: Is there a "bound" entanglement in nature? *Phys. Rev. Lett.*, 80:5239–5242, Jun 1998.

[31] Michał Horodecki, Paweł Horodecki, and Ryszard Horodecki. Separability of mixed states: necessary and sufficient conditions. *Physics Letters A*, 223(1):1–8, 1996.

[32] Pawel Horodecki and Ryszard Horodecki. Distillation and bound entanglement. *Quantum Information and Computation*, 1(1):45–75, July 2001.

[33] Ryszard Horodecki, Paweł Horodecki, Michał Horodecki, and Karol Horodecki. Quantum entanglement. *Rev. Mod. Phys.*, 81:865–942, Jun 2009.

[34] Clare Horsman, Austin G Fowler, Simon Devitt, and Rodney Van Meter. Surface code quantum computing by lattice surgery. *New Journal of Physics*, 14(12):123011, dec 2012.

[35] Fei Hua, Yanhao Chen, Yuwei Jin, Chi Zhang, Ari Hayes, Youtao Zhang, and Eddy Z. Zhang. Autobraid: A framework for enabling efficient surface code communication in quantum computing. In *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO '21, page 925–936, New York, NY, USA, 2021. Association for Computing Machinery.

[36] Zixin Huang, Chiara Macchiavello, and Lorenzo Maccone. Usefulness of entanglement-assisted quantum metrology. *Phys. Rev. A*, 94:012101, Jul 2016.

[37] Felix Huber, Otfried Gühne, and Jens Siewert. Absolutely maximally entangled states of seven qubits do not exist. *Phys. Rev. Lett.*, 118:200502, May 2017.

[38] Lawrence M. Ioannou. Computational complexity of the quantum separability problem, 2007.

[39] M. Kleinmann, H. Kampermann, T. Meyer, and D. Bruß. Physical purification of quantum states. *Physical Review A*, 73(6), June 2006.

[40] Emanuel Knill, Raymond Laflamme, and Wojciech H. Zurek. Resilient quantum computation. *Science*, 279(5349):342–345, 1998.

[41] Rolf Landauer. The physical nature of information. *Physics Letters A*, 217(4):188–193, 1996.

[42] Daniel Litinski. A game of surface codes: Large-scale quantum computing with lattice surgery. *Quantum*, 3:128, March 2019.

[43] Hoi-Kwong Lo and Sandu Popescu. Concentrating entanglement by local actions—beyond mean values, 1999.

[44] Benjamin K. Malia, Yunfan Wu, Julián Martínez-Rincón, and Mark A. Kasevich. Distributed quantum sensing with mode-entangled spin-squeezed atomic states. *Nature*, 612(7941):661–665, November 2022.

[45] Jarrod R. McClean, Zhang Jiang, Nicholas C. Rubin, Ryan Babbush, and Hartmut Neven. Decoding quantum errors with subspace expansions. *Nature Communications*, 11(1), January 2020.

[46] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[47] S. Pirandola, J. Eisert, C. Weedbrook, A. Furusawa, and S. L. Braunstein. Advances in quantum teleportation. *Nature Photonics*, 9(10):641–652, September 2015.

[48] Sandu Popescu and Daniel Rohrlich. Thermodynamics and the measure of entanglement. *Physical Review A*, 56(5):R3319–R3321, November 1997.

[49] John Preskill. Lecture notes for physics 229: Quantum information and computation. *N.A.*, 1998.

[50] John Preskill. Reliable quantum computers. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969):385–410, January 1998.

[51] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, August 2018.

[52] E.M. Rains. A semidefinite program for distillable entanglement. *IEEE Transactions on Information Theory*, 47(7):2921–2933, 2001.

[53] Robert Raussendorf. Key ideas in quantum error correction. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370(1975):4541–4565, September 2012.

[54] Joschka Roffe. Quantum error correction: an introductory guide. *Contemporary Physics*, 60(3):226–245, 2019.

[55] David Schmid, Katja Ried, and Robert W. Spekkens. Why initial system-environment correlations do not imply the failure of complete positivity: A causal perspective. *Phys. Rev. A*, 100:022112, Aug 2019.

[56] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, Oct 1995.

[57] Ashley M. Stephens. Fault-tolerant thresholds for quantum error correction with the surface code. *Phys. Rev. A*, 89:022321, Feb 2014.

[58] Ashley M. Stephens, William J. Munro, and Kae Nemoto. High-threshold topological quantum error correction against biased noise. *Phys. Rev. A*, 88:060301, Dec 2013.

[59] David K. Tuckett, Stephen D. Bartlett, and Steven T. Flammia. Ultrahigh error threshold for surface codes with biased noise. *Phys. Rev. Lett.*, 120:050505, Jan 2018.

[60] W. G. Unruh. Maintaining coherence in quantum computers. *Phys. Rev. A*, 51:992–997, Feb 1995.

[61] Michael Vasmer and Dan E. Browne. Three-dimensional surface codes: Transversal gates and fault-tolerant architectures. *Phys. Rev. A*, 100:012312, Jul 2019.

[62] F. Verstraete, J. Dehaene, B. De Moor, and H. Verschelde. Four qubits can be entangled in nine different ways. *Phys. Rev. A*, 65:052112, Apr 2002.

[63] G. Vidal and J. I. Cirac. Irreversibility in asymptotic manipulations of entanglement. *Phys. Rev. Lett.*, 86:5803–5806, Jun 2001.

[64] G. Vidal, W. Dür, and J. I. Cirac. Entanglement cost of bipartite mixed states. *Phys. Rev. Lett.*, 89:027901, Jun 2002.

[65] Guifré Vidal. Entanglement monotones. *Journal of Modern Optics*, 47(2–3):355–376, February 2000.

[66] Xin Wang and Runyao Duan. Improved semidefinite programming upper bound on distillable entanglement. *Physical Review A*, 94(5), November 2016.

[67] W. K. Wootters and W. H. Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, October 1982.

[68] William K. Wootters. Entanglement of formation of an arbitrary state of two qubits. *Physical Review Letters*, 80(10):2245–2248, March 1998.

[69] William K. Wootters. Entanglement of formation and concurrence. *Quantum Info. Comput.*, 1(1):27–44, jan 2001.

[70] Ugo Zanforlin, Cosmo Lupo, Peter W. R. Connolly, Pieter Kok, Gerald S. Buller, and Zixin Huang. Optical quantum super-resolution imaging and hypothesis testing. *Nature Communications*, 13(1), September 2022.

[71] Zheshen Zhang and Quntao Zhuang. Distributed quantum sensing. *Quantum Science and Technology*, 6(4):043001, July 2021.

[72] Li-Jun Zhao and Lin Chen. Additivity of entanglement of formation via an entanglement-breaking space. *Phys. Rev. A*, 99:032310, Mar 2019.

Chapter 2

Cost-vector analysis and multi-path routing in quantum networks

> After three days they found him in the temple, sitting among the teachers, listening to them and asking them questions
>
> Luke 2:46

2.1 Statement of work

In this chapter, I introduce a framework for analysing quantum routing protocols called the *cost-vector formalism*. Here, quantum networks are recast as multi-graphs where edges represent two-qubit entanglement resources that *could* exist under some sequence of operations. Each edge is weighted with a *transmission probability* that represents the likelihood of the pair existing and a *coherence probability* which is the likelihood the pair is suitable for teleportation. Quantum routing operations such as entanglement swapping and purification are then interpreted as *contractions on the multi-graph* with relatively simple rules for updating the edge-weights. The cost-vector formalism was initially conceived by Dr. Peter Rohde as a path-finding tool for quantum networks. Dr. Rohde and I made roughly equal contributions in the material from sections 2.2 to 2.4 inclusive, though the writing in this chapter is entirely mine. I introduced the technique of *entanglement stacking* in section 2.4.4. In early 2021, Dr. Rhode had the

idea to extend the cost-vector formalism over time and proposed the *temporal meta-graph*. This is a multi-graph that represents the entanglement resources present in the network at various times *together* with the available memory resources. In order to make this theory compatible with our interpretation that every link of the multi-graph is a potential entanglement resource, I developed a resource theory for quantum memories, which is presented in section 2.5.2. Dr. Rohde proposed the concept of the *asynchronous node* discussed in section 2.5.6 as a tool to facilitate routing algorithms in temporal meta-graphs. The routing algorithms developed in section 2.6 and simulation work conducted in section 2.7 is original, though Dr. Nathan Langford contributed significant feedback in the methodology and subsequent write-up.

2.2 Introduction

In classical networking, communication is carried out by transmitting bits of information encoded in an optical signal. Repeater stations amplify the signal at regular intervals which enables it to travel much further than it would be able to otherwise. Compare this with quantum networking, where the objective is to communicate quantum states that are typically encoded with one or more photons. General repeaters which "amplify" incoming states by duplicating them are impossible in quantum networking, since arbitrary copying of unknown quantum information is prohibited by the *no-cloning theorem*. This issue can be circumvented by instead distributing known entangled states between parties. Entangled states enable quantum information to be transmitted directly via *quantum state teleportation* [3] and, unlike the sensitive quantum data in a message, are fungible resources that can be refined with *entanglement purification* [12] and extended over longer distances with *entanglement swapping* [19]. These two actions (together with entanglement distribution), are the fundamental building blocks of *quantum communication protocols*, which are strategies for efficiently establishing entanglement links of arbitrary quality between parties in a quantum network.

A challenge of studying quantum communication protocols however is that they are *non-deterministic* and tend to have *many points of failure*. At any stage during routing, a key entanglement link could suddenly vanish and dramatically alter subsequent routing decisions. Naturally, the most effective way to benchmark quantum communication

protocols is to simulate them directly [8] [9] [18]. Although this approach is accurate, it is likely to be computationally expensive for large networks with many competing end-users. In this work, we propose a *static* framework for analysing quantum communication protocols that we call the *cost-vector formalism*. Our main insight is that quantum networks can be recast as multi-graphs where each link represents a partially entangled two-qubit state that *could exist* under some sequence of operations. Routing strategies in this framework are described as a sequence of contractions on the multi-graph that correspond to entanglement swapping and purification operations.

In the remainder of this section, we define quantum networking from first principles and introduce the problem of *path-finding* in quantum networks. Given the hardness of this subroutine, we identify two broad categories of path-finding heuristics. In section 2.3 we demonstrate how a simplified noise model allows us to find optimal paths in terms of a quantity called the *coherence probability*, which is the likelihood that a given pair is suitable for teleportation. Following this in section 2.4, we formally introduce our cost-vector formalism. Each link of our network is weighted with two costs: a coherence probability p and a *transmission probability* (or *efficiency*) η which is the likelihood of the pair existing. We demonstrate how entanglement swapping and purification operations can be related to vertex and edge contractions of the multi-graph respectively (See Fig. 2.1). Consequently, we derive rules for updating the cost-vectors in either scenario. In section 2.5 we extend our cost-vector formalism to include the use of quantum memories which enables us to consider entanglement routing over space and time. With the cost-vector formalism fully introduced, the latter part of the paper focuses on demonstrating some of its capabilities. We begin in section 2.6 by introducing some rudimentary multi-path routing algorithms, then we spend section 2.7 testing them in a variety of networking scenarios.

2.2.1 State teleportation and quantum networks

State teleportation is a well-established quantum communications protocol where one party transmits an arbitrary single qubit to another using a shared maximally entangled pair and a classical side channel. When the qubit being teleported is not a separable state, but is entangled with another qubit held by a third party, the entanglement is extended between the outer two qubits. This variant of state teleportation is called

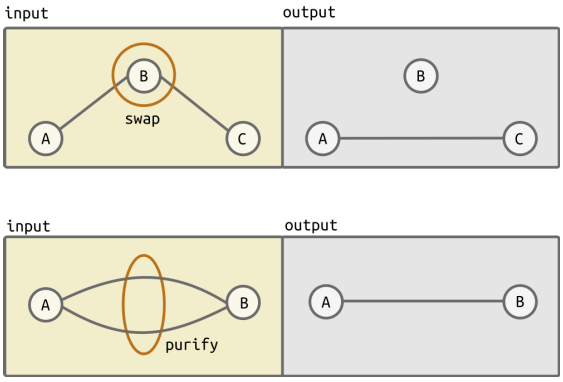

Figure 2.1: In the cost-vector formalism, the state of a network is represented with a multi-graph where edges are partially entangled two-qubit pairs that *could exist*. Top: Entanglement swapping between between edges (A, B) and (B, C) results in a link at (A, C). Bottom: A purification protocol reduces two edges in parallel to produce a *"higher quality"* link.

entanglement swapping [20]. Although there are many types of entanglement, the scope of quantum networking is typically limited to the distribution of *two-qubit entanglement* to enable single qubit teleportation between end-users. No utility is lost in considering this restricted framework since arbitrarily large quantum states may be teleported across the network provided there are sufficiently many distributed entangled pairs.

Formally, a *quantum network* is a collection of quantum computers that are interconnected according to a graph of single qubit channels. These channels are completely positive maps on single qubit density operators that represent the error processes that occur as individual qubits are transmitted between computers, and they are ideally as close to the identity channel as possible. Equivalently, each link of the network may be represented as a completely positive and *trace-preserving* single qubit channel together with a *transmission probability*. Every computer in the network has the ability to generate entangled states, perform arbitrary local operations, store qubits in memory, measure states, and transmit qubits to adjacent quantum computers using the channels connecting them. We also assume that each computer is able to send messages to eachother with a *classical side channel*. Since there are infinitely many maximally entangled two qubit states [1] , we assume by convention that the *target state* which the network aims to deliver is the maximally entangled pair $|\phi^+\rangle := \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Without loss of generality then, the objective of a quantum network is to maximize the throughput

[1] To see this, note that $\Big(R_x(\theta) \otimes I\Big)|\phi^+\rangle$ is a maximally entangled state for every $\theta \in \mathcal{R}$.

of $|\phi^+\rangle$ states distributed between end-users. Recall from section 1.6.2 that $|\phi^+\rangle$ is one of the four *Bell pairs*, which together form a basis for the the two qubit Hilbert space. Additionally, I note from section 1.9.1 that $|\phi^-\rangle$, $|\psi^+\rangle$ and $|\psi^-\rangle$ may be interpreted as $|\phi^+\rangle$ pairs that have suffered a phase-flip error, a bit-flip error or both respectively on one of their qubits

Fidelity as a threshold for error tolerance

In practice, distributing pure $|\phi^+\rangle$ states between end-users is impossible, not only because of the noisy channels connecting them and the asymptotic nature of practical purification protocols, but also because of the device level noise that caps the accuracy states can be purified to. We must therefore refine our objective by insisting that distributed states are close approximations of $|\phi^+\rangle$. Formalising the notion of "approximate" requires the specification of a *distance measure* between states [13], the most common choice being the *state fidelity* with respect to $|\phi^+\rangle$ (see section 1.6.5) . Notably, this pair fidelity can also be related to the *maximum teleportation fidelity* $F_{\max}$, which is the highest attainable qubit fidelity when using the pair ρ for state teleportation [16].

$$F_{\max} = \frac{2F + 1}{3} \tag{2.1}$$

We now amend our definition of the quantum network by clarifying that it must deliver states ρ_{ϕ^+} that are ϵ-*close* with respect to $|\phi^+\rangle$ in fidelity, which is to say

$$1 - F\Big(\rho_{\phi^+}, |\phi^+\rangle\langle\phi^+|\Big) \leq \epsilon \qquad 0 \leq \epsilon \ll 1 \tag{2.2}$$

Swapping vs. distribution

One subtlety of quantum networking that must be addressed is that there are *two ways* of propagating entanglement through the network. Entanglement may be either *distributed* between nodes by forwarding entangled qubits to neighboring computers, or it may be extended by performing *entanglement swapping* on two "adjacent" pairs. As a rule, it is never beneficial to distribute entanglement beyond one's nearest neighbor since repeated transmissions suffer more loss than swapping along the path.

We can demonstrate this claim with a simple thought experiment: Suppose that two parties Alice and Bob want to share entangled pairs but are separated by n consecutive

channels each with transmission probability p_t. They have the option of either transmitting their pairs directly or coordinating the network to distribute and swap entanglement at each step. Let us assume that each swapping operation succeeds with probability p_s. If Alice and Bob attempt direct transmission, their success probability is p_t^n. On the other hand if they distribute and swap, the probability is $n\, p_t\, p_s^{(n-1)}$. If we make the reasonable assumption that p_s is higher than p_t, we see that $n\, p_t\, p_s^{(n-1)} \gg p_t^n$ holds for all $n \geq 2$. Consequently, it is never advantageous to distribute entanglement beyond one's nearest neighbor.

2.2.2 Methods for entanglement distribution

Presently, there are two feasible strategies for distributing entanglement (both of which use entangled photons) between adjacent nodes in a quantum network. The more conventional approach is to use a fiber-optic cable for transmission, however the high attenuation of these cables limits the feasible *'single-shot'* distance to around 100 kilometers [21]. The other approach is to use one or more quantum satellites for distributing entanglement [24]. In this scenario, entangled pairs of photons are generated in space and fired to distant ground stations. This method allows for entanglement to be distributed more reliably at longer distances, however (as we will discuss in Chapter 3) the limited power available on a satellite limits long-term scalability. A hypothetical third option is to use a so-called *quantum sneakernet* where entanglement is loaded onto error-correcting codes that are then physically (and losslessly!) transported to their respective destinations [11]. Although the transmission time of a sneakernet is considerably slower than the speed of light, the immense bandwidth of *classical* sneakernets [15] suggests that a quantum sneakernet has the potential to vastly exceed the entanglement throughput of both repeater and satellite networks. For the time being though, we will limit our attention to photonic-based entanglement transport.

2.2.3 Routing versus path-finding

In any networking scenario, we are frequently interested in *optimal path finding* which, in our context, means finding the sequence of channels that maximizes the rate at which ρ_{ϕ^+} states are delivered between end-users. Here, I preface by drawing a distinction between *path finding* and *entanglement routing*; We take the latter term to mean finding

the optimal distribution strategy given a path of channels between end-users. In general, the throughput of this *linear network* depends both on the order of entanglement swapping [7] and the order of purification operations [14]. Although there are exponentially many distribution strategies for a linear network, heuristic dynamic programs exist that efficiently find protocols with *near optimal* throughput [17] [14].

This result simplifies matters, though we still need a way of determining *which* path through the network is optimal; This is the problem of quantum path-finding. For networks with *lossy* channels, quantum path finding is a *multi-objective optimization*, as we need to *simultaneously* balance between *transmission probability* and the probability of *single-qubit errors*. As multi-objective optimization is a hard problem in general [25], we would like to avoid it wherever possible.

Let us ignore channel loss for the time being to consider a simpler question: For a quantum network of *trace-preserving* channels, does a *single-objective* optimization exist to find the optimal path between a pair of end-users? Specifically, we want to know if there is an additive scalar cost that can be calculated from each channel such that a *path-finding* algorithm over those weights will find the path with the highest yield. Surprisingly, the answer to this question turns out to be *yes* or *no* depending on how we perform entanglement purification.

To qualify this statement, I introduce *two paradigms* for purification strategies that are based on opposite-extreme approaches. In the *link-level paradigm*, all entanglement is purified to ρ_{ϕ^+} before any swapping takes place. In the *end-level paradigm*, all *unprocessed* pairs are first delivered to the end-users before they are purified.

In the link-level paradigm, there is indeed a simple path-finding strategy. Let the scalar cost associated with each channel be the *average number of channel calls* needed to produce an ideal pair over that link. This is closely related to a quantity called the *channel capacity* [10] [23] and is calculated using techniques from *channel coding theory* [22]. Finding the optimal path with respect to this cost is therefore equivalent to finding the path with the *fewest number* of total expected calls required to establish end-to-end entanglement. Although path-finding in this paradigm is easy, a disadvantage is that the routing algorithms are comparatively *slow*. This is because entanglement swapping can *only* be performed after high-quality links are established at every step of the path. Networking in this regime may therefore prove challenging due to the resulting dead-time.

On the other hand, we expect that the latency is *minimized* in the end-level paradigm, since the only precondition for swapping is whether or not the raw resources exist between links. Unfortunately, an additive scalar cost is not possible within this picture. This is demonstrated by Di Franco et. al. who show that optimal path finding with respect to *entanglement negativity* could (in the special case where each channel is diagonal [1]) be re-framed as an instance of a multi-objective optimization. Essentially, this stems from a need to balance the minimization of X and Z errors respectively.

2.3 Optimal path-finding with coherence probability

In the previous section, we made the argument that optimal path-finding in the end-level paradigm is a hard problem, even in the simplified case of diagonal-noise where each channel has some probability of applying any of the three single qubit Pauli errors. Here, we show how further constraining our noise model to *partially-depolarizing* or *partially-dephasing* error channels allows us to develop a path-finding heuristic over a single cost. We begin this section with some definitions before moving onto our main observation that entanglement swapping preserves the structure of partially depolarized and partially dephased states. We conclude by demonstrating how this property can be used to specify a single cost heuristic for these noise models and showing how this cost can be related to the fidelity with respect to $|\phi^+\rangle$.

2.3.1 Definitions

Definition 1. *A single qubit* ***partially depolarizing channel***[2] $\mathcal{E}_d(p, \hat{\rho})$ *is a quantum channel with the action*

$$\mathcal{E}_d(p, \hat{\rho}) = p\hat{\rho} + (1-p)\frac{1}{2}I \tag{2.3}$$

The interpretation of this channel is that the input state passes through as intended with probability p or else is mapped to the maximally mixed state with probability $1-p$.

Definition 2. *The* ***partially depolarized pair*** Δ_p *is a two qubit mixed state of the following form*

[2]Note, I use the terms “partially depolarizing" and “depolarizing" interchangeably.

$$\Delta_p = p|\phi^+\rangle\langle\phi^+| + (1-p)\frac{1}{4}I \tag{2.4}$$

Observe that Δ_p is the state that results from sending one qubit of a maximally entangled pair through a partially depolarizing channel with success probability p.

$$\Delta_p = \mathcal{E}_{\mathrm{d}}(p, |\phi^+\rangle\langle\phi^+|) \tag{2.5}$$

Because of this correspondence, we will refer to p as the *coherence probability* of the state. The definitions for the partially dephasing channel and the partially dephased pair are similar and thus will be treated in brief detail.

Definition 3. *A **partially dephasing channel** $\mathcal{E}_{\sigma_i}(p, \hat{\rho})$ is a single qubit channel with the action*

$$\mathcal{E}_{\sigma_i}(p, \hat{\rho}) = p\hat{\rho} + (1-p)\,\frac{1}{2}(\hat{\rho} + \sigma_i\hat{\rho}\sigma_i) \tag{2.6}$$

Where $\sigma_i \in \{X, Y, Z\}$

Definition 4. *A **partially dephased pair** $\delta_p^{\sigma_i}$ in the σ_i direction (where $\sigma_i \in \{X, Y, Z\}$) is a two qubit mixed state of the form*

$$\delta_p^{\sigma_i} = p|\phi^+\rangle\langle\phi^+| + (1-p)(I \otimes \sigma_i)|\phi^+\rangle\langle\phi^+|(I \otimes \sigma_i) \tag{2.7}$$

As with partially depolarized states, we also have that,

$$\delta_p^{\sigma_i} = \mathcal{E}_{\sigma_i}(p, |\phi^+\rangle\langle\phi^+|) \tag{2.8}$$

Finally, we conclude this section of definitions by defining a term that encompasses our restricted error model.

Definition 5. *In the **decoherence model of quantum networking**, either all network channels are partially depolarizing channels, or all channels are partially dephasing channels.*

2.3.2 Swapping on partially depolarized and partially dephased pairs

In this section, we demonstrate that entanglement swapping between two states in the decoherence model of networking with coherence probabilities p_1 and p_2 respectively results in another partially depolarized state with coherence probability $p_1 p_2$.

Definition 6. *Let $\rho_{1,2}$ and $\sigma_{3,4}$ be two-qubit density operators with subscripts denoting qubit labels. The entanglement swapping operation $\mathcal{E}_{\text{swap}}$ on these states is defined*

$$\mathcal{E}_{\text{swap}}(\rho \otimes \sigma) = \frac{\langle\phi^+|_{2,3}(\rho_{1,2} \otimes \sigma_{3,4})|\phi^+\rangle_{2,3}}{\text{Tr}(\langle\phi^+|_{2,3}(\rho_{1,2} \otimes \sigma_{3,4})|\phi^+\rangle_{2,3})} \tag{2.9}$$

Where $|\phi^+\rangle_{2,3}$ is short-hand for the (partial) projection $I_1 \otimes |\phi^+\rangle_{2,3} \otimes I_4$

In principle, entanglement swapping will not always result in a projection of $|\phi^+_{2,3}\rangle$ but can include projections onto the other three Bell states as well. These other three scenarios are detectable and correctable with local operations, and so are not formally incorperated in our definition of swapping.

Theorem 1. *Let Δ_{p_1} and Δ_{p_2} be two partially-depolarized entangled pairs of coherence probabilities p_1 and p_2 respectively. An entanglement swapping operation on these states results in another partially depolarized state with coherence probability $p_1 p_2$.*

Proof. By definition,

$$\begin{aligned} \Delta_{p_1} &= p_1|\phi^+\rangle\langle\phi^+| + (1-p_1)\tfrac{1}{4}I \\ \Delta_{p_2} &= p_2|\phi^+\rangle\langle\phi^+| + (1-p_2)\tfrac{1}{4}I \end{aligned} \tag{2.10}$$

Using the shorthand $[\phi^+] := |\phi^+\rangle\langle\phi^+|$:

$$\begin{aligned} &\mathcal{E}_{\text{swap}}(\Delta_{p_1} \otimes \Delta_{p_2}) = p_1 p_2 \; \mathcal{E}_{\text{swap}}([\phi^+] \otimes [\phi^+]) \\ &+ p_1(1-p_2)\;([\phi^+] \otimes \frac{1}{4}I) + (1-p_1)p_2 \; \mathcal{E}_{\text{swap}}(\frac{1}{4}I \otimes [\phi^+]) \\ &+ (1-p_1)(1-p_2)\; \mathcal{E}_{\text{swap}}(\frac{1}{4}I \otimes \frac{1}{4}I) \end{aligned} \tag{2.11}$$

Evidently $\mathcal{E}_{\text{swap}}([\phi^+] \otimes [\phi^+]) = [\phi^+]_{1,4}$, and it is straightforward to verify that the other three terms are maximally mixed states. The resulting state is therefore

$$\mathcal{E}_{\text{swap}}(\Delta_{p_1} \otimes \Delta_{p_2}) = p_1 p_2[\phi^+] + (1 - p_1 p_2)\frac{1}{4}I \tag{2.12}$$

Which, as asserted earlier, is a partially depolarized pair with coherence probability $p_1 p_2$.

□

It is similarly straightforward to show that if $\delta_{p_1}^{\sigma_i}$ and $\delta_{p_2}^{\sigma_i}$ are partially dephased pairs with respect to σ_i, a swapping operation on these two states will result in a partially dephased pair with coherence probability p_1, p_2. We omit this proof for brevity and, as a rule, tend to omit proofs for all scenarios involving partially-dephased states since they are virtually identical.

Corollary 1. *The entanglement swapping operation on partially depolarized states is* ***associative****, meaning that for three partially dephased states $\Delta_{p_1}, \Delta_{p_2}, \Delta_{p_3}$ we have that,*

$$\begin{aligned}\mathcal{E}_{swap}(\mathcal{E}_{swap}(\Delta_{p_1}, \Delta_{p_2}), \Delta_{p_3})) = \\ \mathcal{E}_{swap}(\Delta_{p_1}, \mathcal{E}_{swap}(\Delta_{p_2}, \Delta_{p_3}))\end{aligned} \tag{2.13}$$

Proof. From theorem 1, we know that a swapping operation on partially depolarized pairs with coherence probabilities p_1, p_2 results in a partially depolarized pair with coherence probability $p_1 p_2$. It follows immediately from the associative property of multiplication that entanglement swapping on these states is also then associative. □

2.3.3 An additive path-finding cost

Having established the fact that swapping operations are essentially multiplicative on the coherence probabilities of partially depolarized and partially dephased states, we are now in a position to consider what the overall coherence probability is when routing over an an arbitrary network path. We recall from our discussion in sec. 2.2.1 that the first step in any routing protocol is to distribute entanglement between each pair of nodes. Assuming hypothetically that all pairs arrive at their destinations, the coherence probability of a state swapped through a network path is given by the following theorem.

Theorem 2. *Let $\mathcal{E}_d(p_1), \mathcal{E}_d(p_2), \cdots, \mathcal{E}(p_n)$ be a path in a quantum network connecting two end-users. The coherence probability of the path is*

$$P_{path} = p_1 \times p_2 \times \cdots \times p_n \tag{2.14}$$

Proof. We begin our routing protocol by assuming that entangled pairs are distributed between each link in the path. The initial state of the system is therefore

$$\Delta_{p_1} \otimes \Delta_{p_2} \otimes \cdots \otimes \Delta_{p_n} \tag{2.15}$$

By corollary 1 we know that entanglement swapping is associative over partially depolarized states. Applying swapping on the terms from left to right

$$\mathcal{E}_{\text{swap}} \cdots \mathcal{E}_{\text{swap}}(\mathcal{E}_{\text{swap}}(\Delta_{p_1} \otimes \Delta_{p_2}), \Delta_{p_3}), \cdots \Delta_{p_n}) \tag{2.16}$$

And this is easily seen by inspection to produce the state $\Delta_{P_{path}}$ between the end-users.

□

Theorem 2 tells us that the coherence probability of a pair swapped through a path is the product of the coherence probabilities of the constituent links. Finding the optimal path in terms of the coherence probability is therefore equivalent to finding the path whose links multiply to give the highest overall success probability. Optimal path-finding algorithms however require *additive* (not multiplicative) costs. To develop an additive cost over success probabilities we observe that,

$$\log\Big(\prod_{i=1}^{n} p_i\Big) = \sum_{i=1}^{n} \log(p_i) \tag{2.17}$$

Which indicates that coherence probabilities can be converted into an additive cost by taking the logarithm of each term. Notice however that $\log(p_i) \leq 0$ since $0 < p_i \leq 1$. Since path-finding algorithms require (by convention) the shortest path to have the smallest overall cost, we require a sign flip on each term. The additive cost with respect to the coherence probability is therefore

$$c_i = -\log(p_i) \tag{2.18}$$

2.3.4 Relating coherence probability with state fidelity

In this section we show how, using basic algebra, the coherence probabilities of partially depolarized and partially dephased pairs are related to the fidelity with respect to the maximally entangled state $|\phi^+\rangle\langle\phi^+|$. This is a useful conversion technique since fidelity is a familiar way to gauge the quality of an entangled pair whereas the coherence probability is not.

Theorem 3.

$$F(\Delta_p, |\phi^+\rangle\langle\phi^+|) = \frac{1+3p}{4} \tag{2.19}$$

Proof.

$$\begin{aligned}\Delta_p &= p|\phi^+\rangle\langle\phi^+| + (1-p)\frac{1}{4}I \\ &= p[\phi^+] + (1-p)\frac{1}{4}([\phi^+] + [\phi^-] + [\psi^+] + [\psi^-]) \\ &= \frac{1+3p}{4}[\phi^+] + (1-p)\frac{1}{4}([\phi^-] + [\psi^+] + [\psi^-])\end{aligned} \tag{2.20}$$

And since $[\phi^+]$ is orthogonal to the other three Bell pairs the fidelity of this state with respect to $[\phi^+]$ is $(1+3p)/4$ as required. □

Theorem 4.

$$F(\delta_p^{\sigma_i}, |\phi^+\rangle\langle\phi^+|) = \frac{1+p}{2} \tag{2.21}$$

Proof.

$$\begin{aligned}\delta_p^{\sigma_i} &= p[\phi^+] + (1-p)\frac{1}{2}\Big([\phi^+] + (I\otimes\sigma_i)[\phi^+](I\otimes\sigma_i)\Big) \\ &= \frac{1+p}{2}[\phi^+] + (1-p)\frac{1}{2}(I\otimes\sigma_i)[\phi^+](I\otimes\sigma_i)\end{aligned} \tag{2.22}$$

Similar to the previous theorem, the first term is the only element that contributes, giving us an overall fidelity of $(1+p)/2$ as required. □

2.4 The cost-vector formalism

Non-determinism is a pervasive fact of quantum networking. Entangled photons can be lost in transmission, entanglement swapping between photons has a maximum success rate of 50% [20], and entanglement purification is inherently probabilistic. The consequence of this non-determinism is that entanglement routing is a *highly volatile* process with many points of failure. Although the performance of routing protocols can be gauged by simulating the network, this may prove costly for large networks with many competing

end-users. Here, we present an alternative *static* picture of quantum networking that we call the *cost-vector formalism*. At the core of this formalism is the insight that every pair that *could* exist in the network (by following some sequence of swapping and purification operations) can be characterised with a *transmission probability* (*or efficiency*) η and a *coherence probability* p. The former is the probability that the pair can be successfully established between the end-users while the latter is the probability that the pair is actually suitable for teleportation. The tuple (η, p) is defined as the *cost-vector* of the pair. The main utility of this formalism is that quantum routing protocols can be re-cast as *edge reductions on a multi-graph*. Specifically, we will show that swapping operations correspond to path contractions while purification operations collapse multiple edges on the same vertices to a single edge. We begin this section by deriving these contraction rules for a specialised networking scenario and conclude with a holistic analysis about our framework.

2.4.1 Entanglement distribution

The first step in our cost-vector formalism is to construct the multi-graph corresponding to the initial state of our network. We recall from our discussion in section 2.2.1 that entangled pairs should be distributed no further than the nearest neighbors. We therefore imagine the *initial state* of the network as the point immediately after attempting entanglement distribution between each neighboring pair of vertices. More generally, we imagine this taking place over a small time interval Δt. Depending on the length of Δt, the computers of the network may have attempted to deliver entangled pairs one or more times. Each attempt is then represented in the multigraph as an edge connecting the corresponding vertices with a *cost-vector* that depends on the channel. With this multi-graph, we can now begin routing entanglement by contracting the edges with the swapping and purifying protocols we're about to describe.

2.4.2 Entanglement swapping

Suppose we have a network path consisting of n sequential channels. For the sake of example, let us suppose these are all partially depolarizing channels with coherence probabilities $p_1, p_2, \cdots, p_n$ and transmission probabilities $\eta_1, \eta_2, \cdots \eta_n$ respectively. The cost-vectors corresponding to these channels are then $(\eta_1, p_1), (\eta_2, p_2), \cdots, (\eta_n, p_n)$. When two

pairs with transmission probability η_1 and η_2 are *deterministically swapped*, the transmission probability of the output pair is $\eta_1\eta_2$ since both pairs must exist for the swapping to take place. The transmission probability for swapping along the entire path is therefore $\eta_1 \times \eta_2 \times \cdots \times \eta_n$. More generally, we might imagine that each swapping operation has some identical probability of failure η_{swap}, which gives a new overall success probability of

$$\eta_{\text{swap}}^{n}(\eta_1 \times \eta_2 \times \cdots \times \eta_n) \tag{2.23}$$

We recall from theorem 2, that the coherence probability of a pair distributed along a path is the product of the coherence probabilities of the channels in that path. The cost-vector of the reduced path is therefore

$$\left(\eta_{\text{swap}}^{n}(\eta_1 \times \eta_2 \times \cdots \times \eta_n),\ \ p_1 \times p_2 \cdots \times p_n\right) \tag{2.24}$$

2.4.3 Entanglement purification

An entanglement purification protocol takes a number of partially entangled states and uses local operations and classical communications to reduce the initial ensemble to a smaller number of states that more closely resemble some maximally entangled reference state. Considerable work has been done to find protocols that maximize the attainable yield of target states for finitely many initial states or bound the yields that are attainable in asymptotic settings. A notable result in this area is that there is a one-to-one correspondence between purification protocols on entangled pairs and CSS codes [5]. Here, we limit our attention to the *Bennett protocol* [4] which is the simplest instance of a CSS-type purification; It is capable of *detecting* the presence of a single Pauli error (for example the bit-flipped pair $(I\otimes X)|\phi^+\rangle = |\psi^+\rangle$). The Bennett protocol essentially works by coupling two pairs together and destructively measuring one of them to learn whether there was a Pauli error on either of the pairs. If a Pauli error was detected, the remaining pair must be discarded since there is complete uncertainty about where the error could have occurred. On the other hand if no error was detected, the remaining pair is kept. In this section, we show how the Bennett protocol acting on *partially dephased states* can be easily adapted to the cost-vector formalism. In short, this is because partially

dephased states only suffer one kind of Pauli error which makes them naturally suited for this type of purification. Following this, we briefly consider Bennett purification on partially *depolarized states* and demonstrate how it can be adapted to the cost-vector formalism with an additional post-processing assumption.

Partially dephased pairs

Let us begin by considering two *partially dephased pairs* with pair fidelities F_1 and F_2 The fidelity of the pair that passes the Bennett purification is given by

$$F' = \frac{F_1 F_2}{F_1 F_2 + (1 - F_1)(1 - F_2)} \tag{2.25}$$

and the success probability of the protocol is given by the denominator:

$$\eta_{\text{pur}} = (F_1 F_2 + (1 - F_1)(1 - F_2)) \tag{2.26}$$

We recall from section 2.3.4 that that the fidelities for partially dephased and partially depolarized states can both be related to coherence probabilities. Substituting F_1 with $(1 + p_1)/2$ and F_2 with $(1 + p_2)/2$ respectively (according to theorem 4) gives us

$$\begin{aligned} \frac{1 + p'}{2} &= \frac{(1 + p_1)(1 + p_2)}{2 + 2p_1 p_2} \\ p' &= \frac{p_1 + p_2}{p_1 p_2 + 1} \end{aligned} \tag{2.27}$$

Here, p' is the coherence probability of the new partially dephased state. Relating the success probability in terms of coherence probabilities:

$$\eta_{\text{pur}} = \frac{1}{2}(1 + p_1 p_2) \tag{2.28}$$

And we are now ready to present how cost-vectors are updated when two parallel edges are contracted by a Bennett purification. Let (η_1, p_1) and (η_2, p_2) be the cost-vectors of two partially dephased pairs that are shared a pair of end-users. Evidently, the coherence probability of the resulting purification is given by eq. 2.27. The overall success probability is the probability of the purification protocol taken with the product of each transmission probability (since both pairs must exist for the purification to be

attempted). The resulting cost-vector is therefore

$$\left(\frac{1}{2}\eta_1\eta_2(1+p_1p_2),\ \ \frac{p_1+p_2}{p_1p_2+1}\right) \tag{2.29}$$

Partially depolarized pairs

Similar to previous section, we begin by considering two partially depolarized pairs with pair fidelities F_1 and F_2. The fidelity of the output pair in this case is

$$F' = \frac{F_1F_2 + \frac{5}{9}(1-F_1)(1-F_2)}{F_1F_2 + \frac{1}{3}F_1(1-F_2) + \frac{1}{3}F_2(1-F_1) + \frac{5}{9}(1-F_1)(1-F_2)} \tag{2.30}$$

Unlike in the previous example where the output pair was a partially dephased state like its inputs, the output pair here does not have the form of eq 2.4. Rather, it has a skewed distribution of errors. This is because the Bennett protocol only detects one type of Pauli error at a given time (X or Z). For the cost-vector formalism to work, we must convert this state back to a partially depolarised pair. This can be done with *entanglement twirling* which equalises the noise terms using random local operations. The drawback of this approach however is that states with biased noise are much easier to purify than states with no bias at all. This means we have to expend some of our distillable entanglement in order to be consistent with our cost formalism in this scenario.

Substituting F_1 with $(1+3p_1)/4$ and F_2 with $(1+3p_2)/4$ respectively (according to theorem 3) gives us,

$$\begin{aligned}\frac{1+3p'}{4} &= \frac{3-p_2+p_1(7p_2-1)}{4+4p_1p_2} \\ p' &= \frac{2-p_2+p_1(6p_2-1)}{3+3p_1p_2}\end{aligned} \tag{2.31}$$

Where, as before, p' is the coherence probability of the new state. Substituting again, the success probability of the purification is

$$\frac{1}{2}(1+p_1p_2) \tag{2.32}$$

And the resulting cost-vector is

$$\left(\frac{1}{2}\eta_1\eta_2(1+p_1p_2),\ \ \frac{2-p_2+p_1(6p_2-1)}{3+3p_1p_2}\right) \tag{2.33}$$

2.4.4 Entanglement stacking

In the previous section, we demonstrated how entanglement purification can be used in our cost-vector formalism to combine concurrent edges into a single edge with a higher coherence probability. Here, we introduce a complimentary strategy called *entanglement stacking* which combines concurrent edges into a single link with a higher transmission probability. First, let us recall that the edges of our multi-graph represent entangled pairs *that could exist* in the network. We can choose to represent a collection of edges that are shared by a pair of users as a single link with a higher overall success probability. This simplifies the graph topology, but comes at the expense of some accuracy since removing edges means disregarding the possible existence of other legitimate pairs. Unlike with swapping and purification, we emphasise that there is no physical process that takes place during entanglement stacking.

Here, we propose a rudimentary entanglement stacking protocol on two edges with cost-vectors $v_1 = (\eta_1, p_1)$, and $v_2 = (\eta_2, p_2)$ respectively. For the sake of example, we suppose these are cost-vectors specifying partially depolarized states. Let η' denote the transmission probability of the stacked edge, and let it be defined as the probability that at least one of the two pairs are successfully transmitted.

$$\eta' = \eta_1\eta_2 + \eta_1(1-\eta_2) + \eta_2(1-\eta_1) \tag{2.34}$$

If one pair is transmitted, the end-users receive the v_1 state with probability $\gamma_1 = \frac{\eta_1}{\eta_1+\eta_2}$ and the v_2 state with probability $\gamma_2 = \frac{\eta_2}{\eta_1+\eta_2}$ The resulting mixed state is therefore:

$$\gamma_1(p_1|\phi^+\rangle\langle\phi^+| + \frac{1}{4}I(1-p_1)) + \gamma_2(p_2|\phi^+\rangle\langle\phi^+| + \frac{1}{4}I(1-p_2)) \tag{2.35}$$

Expanding and rearranging gives us,

$$(\gamma_1p_1 + \gamma_2p_2)|\phi^+\rangle\langle\phi^+| + (\gamma_1(1-p_1) + \gamma_2(1-p_2))\frac{1}{4}I \tag{2.36}$$

Which is another partially depolarized state with coherence probability

$$p' = \gamma_1 p_1 + \gamma_2 p_2 \tag{2.37}$$

The cost-vector of the stacked path is therefore

$$\Big(\eta_1\eta_2 + \eta_1(1-\eta_2) + \eta_2(1-\eta_1),\ \ \gamma_1 p_1 + \gamma_2 p_2\Big) \tag{2.38}$$

Although we do not treat on this subject any further, we remark that there appears to be good potential for developing more general stacking protocols. Rather than stacking up to a single pair for example, it may be preferable to develop a scheme for stacking n pairs into $m < n$ pairs as a more nuanced simplification.

2.5 Entanglement routing over time

Although the cost-vector formalism in its current iteration might be sufficient for routing between a small number of end-users, we will later see that congestion quickly becomes an issue as the number of users increases. Network congestion occurs because of *path collisions* where two or more parties are separated by a common *bottleneck*. Since it is impossible for all parties to send messages through a bottleneck simultaneously, some parties must wait until the channel frees up or another path becomes available. In the context of quantum networking, path-collisions occur when two or more parties need to swap over the same entanglement link. Similarly, these kinds of collisions are resolved by having parties wait until new links are established. This waiting action can (and should) be treated as a quantum channel since holding qubits in memory always introduces a small amount of unintended noise. In this section, we demonstrate how our cost-vector formalism can be extended to networking scenarios where states may be stored in *quantum memories*. In brief, our main insight is that multi-graphs representing distributed entanglement are extended in time by adding duplicate layers that describe the network in increments of Δt. Edges between layers represent *memory channels* that carry quantum states across time. Temporal routing protocols are then recast as edge-reductions on this larger multi-graph.

2.5.1 Memory channels

We recall from section 2.2.1 that a quantum network is defined as a collection of quantum computers that are interconnected according to a graph of single qubit error channels. These *transport channels* allow qubits to be moved across the computers of the network. Quantum memories on the other hand can be thought of as channels that *move qubits through time*. A precise, mathematical description of this is beyond the scope and utility of this chapter, so we simplify by supposing that every qubit can be assigned a label indicating its position in time. For example $\rho_{t=x}$ would correspond to the state ρ at a time x where time is measured in units of Δt (See sec. 2.4.1). The action of an *ideal memory channel* $\mathcal{E}_{\text{mem}}$ is then defined

$$\mathcal{E}_{\text{mem}}(\rho_{t=x}) = \rho_{t=x+1} \tag{2.39}$$

In practice, memory channels are imperfect since they represent real-world processes. As with transmission channels, we consider a simplified picture of quantum memories by supposing they are either *partially depolarising channels* or *partially dephasing channels* with a coherence probability p and a transmission probability η.

2.5.2 Transitory pairs

The fact that we can specify a p and η for our memory channels strongly suggests there is some way of incorporating memories into the cost-vector formalism. It is not immediately obvious though how this ought to be done. We recall from section 2.4 that the edges of our multi-graph represent entangled pairs that could exist in the network by following some sequence of purification and swapping operations. At face value, this seems to rule out the possibility that memory channels could be represented with edges. In this section however, we propose a fictitious category of states called *transitory pairs* that allow us to treat memory channels as states that are partially entangled *over time*. From this, we show how memory usage can be recast as entanglement swapping.

To begin, let us suppose we have an ideal memory channel $\mathcal{E}_{\text{mem}}$ as defined in equation 2.39. We now imagine the hypothetical scenario where this channel is applied to one half of a $|\phi^+\rangle$ pair at some initial time $t = 0$. This corresponds to a *non-physical* process where one of the particles is moved forward in time while the other stays behind. The

resulting *"state"* is therefore

$$|\phi^+_{\Delta t}\rangle := \frac{1}{\sqrt{2}}\Big(|0\rangle_{t=0}|0\rangle_{t=1} + |1\rangle_{t=0}|1\rangle_{t=1}\Big) \tag{2.40}$$

More generally, if the memory channel is partially depolarizing (dephasing), the state will be a partially depolarized (dephased) pair where the qubits are seperated by one time unit. For example, a partially depolarizing channel applied to one half of $|\phi^+\rangle$ would result in the object,

$$\rho_{\Delta t} = p[\phi^+]_{(t_1=0,\, t_2=1)} + (1-p)\frac{1}{4}I_{(t_1=0,\, t_2=1)} \tag{2.41}$$

Where t_1 and t_2 are the positions in time of the first and second particle respectively. Objects of this type, where the qubits of the underlying state are positioned at different times, we call *transitory pairs*. This name was chosen to emphasise the fact that these pairs are partway through a larger time-evolution and, as such, do not currently represent resources that can be used for teleportation. To illustrate this point more clearly, consider the time-evolution

$$\mathcal{E}_{\text{mem}} \otimes \mathcal{E}_{\text{mem}}\Big(|\phi^+\rangle_{(t_1=0,t_2=0)}\Big) = |\phi^+\rangle_{(t_1=1,t_2=1)} \tag{2.42}$$

Clearly, this is a valid physical process since it pushes the entire $|\phi^+\rangle$ state one step forward in time. Nevertheless, we can decompose this process to obtain a *transitory pair* midway through the calculation.

$$(\mathcal{E}_{\text{mem}} \otimes \mathcal{I})(\mathcal{I} \otimes \mathcal{E}_{\text{mem}})\Big(|\phi^+\rangle_{(t_1=0,t_2=0)}\Big) \tag{2.43}$$

$$= (\mathcal{E}_{\text{mem}} \otimes \mathcal{I})|\phi^+_{\Delta t}\rangle \tag{2.44}$$

For a transitory pair to be converted into a teleportation resource, it must find one or more memory channels that can bring both of its qubits to the same position in time. If this is not possible (because there are no available memories), the state is considered lost.

2.5.3 Temporal swapping

In the previous section, we defined the transitory pairs and suggested that using memory channels is equivalent to a swapping operation on transitory pairs that have been *"distributed"* through memory channels. Let us begin by first defining the swapping operation over transitory pairs. Our intuition informs us that a swapping operation should only be possible if the two qubits being measured exist at the same time. Therefore let $\rho_{t_1=a,t_2=b}$ and $\sigma_{t_1=b,t_2=c}$ be partially entangled transitory pairs where the second qubit of ρ and the first qubit of σ are both found at time b. The *temporal swapping operation* on these two states is defined by the action

$$\mathcal{T}(\rho_{(t_1=a,\, t_2=b)} \otimes \sigma_{(t_1=b,\, t_2=c)}) = \mathcal{E}_{\text{swap}}(\rho \otimes \sigma)_{(t_1=a,\, t_2=c)} \tag{2.45}$$

For the remainder of this section, we demonstrate the equivalence between memory usage and temporal swapping over transitory pairs. Imagine that two parties Alice and Bob share a $|\phi^+\rangle$ state and want to keep this pair in memory for one unit of time using partially depolarizing memory channels with coherence probabilities p_A and p_B respectively. The resulting state is easily verified to be

$$\mathcal{E}_d(p_1) \otimes \mathcal{E}_d(p_2)(|\phi^+\rangle\langle\phi^+|) = \Delta_{p_1 p_2} \tag{2.46}$$

Now let us treat the scenario from the perspective where, instead of applying the channel directly, we use the vertex contraction rule developed for swapping in the cost-vector formalism to determine the final state. Our swapping strategy is presented diagrammatically in figure 2.2. Initially, a single link exists between Alice and Bob at $t = 0$ with the cost-vector $(\eta = 1, p = 1)$. The first swapping operation (marked by the circle on $B_{t=0}$) results in a partially depolarized transitory pair between Alice at $t = 0$ and Bob at $t = 1$. Using the update rule derived in equation 2.24, the transitory pair has the cost-vector $(1, p_A)$. Swapping again on Bob's vertex and applying the same update rule results in a partially depolarized pair at $t = 1$ with a cost vector of $(1, p_A p_B)$, which is the same state described in equation 2.46. From this, we conclude that temporal swapping operations are equivalent to uses of memory channels.

One counter-intuitive feature of our time-dependent formalism is that network paths do not have to descend monotonically but can weave forwards and backwards in time. Al-

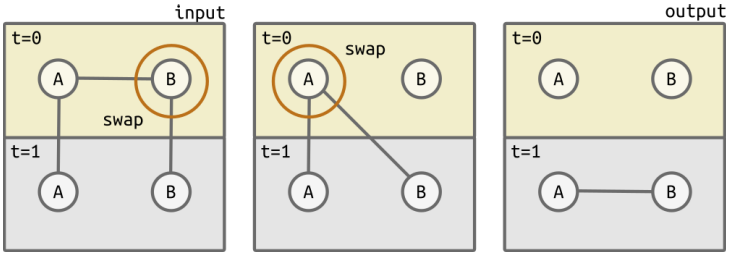

Figure 2.2: A simple routing protocol consisting of two temporal swaps. The initial state of the network is presented in the left-most square. At time $t = 0$, Alice and Bob share an entanglement link between them and each posses a transitory link which indicates availible quantum memory. The protocol begins by swapping the links at Bob's vertex at $t = 0$. The resulting entanglement is a transitory pair that extends from Alice to Bob one time-step into the future. Swapping the entanglement at Alice's position results in a genuine resource between Alice and Bob at $t = 1$.

though this might appear nonsensical at first, *some* of these paths nevertheless represent physically realisable quantum routing protocols. To illustrate, consider figure 2.3, where a network path between nodes $A_{t=1}$ and $D_{t=1}$ appears to jump up briefly to $t = 0$. What the path indicates in reality is that $B_{t=0}$ and $C_{t=0}$ have an entangled pair that can be stored in memory for one unit of time. During this time, $(A_{t=1}, B_{t=1})$ and $(C_{t=1}, D_{t=1})$ are able to establish entanglement links. If $(B_{t=0}, C_{t=0})$ had decided to store their pair in memory, a continuous path of entanglement links would exist between A and D at the $t = 1$ time-layer which could be swapped to give them end-to-end entanglement.

There are however instances of temporal paths that *do not* represent realisable routing protocols, some examples of which are presented in the top part of fig. 2.4. Essentially, there are two conditions that must be met for a path through a temporal network to be valid. The first condition is that the ends of the path must exist at the same time layer[3]. The second condition is that the path cannot push into the future *beyond* the end points of the path. An example of this condition being violated is given in Fig. 2.4 (II), where a path between $A_{t=0}$ and $D_{t=0}$ utilises a resource at $t = 1$. Such a path breaks causality by implying that a resource available in the future could be used to establish an entanglement link in the present.

[3]If the ends of the path are at different time layers, then either Alice or Bob will have to discard their half of the pair before they finish the communication. Consequently, establishing entanglement is impossible in this scenario.

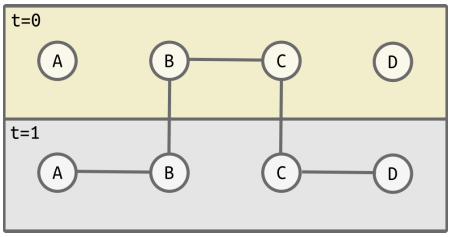

Figure 2.3: The path of this temporal network, which starts at $A_{t=1}$ and ends at $D_{t=1}$ appears to propagate backwards in time if followed from left to right.

Demonstrating necessity and sufficiency for temporal path conditions

Let us pause to formally demonstrate how these two conditions are necessary and sufficient for path validity. Let V be the set of all valid paths in a temporal network and let S be the set of paths (in the same network) that satisfy conditions 1 and 2 from the previous section. Our objective then is to demonstrate that $P \in S \Leftrightarrow P \in V$.

I will begin by proving the reverse case: that $P \in V \Rightarrow P \in S$. For $P \in V$ to be true, two criteria must be met: The path P must generate a valid entanglement resource when the links of P are swapped, and the communication protocol must respect causality; Specifically, an entanglement link generated between two users at a time t cannot require entanglement generated at a later time $t+\delta$. For the first criteria to be met, it is necessary that the end points of the path are on the same time-layer, as otherwise the resulting link would not correspond to a physical resource. For the second criteria, it is necessary for the path to stay at or above the bottom-most time layer (where the end-points of the path are). Since both of these criteria are also the conditions for $P \in S$, we see that $P \in V \Rightarrow P \in S$ as required.

Let us now consider the forward case: that $P \in S \Rightarrow P \in V$. If $P \in S$ then P is a path where the endpoints of the path are on the same time-layer. This implies that swapping the links along path P will result in a link that connects the end-users at the same time-layer, which is a valid physical resource. Moreover, $P \in S$ means that P is a path that never dips below the end-points which in turn means that there is no causal violation when implementing the protocol. Consequently, we see that P is a valid path, so $P \in S \Rightarrow P \in V$ as required. This concludes the proof that $P \in S \Leftrightarrow P \in V$.

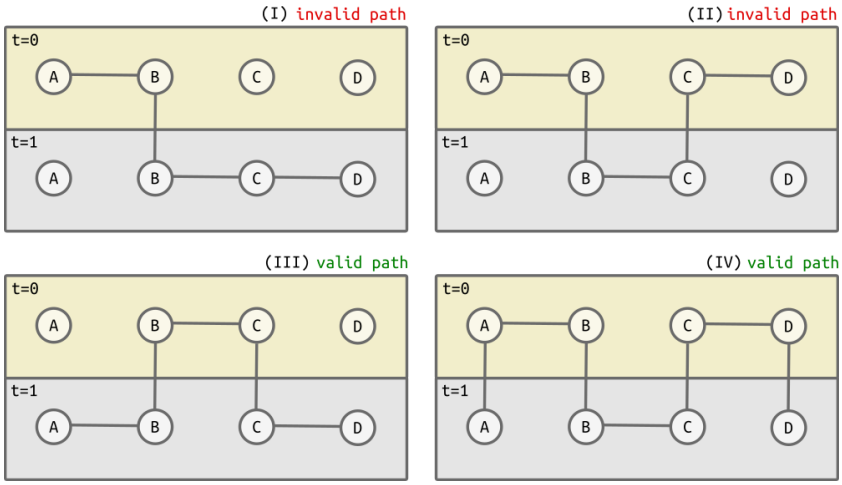

Figure 2.4: Four examples of valid and invalid temporal paths connecting two vertices A and D. Two conditions must be met for a path to be valid: The start and end points of the path must exist at the same time, and the path cannot ever descend *below* the ends of the path. Sub-figure (I) is an invalid path because the end points are not at the same time layer. Sub-figure (II) is an invalid path because a link between $B_{t=1}$ and $C_{t=1}$ cannot be used to generate a link between $A_{t=0}$ and $D_{t=0}$ (In other words, future resources cannot be used to route entanglement in the past). Sub-figure (III) is a valid path and is reproduced from fig 2.3. Sub-figure (IV) is a valid path made to resemble path (II). The difference now is that memory links connect A and D which allows them to store their pairs at $t = 0$ and swap at $t = 1$.

2.5.4 Temporal purification

Entanglement purification, like entanglement swapping, can be defined over transitory pairs but only in the special case of *one-way purification*. This is a subset of protocols where classical communication is only permitted in a single direction between the two parties. A purification on temporal links carries the interpretation that one party performs their half of the protocol with the intent to signal their measurement outcomes to a recipient at some later time.

2.5.5 Constructing the temporal meta-graph

We recall from section 2.4.1 that the state of a network after some time Δt can be represented as a multi-graph G where each edge is a potential entanglement link weighted with a corresponding cost-vector. Our objective now is to construct multi-graphs that encode *entanglement and memory resources* over n intervals of Δt. We call these structures *temporal meta-graphs* to distinguish them from the multi-graphs that only represent entanglement resources. To build a temporal multi-graph, we begin by initialising n copies of G where each graph represents the entanglement generated in increments of Δt. Following this, we connect each node to itself at the next time layer with a number of transitory pairs equal to the number of available memories. An example construction is illustrated in fig 2.5.

2.5.6 Pathfinding in temporal-metagraphs with asynchronous nodes

Routing protocols for quantum networks will almost certainly rely on path-finding subroutines to find optimal paths between two vertices according to some heuristic. This is an issue for temporal meta-graphs since multiple vertices can represent the same user at different times. We present one possible solution to this problem by introducing so-called *asynchronous nodes*. These are vertices that are temporarily added to the meta-graph for the duration of a path-finding protocol and are exclusively connected to the vertices representing a single user (See Fig. 2.6 and Fig. 2.7 for examples). By adding asynchronous nodes that correspond to a pair of users, path-finding algorithms are able to discover the optimal path between them over the entire temporal meta-graph.

One issue with this approach however is that a found path *is not necessarily valid* according to the two conditions given in section 2.5.3. Specifically, the end-points of a

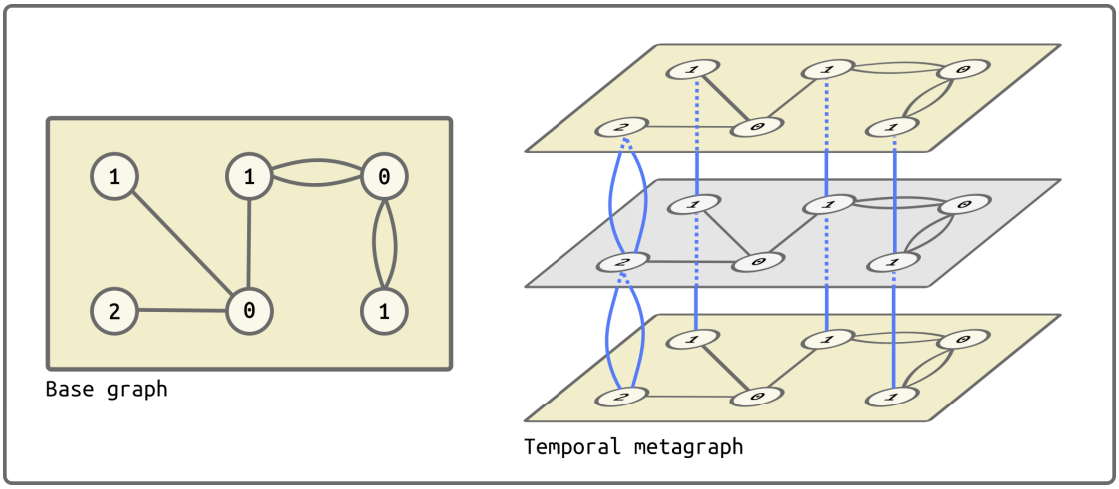

Figure 2.5: Left: An arbitrary multi-graph where edges are potential entanglement links. The numbers of the vertices indicate how many quantum memories exist at that location. Right: A temporal multi-graph corresponding to the base graph over three time-layers. Three instances of the original graph are connected by transitory pairs corresponding to the number of memories at each vertex.

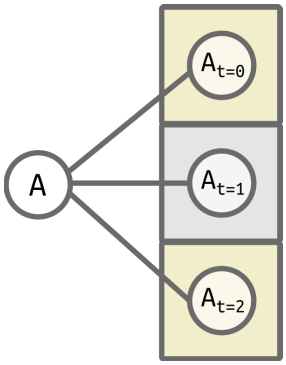

Figure 2.6: An example of an *asynchronous node* (left) for some network party Alice. This node has edges between all vertices that represent Alice at different times.

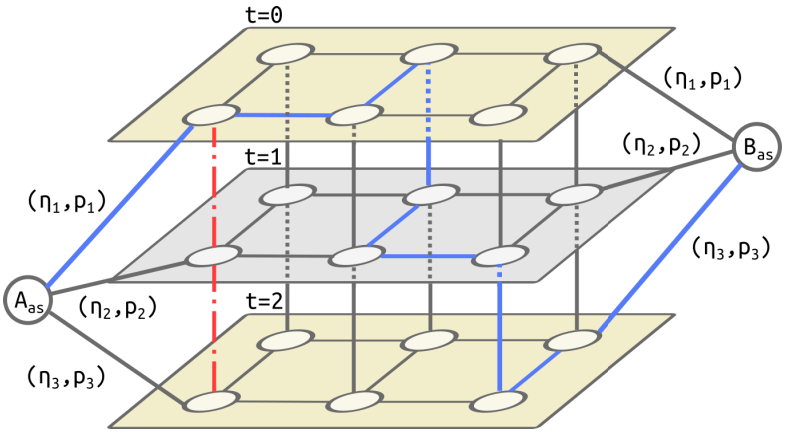

Figure 2.7: A temporal meta-graph with two asynchronous nodes A_{as} and B_{as} representing parties A and B. The edges connecting the asynchronous vertices may be weighted with arbitrary cost-vectors depending on how one wants to prioritise access to different time layers. An example (though not necessarily optimal) path between the asynchronous nodes is highlighted in blue. Although this path connects A_{as} and B_{as}, it does not currently correspond to a physically realisable communication protocol since the "start" of the path (the first vertex after A_{as}) is at time $t = 0$ while the "end" is at $t = 2$. Nevertheless, the path can be treated as valid provided that Alice is promised a secure memory channel from $t = 0$ to $t = 2$ (this is marked with a red dash-dotted line).

found path may exist at different times or the path may dip below the end-points (See 2.7). Both of these problems are rectified if Alice and Bob are both allocated additional memory channels *in advance* to ensure the ends of their path terminate at the same time and at the appropriate depth. A relatively easy (though simplistic) implementation of this solution is to assume that Alice and Bob each have a *reserve supply* of quantum memories that are not included in the description of the temporal meta-graph. This way, if an invalid path between asynchronous nodes is discovered, the reserve memory can be automatically allocated to ensure the path is physically realisable.

The edges connecting asynchronous nodes to their counterpart vertices may also be weighted with cost-vectors in order to prioritise access to certain time-layers. For example, it may be that the asynchronous edges pointing to lower time-depths are highly penalised to create an incentive for faster routing. Alternatively, if a layer is known to be congested, it may be preferable to prioritise paths that start at a different layer. The flexibility of this approach may prove advantageous in the development of quantum routing algorithms that incorporate load balancing.

2.6 Multi-path routing

A quantum network, like any other communications network, may be circuit-switched or packet-switched. A *circuit-switched* network is one where the end-users reserve network paths for the duration of their communication. This model is well-suited for routing entanglement in the *link-level* paradigm, since high fidelity pairs must be established along a fixed path. Compare this to the *packet-switching* picture where data is routed over many different channels subject to availability. This approach is favored by the *end-level paradigm* where there is flexibility in the paths that can be taken. Moreover the packet-switching paradigm allows for the possibility of routing over *multiple paths simultaneously*. In this section, we introduce rudimentary greedy algorithms for multi-path routing in the end-level paradigm of our cost-vector formalism involving one or more pairs of end-users. We emphasise that these algorithms are unlikely to be optimal in most scenarios, particularly in cases where multiple users must route together. Nevertheless, we will see in the latter part of this chapter how even these basic protocols can allow us to make a number of important insights into the design principles of quantum networks.

2.6.1 One user-pair

Our greedy algorithm for multi-path routing on one user-pair is outlined in the textblock labeled Alg. 1. We begin with a cost-vector multi-graph G, a pair of end-users and a small positive number ϵ which indicates the pair infidelity we hope to attain. Essentially, the strategy of this algorithm is to find the best path on the graph G with respect to the coherence probability p. Once this path is found, it is reduced to a single link by swapping the intermediate vertices. If the fidelity of this link is within the threshold, (i.e. if $F(\text{link}, |\phi^+\rangle \geq 1 - \epsilon)$ the protocol ends. Otherwise, the next best path is reduced and the two links are purified to one using the Bennett protocol discussed in section 2.4.3. This repeats until either the resulting link exceeds the fidelity threshold or there are no more paths between the users.

2.6.2 Multiple user-pairs

Our algorithm for multi-path routing with multiple users is similar to our previous algorithm, and is presented in Alg. 2. The only significant difference is that we attempt to

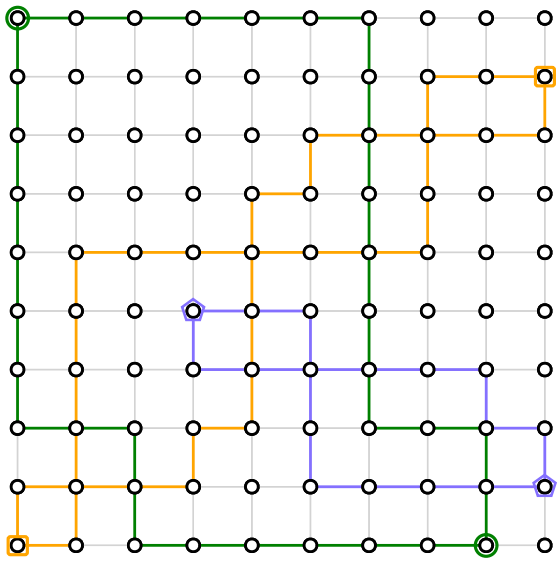

Figure 2.8: The outcome of Alg 1 when applied to an instance of a 10×10 grid with three randomly selected pairs of end-users (marked with circles, squares and pentagons respectively). All edges are equally weighted.

input : cost-vector multi-graph G, user_pair, threshold ϵ.
output: (if successful) a partially entangled link between the users of user_pair with pair fidelity above $1 - \epsilon$

begin
  while *paths exist between user_pair* **do**
    p = `best_path_wrt_coherence(`G, *user_pair*`)`
    link = `swap_path(`p`)`
    if *user_pair already has link* **then**
      link = `purify(`*link, old_link*`)`
    if $F(\text{link}, |\phi^+\rangle) \geq 1 - \epsilon$ **then**
      return link *(success)*
  return link *(fail)*

Algorithm 1: A greedy algorithm for multi-path entanglement routing between a single pair of end-users. The subroutine `best_path_wrt_coherence` is a path-finding algorithm that finds the best path between a given pair of users with respect to the coherence probability p. In the `swap_path` method, a path p is reduced to a single link by swapping each adjacent pair of links along p. Similarly, the `purify` method takes two links and applies the purification operation described in either eq 2.29 (for dephasing noise) or eq 2.33 (for depolarizing noise)

```
input  : cost-vector multi-graph G, end_users, threshold ε.
output: (if successful) partially entangled links between every end-user each
        with pair fidelity above 1 − ε
begin
    while end_ users have paths do
        for end_ users with paths do
            p = best_path_wrt_coherence(G, end_ user)
            link = swap_path(p)
            if end_ user already has link then
                link = purify(link, old_ link)
            if F(link, |φ+⟩) ≥ 1 − ε then
                remove end_user from list # they've got a good pair!
    if all users have good pairs then
        return links (success)
    else
        return links (fail)
```

Algorithm 2: A greedy algorithm for multi-path entanglement routing between multiple end-users.

balance requests by allocating paths to each user sequentially. In other words, we find a path for the first users, then find a path for the second users, and so on, before looping through the users again. This continues until either each user is able to purify a link to the fidelity threshold or there are no more paths remaining between any of the users.

2.7 Benchmarking

Characterising the performance of a routing algorithm, whether quantum or not, is an exceptionally difficult research challenge. The first and perhaps most significant obstacle is that it is not obvious how performance ought to be scored, since there are many (if not innumerably many) valid options for doing so. The second problem is that the performance of a routing algorithm will inevitably depend on the topology of the network and the underlying *demand function* which models how and when users make requests to the network. This is less of a concern in the classical context since simulated networks can be modeled after networks in the real world. We, on the other hand, have no such luxury since quantum networks do not currently exist at scale. For simplicity therefore, we limit ourselves to a toy model of quantum networking where the vertices of the network are arranged in a grid with equally weighted edges (See Fig. 2.8) and where

the demands of the network are specified by a random selection of n user-pairs who request entanglement links without imposing any demands on the required quality. We quantify the performance of our greedy algorithms by considering the average fidelity of the delivered pairs in tandem with the average efficiency at which they are delivered.

We begin in section 2.7.1 by investigating the average case performance of our multi-path algorithm for a single pair of users. Although this may seem to be a mundane exercise, we observe a surprising amount of combinatorial complexity which demonstrates the effectiveness of random sampling over naive analysis for quantifying network performance. Following this in section 2.7.2, we study the effectiveness of our routing algorithm when multiple users are competing simultaneously. In doing this, we identify a *saturation point* in which adding more users results in diminishing returns on performance. Finally, in sections 2.7.3 and 2.7.4 we investigate how network congestion is ameliorated by either scaling the network in size or extending the network in time respectively. Our findings indicate the latter approach is considerably more effective.

2.7.1 Benchmarking multi-path purification with single-user networks

We begin by studying the average case performance of Alg. 1 for a single pair of users in a square lattice graph (See Fig. 2.8) of variable size where each edge represents a *partially dephased* pair with a cost-vector of $(\eta = 0.794,\ p = 0.897)$. These seemingly arbitrary values are an artifact of an earlier version of this work where we described our costs in units related to decibels[4]. In each trial, we randomly select a user pair from the network and use Alg. 1 to establish an entanglement link using a number of paths up to some specified maximum. We present our results for this scenario in Fig. 2.9. The average fidelity and transmission probability of the distributed pair are given in Fig. 2.9(a). Because the Bennett purification protocol is non-deterministic, we see the fidelity increase with successive purifications at the cost of decreasing transmission probability. We also observe that both the fidelity and efficiency decrease with grid size since the average distance between random users increases. For a square lattice, the average number of steps between two vertices (also known as the Manhattan distance) is $\frac{2}{3}n$ (See App. 2.9

[4]Decibels are a common way to describe the attenuation of a classical channel. When I started this project, I initially thought it would be best to describe cost-vector channels in terms of decibels as well. This probably would have worked if I chose to have one decibel rating for η and one for p. Instead, I chose to have one decibel rating for η and for the fidelity F, which was extremely awkward and confusing. In later revisions I decided to walk this idea back completely. The cost-vector $(\eta = 0.794, p = 0.897)$ corresponds to $1dB$ loss in both η and F respectively.

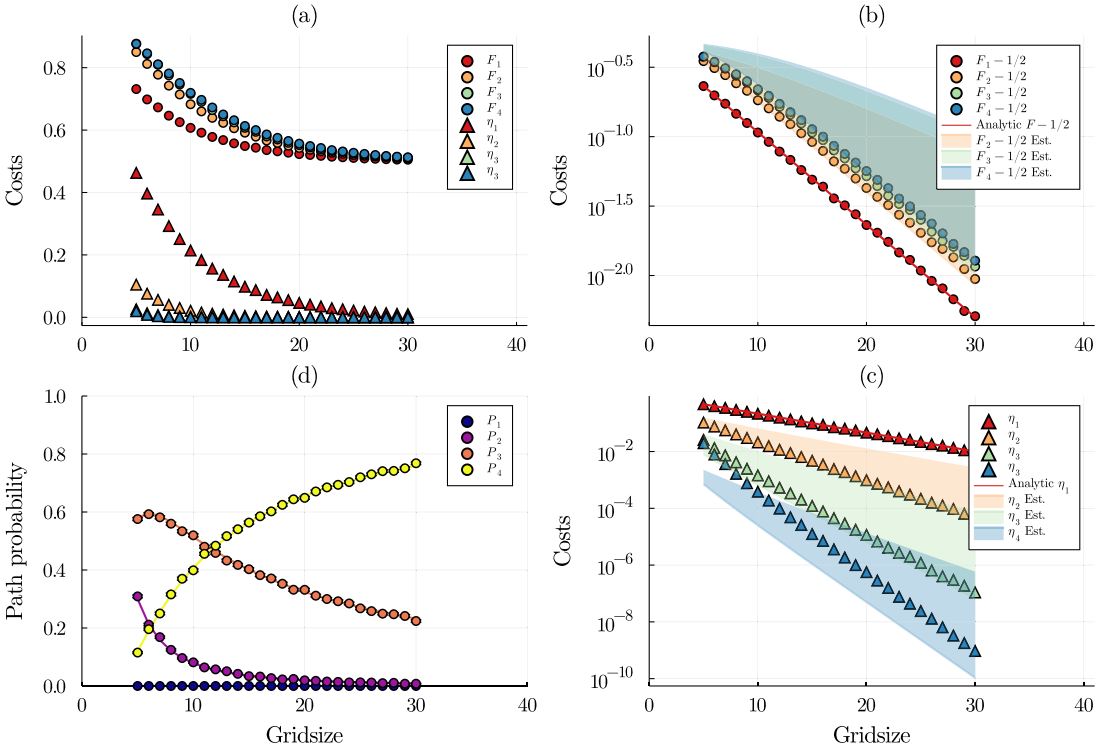

Figure 2.9: Single-user network performance versus grid size for different numbers of maximum allowed (edge-disjoint) paths. For each trial run, one random user pair was selected on a $n \times n$ rectangular lattice and was routed using `greedy_multi_path`. Each edge is weighted with the cost-vector $(\eta = 0.794, p = 0.897)$, and each data point is the average of 5000 random trials. **(a)** Average routing costs in terms of efficiencies η_j and fidelities F_j, where j is the maximum number of allowed paths. **(b, c)** Average costs plotted on a log scale for fidelity $(F_j - \frac{1}{2})$ (b) and efficiency (c). A solid line is plotted for the one-path case where an exact analytical solution is known. For the multi-path cases, the shaded areas show estimated expected scalings for each case, based on analytical approximations. **(d)** Probability P_j that a user finds j paths in the case where up to four paths may be purified. Analytical curves for each path probability are overlaid.

for a derivation). By plotting the fidelity and transmission probabilities on a log scale (Figs. 2.9(b,c)) we confirm that, as the size of the grid increase, the fidelity and efficiency decay exponentially towards their asymptotic values of $\frac{1}{2}$ and 0 respectively. We observe minor deviations from this trend (particularly in Fig. 2.9(c)) for small grid sizes which we attribute to *boundary effects*. Specifically, the users in small grids are more likely to be located on the edges or corners of the lattice which limits the number of paths they can use for routing.

We now focus on analytically validating our data, which proves to be a challenging task even in this relatively simple scenario. For a single path, Fig. 2.9 shows that our

numerical data agrees exactly with a simple analytical prediction where the overall cost of the path is calculated as a function of the average Manhattan distance. Allowing for multiple paths however introduces three effects that complicate the analysis. Firstly, the lengths of available network paths depend on whether the end users lie in the same row or column. To illustrate, let L be the Manhattan distance between two users where neither party is located on a boundary of the lattice. The four shortest edge-disjoint paths connecting them will have lengths $(L, L+2, L+2, L+8)$ for users in the same row or column, and $(L, L, L+4, L+4)$ otherwise. The chances of selecting a user pair in the same row or column diminishes quadratically with grid size, and this fraction can be neglected if the grid is sufficiently large. The second complicating effect is that users on the network boundaries have fewer edges and therefore may not be able to use the maximum number of paths permitted by the experiment. Similarly, this effect can be seen in Fig. 2.9(d) to diminish with grid size.

The third and most significant complication is the non-linear update rule of the Bennett purification (Eq. 2.29). Unlike the previous two effects, the average transmission probability and fidelity of a purified pair cannot be straightforwardly related to the average Manhattan distance, especially when taking the previous two effects into account. Because of this, we elected to compare our multi-path data against two estimates that roughly bound the expected values. These estimates, which define the edges of the shaded regions in Figs. 2.9(b, c), are calculated by purifying a set of paths with lengths given by the edge-disjoint path sets described above—$(L, L+2, L+2, L+8)$ for users in the same row or column, and $(L, L, L+4, L+4)$ otherwise—with L taken to be the average Manhattan distance for a given lattice size.

2.7.2 Multi-user, multi-path routing

Competition between end-users is significant in quantum networks where entanglement resources are scarce. Here, we benchmark the performance of our multi-party algorithm (Alg. 2) for a 10×10 grid lattice with a variable number of random end-users up to the fully saturated limit of 50 user pairs. As in the previous section, each link represents a partially dephased pair with a cost vector of $(\eta = 0.794, p = 0.897)$ though this time we do not impose any artificial constraints on the number of paths that communicants can use. The results of our experiment are shown in Fig. 2.10.

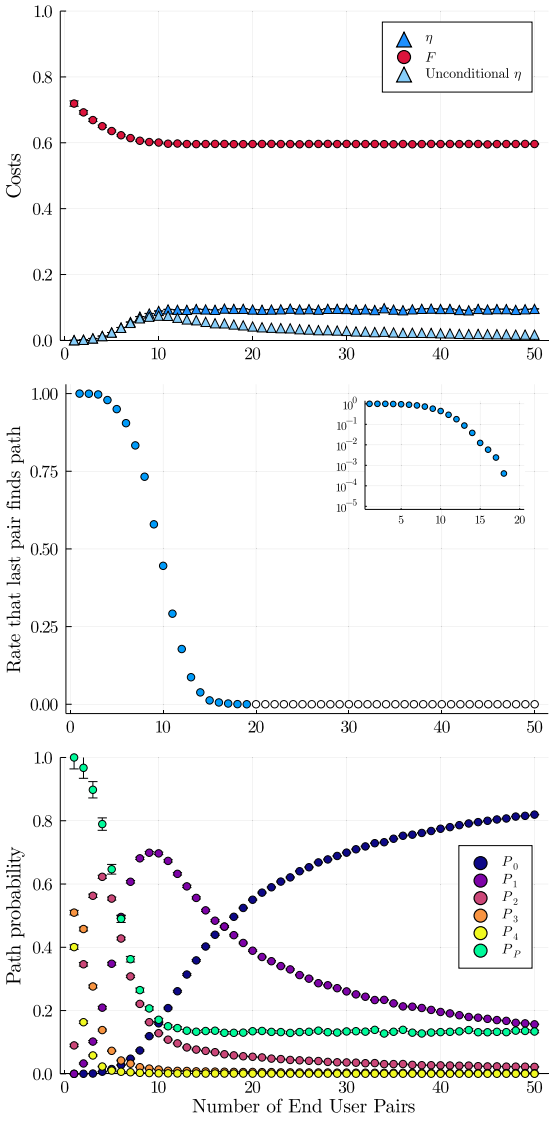

Figure 2.10: Network performance with the number of competing end users. For each sample, n random userpairs were selected on a 10×10 grid lattice and were routed using `greedy_multi_path`. Each edge is weighted with the cost-vector ($\eta = 0.794, p = 0.897$), and each data point was collected with 5000 samples. (top) Average routing costs in terms of average conditional efficiency η, average conditional fidelity F, and average unconditional efficiency. (middle) The probability that the last userpair queued in `greedy_multi_path` finds at least one path. Unfilled circles are data points where no path was found in the 5000 samples. [top right] log plot. (bottom) Probability P_j that a user finds j paths. P_P is the probability of purification, that is, the likelihood a userpair finds at least two paths between them.

Fig. 2.10 (top) shows how the average transmission probability and pair fidelity change with the number of competing user pairs. Starting from the maximal fidelity expected for the 10×10 single-pair case already analysed in Fig. 2.9, we see the fidelity initially decrease, before flattening out to a steady-state value after approximately 10 user pairs. By contrast, while the efficiency initially increases, it reaches a maximum value at around 10 user pairs, before decaying away towards zero. The initial change reflects the fact that as more users are introduced into the network, fewer user pairs are able to access multi-path routing, and without the ability to exploit the post-selective purification process, this leads to higher efficiencies, but lower fidelities. In Fig. 2.10 (bottom), we see that the rates associated with users finding multiple paths all decrease rapidly, and the proportion of users finding only one path increases to a maximum at around 10 user pairs. However, we also see a new effect, arising due to multiuser competition, namely that the proportion of users finding no paths (P_0) increases rapidly after the same point. Thus while the fidelity appears to reach a steady-state value even for large numbers of users, the decaying overall transmission probability for a randomly selected user pair highlights that the network becomes less and less effective at distributing entanglement. To isolate the different factors at play, we also plot a *conditional efficiency cost* in the top figure, where the transmission probability is averaged only over those users who are able to find at least one path. Now, the transmission probability also reaches a steady state value after around 10 user pairs. Our multi-user algorithm (Alg. 2) cycles sequentially through all randomly chosen user pairs (in a fixed order), looking first for one path each, then more paths. Fig. 2.10 (middle) shows that the rate at which the final user finds at least one path decays very rapidly towards zero, and shows that even though it is possible to choose up to 50 non-colocated user pairs in a 10×10, overall network congestion increases very rapidly and the network capacity effectively reaches full saturation at many fewer user pairs. And once new user pairs can no longer find any new paths, adding users no longer affects how many paths can be found by the other user pairs and the distribution over path numbers remains constant. Indeed, by aggregating the multiple-path rates into a single probability of a user pair being able to access purification, P_P, we see that this probability drops rapidly and monotonically from 1 to a low steady-state value at around 10 user pairs, with the curve closely tracking the shape of the efficiency curve observed in the top panel. These results are consistent with the change in purification rate being

the main effect driving the changing path costs in this multiuser setting.

2.7.3 Network scaling effects

In the previous sections, we demonstrated how competition can hinder the performance of multi-path entanglement routing. Here, we consider strategies to ameliorate the effects of competition, which we will demonstrate for the case of network congestion. A simple, if perhaps brute-force strategy for mitigating competition is to increase the size of the network. To explore this scenario, shown in Fig. 2.11, we start with the same fully saturated 10×10 rectangular lattice network with fifty randomly chosen end-users that was studied in Fig. 2.10, and analyse performance as we increase the size of the network. As usual, each edge is weighted with a cost vector of $(\eta = 0.794, p = 0.897)$. In this scenario, we expect that as we decrease the relative density of user pairs, more links will be routed on average in exchange for a larger average path length. Fig. 2.11 (top) is a logarithmic plot showing the average routing costs in terms of η and $F - \frac{1}{2}$ versus grid size, the clear linear trends showing the expected exponentially decaying success probabilities with average path length. What is perhaps somewhat surprising is that this effect completely dominates any improvement in congestion that might have been expected to result from increasing the number of paths available for path routing.

As seen in Fig. 2.11 (bottom), and earlier, for the fully saturated 10×10 lattice, there is more than an 80% chance that a randomly chosen user pair will not be able to find any viable communication pathways. In Fig. 2.11 (bottom), we also see that the proportion of users that do find paths increases rather rapidly with gridsize. Despite this, Fig. 2.11 (middle) shows that the rate that the last (50th) user is able to find a path does not start to increase significantly until we reach a grid size around 50×50 (a network 25 times larger than the initial 10×10 case). Even then, this increases rather slowly, not even exceeding 50% until a lattice size around 130×130, by which point the end-user pairs already occupy just a fraction of a percent of the total nodes. The proportion of users finding the most paths (3 or 4) increases even more slowly. Indeed, across the full range, P_3 and P_4 combined only reach around 5%, whereas P_2 (still increasing) reaches in excess of 80%, substantially larger than the maximum P_2 reached in Fig. 2.10.

By comparison, the probability a random user pair will have the connectivity required to find 3 or 4 paths in principle, is already much larger than 99%. These results indicate

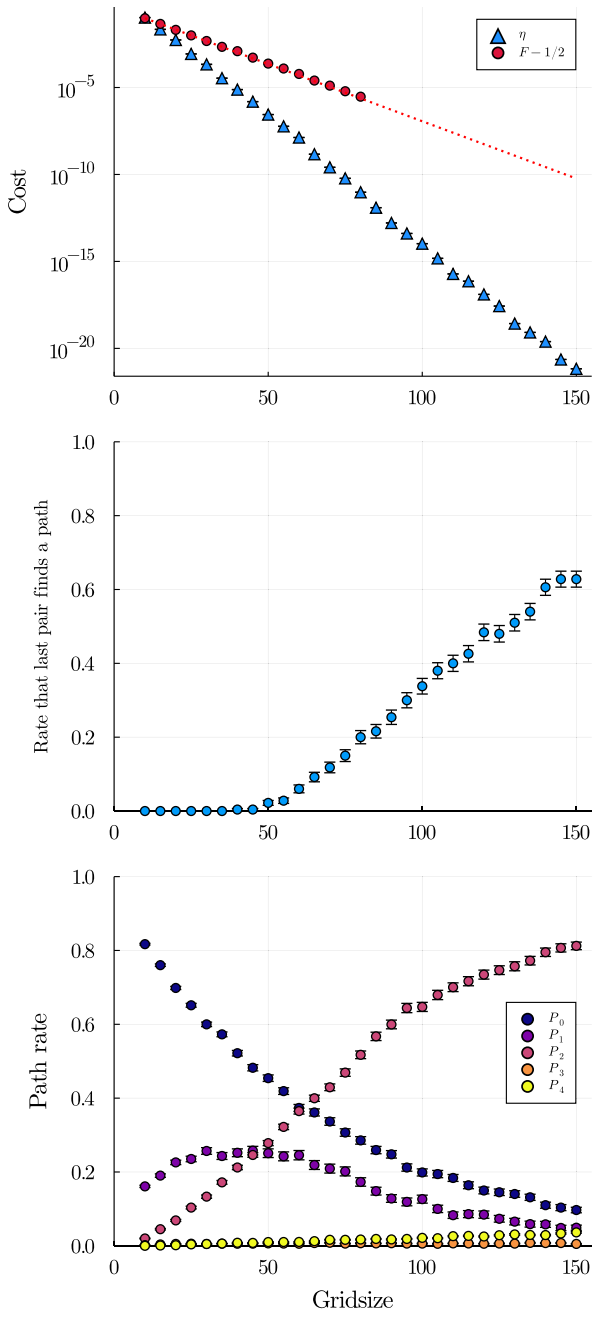

Figure 2.11: Routing data for fifty randomly chosen userpairs using `greedy_multi_path` on a grid lattice with variable size. Each edge is weighted with the cost-vector ($\eta = 0.794, p = 0.897$), and each data point was collected with 500 samples. (top) Average multipath routing costs, shown in log scale, for the efficiency η, and fidelity F of the Bell pair (or rather, $F - \frac{1}{2}$). Note that, in log scale, the fidelity data does not show across the full range, because the fidelity rapidly saturates to its asymptotic value of 0.5 beyond the floating point precision used to store the data. (middle) Probability that the last pair is able to find at least one path between them. (bottom) Probabilities of different path finding scenarios such that P_j is the case that j paths were found between the user pair.

that it is much more difficult to unlock paths by expanding the size of a network than it is to increase congestion by adding user pairs. Furthermore, the exponential increase in path costs associated with a larger network, more than counterbalances any potential improvement in multipath routing that the larger network enables. Even when the user-pairs make up a tiny fraction of the total number of nodes, we still find significant competitive effects that limit the multi-path routing capacity. Increasing lattice size is therefore clearly an ineffective strategy for solving network congestion in multipath entanglement networks. In the next section, we explore the alternative paradigm for mitigating congestion with temporal multiplexing, rather than spatial multiplexing.

2.7.4 Network Performance with Time-depth

In this section, we use the temporal meta-graph formalism developed in Sec. 2.5 to study how extending a network in time ameliorates congestion. To be consistent with our methodology in the previous section, we again start with a fully saturated 10×10 lattice graph with 50 user-pairs and analyse the performance of our multi-user algorithm (Alg. 2) as we increase the temporal depth of the meta-graph. In Sec. 2.5.6 we enable path-finding over temporal meta-graphs by temporarily introducing two asynchronous nodes that correspond to a pair of end-users independent of time. Paths found between asynchronous nodes may not necessarily be valid, so we compensate by assuming that each user has a sufficient supply of lossless quantum memories in reserve. Additionally, we assume that every computer in our network has one bit of lossless quantum memory as a publicly available resource. In our previous experiments, a random user pair could use up to a maximum of four paths owing to the fact that the maximum degree of a vertex in a square lattice is four. In order to maintain a fair comparison, we therefore impose a four path limit for each pair in this experiment as well.

We present our results in Fig. 2.12. The top part of this figure shows the average routing costs of the network versus the maximum depth of the temporal meta-graph. The two dotted lines show the asymptotic "competition-free" values for average efficiency η and fidelity F, respectively, calculated by considering the single-user case with up to five temporal layers available for redundancy. Fig. 2.12 (bottom) shows the likelihood that a random user-pair will be able to locate zero, one, or more paths between themselves. Here, we find that only a few temporal slices are required to completely eliminate the

congestion caused by multi-user competition; Only four time steps are needed for P_0 (the probability of finding no paths) to become negligibly small. Beyond this, we see that network performance continues to improve with increasing temporal depth; Fewer than fifteen layers are necessary to effectively guarantee that every user is able to find the maximum number of allocated paths.

Although our results are conclusively in favour of extending the network in time to improve congestion (as opposed to increasing its scale) it is not clear to what extent the *publicly available* quantum memories contributed to this improvement since users also had the option to begin routing at later times. We study the contribution of quantum memories by considering the average maximum time-depth reached by our multi-user algorithm for two different 10×10 grid networks. The first network has one memory channel available per computer while the other has no public memory channels. If memory channels significantly improve network congestion, then we expect the average maximum time-depth of our routing algorithm to be *smaller* for the network with memories. What we find in Fig. 2.13, however, is that there is no significant difference between the maximum time-depth reached by the two networks. This suggests that, at least for our greedy algorithm, quantum memories do not significantly improve network performance. Rather, our initial investigation indicates that the effectiveness of temporal routing is primarily because of the freedom to defer path-finding to a less congested layer.

2.8 (Appendix) Multi-path routing on realistic network topologies

Throughout this chapter, we limited our attention to nearest-neighbor 2d grids which (while informative) are not especially realistic. In this section, I offer some hypotheses about how the performance of our multi-path routing algorithm(s) (algs. 1, 2) might change under a more refined network model.

Historically, the conventional wisdom was that a typical network is *scale-free*, which means that the distribution of node degrees follows a power law [2]. Specifically, the probability $P(k)$ of randomly selecting a network node with degree k is proportional to k^γ where γ is a positive constant (usually set between 2 and 3). A recent result from Broido et. al. [6] however indicates that scale-free networks are rare in practice, and that

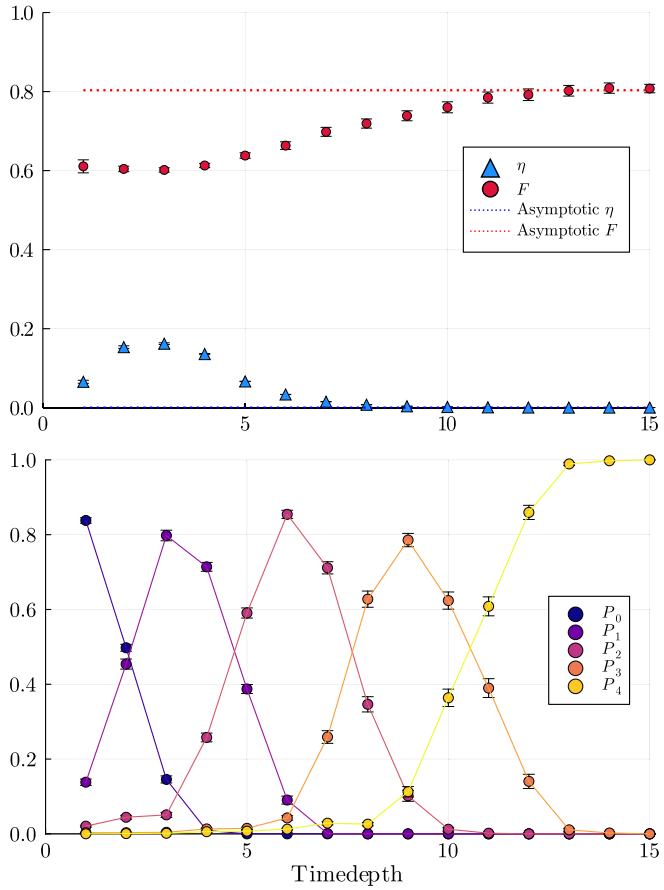

Figure 2.12: Network performance versus the maximum allowed time-depth. The underlying network is a 10×10 grid lattice that is extended to n distinct temporal layers and 1 asynchronous layer that represents the nodes irrespective of time. Each node is equipped with a lossless quantum memory and so has a directed edge to itself at the next time layer up to the maximum depth. For each sample, a random user pair is chosen at the asynchronous layer and is routed through the temporal network using `greedy_multi_path`. Each edge (Except for those corresponding to lossless memory) Each edge is weighted with the cost-vector $(\eta = 0.794, p = 0.897)$, and each data point was collected with 1000 samples. (top) Average routing costs in terms of the efficiency η, and fidelity F of the Bell pair. The dotted lines are the costs for η and F in the limit where it is assumed each userpair has a time layer to themselves. (bottom) Probabilities of different path finding scenarios such that P_j is the case that j paths were found between the userpair.

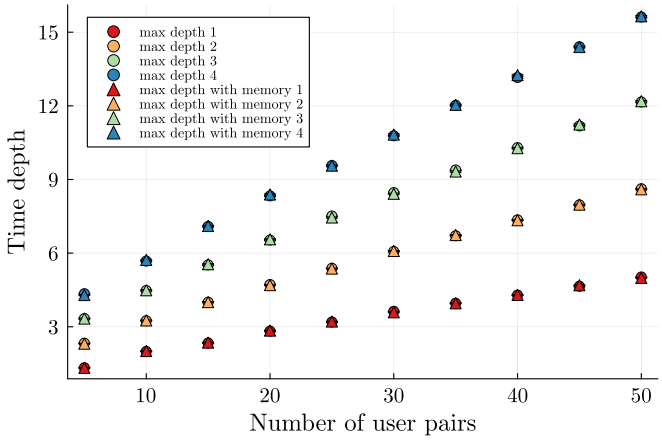

Figure 2.13: A comparison of the average maximum time depth reached by `greedy_multi_path` for two different networks; Both are 10×10 grids extended in time by 20 layers but one has lossless quantum memories on each node while the other has none. If a node has a memory, there exists a directed edge pointing the node to itself at the next time-layer. The routing between temporal layers is unrestricted, meaning that users may start and finish at any level they like so long as it respects causality. Earlier times are prioritised over later times, and asynchronous routing is prioritised over memory channels. Each edge (Except for those corresponding to lossless memory) Each edge is weighted with the cost-vector $(\eta = 0.794, p = 0.897)$, and each data point was collected with 1000 samples of randomly chosen user pairs. Data analysis confirmed that the maximum time depth of 20 was never reached in any of the samples.

typical networks are more accurately described by *log-normal* degree distributions.

Regardless, a common feature of these "realistic" networks is the relatively large proportion of so called *hub nodes*. These are vertices with high degree that are often included in the shortest-paths connecting pairs of nodes. One potential consequence of having many hub nodes is the emergence of "small world" behavior. A network is said to be *small-world* if the *typical distance* L between a random pair of nodes grows logarithmically with the number of nodes n in the network.

Let G and S be quantum networks (with n nodes each) where G is a 2d grid and S is a small world network. For the sake of consistency, let us suppose that all channel costs are identical and that one entanglement link initially exists between each adjacent pair of nodes in the respective network graphs. The first, (and most obvious) difference between these networks is that the average path length of S is asymptotically shorter than that of G [5]. Consequently, we expect the average quality of end-to-end entanglement to be higher in S than in G.

It is unclear however, whether S will have a higher entanglement throughput than G on average. Analyzing throughput is a difficult task since throughput depends not only on the number of pairs that are routed, but also on the *quality* of those pairs. To simplify, let us only consider the question of whether S or G can route more pairs on average. The answer to this will likely depend on *how many hub nodes* need to be visited on average for a shortest path in S. If the answer is more than one, then I suspect G will have a higher throughout than S. My reasoning here is that removing an edge in S that connects two hub nodes seems more "consequential" than removing one (or even many!) edges in G. This is especially true if users need to visit more than one hub node on average, since the edges between hub nodes will likely become serious bottlenecks. On the other hand, if only one hub node needs to be visited on average, then it's possible that path collisions between users will be relatively rare.

It's worth pointing out though that the *one-link-per-neighbor* assumption is fairly contrived, especially in the context of small-world networks. Consider the internet for example, which despite its size has only about a dozen *tier-one* service providers. Clearly, the channels that connect these providers have orders-of-magnitude higher capacity than the channels that connect us to our local service providers. The same will certainly be

[5] The average path length of S is proportional to $\log(n)$ by definition while the average path length of G is $\frac{2}{3}n$. See appendix 2.9 for a proof of the latter

true for quantum networks. If the hub nodes in S are allowed to have multiple edges shared among themselves, it seems virtually guaranteed that the throughput of S will be higher than that of G. The question then becomes a matter of *how many* are necessary, which I leave as an open question.

2.9 (Appendix) Average L_1-distance between random user-pairs on a square lattice

The number of ways N, to pick two distinct vertices in an $n \times n$ grid lattice is equivalent to the number of ways to pick two distinct pairs of positive integers no greater than n:

$$N \equiv n^2(n^2 - 1) \tag{2.47}$$

The average L_1 distance between two distinct vertices is therefore:

$$\langle L_1 \rangle = \frac{\sum_{x_1,x_2,y_1,y_2}^{n} |x_1 - x_2| + |y_1 - y_2|}{N} \tag{2.48}$$

Where $(x_1, y_1), (x_2, y_2)$ are the coordinates of the vertices in the lattice. Expanding the series to separate x and y:

$$\langle L_1 \rangle = \frac{n^2(\sum_{x_1,x_2}^{n} |x_1 - x_2| + \sum_{y_1,y_2}^{n} |y_1 - y_2|)}{N} \tag{2.49}$$

$$= \frac{n^2(2\Delta)}{N} \tag{2.50}$$

Where

$$\Delta \equiv \sum_{x_1,x_2}^{n} |x_1 - x_2| = \sum_{y_1,y_2}^{n} |y_1 - y_2| \tag{2.51}$$

Let $\Delta = \Delta^+ + \Delta^-$ where Δ^+ contains the terms such that $x_2 < x_1$, and Δ^- the ones

where $x_1 < x_2$. By symmetry we see that, $\Delta^+ = \Delta^-$. Then,

$$\begin{aligned}\Delta = 2\Delta^+ &= 2\sum_{x_1=1}^{n-1}\sum_{x_2=x_1+1}^{n}(x_2 - x_1) \\ &= \frac{n(n^2-1)}{3}.\end{aligned} \tag{2.52}$$

By substitution, we find that

$$\langle L_1 \rangle = \frac{2}{3}n \tag{2.53}$$

Bibliography

[1] Amir R. Arab. On diagonal quantum channels. *Reports on Mathematical Physics*, 88(1):59–72, August 2021.

[2] Albert-László Barabási. Scale-free networks: A decade and beyond. *Science*, 325(5939):412–413, July 2009.

[3] Charles H. Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K. Wootters. Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. *Physical Review Letters*, 70:1895, 1993.

[4] Charles H. Bennett, Gilles Brassard, Sandu Popescu, Benjamin Schumacher, John A. Smolin, and William K. Wootters. Purification of noisy entanglement and faithful teleportation via noisy channels. *Physical Review Letters*, 76:722, 1996.

[5] Charles H. Bennett, David P. DiVincenzo, John A. Smolin, and William K. Wootters. Mixed-state entanglement and quantum error correction. *Physical Review A*, 54(5):3824–3851, November 1996.

[6] Anna D. Broido and Aaron Clauset. Scale-free networks are rare. *Nature Communications*, 10(1), March 2019.

[7] Alena Chang and Guoliang Xue. Order matters: On the impact of swapping order on an entanglement path in a quantum network. In *IEEE INFOCOM 2022*

- *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–6, 2022.

[8] Tim Coopmans, Robert Knegjens, Axel Dahlberg, David Maier, Loek Nijsten, Julio Oliveira, Martijn Papendrecht, Julian Rabbie, Filip Rozpędek, Matthew Skrzypczyk, Leon Wubben, Walter de Jong, Damian Podareanu, Ariana Torres Knoop, David Elkouss, and Stephanie Wehner. Netsquid, a discrete-event simulation platform for quantum networks, 2020.

[9] Axel Dahlberg and Stephanie Wehner. Simulaqron – a simulator for developing quantum internet software. *Quantum Science & Technology*, 4:015001, 2018.

[10] I. Devetak. The private classical capacity and quantum capacity of a quantum channel, 2004.

[11] Simon J. Devitt, Andrew D. Greentree, Ashley M. Stephens, and Rodney Van Meter. High-speed quantum networking by ship. 2014.

[12] W Dur and H J Briegel. Entanglement purification and quantum error correction. *Reports on Progress in Physics*, 70(8):1381–1424, jul 2007.

[13] Alexei Gilchrist, Nathan K. Langford, and Michael A. Nielsen. Distance measures to compare real and ideal quantum processes. *Phys. Rev. A*, 71:062310, Jun 2005.

[14] Kenneth Goodenough, David Elkouss, and Stephanie Wehner. Optimizing repeater schemes for the quantum internet. *Phys. Rev. A*, 103:032610, Mar 2021.

[15] Jim Gray, Wyman Chong, Tom Barclay, Alex Szalay, and Jan vandenBerg. Terascale sneakernet: Using inexpensive disks for backup, archiving, and data exchange, 2002.

[16] Pawel Horodecki, Michal Horodecki, and Ryszard Horodecki. General teleportation channel, singlet fraction and quasi-distillation, 1999.

[17] Liang Jiang, Jacob M. Taylor, Navin Khaneja, and Mikhail D. Lukin. Optimal approach to quantum communication using dynamic programming. *Proceedings of the National Academy of Sciences*, 104(44):17291–17296, October 2007.

[18] Takaaki Matsuo. Simulation of a dynamic, ruleset-based quantum network, 2019.

[19] Jian-Wei Pan, Dik Bouwmeester, Harald Weinfurter, and Anton Zeilinger. Experimental entanglement swapping: Entangling photons that never interacted. *Phys. Rev. Lett.*, 80:3891–3894, May 1998.

[20] Jian-Wei Pan, Dik Bouwmeester, Harald Weinfurter, and Anton Zeilinger. Experimental entanglement swapping: Entangling photons that never interacted. *Physical Review Letters*, 80:3891, 1998.

[21] Nicolas Sangouard, Christoph Simon, Hugues de Riedmatten, and Nicolas Gisin. Quantum repeaters based on atomic ensembles and linear optics. *Rev. Mod. Phys.*, 83:33–80, Mar 2011.

[22] Marco Tomamichel, Mario Berta, and Joseph M. Renes. Quantum coding with finite resources. *Nature Communications*, 7(1), May 2016.

[23] Mark M. Wilde. *Quantum Information Theory*. Cambridge University Press, April 2013.

[24] Juan Yin, Yuan Cao, Yu-Huai Li, Sheng-Kai Liao, Liang Zhang, Ji-Gang Ren, Wen-Qi Cai, Wei-Yue Liu, Bo Li, Hui Dai, Guang-Bing Li, Qi-Ming Lu, Yun-Hong Gong, Yu Xu, Shuang-Lin Li, Feng-Zhi Li, Ya-Yun Yin, Zi-Qing Jiang, Ming Li, Jian-Jun Jia, Ge Ren, Dong He, Yi-Lin Zhou, Xiao-Xiang Zhang, Na Wang, Xiang Chang, Zhen-Cai Zhu, Nai-Le Liu, Yu-Ao Chen, Chao-Yang Lu, Rong Shu, Cheng-Zhi Peng, Jian-Yu Wang, and Jian-Wei Pan. Satellite-based entanglement distribution over 1200 kilometers. *Science*, 356(6343):1140–1144, 2017.

[25] Sandra Zajac and Sandra Huber. Objectives and methods in multi-objective routing problems: a survey and classification scheme. *European Journal of Operational Research*, 290(1):1–25, April 2021.

Chapter 3

Resource Estimation for Satellite Networks

> And he brought him outside and said "Look towards heaven and number the stars if you are able to number them." Then he said to them "So shall your descendants be."
>
> Genesis 15:5

3.1 Statement of work

In this chapter, I present upper bounds for the attainable rates at which maximally entangled two qubit states can be established between distant *surface codes* when supplied by a quantum satellite. To this end, I develop a closed form expression relating the *satellite power* to this pair production rate. At various points, I simplify the analysis by making generous assumptions that inflate the attainable pair generation rates. (hence, why this work is about upper-bounds). These assumptions include: lossless photon capture at the ground stations, purely dephasing noise (which simplifies purification), ideal weather conditions, and 100% allocation of satellite power for pair production. All work presented here is original (unless explicitly indicated), and was conducted under the supervision of Dr. Simon Devitt and Dr. Peter Rhode. I thank my collaborator S. Srikara for helping me to develop the arguments of sections 3.6.1 and 3.7, and for compiling the survey of commercial satellites presented in Table 3.2.

For the sake of transparency, I note that Srikara completed a *related* project that was motivated in part by the results I present here. Specifically, he conducted a numerical analysis to determine the feasibility of *up-link* entanglement distribution. In this scenario, entangled photons are generated on the ground and sent up to a satellite where a swapping operation takes place to extend the entanglement between distant ground stations. This is the opposite of the *down-link* case that we consider in this project, where entangled pairs of photons are transmitted *from* an overhead satellite.

3.2 Introduction

In our first chapter, we established that the Hilbert space of a quantum state grows exponentially with the number of particles in the ensemble. This suggests that a *cluster* of quantum computers has more computational power than the *sum of its parts*. If this is the case in practice, there is a great incentive to *network* quantum computers for *distributed quantum computation* (DQC) [6]. For this to be possible, a high rate of entangled pairs must be supplied between modules so the *teleportation* of states (or multi-qubit gates) may be performed. In the setting of this chapter, we imagine that a cluster consists of two or more *surface-code* quantum computers where each module is separated by *continental* distances.

Although fibre-optic repeater networks can in principle distribute entanglement *arbitrarily far* [27], the high attenuation rate of fibre-optic cables limits the distance that *repeater stations* may be separated. With current technology, it is expected that such stations will be required every 30 to 100 kilometers [11, 16, 19, 24, 1, 23, 9]. This is likely to be prohibitively expensive, especially for DQC at continental scales.

A *more viable* alternative may be use to use *quantum satellites* which generate entangled pairs of photons with an on-board SPDC source and beam the particles to distant ground stations. A growing body of theoretical and experimental research indicates that such satellites are viable for *quantum key distribution* [5, 17]. Notably, the *Micius* or *Mozi* satellite (launched in 2016) is able to distribute entangled pairs over distances of 1200km at a rate around one kilohertz [32].

Our objective for this work is to determine upper bounds for the rates at which maximally entangled $|\phi^+\rangle$ states can be established between distant *surface code* qubits.

Equivalently, this is the rate that single qubits (or two qubit gates) can teleported between modules. Due to the difficulty of establishing entanglement (particularly *encoded entanglement* between surface codes), this is likely to be the *rate limiting factor* (and hence the clock speed) of the distributed quantum computer.

A simplified diagram of our protocol is illustrated in figure 3.1. An overhead satellite distributes entangled pairs between ground stations that are separated by *statewide*, *continental* or *transcontinental* distances. The surviving pairs are recursively purified until they are suitable resources for a *lattice surgery* operation. After repeating lattice surgery a number of times to ensure fault-tolerance, the logical state $|\phi^+\rangle_L$ is established between the two surface codes.

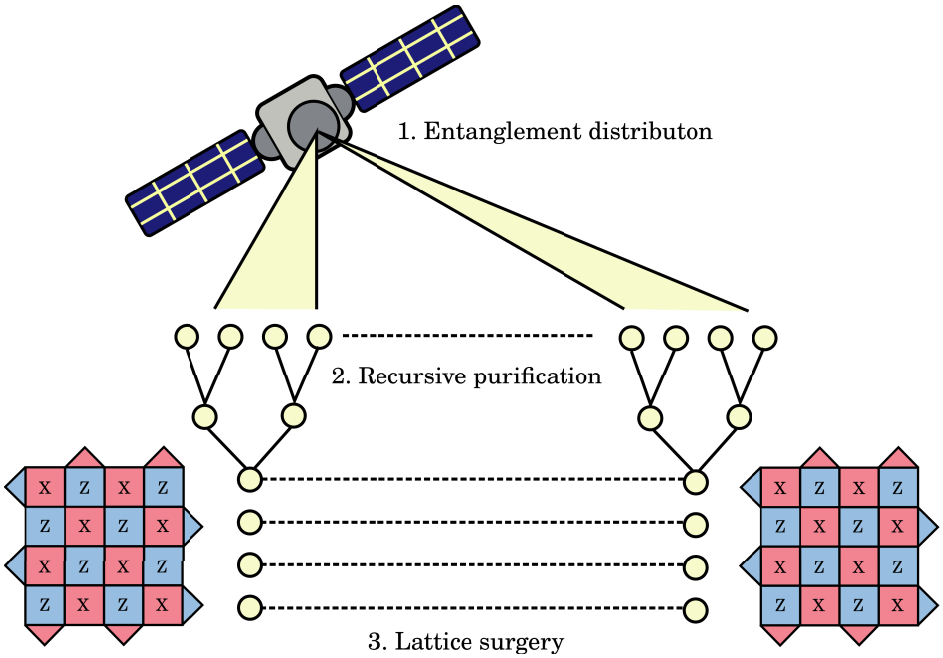

Figure 3.1: A basic schematic of our protocol: An overhead satellite distributes entangled photon pairs between two distant ground stations. We assume an average double down-link attenuation η due to atmospheric and free-space effects with ideal weather conditions, and we assume perfect photon capture, and conversion at the ground stations. To compensate for entanglement degradation, a recursive purification protocol is used to refine χ pairs into a single pair of suitable quality. Finally, the purified pair is used to implement a fault-tolerant lattice surgery operation to create a logical Bell pair between the code patches.

3.3 Logical pairs through lattice surgery

The *lattice surgery* operation (See section 1.8.1) can be used to prepare a *maximally entangled pair* between two surface code qubits. To implement it, we begin with the initial state $|0+\rangle_L$, which we assume has been prepared in advance. Next we perform a *syndrome extraction cycle* (stabilizer check) over the two patches as if they were one elongated patch. In the event where the patches are not *spatially* adjacent, d maximally entangled pairs can be expended to implement this operation. This affects a ZZ parity check between the two qubits, resulting in the state $|\phi^+\rangle$ or $|\psi^+\rangle$ depending on whether a $+1$ or a -1 eigenvalue was measured respectively. In the absence of noise, this would ideally be done in a single code cycle. In practice however, the lattice surgery operation needs to be repeated for d cycles, since the measurements that implement the gate teleportation could introduce *undetected errors*. A simple diagram of this process is presented in figure 3.2.

When entangled pairs are used to implement lattice surgery between distant surface codes, any single qubit errors that the pairs have suffered will be *introduced* into the physical qubits of the code. Consequently, the rate at which entangled pairs suffer errors must not exceed the *error threshold* of the code [1]. From section 1.9.1, we know that any two qubit state may take the form

$$\rho := \sum_{i,j=1}^{4} \chi_{i,j} \Big(I \otimes \sigma_i \Big) |\phi^+\rangle\langle\phi^+| \Big(I \otimes \sigma_j \Big) \tag{3.1}$$

Where $\sigma_i, \sigma_j \in \{I, X, Y, Z\}$. We recall that the *off-diagonal* terms of this mixture represent the *non-observable* error contributions and may be ignored for the sake of simplicity. What we are left with then is a probabilistic mixture of the target state $|\phi^+\rangle$ with three error terms.

$$\rho = \chi_{1,1} |\phi^+\rangle\langle\phi^+| + \chi_{2,2} |\phi^-\rangle\langle\phi^-| + \chi_{3,3} |\psi^+\rangle\langle\psi^+| + \chi_{4,4} |\psi^-\rangle\langle\psi^+| \tag{3.2}$$

The fidelity $F(\rho)$ of this mixed state with respect to the target pair $|\phi^+\rangle$ is easily seen to be $\chi_{1,1}$, which is the probability that ρ is found to be without any error. If ϵ is the

[1] Interestingly, one year after I conducted this resource analysis, Ramette et. al. presented numerical evidence suggesting that fault-tolerant lattice surgery can be performed using pairs with up to an order of magnitude *more* error than the code threshold [26]. This insight will be revisited in greater detail in the subsequent chapter.

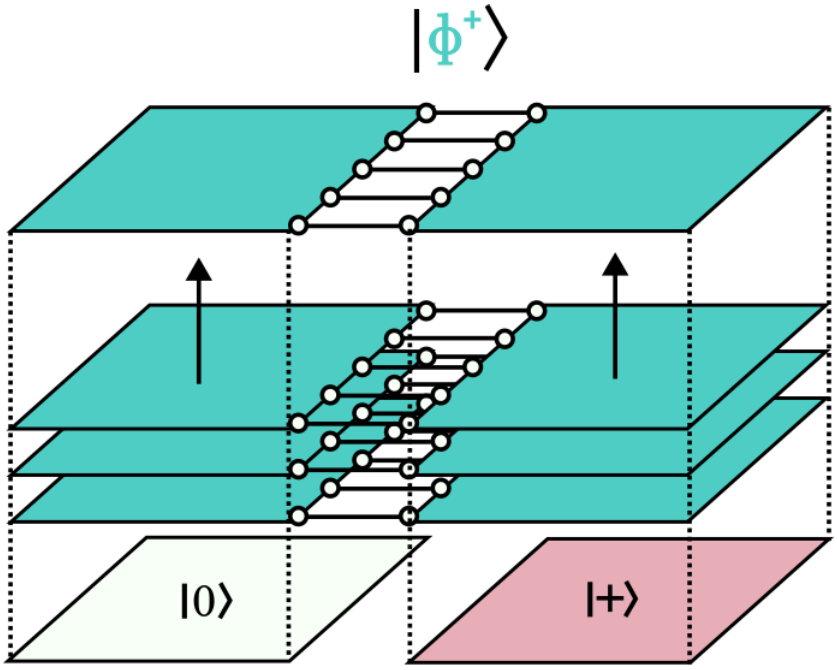

Figure 3.2: A space-time diagram demonstrating how a logical Bell state is prepared between two surface codes. Time flows from the bottom of the diagram to the top. Each square slice represents a surface code at a different instant in time. At the start of the protocol, the state is initialised to $|0\rangle \otimes |+\rangle$. Following this, a total of d remote lattice surgery operations are performed using d entangled pairs at each time. After the final surgery, the surface codes are split (not pictured) and the resulting state is (up to a possible local correction) $|\phi^+\rangle$

threshold of the surface code, we say that ρ is *suitable* for lattice surgery if the *infidelity* does not exceed ϵ. In other words

$$1 - F(\rho) \leq \epsilon \tag{3.3}$$

For this analysis, we assume an optimistic code threshold of $\epsilon = 0.01$, in which case we require a pair fidelity of at least $F_{\text{id}} = 0.999$.

3.4 Choosing code distance

Since each round of lattice surgery requires d pairs (where d is the distance of the surface code), our resource analysis will inevitably depend on our choice of *code distance*. Higher distances result in more robust codes, which consequently enable longer computations. Our objective then is to choose a code distance that is *just large enough* for a suitably complex calculation that would *warrant* the infrastructure needed for a distributed

quantum computation. The program that we selected based on this criteria is an implementation of *Shor's factorization algorithm* by *Beauregard* et. al. for *RSA encryption breaking* [2].

To briefly summarise, *RSA* (named after the initials of its inventors) is a widely used cryptography protocol that enables two parties *Alice and Bob* to safely communicate over a *public channel* in such a way that a third party *Eve* cannot interpret their messages. First, Alice transmits a number to Bob that is called a *public key*. Using this public key, Bob encodes his message in a way that is *virtually impossible* for Eve to determine what the original message was (except by guesswork). Bob then transmits the encoding to Alice who uses a *private key* (known only by her) to decipher the message. Crucially, the hardness of RSA is based on an assumption that a sufficiently large composite number cannot be *quickly factorized* into its primes. This premise was proved false with the discovery of *Shor's algorithm* which is *exponentially faster* than any known classical algorithm. Consequently, RSA encryption is now considered *compromised* since encoded messages that were once *thought* to be safe are now potentially vulnerable with the advent of a sufficiently powerful quantum computer.

Ha et. al. report that the Beauregard algorithm can tolerate *logical errors* at a rate of $P_L = 4.28 \times 10^{-21}$ when factoring 2048 bit integers [15]. Given this P_L, we can now work backwards to determine the minimum code distance d. To start, let P_C be the probability that a surface code suffers an undetectable *logical error* during a *syndrome extraction cycle* (i.e. stabilizer check). This P_C can be related to the surface code distance d using a formula provided by Devitt et. al. [8]

$$P_C = \alpha(\beta p)^{\frac{d+1}{2}} \tag{3.4}$$

Here, p is the *unbiased* error rate of the physical qubits, while α and β are experimental parameters. Based on numerical data, Devitt et. al. propose the parameter values $\alpha = 0.3$ and $\beta = 70$, which we adopt as well. For the Bell state preparation [2] to succeed, we require that *every* code cycle succeeds (of which there are $2 \times d$). Assuming that P_C is small, we can approximate the *overall success probability* of the Bell preparation by dropping the higher order terms:

[2] as described in section 3.3

$$(1 - P_C)^{2d} \approx (1 - 2d\, P_C) \tag{3.5}$$

Consequently, the *overall failure rate* of pair production as a function of code distance is given:

$$2d\alpha(\beta p)^{\frac{d+1}{2}} \tag{3.6}$$

If we set P_L to be this overall failure rate, then our objective now is to solve for d in the expression $P_L = 2d\alpha(\beta p)^{\frac{d+1}{2}}$. Assuming an unbiased error rate of $p = 0.001$, we find that a code distance of $d \approx 34$ is sufficient for implementing this factorization.

For the sake of transparency however, I would like to point out that the code distance we used in our publication with PRR is $d = 37$. The reason for this discrepancy is that I mistakenly thought that *four* surface code cycles needed to succeed instead of just two; At the time, I did not realise that the initial $|0\rangle \otimes |+\rangle$ logical state could be prepared using *transversal* operations in a single (neglible) timestep. The code distance used throughout this chapter is therefore $d = 34$. Let us pause to evaluate the significance of this difference on our final result. Skipping ahead slightly, we see from equation 3.20 that the rate at which a satellite needs to produce entangled pairs is proportional to d^2. For $d = 37$ and $d = 34$, we see that the corresponding d^2 values are 1369 and 1156 respectively. This indicates a 16.6% decrease in the required pair generation rate. Since power is proportional to this generation rate (see eq. 3.22), we therefore expect a 16.6% decrease in the required satellite power from what we previously reported in PRR. Although this percentage difference sounds like a lot on paper, in practice, it doesn't substantively change the conclusion of this project [3].

3.5 Calculating purification overhead

An entangled pair distributed through a noisy channel naturally loses some of its entanglement through *decoherence*. This is also true for an entangled pair of photons that

[3] If the reader is curious to know specifics, the estimates for the attainable logical pair generation rates (see table 3.3) changed in the following way: Statewide distances: 2×10^6 Hz $\rightarrow 2 \times 10^6$ Hz (No change within rounding). Continental distances: 1×10^4 Hz $\rightarrow 2 \times 10^4$ Hz. Transcontinental distances: $6 \times 10^2 \rightarrow 7 \times 10^2$ Hz

pass through the *atmosphere*. As discussed in section 3.3, these errors must be corrected before the pairs can be used for lattice surgery. This is done with a *purification protocol* which takes a number of low-fidelity pairs and (using local operations) produces a higher fidelity pair with some overall success probability [10].

Let us define the *purification factor* χ as the number of initial (non-ideal) pairs that are required to generate *one* ideal pair with a confidence of S. In other words, for every group of χ low quality pairs, we expect to produce one high quality pair with probability S. In general, this purification factor will depend on the *combined* free-space channel $\mathcal{E}$ [4] over which the pair is distributed, the required output fidelity F_{id}, the required success rate S, and the choice of purification protocol.

Naturally, we expect the channel $\mathcal{E}$ to *change* as the satellite tracks overhead, since the photon trajectories must be adjusted. We simplify by assuming that every distributed pair ρ_{in} is *identical*. In general, the amount of noise that ρ_{in} has suffered will depend on *how* the entanglement is *encoded*. There are various physical degrees of freedom available to photons that *can* be entangled such as polarization [18], time-bin [29], spatial mode, and orbital angular momentum (OAM) [21] to name a few. Belencia et. al. indicate that spatial mode and OAM encodings are not suitable for quantum satellite communications as they are highly susceptible to atmospheric turbulence [3]. Rather, they express a preference for *polarization* or *frequency* encodings citing *higher robustness* as the reason. In the case of polarization, corroborating theoretical work from Bonato et. al. appears to support this claim [5].

For the sake of argument then, we assume (based on numerical data from the *Mozi* project [32]) that our satellite distributes *polarization encoded* entangled pairs that arrive at the ground stations with an overall fidelity of $F = 0.87$. As there is *no tomographical data* available for these states, we make the generous assumption that the pair has suffered *completely biased* noise. Specifically, we define ρ_{in} to be

$$\rho_{\text{in}} = 0.87\,|\phi^+\rangle\langle\phi^+| + 0.13\,|\psi^+\rangle\langle\psi^+| \tag{3.7}$$

A second (and more important) reason for considering biased noise is that it *greatly simplifies* our subsequent analysis of *entanglement purification*. This is because biased

[4] The combined channel is $\mathcal{E} := \mathcal{E}_1 \otimes \mathcal{E}_2$ where $\mathcal{E}_1$, and $\mathcal{E}_2$ are the single qubit channels for *either arm* of the path from satellite to ground.

errors only propagate in *one direction* under the action of an entangling gate [12]. For example, consider that a single qubit X error can only travel *down* a CNOT (from control to target) while a Z error can only travel up. With biased noise, we have only one type of error and consequently, errors can only *flow* in one direction.

For our purification, we use recursive instances of the *parity-check* protocol by Bennett et. al. [4], which is notable for its short depth circuit and high yield in the case of biased noise. Each *round* of this algorithm takes two pairs of the form:

$$F|\phi^+\rangle\langle\phi^+| + (1-F)|\psi^+\rangle\langle\psi^+| \tag{3.8}$$

And, with probability $F^2 + (1-F)^2$, returns a pair of the *same form* but with an improved fidelity of: [5]

$$f(F) = \frac{F^2}{F^2 + (1-F)^2} \tag{3.9}$$

This of course assumes that our purification circuits are themselves implemented without noise. It is easily verified that for an initial fidelity of $F = 0.87$, and a *minimum* target fidelity of $F_{\text{id}} = 0.999$, *two rounds* of the parity check purification (using a total of four pairs) are sufficient to obtain a *single* pair with an overall success probability $P = 0.573$. If the purification fails however, *all* or *part* of the input entanglement is lost. In order to compensate for this, we *multiplex* our purification circuits by running multiple instances of it in parallel. This way, we can improve our *overall* confidence of obtaining at least one pair at the expense of potential redundancy.

For the sake of transparency, I will pause for a moment to describe a *nuance* that I failed to consider while I was conducting this resource estimation. The issue is that some entanglement may be salvaged depending on *what step* of the purification has failed. Specifically, if a purification fails in the *first round* of our protocol, the other *surviving pair* can (in principle) be rerouted to another module to be used as a resource for the *second round* of purification. In the analysis that I present below however, I treat *each* purification circuit as a *black box* that can only either succeed with probability P or fail with probability $1-P$. This may be fine if we assume that routing entanglement between modules is *expensive* in some way and therefore not worth the trouble.

[5] Strictly speaking, we only an see improvement if the initial fidelity F is greater than $\frac{1}{2}$.

Returning to our work, let $\mathcal{B}(P,k)$ be a binomial distribution where P is the success probability of our purification and where k is a number of trials. Let $\Phi(\mathcal{B},x)$ denote the *cumulative distribution function* of this binomial up to $x \leq k$ successful events. The probability $\mathcal{P}_1$ that *at least* one purification event is successful is then

$$\mathcal{P}_1 := 1 - \Phi\Big(\mathcal{B}(P,k), 1\Big) \tag{3.10}$$

Consequently, the *minimum* number of purification modules K that are needed to produce *at least* one high-quality pair with a confidence of S is

$$K \equiv \min_k(\mathcal{P}_1 \geq S) \tag{3.11}$$

For each of the K purification modules, we require 4 low-quality pairs since we use two rounds of the parity check purification per module. Consequently, the purification factor χ is equal to $4K$. Setting our confidence to an arbitrarily high value of $S = 0.999$, we can use equation 3.11 to determine that $K = 9$ and therefore $\chi = 36$. This means that for a code of distance d, we require $n_{\text{g}} := d\chi$ pairs to be established on the ground for a lattice surgery operation to *likely* succeed.

In hindsight, I note that my chosen S value is perhaps *too small*. This is because for a distance $d = 37$ code, the probability of purifying d pairs is S^d, which is calculated to be around 0.964. As a result, we see there is an *insufficient amount* of entanglement for lattice surgery around 4.6% percent of the time. When this happens, the surface codes will need to wait an *additional round* to try and collect the necessary entanglement. Although this is only a *marginal increase* in the overall preparation time, it is presently unclear to me how significant these *gaps* are with respect the *fault-tolerance* of the state preparation. I presume that this is *not* a significant issue.

3.6 Incorporating attenuation

There is a significant chance that *one* or *both* photons transmitted from a satellite will fail to reach their destination. Let the *double attenuation rate* η be the probability that *both* photons of an entangled pair are successfully transmitted. Our objective now is to determine the *number* of pairs n_s the satellite must generate in order to establish

the n_g pairs required on the ground. As before, we suppose this must succeed with an arbitrarily high confidence of $S = 0.999$ [6]. Following this, we present estimates for η at various distances based on numerical data available from the literature.

3.6.1 Logical pair generation rate

The number of pairs x that reach the ground after k attempts is a random variable that follows a binomial distribution, however since k is assumed to be very large, we can approximate this as a *normal distribution* with a mean of $k\eta$ and a variance of $k\eta(1-\eta)$.

$$x \sim \mathcal{N}\Big(k\eta,\ k\eta(1-\eta)\Big) \tag{3.12}$$

Similar to our previous calculations, the probability $\mathcal{P}_{n_g}$ that at least n_g pairs are established with k attempts is

$$\mathcal{P}_{n_g} \equiv 1 - \Phi\Big(\mathcal{N}, n_g\Big) \tag{3.13}$$

The number of pairs the satellite needs to distribute is therefore the *minimum* value of k such that the probability of exceeding n_g many successes is *greater than* or *equal* to the confidence threshold S:

$$n_s = \min_k \Big(\mathcal{P}_{n_g} \geq S\Big) \tag{3.14}$$

A further simplification is possible with *Markov's inequality*, which gives an upper bound for the probability that a random variable X of some distribution is greater than a constant a

$$P(X \geq a) \leq \frac{E(X)}{a} \tag{3.15}$$

Let us use this inequality to *bound* the probability that our random variable x meets or *exceeds* the required number of pairs n_g. In other words, our intention is to use the above expression to find a bound for $P(x \geq n_g)$.

[6] Unlike in our previous calculation, we will soon demonstrate that this choice of S does not matter particularly much

$$P(x \geq n_g) \leq \frac{E(x)}{n_g} \tag{3.16}$$

The *expectation value* of x (or the average number of pairs that make the journey) is easily seen to be $\eta\, n_s$. Additionally, note that $P(x \geq n_g) = \frac{E(x)}{n_g}$ is the same thing *by definition* as our confidence threshold S. Performing these substitutions, we find that:

$$S \leq \frac{\eta\, n_s}{n_g} \tag{3.17}$$

And since $S = 1 - \epsilon$ where ϵ is small, we can perform the following approximation

$$1 \approx \frac{\eta\, n_s}{n_g} \tag{3.18}$$

Which (rearranging for n_s and substituting $n_g = d\chi$) gives us

$$n_s = \frac{d\chi}{\eta} \tag{3.19}$$

Which is the *approximate* number of pairs that the satellite needs to distribute for *one round* of lattice surgery to succeed.

Let R_S be the *rate* at which our satellite produces entangled pairs and let R_L be the rate at which *logical pairs* are established between our surface codes. Since $\frac{d^2\chi}{\eta}$ pairs must be generated by the satellite for the d lattice surgery operations needed for the *fault-tolerant lattice surgery*, we see that these rates are related to each other in the following way:

$$R_S = \frac{d^2\chi}{\eta} R_L \tag{3.20}$$

3.6.2 Estimating η

Having established the relationship between R_S and R_L, I now focus on choosing a suitable average-case value for η. Bonato et. al. modeled η in the context of quantum key distribution (QKD) [5], as did Mazzarella et. al. [22] and Khatri et. al. [17]. The latter results are especially relevant to us, as Khatri et. al. conducted extensive numerical simulations to determine (among various other things) average η values between major cities

for a large quantum satellite *network*. To this end, they first designed and optimized a 400 satellite constellation with the objective of balancing coverage and dollar-cost. They then ran this constellation over a period of 24 simulation hours with ideal weather conditions and a standard day-night cycle. We elected to use their three most optimistic attenuation rates for a selection of *state*, *continental* and *transcontinental* distance categories (the values of which are presented in table 3.1).

Classification	City pairs	Average loss (dB)
State (500-999 km)	Toronto - New York City	45.1
Continental (1000-4999 km)	Sydney - Auckland	65.6
Transcontinental (5000 km+)	New York City - London	79.1

Table 3.1: The most optimistic (average) double attenuation rates for three city pairs simulated by Khatri et. al. with their proposed satellite network [17].

The primary factor that contributes to down-link attenuation is *beam widening*. This happens naturally as the beam propagates through a vacuum, but is exacerbated by atmospheric diffraction. We might intuitively expect that the atmospheric effects contribute more to the *overall* attenuation than the free-space effects. Counter-intuitively however, Khatri et. al. present evidence that *free-space widening* is the more significant contribution (at least for their constellation). This is because the path segments where the atmospheric effects occur are *much shorter* than the free-space parts.

Another potentially relevant effect is *beam wandering*, where the middle of the photon's Gaussian wave-packet shifts by some amount. This is caused by the particles of the atmosphere *scattering* the beam. In this *downlink* case where photons are being sent *down* from a satellite, this effect turns out to be negligible, as a shifting profile doesn't matter so much when it takes place towards the *end* of a transmission.

3.7 Required Pair Generation Rate and Satellite Power

In this section, we relate R_L (the rate at which logical pairs are established) to the available satellite power P_s. The rate at which a satellite can generate *physical* pairs (R_s) depends on the brightness of the SPDC source (N_p), the power consumption of each source (P_r), and the power available to the satellite (P_s). (Here, we assume that all power is exclusively allocated to pair production).

Satellite Name	Power	Budget (10 million USD)
GSAT-11	13.6 kW	7.43
GSAT-31	4.7 kW	6.46
GSAT-7A	3.3 kW	6.32-10.11
GSAT-29	4.6 kW	2.08
GSAT-30	6 kW	6.46

Table 3.2: A survey of Indian communication satellites launched between 2018 and 2019 with power ratings and costs

$$R_s = \frac{P_s N_p}{P_r} \tag{3.21}$$

Rearranging gives us the required satellite power for a particular generation rate.

$$P_s = \frac{R_s P_r}{N_p} \tag{3.22}$$

Finally, we substitute P_s with its definition in equation 3.20 to obtain an expression relating the satellite power to the rate of logical pair production.

$$P_s = \frac{d^2 R_L \, \chi P_r}{N_p \eta} \tag{3.23}$$

To determine a realistic estimate for a maximum satellite power, we present a short survey of Indian communication satellites in table 3.2. The most powerful of these, the *GSAT-11*, has a considerably larger power rating than the others at a comparable budget to the others. Based on this data, we thereofre conclude that a satellite with a power rating of 10 kW (is on the order of) the most powerful commercial satellite possible with current technology.

The brightest available Bell-pair source reported at the time of writing is a waveguide integrated AlGaAson micro-resonator [31]. Its high output combined with a form factor on the order of micrometers makes it a promising candidate as a satellite-based entanglement source. From the experimental data, the highest reported production rate was 4×10^6 pairs per second at a power of $15\mu W$. According to the report, increasing the power beyond this point would "exceed the lasing threshold of the micro-resonator" which consequently would reduce the "overall entanglement visibility." With this information, we set N_P and P_r to the aforementioned values of brightness and power per source respectively.

Architecture	Average Gate Time	Rate
Superconducting qubits [25]	50 ns	2×10^7 Hz
NV Diamond [7]	0.05 μs	2×10^7 Hz
Ion trap [28]	1.6 μs	6.25×10^5 Hz
NMR Spins [20]	$1ms$	1×10^3 Hz

Distance Category	Rate
State	2×10^6
Continental	2×10^4
Transcontinental	7×10^2

Table 3.3: Sample of average gate times for common qubit architectures

3.8 Results and Discussion

Our numerical data is presented in figure 3.3. Here, we have plotted the rate that logical Bell pairs can be generated versus the required satellite power (eq. 3.23) for *state*, *continental* and *transcontinental* distances (Table 3.1). The dashed vertical line indicates the *approximate maximum power* of a commercial satellite. The *fastest possible* production rates for each distance are found at the points where the three curves intersect this vertical.

These maximum values are summarised in table 3.3. For state distances, the maximum rate is around 2×10^6 s^{-1}. For continental distances this is around 2×10^4 s^{-1}, and for transcontinental, around 7×10^2 s^{-1}. To contextualise this data, I also present in the table a selection of *average gate times* for some common qubit architectures. Here we see that the maximum achievable rate at the statewide distance is *comparable* to the operating speed of a trapped-ion quantum computer. Continental and transcontinental DQCs however are considerably slower and are more akin to an *NMR-spin* computer.

Although the attainable rates for our continental and transcontinental distances seem somewhat limited, our estimates indicate that distributed quantum computing may be feasible at statewide distances. The pressing issue however is that the rates we propose are *highly optimistic* upper bounds based on various generous assumptions. This suggests long-term issues for scaling satellite-based DQC.

Let us now analyse the ways we might improve pair production. Our options are to increase the throughput of the satellites, decrease the *required* pair generation rate, improve the efficiency of our purification, or reduce the *relative* attenuation of the down-link channel. In the first case, the *only* possibilities are to increase the satellite power

or to improve the brightness of the photon pair source. It is unlikely that the power available to a commercial satellite will dramatically increase, though the brightness of the entanglement sources may improve over time. From equation 3.20 we see that the code distance $d = 34$ contributes around *three orders of magnitude* to the required satellite power. Since this is the *minimum* distance required to run the factorization algorithm discussed in 3.4, it is impossible for this value to be brought any lower. That said, an alternative *qubit encoding* may someday have a more efficient two qubit operation that requires fewer entanglement links; For the time being, we can only speculate.

Our purification factor χ contributed around one order of magnitude, and is unlikely to be dramatically improved given the optimistic assumption made about the noise being *entirely* biased.

I thank *Craig Gidney* for the observation that it may be possible to avoid non-deterministic purification by *instead* performing error correction on multiple blocks of surface codes [7]. The main issue with this approach however is that it *lengthens* the preparation time considerably and is therefore unlikely to improve pair generation rates.

By far, the most significant contribution to the required rate of photon pairs is the *attenuation*. The optimistic average attenuations we selected from Khatri et. al. contribute between five and eight orders of magnitude to the satellite power depending on the *distance* between stations. As attenuation is primarily caused by *beam widening* in free space, there are limited options for mitigating this effect. One notable strategy is to store *half* of the pair in the satellite while the other half is transmitted. When two pairs are established between separate stations, an *entanglement swapping operation* is performed to *extend* the entanglement between the ground stations. This approach *halves* the attenuation rate, which would result in considerable improvements in pair production rate. This approach however appears infeasible with present technology, as the satellite would need to reliably store, control and measure an enormous volume of pairs.

Throughout this chapter, we made a number of generous assumptions to simplify our discussion. It is worth considering then how our analysis could be refined. Starting from the beginning, our first assumption was that photon capture is an entirely lossless process. If photon capture is known to have an average attenuation of η_c, then this loss is easily incorporated into the double down-link attenuation using the substitution

[7] See [13] for an example a *five-qubit* code implemented in this way.

$\eta \to \eta \times \eta_c^2$. Our second assumption was that the channel noise suffered by the photon pairs was purely dephasing. To improve this part of the analysis, we would ideally obtain tomographical data of the incoming states (Later, in chapter 4, we will see how a *custom purification protocol* can be tailored to help optimize entanglement yield for specific noise). This is perhaps unrealistic though since the transmission channels will change as the satellite moves. A more reasonable approach might be to consider the *worst case scenario* which is purely depolarizing noise. In this case, it is likely that the purification factor χ will need to be estimated numerically due to the mirad ways that X and Z errors can propagate. As previously mentioned, the major contributing factor in this analysis is the double down-link attenuation η. This is a difficult quantity to accurately estimate since it depends on the satellite position, time of day[8], and weather. The opinion of the author is that more refined estimates of η will need to be studied on a case-by-case basis. Finally, we were generous to assume that a given quantum satellite can use 100% of its available power to generate maximally entangled photons pairs. Moreover, we assumed these pairs would be generated using many instances of the most cutting-edge entanglement sources. These are demanding (if not impossible) engineering requirements, though it is not clear to the author how the estimates for satellite power (and therefore pair production) might be refined. Part of the challenge is that quantum satellite are a topic of on-going research which makes it difficult to anticipate what is theoretically possible. For the time being then, I leave this as an open question.

So far, we have exclusively considered a *down-link* transmission model where entanglement is generated on satellites and distributed between ground stations. The opposite case is *up-link*, where entangled pairs are prepared on the ground and distributed to a satellite that *swaps* the pairs to project the entanglement between the stations. The main advantage of this approach is that there is more power available to a ground station than a satellite. A key drawback however is that up-link *attenuation rates* are considerably higher since *beam wandering* is a more significant effect in this paradigm. Another challenge of implementing up-link communication is that photons from either ground station need to arrive at the satellite *close to simultaneously*. Despite these difficulties, a pre-print by my colleague *S. Srikara* has demonstrated a proof of concept for up-link distribution and reports pair fidelities and attenuation rates that are surprisingly

[8]More stray photons increases the value of η, since it becomes more difficult to identify the entangled photons at the ground stations.

comparable to (but still worse than) the downlink case [30].

An interesting *speculative* alternative to quantum satellites is the *quantum sneakernet* concept. Here, the idea is to entangle large surface (or other) codes and then *physically transport* the encoded qubits to their intended destinations [8]. Such communication channels have the potential to reach *extremely high bandwidths* despite low latency [14].

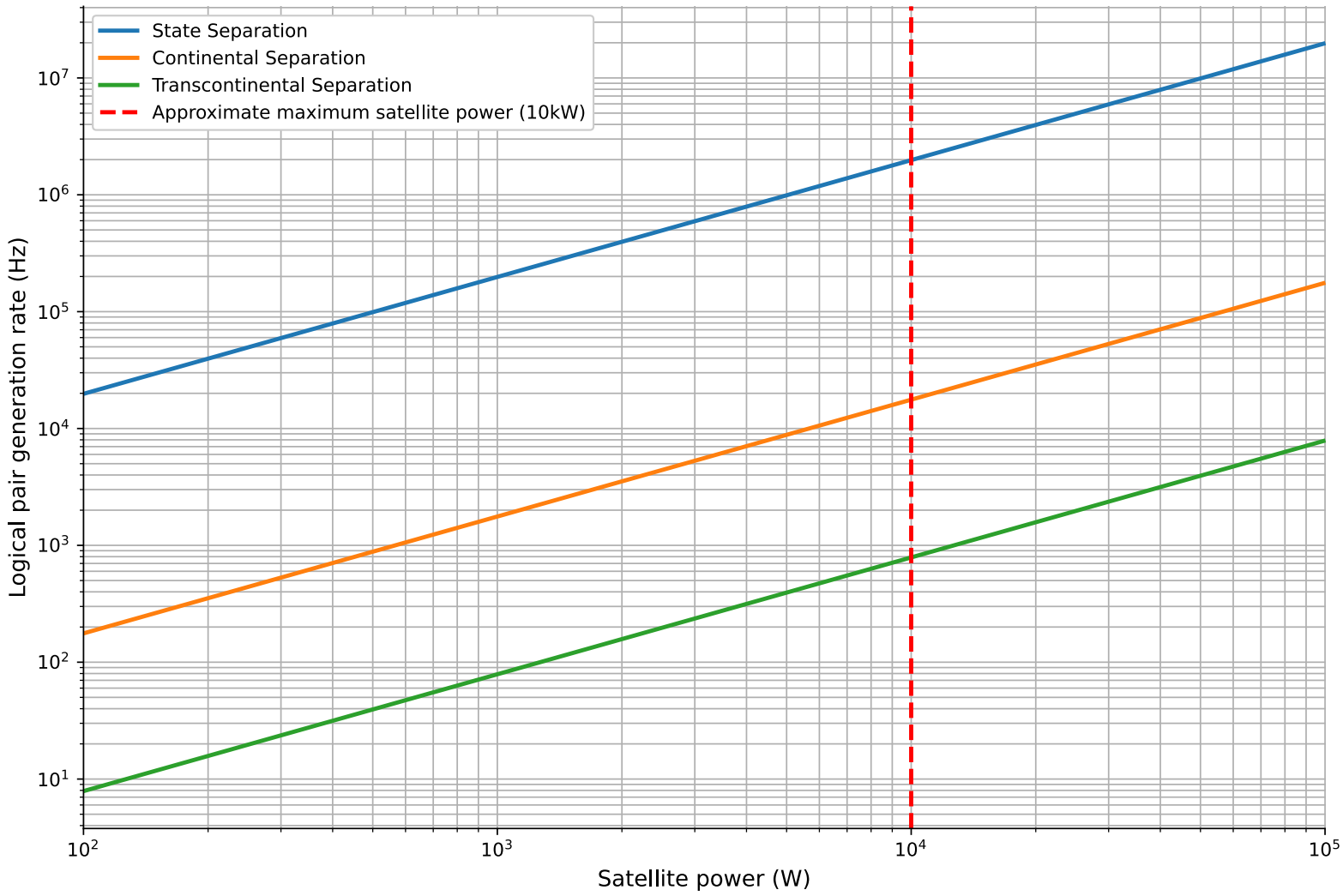

Figure 3.3: The rate at which logical surface code Bell pairs can be generated versus the available satellite power (eq. 3.23) for three different distance ratings (Table 3.1). The vertical dashed line indicates the approximate maximum power of a commercial satellite.

Bibliography

[1] Koji Azuma, Kiyoshi Tamaki, and Hoi-Kwong Lo. All-photonic quantum repeaters. *Nature communications*, 6(1):1–7, 2015.

[2] Stephane Beauregard. Circuit for shor's algorithm using 2n+3 qubits. 2002.

[3] Alessio Belenchia, Matteo Carlesso, Ömer Bayraktar, Daniele Dequal, Ivan Derkach, Giulio Gasbarri, Waldemar Herr, Ying Lia Li, Markus Rademacher, Jasminder Sidhu, Daniel K.L. Oi, Stephan T. Seidel, Rainer Kaltenbaek, Christoph Marquardt, Hendrik Ulbricht, Vladyslav C. Usenko, Lisa Wörner, André Xuereb, Mauro Paternostro, and Angelo Bassi. Quantum physics in space. *Physics Reports*, 951:1–70, 2022. Quantum Physics in Space.

[4] Charles H. Bennett, David P. DiVincenzo, John A. Smolin, and William K. Wootters. Mixed-state entanglement and quantum error correction. *Physical Review A*, 54(5):3824–3851, November 1996.

[5] C Bonato, A Tomaello, V Da Deppo, G Naletto, and P Villoresi. 11(4):045017, apr 2009.

[6] Marcello Caleffi, Michele Amoretti, Davide Ferrari, Jessica Illiano, Antonio Manzalini, and Angela Sara Cacciapuoti. Distributed quantum computing: a survey. *Computer Networks*, page 110672, 2024.

[7] Yi Chou, Shang-Yu Huang, and Hsi-Sheng Goan. Optimal control of fast and high-fidelity quantum gates with electron and nuclear spins of a nitrogen-vacancy center in diamond. *Physical Review A*, 91(5):052315, 2015.

[8] Simon J. Devitt, Andrew D. Greentree, Ashley M. Stephens, and Rodney Van Meter. High-speed quantum networking by ship. 2014.

[9] Simon J Devitt, William J Munro, and Kae Nemoto. High performance quantum computing. *arXiv preprint arXiv:0810.2444*, 2008.

[10] W Dur and H J Briegel. Entanglement purification and quantum error correction. *Reports on Progress in Physics*, 70(8):1381–1424, jul 2007.

[11] Austin G Fowler, David S Wang, Charles D Hill, Thaddeus D Ladd, Rodney Van Meter, and Lloyd CL Hollenberg. Surface code quantum communication. *Physical review letters*, 104(18):180503, 2010.

[12] Austin G. Fowler, David S. Wang, and Lloyd C. L. Hollenberg. Surface code quantum error correction incorporating accurate error propagation. *Quantum Info. Comput.*, 11(1):8–18, jan 2011.

[13] Craig Gidney. 5 qubit stim distillation circuit, 2022.

[14] Jim Gray, Wyman Chong, Tom Barclay, Alexander Szalay, and Jan vandenBerg. Terascale sneakernet: Using inexpensive disks for backup, archiving, and data exchange. *CoRR*, cs.NI/0208011, 08 2002.

[15] Jinyoung Ha, Jonghyun Lee, and Jun Heo. Resource analysis of quantum computing with noisy qubits for shor's factoring algorithms. *Quantum Information Processing*, 21, 02 2022.

[16] Liang Jiang, Jacob M Taylor, Kae Nemoto, William J Munro, Rodney Van Meter, and Mikhail D Lukin. Quantum repeater with encoding. *Physical Review A*, 79(3):032325, 2009.

[17] Sumeet Khatri, Anthony J. Brady, Renée A. Desporte, Manon P. Bart, and Jonathan P. Dowling. Spooky action at a global distance: analysis of space-based entanglement distribution for the quantum internet. *npj Quantum Information*, 7(1), 1 2021.

[18] A. Lamas-Linares, J. C. Howell, and D. Bouwmeester. Stimulated emission of polarization-entangled photons. *Nature*, 412(6850):887–890, August 2001.

[19] Ying Li, Sean D Barrett, Thomas M Stace, and Simon C Benjamin. Long range failure-tolerant entanglement distribution. *New Journal of Physics*, 15(2):023012, 2013.

[20] Dawei Lu, Aharon Brodutch, Jihyun Park, Hemant Katiyar, Tomas Jochym-O'Connor, and Raymond Laflamme. Nmr quantum information processing, 2015.

[21] Alois Mair, Alipasha Vaziri, Gregor Weihs, and Anton Zeilinger. Entanglement of the orbital angular momentum states of photons. *Nature*, 412(6844):313–316, July 2001.

[22] Luca Mazzarella, Christopher Lowe, David Lowndes, Siddarth Koduru Joshi, Steve Greenland, Doug McNeil, Cassandra Mercury, Malcolm Macdonald, John Rarity, and Daniel Kuan Li Oi. Quarc: Quantum research cubesat—a constellation for quantum communication. *Cryptography*, 4(1), 2020.

[23] WJ Munro, KA Harrison, AM Stephens, SJ Devitt, and Kae Nemoto. From quantum multiplexing to high-performance quantum networking. *Nature Photonics*, 4(11):792–796, 2010.

[24] Sreraman Muralidharan, Jungsang Kim, Norbert Lütkenhaus, Mikhail D Lukin, and Liang Jiang. Ultrafast and fault-tolerant quantum communication across long distances. *Physical review letters*, 112(25):250501, 2014.

[25] Atsushi Noguchi, Alto Osada, Shumpei Masuda, Shingo Kono, Kentaro Heya, Samuel Piotr Wolski, Hiroki Takahashi, Takanori Sugiyama, Dany Lachance-Quirion, and Yasunobu Nakamura. Fast parametric two-qubit gates with suppressed residual interaction using the second-order nonlinearity of a cubic transmon. *Physical Review A*, 102(6):062408, 2020.

[26] Joshua Ramette, Josiah Sinclair, Nikolas P. Breuckmann, and Vladan Vuletić. Fault-tolerant connection of error-corrected qubits with noisy links, 2023.

[27] Qiao Ruihong and Meng Ying. Research progress of quantum repeaters. In *Journal of Physics: Conference Series*, volume 1237, page 052032. IOP Publishing, 2019.

[28] VM Schäfer, CJ Ballance, K Thirumalai, LJ Stephenson, TG Ballance, AM Steane, and DM Lucas. Fast quantum logic gates with trapped-ion qubits. *Nature*, 555(7694):75–78, 2018.

[29] Christoph Simon and Jean-Philippe Poizat. Creating single time-bin-entangled photon pairs. *Phys. Rev. Lett.*, 94:030502, Jan 2005.

[30] S. Srikara, Hudson Leone, Alexander S. Solnstev, and Simon J. Devitt. Quantum entanglement distribution via uplink satellite channels, 2025.

[31] Trevor J. Steiner, Joshua E. Castro, Lin Chang, Quynh Dang, Weiqiang Xie, Justin Norman, John E. Bowers, and Galan Moody. Ultrabright entangled-photon-pair generation from an AlGaAs-on-insulator microring resonator. *PRX Quantum*, 2:010337, Mar 2021.

[32] Juan Yin, Yuan Cao, Yu-Huai Li, Sheng-Kai Liao, Liang Zhang, Ji-Gang Ren, Wen-Qi Cai, Wei-Yue Liu, Bo Li, Hui Dai, Guang-Bing Li, Qi-Ming Lu, Yun-Hong Gong, Yu Xu, Shuang-Lin Li, Feng-Zhi Li, Ya-Yun Yin, Zi-Qing Jiang, Ming Li, Jian-Jun Jia, Ge Ren, Dong He, Yi-Lin Zhou, Xiao-Xiang Zhang, Na Wang, Xiang Chang, Zhen-Cai Zhu, Nai-Le Liu, Yu-Ao Chen, Chao-Yang Lu, Rong Shu, Cheng-Zhi Peng, Jian-Yu Wang, and Jian-Wei Pan. Satellite-based entanglement distribution over 1200 kilometers. *Science*, 356(6343):1140–1144, 2017.

Chapter 4

Resource estimation for lattice surgery in trapped-ion systems

> You are the salt of the earth; but if salt has lost its taste, how shall its saltiness be restored? It is no longer good for anything except to be thrown out and trodden under foot by men.
>
> Matthew 5:13

4.1 Statement of work

In this chapter, I present estimates for the number of ions needed to implement fault-tolerant lattice surgery between spatially separated trapped-ion surface codes. Additionally, I determine attainable lattice surgery rates given a number of dedicated "communication ions" per logical qubit. Because this analysis depends *heavily* on the rate that syndrome extraction cycles take place, I survey the state-of-the-art and synthesise my findings to propose *three* possible cycle times that could be realistically implemented provided certain technological milestones are met.

Although this work shares a number of similarities with chapter 3, there are several key ways in which this project differentiates itself; One significant difference is that in this chapter, entanglement is established between a *finite pool* of resources. This changes how lattice surgery rates are calculated. Another (more general) difference is that the *resolution* of this work is higher; In other words, there is a greater emphasis on *realism* as

opposed to the *optimism* of the previous chapter. An example of this realism can be seen in section 4.5.4 where I use an existing genetic algorithm to find an optimal purification protocol with respect to a particular partially entangled state.

All work presented is entirely original, and was conducted under the supervision of Dr. Simon Devitt. Dr. Thinh. Le. contributed significant feedback and helped me to reword the introduction, methodology, and appendix sections of this chapter. I thank S. Srikara, Ilia Khait, David Elkouss, Stefan Krastanov, Vaishnavi Addala, and Kenneth Goodenough for other feedback and discussions.

4.2 Introduction

Trapped ions are among the best studied and most technologically mature type of qubits to date; Their long coherence times and high-fidelity gates alone justify them as a candidate qubit for scalable quantum computing [7]. How a quantum computer is scaled will depend on its underlying architecture. Broadly speaking, an architecture may be *monolithic* [32] or *modular* [45]. A monolithic architecture is scaled by increasing the size of the chip, while a modular architecture is scaled by increasing the number of chips. Physical constraints and routing overheads generally cap the number of qubits that a monolithic architecture can reasonably support, which makes modularity something of an informal requirement when scaling quantum computers. Another requirement for scalability is error correction [50]. Industrial applications for quantum computing require programs to run for hours or even days. Since quantum operations introduce small amounts of error into the ion states, the computational qubits must therefore be encoded and periodically corrected for lengthy computations to succeed.

Based on these considerations, we expect that scalable quantum computers will be both *modular* and *error corrected*. One obvious disadvantage of the modular architecture is that two-qubit operations are *not intrinsically possible* between qubits that live in separate modules. Instead, some amount of entanglement has to be shared between modules as a resource for *state* or *gate teleportation* [12]. This process is complicated by the fact that entanglement distribution is *probabilistic* and *noisy*. To say that distribution is probabilistic means there is a significant chance of failure when attempting to entangle two physical qubits. For inter-modular two qubit operations to be reliable, this means

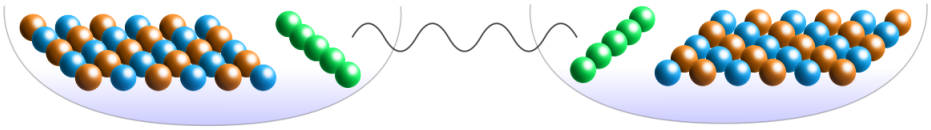

Figure 4.1: A collection of trapped ions in two separate Elementary Logical Units (ELUs). Some of the ions in each trap are used to encode a logical surface code qubit while other *communication ions* collect and refine entanglement to be used for a two-qubit *lattice surgery* operation.

that a large number of *communication qubits* will be required as an overhead to ensure that enough entanglement is collected. On the other hand, to say that distribution is noisy means that errors are inadvertently introduced into the entangled states which must be corrected before any teleportation can take place. This correction is done through *entanglement purification* [3] which non-deterministically reduces a large ensemble of weakly entangled pairs into a smaller ensemble of strongly entangled pairs.

In this chapter, I conduct resource analysis to estimate the number of ions needed to implement *reliable* two qubit operations between logically encoded qubits in different modules at different speeds. Specifically, I limit my attention to the *lattice-surgery* operation between *surface-code* encoded qubits (See Fig. 4.1 for a simplified schematic). This analysis is significantly influenced by the speed at which lattice surgery can be performed. Faster operations are desirable of course, but myriad factors limit the attainable rate. To ground this analysis, I survey the state-of-the-art in trapped-ion technologies and synthesise this information to propose *three possible surgery times* together with a rubric of technological milestones that are necessary to achieve each one. Additionally, I determine the attainable rates for lattice surgery operations given a fixed number of ions.

4.3 Background

4.3.1 Higher threshold rates

In chapter 3, I conducted resource analysis for satellite-based distributed quantum computing and made the common-sense assumption (based on prior work [21]) that the entangled pairs used for remote lattice surgery could not have an infidelity exceeding the threshold of the surface code. A recent result from Ramette et. al. however demonstrated

that lattice surgery is possible with pairs having an infidelity up to *ten times* higher than this threshold [48]. The reason for this unexpected robustness is that the error contributions from the entangled pairs are more or less confined to a single spatial dimension (Specifically, the edges of the two codes on which the surgery operation is performed). Consequently, their numerical analysis demonstrates that these lines of elevated activity are *well approximated* as 1-dimensional *repetition codes* [2]. Such codes are well known for having an exceptionally high error threshold of around 11% for unbiased noise [15].

Although Ramette et. al. do not mention this in their paper, I find it likely that the pair threshold is *even higher* than what they report, since they assume that each of the two surface code decoders are run *independently*. This approach however does not take advantage of the fact that the errors introduced by an entangled pair are *correlated* on either side. By taking these correlations into effect, it is plausible that a more powerful decoder could be engineered that would in turn allow for noisier pairs to be used. As this is an unsubstantiated speculation however, I will only work under the assumption that the infidelity of the entangled pairs may be up to ten times larger than the surface code threshold.

4.3.2 Trapped-ion architectures

Monroe et. al. [41] were among the first to perform a systematic study of a modular trapped-ion architecture, and some of our our terminology follows from their work. The modular trapped-ion computer is a collection of Elementary Logic Units (ELUs) which serve as local processors and or memory banks. In the Monroe framework, each ELU is a linear Coulomb crystal containing an identical quantity of ions. Some of these are *communication ions* which are coupled to photonic interconnects and can be used to create entangled pairs between ELUs. Still another fraction of the ions may be used for sympathetic cooling, which is when one ion is brought in proximity to another to absorb some of its vibrational energy.

An attractive advantage of the linear crystal is that it is *all-to-all* connected: No rearrangement is needed to implement a two-qubit gate between any pair of ions, though there's an effective limit to the crystal's length. Monroe et. al. believed this limit to be around 100 ions which, ten years later, seems to be a well justified estimate since the longest linear crystal today is around 30 computational qubits [11]. Further corroborating

evidence for this limit is discussed in greater detail by Murali et. al. [43]. Later research from the same Munroe group indicates that their initial 100 ion estimate might have been too optimistic [10]. On the other hand, Ratcliffe et. al. [49] propose a *micro-trap* based architecture with all-to-all connectivity which they claim can be scaled beyond a linear trap. This is accomplished with a *non-adiabatic gate* which, at the cost of higher power, circumvents the need to use conventional side-band resolving gates which are slow and more susceptible to vibrational noise. A two dimensional extention to the microtrap architecture has also been studied in some detail [39].

In practice, we do not need to restrict ourselves by limiting the ELU to a single linear crystal. One possible workaround for further scaling is to have multiple crystals "chained together" in the same ELU to form a *segmented linear trap* [28]. The crystals in this chain can be separated, combined, and rotated to move a qubit in one crystal to any other crystal in the trap. Though the segmented linear trap has a larger qubit capacity, it lacks the all-to-all connectivity of a single crystal – meaning that some amount of physical routing is required for general computation. On average, the number of swap operations needed to move quantum information from one end of the chain to the other scales linearly with the number of modules and the number of qubits per segment. Monroe et. al. believe more optimistically that a maximum of 1000 ions should be possible in an segmented linear trap [41]. Recent work indicates that stabilizing ions with optical tweezers could significantly improve the scalability of long ion-crystals [56] [55].

A still more general option for an ELU architecture is a Quantum Charged Coupling Device (QCCD) [28] [47]. This is a two-dimensional configuration of ion chains which may be routed along a fixed network of corridors and junctions. Theoretically, QCCD architectures are monolithic, well connected, and faster than segmented linear traps, though they are difficult to fabricate. Presently, the largest QCCD is *Quantiniuum's* 32 qubit *race-track* computer, which is a linear ion trap with periodic boundary conditions [42]. Various theoretic proposals exist for scaling beyond this current best: Malinowski et. al. report on a QCCD architecture called *WISE* that addresses a crucial *wiring problem* and is capable of supporting around 1000 ions [37]. Valentini et. al. developed the so-called *Quantum Spring Array* (QSA) where no inter-trap routing is required [62]. Sterk et. al. propose an architecture that specifically addresses the problem of power dissipation when scaling QCCDs [59]. Mehta et. al. suggest the use of planar-fabricated optics for

further improvements [40]. For a survey on the technical challenges of scaling QCCDs, see Murali et. al. [43]. True two and three dimensional ion crystals (also called Wigner crystals) may also be possible to engineer in the future on the scale of hundreds or perhaps even thousands of ions [66] [63].

4.3.3 Trapped-ion surface code implementations

Specialised architectures for *quantum error correction* will likely be easier to realise in the near term since error correction is predictable, repetitive, and often nearest neighbor (as is the case for the surface code). Recent experimental efforts to build trapped-ion surface codes are encouraging, though limited in scope. Erhard et. al. for example used a ten ion quantum computer to perform quantum state teleportation between surface codes of distance two [17] while Egan et. al. implemented the closely related Bacon-Shor code [16] based in part on the theoretical results from [14].

There is also a growing body of theoretical work around this objective of building trapped-ion surface codes. LeBlond et. al. presented software for compiling surface code operations to trapped-ion hardware [31]. Trout et. al. conducted extensive simulations of a distance 3 surface code implemented in a trapped-ion linear array [61], and reported a pseudothreshold of 3×10^{-3} with syndrome cycle times ranging from around 3 to 8 milliseconds. Similarly, Li et. al. studied a surface code modeled on a segmented linear trap [34]. For segment lengths of around 15 qubits, they report an error tolerance of 0.12% but say nothing about syndrome extraction times. Lekitsch et. al. [33] present a proof-of-concept for a monolithic surface code architecture that closely resembles a QCCD. A crucial difference however is that the Lekitsch architecture relies on *global fields* for a majority of the operations instead of individual lasers, which they argue is more feasible for monolithic scaling since it circumvents the need to align many optical elements with high precision. Their proposed cycle times are around $300\mu s$.

Although the focus of this chapter is on the surface code, we note that there is a strong interest for the so-called *color code* within the trapped-ion community [46, 62, 52]; This is a stabilizer code that is closely related to the surface code [30]

4.4 Estimating surface code cycle times

The rate we are able to do lattice surgery is upper bounded by the rate at which *syndrome extraction cycles* can be performed. Some care is therefore required to establish reasonable estimates for the attainable cycle times in trapped-ion systems. Broadly speaking, there are three processes that need to be accounted for. The first is the entangling operations that are required to couple the syndrome qubits with the appropriate data qubits. Ions may or may not need to be routed for this depending on the underlying architecture. The second process is the measurement of the syndrome qubits. As we will soon see, this may require separating the ions a short distance from each other in order to avoid measurement induced decoherence. The third and final process is ion-cooling, which is necessary to ensure high fidelity two qubit operations. In brief, the time required for a surface code cycle will depend both on the underlying choice of architecture, and also on various technical factors such as the speeds of single and two-qubit gates, measurements, ion-shuttling and cooling. We consider each of these factors in the following subsections and conclude our review by establishing three *cycle time paradigms* we expect are feasible provided that specific technological milestones are met.

4.4.1 Trapped-ion gates

The review of Bruzewicz et. al. presents a thorough comparison of single and two qubit trapped-ion gate times [8]. Typical single qubit gates with fidelities greater than surface code threshold are reported between $2\mu s$ and $12\mu s$, though lower fidelity operations have been demonstrated on the order of nanoseconds. Two qubit gates are generally slower and lower fidelity. For the sake of argument, let us assume that we are willing to tolerate operational two-qubit error rates of up to 0.1%, which sits comfortably below the surface code threshold. Typical two-qubit gates with fidelities *close* to this rate are clocked between $1.6\mu s$ and $100\mu s$. It is not unreasonable to assume therefore that the time it takes to implement the gates in a surface code cycle is around $10\mu s$.

4.4.2 Trapped-ion measurements

Single-qubit trapped-ion measurements are commonly implemented via *state-dependent fluorescence*. In this method, laser light is directed at an ion which exclusively couples

the $|1\rangle$ state to a 'cycling transition' that scatters numerous easily detectable photons. Likewise, the absence of photons indicates a measurement of the $|0\rangle$ state [8]. Although fluorescence measurements are relatively fast (on the order of $10\mu s$ [13] [44]) and high fidelity, the scattered photons (both from the laser and from the irradiated ion) are likely to decohere nearby qubits that aren't also being measured. This is a significant problem for error correcting circuits which all rely on mid-circuit measurements. Broadly speaking, there are two complementary strategies for mitigating measurement induced decoherence. The first is to incorporate techniques that suppress the decoherence, and the second is to move the ions some distance away to be measured safely. Both of these strategies will be discussed in the following subsections.

Techniques for suppressing decoherence

One way to limit measurement induced decoherence is to shorten the amount of time the qubit(s) are illuminated. This comes at the cost of measurement fidelity since there are fewer scattered photons to be detected. Naturally, faster and higher fidelity measurements will be possible with improvements in the photon collection rate and photo-detector efficiency. See the introduction of Wolk et. al. for a brief summary of techniques used to improve sparse detection fidelities [64].

Another approach for protecting against decoherence is to use quantum logic spectroscopy [54]. This is when the information of one qubit is transferred onto an ion of a different species that when measured emits off-resonant photons which are unlikely to disturb the states of neighboring qubits. An accidental benefit of this approach is that preexisting cooling ions may be used for this purpose. The disadvantages of quantum logic spectroscopy are that it is more difficult to maintain coherent control of multiple ion-species simultaneously, and that it requires additional ions (at most double for a one-to-one pairing). A promising alternative might be to use ions of the same species but have the data and measurement qubits encoded in different energy levels of the ion [19] [67]. An alternative strategy would be to suppress the decoherence effects altogether so that additional measurement ions are not required. Gaebler et. al. [22] demonstrate a technique for reducing measurement cross-talk errors by an order of magnitude using tailored *micromotion* which may reduce and potentially eliminate the need for logic spectroscopy or shuttling altogether. Micromotion is a time-dependent, driven motion

that naturally occurs as a result of ion confinement. Too much micromotion however is known to push ions outside the *Lamb-Dicke regime*, making high fidelity gates infeasible. Using tailored micromotion to hide ions from fluorescence induced measurements may therefore require more cooling than usual.

Ion-shuttling speeds

Perhaps the most intuitive strategy for mitigating measurement induced decoherence is to move the ions a safe distance away before measuring them. Ideally we'd like to complete this operation as fast as possible, but faster shuttling introduces more thermal noise which can have a detrimental effect on two-qubit gates in particular. Broadly speaking, the infidelity of a Mølmer-Sørensen gate (See section 4.4.3) applied between two qubits in an ion-chain is known to depend on both the temperature of the chain and its displacement in phase space. These effects have been fully characterised for two different error metrics [51]. In the ion-shuttling literature, the amount of heat imparted in transport is commonly characterised in terms of how much the expected *energy quanta* of a particular motional mode increases. In the absence of noise, a linear crystal can withstand several quanta of phonons before there is an appreciable drop in the fidelity of a Mølmer Sørensen gate. As noise is introduced however, this tolerance drops [4]. The fastest and quietest reported shuttling operation at the time of writing is also from Sterk et. al. who demonstrate a $210\mu m$ one way ion transport in $6\mu s$ with a maximum gain of 0.36 ± 0.08 quanta for an average speed of 35 $\mu m\ \mu s^{-1}$ [60]. Slower, but more conservative routing was also demonstrated in $55\mu s$ with a gain in 0.1 quanta [5].

Estimates for shuttling times

How far do ions need to be separated for the effects of measurement induced decoherence to be considered negligible? At the shorter end, Pino et. al. report a QCCD architecture where a shuttling distance of $110\mu m$ resulted in cross-talk errors between 3.5×10^{-3} and 1.5×10^{-2} [47]. Similarly, Crain et. al. show that a separation of $370\mu m$ results in cross-talk errors of 2×10^{-5} [13]. If we assume a distance of $300\mu m$ is tolerable, then with the shuttling speed reported by Sterk et. al. we can assume that a two-way shuttling time of around $10\mu s$ is sufficient for eliminating decoherence effects.

Multi qubit measurements

Strategic multi-qubit fluorescence measurements may be an effective way to reduce shuttling time further since high fidelity measurements are possible on *groups* of proximate ions. Zhukhas et. al. for example present a four ion readout with an overall error lower than twenty parts per million [68]. Similarly, IonQ's *Harmony* device is able to perform an 11 qubit readout with a cross-talk error lower than 10^{-4} [65]. Assuming their linear trap is around 100 micrometers long, this would suggest that ion separations below $10\mu m$ or so are sufficient for high fidelity simultaneous measurements.

4.4.3 Cooling trapped ions

All gates and operations of trapped ion quantum computers require low temperatures, but this is especially true of the two qubit gates since they depend on vibrational coupling which is highly sensitive to noise. In the early days of trapped ion-quantum computing, two qubit gates required temperatures close to the ground state energy. The breakthrough discovery of Mølmer and Sørrensen [57] shifted this paradigm by introducing a gate that could operate at the *Doppler temperature* – the temperature regime attainable with *Doppler cooling*. In the following subsections, we present a brief review of the cooling techniques used to bring *collections of ions* to Doppler and sub-Doppler temperatures – endeavoring to report approximate cooling times wherever possible. Although cooling single ions is considerably easier than cooling ion crystals (since there are fewer motional modes to be addressed [27]), it is unlikely that single-ion cooling will be a leading technology in the context of quantum computation. This is because virtually all quantum architectures (with the possible exception of QCCDs) keep their computational ions organised in crystals.

At a high level, Doppler cooling works by shining a laser on an ion with a frequency just below what the ion will absorb. When the ion moves towards the laser, the incoming light is blue-shifted with respect to the ion which causes it to absorb a photon and slow down. Aside from Doppler cooling, other established laser based cooling techniques that operate under similar principles include resolved sideband cooling, Raman sideband cooling, and Electromagnetically Induced Transparency cooling (EIT) with the fastest of these being EIT. Feng et. al. report cooling a 40 ion chain to a near ground state energy in under $300\mu s$ [20] while Jordan et. al. reach similar temperatures for a 100 ion

Penning trap within $200\mu s$. Some disadvantages of EIT are that it has a limited range of motional frequencies it can cool, and it is slow at cooling low frequency excitations [26].

Sympathetic cooling, where cold ions are brought into physical contact with computational ions, has been discussed in some detail in the previous sections. The major disadvantage of this approach is that it is relatively slow compared to other cooling methods and requires additional resource overheads [36]. A recent experimental demonstration showed that ion chains up to length 28 could be cooled to the global Doppler cooling limit using only two dedicated cooling ions of the same species [38]. This paper reported *relaxation times* (defined as the time required for the noise to settle within 5% of noise of the initial state) between 10 and 100 ms. Though these numbers are somewhat discouraging, one promising direction for further study is *persistent cooling* where a number of sympathetic cooling ions are brought in perpetual contact with computational ions. As the computation proceeds, the cooling ions are continuously chilled with Doppler cooling. This technique could lift the requirement for cooling processes that halt the computation. Lin et. al. present an analysis of the dynamics of a linear array where a small subset of the ions are continuously cooled [35]. Additionally, a theoretic proposal for sympathetic cooling between one ion and a pre-cooled resource ion can be accomplished on the order of tens of microseconds, which may find some applicability in this context [53].

Rapid exchange cooling is a recently proposed alternative to sympathetic cooling that was suggested by Fallek et. al. in the context of QCCD [18]. Here, coolant ions in a continuously chilled bank are shuttled to and from the computational ions. The authors of this work perform a proof-of-concept experiment in which two calcium ions are cooled with a round-trip shuttling time of $107.3\,\mu s$ which, in their words, is "an order of magnitude faster than typical sympathetic cooling durations."

4.4.4 Cycle time paradigms

At the beginning of this section (4.4), we mentioned that estimating attainable cycle times is crucial to our resource estimation task. This is because the cycle time has a *direct bearing* on the number of communication ions we require per ELU; Faster cycles mean that more ions are required to collect the necessary entanglement in a shorter period. In this section we synthesise our findings from the previous review by proposing

Cycle time	System assumptions
$1000\mu s$	Cycle time is comparable to theoretic proposals in literature.
$100\mu s$	Less than one dedicated round of cooling per cycle. EIT cooling with sub-quanta shuttling required. Multi-qubit measurement stages recommended. Will likely incorporate at least one suppression technique discussed in Sec. 4.4.2.
$10\mu s$	Virtually no shuttling allowed. Persistent cooling that doesn't pause syndrome extraction cycle is essential. Multiple suppression techniques from Sec. 4.4.2 will likely be used together. Purification circuits are low-depth with one round of measurement.

Table 4.1: A summary of three cycle time paradigms for trapped-ion surface codes and the various technological milestones required for each speed.

three *cycle time paradigms* $(1000\mu s, 100\mu s, 10\mu s)$ that we could reasonably expect to see achieved for trapped-ion surface codes given various technical assumptions. For a short-hand summary of these paradigms see Table 4.1. The first and slowest cycle time we propose is around $1000\mu s$. This is several times faster than what was simulated by Trout et. al. [61] and around three times slower than the architecture proposed by Lekitsch et. al. [33]. This time scale permits some flexibility in routing and cooling options making it especially suitable for segmented-linear trap and QCCD architectures which require extensive use of both. Here, we expect one or more stages of cooling per cycle and shuttling to avoid measurement induced decoherence. The second time proposed is $100\mu s$. This is a more optimistic regime, being several times faster than the EIT cooling times reported in Section 4.4.3. Because of this, we require that fewer than one dedicated round of cooling is made per clock cycle. There is considerably less flexibility permitted for routing or shuttling at this scale. Dedicated zones for multi-qubit measurement (Section 4.4.2) will likely become significant time and heat savers as will sub-quanta shuttling. Architectures that are likely to be viable in this paradigm include linear-traps, Wigner crystals, and QCCDs. It is likely as well that at least one strategy for mitigating measurement induced decoherence will be employed (Section 4.4.2). The final, and most optimistic regime is $10\mu s$. At this timescale, our clock cycle matches the two qubit gate times reported in Sec. 4.4.1 meaning that no processes are allowed which interrupt syndrome extraction. Persistent cooling is an absolute necessity here, as is an architecture that doesn't require any routing or shuttling. Linear traps, Wigner crystals or micro-trap based architectures are the most likely candidates for this regime.

4.4.5 Entangling ion pairs

Any modular architecture requires some means of communicating quantum information between the constituent ELUs. A common approach is to establish *maximally entangled pairs* of qubits between dedicated communication qubits to be used for quantum state teleportation. First, an ion is pulsed to create an entangled pair between an internal state of the ion and an emitted photon. Then, photons emitted from two different ELUs are routed together and fused with an polarization resolving Bell measurement that entangles the seperated ions. This is illustrated in fig. 4.2.

The maximum rate at which ion-ion entanglement can be *attempted* is fixed by a constant called the photon scattering rate; This sits at around 100 MHz, though myriad other experimental factors further limit the attempt rate to around 1 MHz (for specifics, see [58]). The fastest and highest quality ion-ion distribution rate at the time of writing comes from the same paper cited and reports 94% fidelity Bell pairs with a success probability of 2.18×10^{-4} for an average pair rate of $182s^{-1}$. The best ion coupling at the time of writing comes from the same paper cited previously, and reports 94% fidelity pairs with a success probability of $p_c = 2.18 \times 10^{-4}$ per attempt.

4.5 Methodology

4.5.1 The lattice surgery cycle

Given a surface code cycle time T, there are three steps that need to be completed within this T for lattice surgery to be implemented. The first is entanglement distribution, the second is purification, and the third is the joint syndrome extraction. All of these processes are illustrated in fig. 4.3. A natural strategy is to complete these steps *sequentially*. This means we divide our time window T into three parts which are then allocated to each process. The disadvantage of this method however is that we cannot make use of the full time-window for any of the three steps.

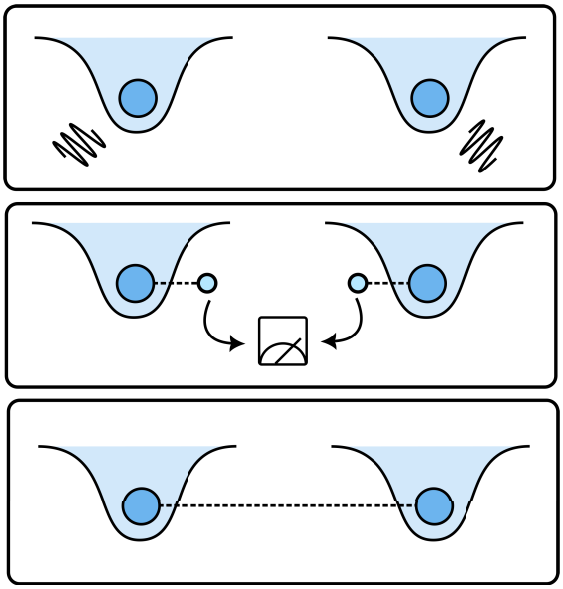

Figure 4.2: A cartoon illustrating the basic protocol for establishing ion-ion entanglement. (Top): Two ions in separate traps are stimulated with electromagnetic pulses. (Middle): If we are lucky, the two ions *simultaneously* emit photons that are entangled with their respective ions. When this happens, the routed together and interfered. (Bottom): Provided this *fusion* (or swapping) operation was successful, the entanglement is extended between the two ions.

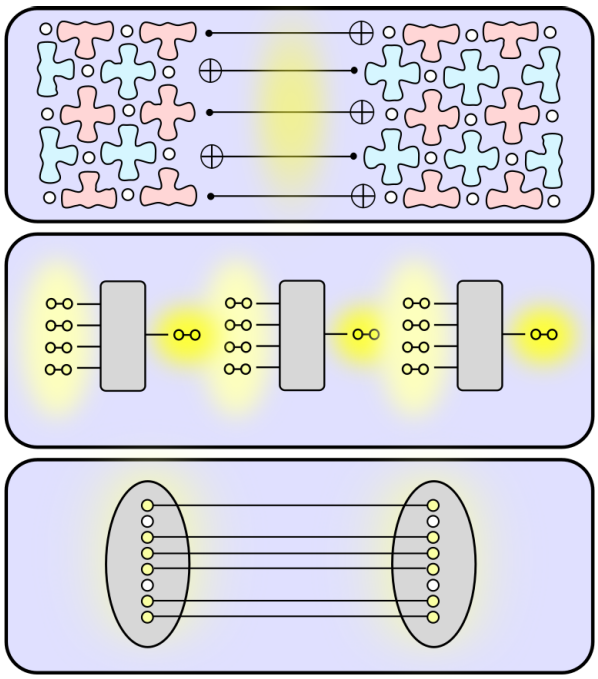

Figure 4.3: Three concurrent stages for a single lattice surgery operation that each must succeed within the cycle time T. (Bottom) Entanglement is established between pairs of communication ions in separate ELUs. (Middle) Distributed pairs in storage ions are refined via entanglement purification. (Top) Refined pairs are used to teleport the mediating gates needed for a lattice surgery between two surface code patches.

4.5.2 A heuristic for fault-tolerant lattice surgery

Our primary objective for this work is to estimate the number of trapped-ions needed to perform *fault-tolerant* lattice surgery between surface codes at the rates specified in Table 4.1. So far however, we have not addressed the question of what it means for lattice surgery to be fault-tolerant. Here, we formalise a definition by introducing a heuristic consisting of three criteria that must all be met for lattice surgery to be considered fault-tolerant. These conditions correspond to *three subroutines* of lattice surgery, which are depicted in Fig. 4.4 (Note that these correspond to the steps of fig 4.3). Essentially, the point of this heuristic is to ensure that each of of these necessary subroutines succeeds with high probability.

The first criteria is the promise of a *purification protocol* that takes n Bell pairs of some initial fidelity F_{in} and, with some probability p, returns one pair at or above the required fidelity threshold F_{ideal}. The second condition is that each purification circuit is *multiplexed* (run in parallel copies) until the probability of getting at least one pair from the lot exceeds P_{pair}. The final condition is that we are able to collect enough entanglement within the cycle time T to implement d instances of the multiplexed purification protocol (one for each "stitch" of the lattice surgery). We stress that this collection *must* take place within the time T so that enough entanglement is acquired to be used for the next round of lattice surgery. The probability that this collection succeeds must equal or exceed a threshold P_{LS}.

Minimum number of communication ions required given cycle time

Let p be the success probability of a purification circuit. The probability of obtaining at least one success out of n trials is $1-(1-p)^n$. Naturally the minimum number of purification circuits needed to produce at least one pair with a confidence of P_{pair} is then

$$K \equiv \min_n \left[1-(1-p)^n \geq P_{\text{pair}} \right] \tag{4.1}$$

If the purification circuit takes N_p raw pairs as input and returns one pair as output, the total number of raw pairs needed for the lattice surgery according to our heuristic is

$$N_{\text{LS}} = DN_pK \tag{4.2}$$

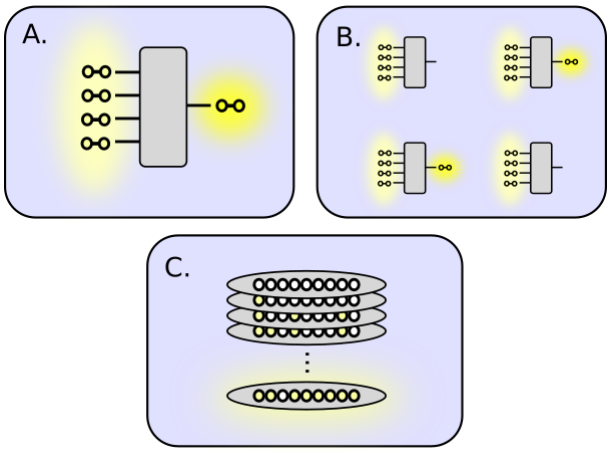

Figure 4.4: Illustrations of the three conditions we require for our lattice surgery to be considered fault-tolerant: (A): We are given an $n \to 1$ purification protocol where the output pair meets or exceeds fidelity F_{ideal}. (B): We duplicate the circuit until the probability of getting at least one purified pair exceeds P_{pair}. (C): We have enough communication ions such that after after a collection time T, we are sure up to a confidence of P_{LS} that we have enough entangled implement D instances of the protocol described in (B).

If T is the surface code clock cycle (the time it takes to perform a round of syndrome extraction) and if R is the rate at which entanglement can be attempted between pairs of ions, then we have $A = TR$ attempts to collect N_{LS} pairs. Suppose each ELU has $N_{\text{ions}} > N_{\text{LS}}$ communication ions. During a collection attempt, each of the $v \leq N_{\text{ions}}$ vacant (unentangled) ions are pulsed and may become entangled with probability p_e. Entangled ions are not pulsed in subsequent rounds, and we assume the entanglement does not degrade as it waits. Our first objective is to determine the probability that N_{LS} pairs can be collected in A attempts. The probability that a single ion pair is entangled after A attempts is given by:

$$P_{\text{onepair}} = 1 - (1 - p_e)^A \tag{4.3}$$

Let $X \sim \mathcal{B}(N_{\text{ions}}, P_{\text{onepair}})$ be the binomial random variable representing the number of ion pairs out of the initial N_{ions} that are entangled after A rounds. The minimum number of communication ions needed to collect at least N_{LS} pairs in a code cycle with confidence P_{LS} is then

$$\min_{N_{\text{ions}}} \left[P(X \geq N_{\text{LS}}) \geq P_{\text{LS}} \right] \tag{4.4}$$

Maximum attainable rate given given number of communication ions

Suppose now that N_{ions} is fixed. Let A_{min} be the minimum number of attempts needed to populate the N_{LS} ions needed for lattice surgery with an overall confidence of P_{LS}.

$$A_{\text{min}} = \min_{A} \left[P(X \geq DN_pK) \geq P_{LS} \right] \tag{4.5}$$

The maximum attainable rate for our fault-tolerant lattice surgery given N_{ions} is then just A_{min}/R

4.5.3 Device parameters and assumptions

From Stephenson et. al. [58], we assume that we can pulse ions at a rate of 1MHz where each pulse has a 2.18×10^{-4} chance of producing an entangled ion-ion pair of fidelity 0.94. We assume that our surface codes have an operational error rate of 0.1% which, from the results of Ramette et. al. [48], means we can tolerate Bell pairs with an infidelity of 0.01. We assume that the routing and circuits within the purification stage take a negligible amount of time compared with the entanglement collection. Single and two qubit gate error are approximated as single and two-qubit depolarizing channels that occur with probabilities 1×10^{-5} and 5×10^{-5} respectively. Measurement errors are taken as bitflip channels that occur with probability 1×10^{-5}. Although ion-trapping lifetimes are extremely good (hours, and even months in extreme cases), they are not indefinite. We do not consider ion loss or replacement in our resource estimation. Neither do we consider leakage errors that are known to accumulate with consecutive surface code cycles [6]. Though a linear crystal architecture has all-to-all connectivity in theory, it may not be possible in practice to perform arbitrary simultaneous two qubit gates as we have assumed. Nevertheless, we note promising results from the current state-of-the-art [24].

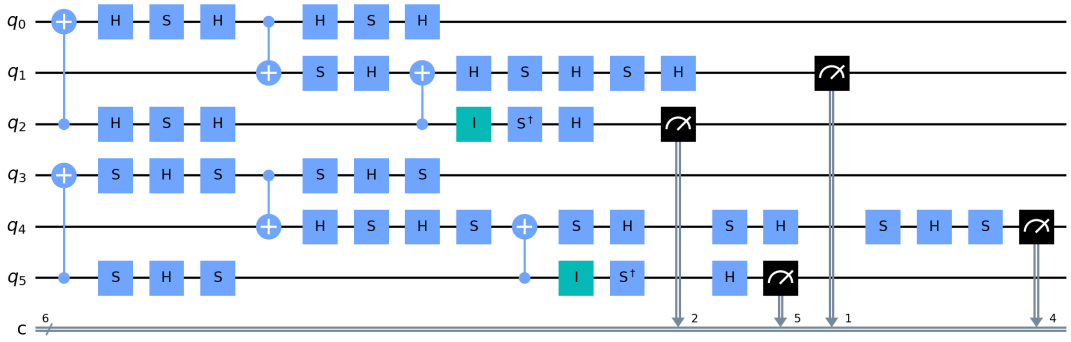

Figure 4.5: A high yield purification circuit discovered by the genetic algorithm from Addala et. al. [1]. This circuit takes three partially entangled pairs as input $\{(q_0, q_3), (q_1, q_4), (q_2, q_5)\}$ and non-deterministically returns a higher fidelity pair at (q_0, q_3). The protocol succeeds if $c_1 = c_4$ and $c_2 \neq c_5$ (in other words, the measurement outcome of qubit q_1 is coincident with the measurement of q_4 and the measurement of q_2 is *anti*-coincident with q_5). When this circuit is run with three copies of the *Stephenson pair* (see Appendix) as input together with the noise parameters described in sec. 4.5.3, the purification produces a fidelity $F = 0.9904$ pair with probability $p = 0.819$

4.5.4 Optimizing entanglement purification with device level noise

Our need for fault-tolerant lattice surgery highlights the importance of a high yield pair purification protocol. Though all such protocols theoretically asymptote to unit fidelity, the practical reality is that device level-noise imposes a cap on the pair fidelities that are attainable with purification. It is essential therefore to find a purification protocol that is able to reach our target fidelity of $F_{\text{ideal}} = 0.99$ *despite circuit-level noise*. To this end, we decided to search for high-performing purification protocols using recently developed numerical methods. Goodenough et. al. proposed an exhaustive search over purification protocols by mapping the problem to an enumeration over so called *graph codes* [23], while Addala et. al. [1], built on earlier work from Krastanov et. al. [29] to refine a genetic algorithm for finding purification protocols. We opted to use the genetic optimization over the enumeration because of its ease of use and because the attainable fidelities reported by Addala et. al. were comparable to those reported by Goodenough. We used this genetic algorithm to identify several hundred potentially suitable purification circuits.

From this initial pool of candidates, we simulated each circuit using the noise parameters detailed in section 4.5.3. For added realism, we modeled our initial $F = 0.94$

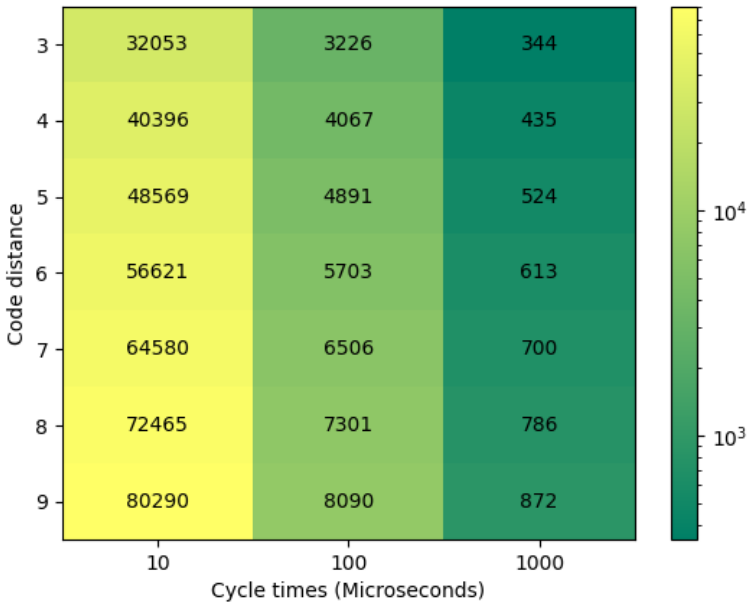

Figure 4.6: Minimum number of communication ions required to ensure that enough entanglement can be supplied within a given cycle time.

entangled pairs after the density matrix of an experimentally realised ion-ion pair reported in the supplementary material of Stephenson et. al. [58] (See the appendix section 4.7.1). The highest yield purification circuit we discovered with this genetic algorithm is presented in fig. 4.5. This protocol takes three *Stephenson pairs* as input and produces one output pair with a fidelity of $F = 0.9904$ with an overall success probability of 0.819. A full discussion of our methodology and findings is presented in the appendix of this chapter in section 4.7.

4.6 Results

Our numerical results are presented as tables in Fig. 4.6 and Fig. 4.7 respectively. In Fig. 4.6 we used Eq. 4.4 to determine the minimum number of communication ions needed to collect sufficiently many entangled pairs for fault-tolerant lattice surgery for a given cycle time T. We considered three cycle times of $10\mu s, 100\mu s$, and $1000\mu s$ according to the technological paradigms we discussed in section 4.4 over a small range of code distances. Our calculations indicate that the number of communication ions required is approximately linear with respect to both the code distance and the cycle time within our selected ranges. As the cycle time decreases in orders of magnitude, we find straight-

forwardly that the number of communication ions increases in orders of magnitude. If we assume that a given ELU may contain around 1000 ions at most, we find that a $D = 9$ code is theoretically supported at a clock cycle $1000\mu s$, while cycle times considerably faster than this appear out of reach.

In fig 4.7, we used Eq. 4.5 to calculate the maximum lattice surgery rates that are theoretically possible according to our fault-tolerant heuristic given various numbers of communication ions and various code distances. The whited out squares in the 100 ion column from $D = 7$ onward indicate that fault-tolerant lattice surgery is not possible at these distances, since the number of required pairs exceeds the number of communication ions available. Similar to what we observed in the previous table, we find that there's a tenfold difference between the 1,000 and 10,000 ion columns, though we note a slight deviation from this trend at the 100 ion column. This behavior occurs because the number of communication ions is close to the number of required pairs. As the required number of pairs approaches the number of available ions, we expect an exponential increase in the number of attempts needed to collect the entanglement since this collection is done without replacement. Our results indicate that with 100 communication ions, we could expect to support a distance 5 or 6 code at a maximum rate of around 100Hz. For 1,000 and 10,000 communication ions we find that larger code distances are possible with rates at around 1KHz and 10KHz respectively.

Our results indicate long term concerns for scalability due to the large number of physical resources required. If $10\mu s$ is about the fastest clock cycle we can hope for, our findings in figure 4.6 indicate that we would need upwards of 40,000 communication ions per ELU even for modestly sized surface codes. The prohibitive cost of this indicates an urgent need for improved optical coupling; As it stands, the low probability of entangling ions in separate ELUs ($p_c = 2.18 \times 10^{-4}$ [58]) is a leading cause for this inflated resource overhead. A natural question then (and the focus for the remainder of this section) is how the number of communication ions scales with *improvements* in the coupling rates. Our results are presented in figure 4.8. Here, we have again used equation 4.4 to calculate the minimum number of ions needed for lattice surgery between a selection of different surface codes while varying the probability p_c of establishing entanglement between a given pair of ions. What we see is that, irrespective of the distances and cycle times we consider, the number of communication ions first decreases as a power law with

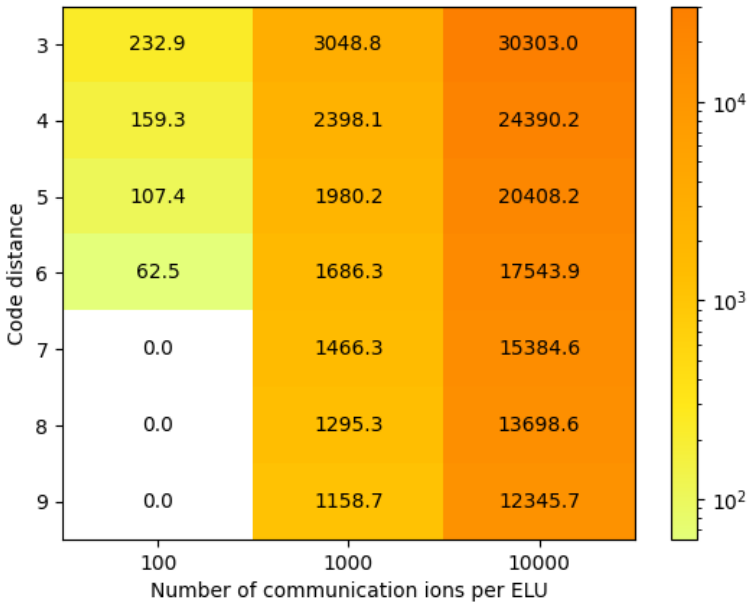

Figure 4.7: Average lattice surgery rates for different code distances and numbers of communication ions

increasing p_c and eventually tapers off to a fixed value that depends on the code distance and the purification protocol. These plateau values are equal to $N_{LS} = dN_pK$, which is the number of unpurified pairs that are required between ion-traps for lattice surgery to succeed. We can easily demonstrate this by examining the limiting case behavior when $p_c = 1$. In this scenario, all ions are guaranteed to be entangled after a single round of attempts. Consequently, we only require N_{LS} many ions to ensure that at least N_{LS} many raw pairs are established between traps.

For the sake of transparency, we note that the number of communication ions reported at the plateaus of fig. 4.8 are 46, 91, and 136 for distances $d = 3, 6$ and 9 respectively. We point this out because for $K = 5$ and $N_p = 3$, these values are equal to $N_{LS}+1$. This extra $+1$ comes from a small programming oversight where eq. 4.4 was calculated using a strict inequality as opposed to a weak inequality. The data of figure 4.8 additionally allows us to pin-point the ion coupling probabilities at which we reach these plateaus and consequently have no need to improve further. For a cycle time of $1000\mu s$, this occours at around $p_c \approx 10^{-2}$, while for $100\mu s$ and $10\mu s$ we find the convergence points at $p_c \approx 0.1$ and $p_c \approx 0.5$ respectively.

Let us suppose for the sake of argument that we are allowed 200 communication ions

per trap; This is high, but (unlike our data in figure 4.6) is not outside the realm of possibility. From figure 4.8, the ion coupling probabilities that are required to perform lattice surgery for a distance 9 surface code at cycle times of $1000\mu s, 100\mu s$ and $10\mu s$ respectively are $p_c \approx 1.5 \times 10^{-3}$, $p_c \approx 1.5 \times 10^{-2}$ and $p_c \approx 1.5 \times 10^{-1}$. These data suggest that ion-coupling rates need to improve by *one or several* orders of magnitude depending on the cycle time one wishes to operate at. One way to help meet this demand is to improve the efficiency of photon collection. Carter et. al. report collection efficiencies of 10% which is roughly an order of magnitude above what was previously possible [9]. Based on this result, it seems that improving entanglement rates by an order of magnitude is feasible target. Whether further improvements are possible however is unclear to us at present. Alternative methods for transporting entanglement via *shuttling* (therefore bypassing the need for improved coupling) are conceivable, yet speculative. Entanglement distribution using neutral atoms to mediate interactions has been discussed in the context of quantum networking [25].

Entanglement purification is another target for improvement. Our $3 \rightarrow 1$ protocol has a 81.9% success rate, which for a confidence threshold of $P_{\text{Pair}} = 0.999$ means that we require $K = 5$ purification circuits (Eq. 4.1) per stitch in the lattice surgery. This is effectively a $15 \rightarrow 1$ purification circuit, which appears rather wasteful. Given that this protocol is likely close to optimal for our initial state, the most likely strategy for reducing these overheads is to improve the fidelity at which pairs are distributed. Ideally, purification is eliminated altogether by delivering pairs a fidelity of $F = 0.99$ or higher.

In this work we estimated the number of communication ions needed to perform lattice surgery between two trapped-ion surface code qubits. To this end, we developed three paradigms for syndrome extraction cycle times that were predicated on various technological milestones and presented a heuristic that establishes what it means for a lattice surgery operation to be fault tolerant. With current inter-trap coupling rates, we found that hundreds, thousands and tens of thousands of communications ions are required for fault tolerant lattice surgery at cycle times of $1000\mu s, 100\mu s$ and $100\mu s$ respectively. The primary factor contributing to these prohibitive overheads is poor ion-coupling rates. Our results indicate the need to improve the coupling probability p_c by at least an order of magnitude for lattice surgery to be possible with only a couple hundred resource ions.

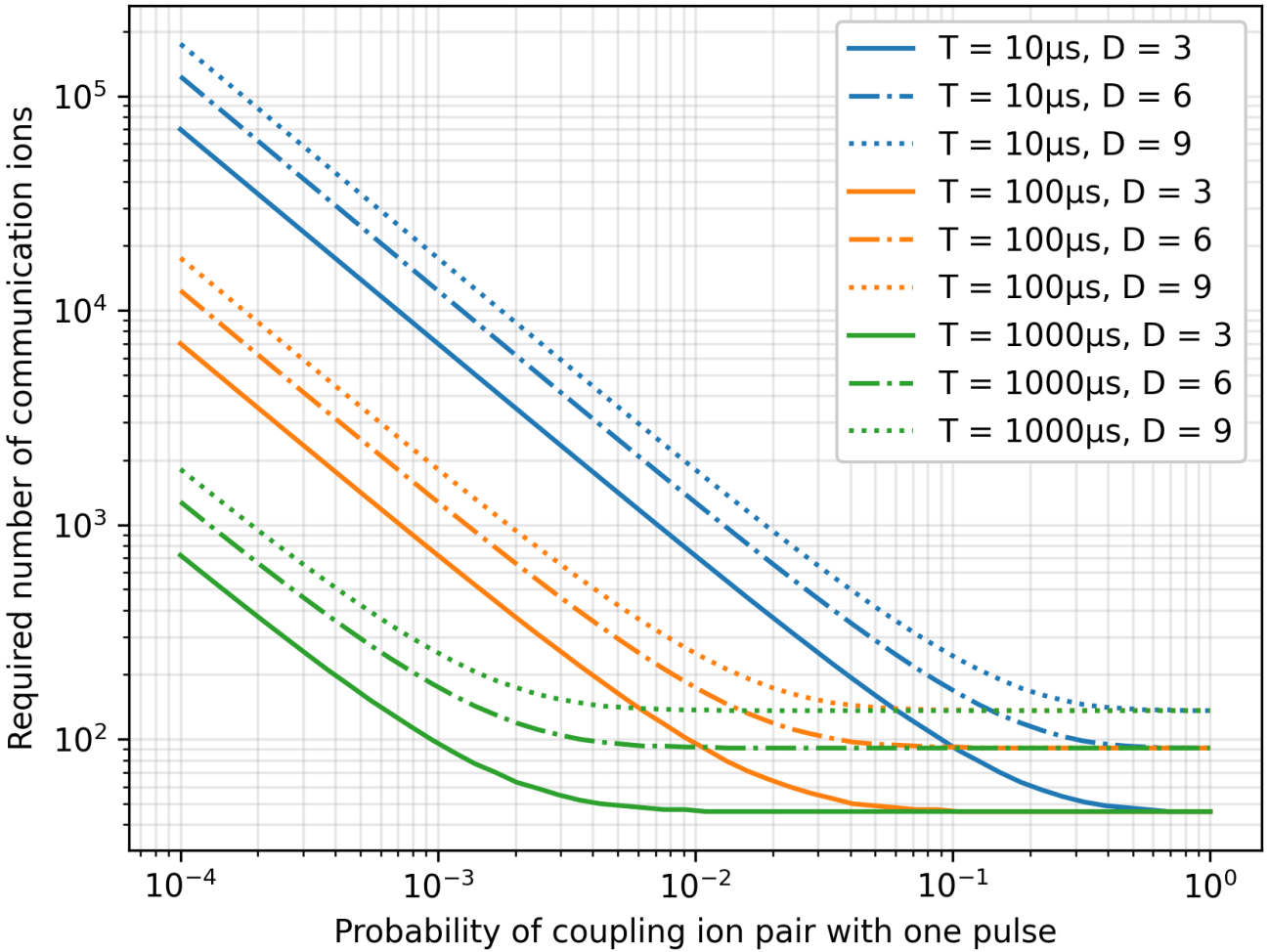

Figure 4.8: A selection of line plots that relate the minimum number of communication ions needed per trap with the probability of entangling two inter-trap ions with a single *'pulse'*. The three line colors represent the three surface code cycle times proposed in table 4.1 while the three linestyles represent various surface code distances. On the left, the data points begin at probability $p_c = 10^{-4}$, which is comparable to the best known coupling odds of $p_c = 2.18 \times 10^{-4}$ reported by Stephenson et. al. [58]. Fewer communication ions are needed as the success probability increases since it becomes easier to establish entanglement links between traps. At high probabilities, the lines converge to the same three values for the respective distances; These values represent the minimum number of unrefined pairs that are required for entanglement distillation to yield enough $F = 0.99$ links for lattice surgery. For $T = 1000\mu s$, $100\mu s$ and $10\mu s$, these values are 46, 91, and 136 respectively.

4.7 Appendix

4.7.1 Noisy entanglement distillation

Implementations $\tilde{\mathcal{D}}$ of any entanglement distillation protocol $\mathcal{D}$ are generally subjected to noise in the sense that $0 \neq \|\tilde{\mathcal{D}} - \mathcal{D}\| \leq \epsilon$ for small ϵ. Here we benchmark the performance of entanglement distillation protocols obtained with genetic algorithm [1] subjected to noise in ion trap systems. The genetic algorithm takes a Bell-diagonal state

$$F|\phi^+\rangle\langle\phi^+| + (1-F)\Big(p_x|\psi^+\rangle\langle\psi^+| + p_z|\phi^-\rangle\langle\phi^-| + p_y|\psi^-\rangle\langle\psi^-|\Big)$$

as input and searches for optimal $n \to k$ purification protocols for that state by iterating an initial randomly generated population of circuits (describing entanglement distillation protocols) over a number of generations. The fitness of the individuals in the population is evaluated with respect to one of several possible objective functions. We optimized with respect to the *average marginal fidelity*, which is the average fidelity of each output pair traced out from the final ensemble. This is determined analytically and may optionally account for single and two qubit depolarizing gate noise along with measurement errors. Each output circuit is returned with its average marginal fidelity and overall success probability.

Although the Bell-diagonal states may at first appear to be a somewhat contrived category of entangled pairs, it turns out that all two-qubit mixed states can be deterministically *twirled* into Bell-diagonal pairs of the same fidelity using local operations and classical communications. This may however cost a small amount of the *distillable entanglement* depending on the input state, though quantifying the amount of entanglement lost and developing recovery techniques appear to be open research directions. For this reason, and because twirling introduces a small amount of noise, we developed our search methodology around the objective of finding purification protocols that could work without twirling.

Our strategy therefore was to perform an initial search for promising looking protocols using $F = 0.94$ Bell-diagonal pairs with $p_x = p_y = p_z = 1/3$ under the parameter values $n = (3, 4, 5)$ and $k = 1$. Our decision to limit our search to $k = 1$ was motivated by our analysis indicating that the $k > 1$ protocols produced output pairs with some

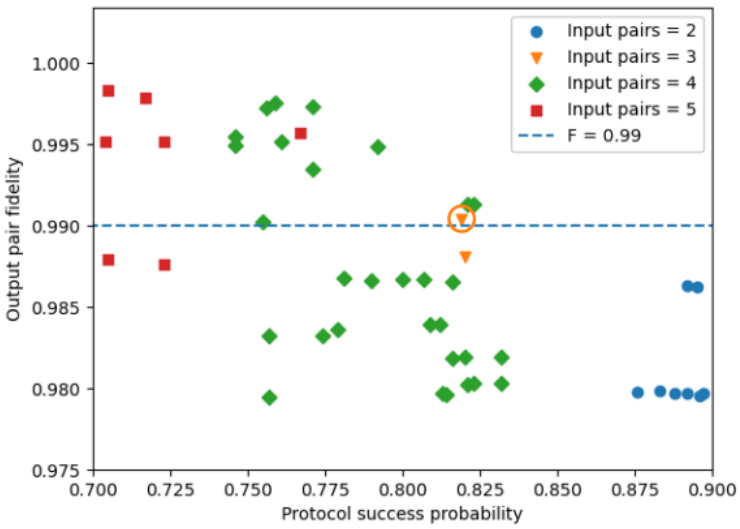

Figure 4.9: A scatter plot of the success probabilities and output pair fidelities of a high performing subset of $n \to 1$ purification protocols that were first identified with the genetic algorithm [1] then simulated under circuit level noise with $F = 0.94$ Stephenson pairs (See eq. 4.7) as inputs. The highest yield purification protocol that exceeds the required fidelity threshold $F_{\text{ideal}} = 0.99$ is circled in orange. This is a $3 \to 1$ purification protocol whose circuit is presented in fig. 4.5.

amount of *mutual entanglement* between them. This is a significant issue for lattice surgery, since it is assumed that each input pair is independent and required us to restrict ourselves to the $k = 1$ case. Simulating beyond $n = 5$ proved to be unnecessary since the average success rate of the protocols can be seen to decrease with increasing n in Fig. 4.9. For an $(n > 4) \to 1$ protocol to have a *higher yield* than the $3 \to 1$ we identified, it would be necessary for the larger purification circuit to have a significantly higher success probability.

Each simulation was performed with a population of 100 circuits evolved for 150 generations. We selected the top performing circuits from each simulation and benchmarked their performance under the same circuit level noise when simulated with $F = 0.94$ *Stephenson pairs* (See sec. 4.5.4) as input. Our numerical results are presented in Fig. 4.9. The broad trend indicates that as the number of pairs increases, the success probability of the protocol decreases while the average success probability increases. None of the protocols we identified were able to exceed the required fidelity threshold of $F_{\text{req}} = 0.999$.

In section 4.5.4, we alluded to an experimentally realised inter-trap ion pair reported by Stephenson et. al. in the supplementary material of their main paper [58]. Strictly

speaking, there are *four pairs* reported which correspond to four possible interferometer detection events. Since these state are all effectively equivalent under local operations and classical communications, we arbitrarily chose to consider the state presented in equation 4.6. NEW: Because our purification protocol works with respect to the target state $|\phi^+\rangle$, it is necessary to rotate our state ρ so its predominant term is $|\phi^+\rangle$. First (for convenience) we apply the following change of basis:

$$\left\{|\beta_1\rangle := |\phi^+\rangle,\ \beta_2 := |\phi^-\rangle,\ \beta_3 := \frac{1}{\sqrt{2}}(0,1,i,0)^T,\ \beta_4 := \frac{1}{\sqrt{2}}(0,1,-i,0)^T\right\}$$

, After this, we perform the rotation $(I \otimes XZ)\ \rho\ (I \otimes XZ)$ which gives us the density operator ρ' in eq. 4.7 This state is easily verified to have a fidelity of $F = 0.93$ with respect to $|\phi^+\rangle$. This is the *Stephenson pair* that we refer to in the main body of our paper.

$$\rho = \begin{pmatrix} 0.01 & -0.00487616 + 0.00349614i & 0.0135924 + 0.00634402i & 0.00374015 - 0.00331833i \\ -0.00487616 - 0.00349614i & 0.569 & 0.0542638 + 0.440672i & -0.012985 - 0.0292471i \\ 0.0135924 - 0.00634402i & 0.0542638 - 0.440672i & 0.416 & -0.0225074 - 0.00473484i \\ 0.00374015 + 0.00331833i & -0.012985 + 0.0292471i & -0.0225074 + 0.00473484i & 0.005 \end{pmatrix} \tag{4.6}$$

$$\rho' = \begin{pmatrix} 0.569 + 0.i & -0.00487616 - 0.00349614i & -0.0292471 + 0.012985i & 0.440672 - 0.0542638i \\ -0.00487616 + 0.00349614i & 0.01 + 0.i & -0.00331833 - 0.00374015i & 0.00634402 - 0.0135924i \\ -0.0292471 - 0.012985i & -0.00331833 + 0.00374015i & 0.005 + 0.i & -0.0225074 + 0.00473484i \\ 0.440672 + 0.0542638i & 0.00634402 + 0.0135924i & -0.0225074 - 0.00473484i & 0.416 + 0.i \end{pmatrix} \tag{4.7}$$

4.8 Tables of constants

In this section we present two tables that detail the most important free parameters and physical constants used throughout the chapter. Table 4.8 is a summary of the free parameters used and table 4.3 is a summary of the constants.

Parameter	Definition
N_{ions}	The number of communication ions in an ELU
d	Code distance
T	Syndrome extraction cycle time

Table 4.2: A summary of the free parameters considered in our analysis.

Parameter	Definition	Value	Justification
p	Purification protocol success probability	0.819	Simulated numerically under circuit level noise
R	Pulse rate	1 MHz	Within the magnitude of what is physically possible [58]
p_c	The probability of entangling two ions with one pulse-attempt	2.18×10^{-4}	State of the art: [58]
F_{ideal}	The fidelity required for pairs used in lattice surgery	0.99	Originally 0.999, but improved thanks to [48]
N_p	The number of pairs required for the purification circuit	3	Fig. 4.5
P_{pair}	The required confidence for multiplexed purification circuits to produce at least one pair	0.999	Arbitrarily high
K	The required number of purification circuits needed to meet multiplexing confidence	5	Substituting appropriate values into eq. 4.1
P_{LS}	The required confidence for collecting sufficient entanglement within a given clock cycle	0.999	Arbitrarily high

Table 4.3: A catalogue of important numerical constants used throughout this paper with justifications.

Bibliography

[1] Vaishnavi L. Addala, Shu Ge, and Stefan Krastanov. Faster-than-clifford simulations of entanglement purification circuits and their full-stack optimization, 2023.

[2] Google Quantum AI and collaborators. Exponential suppression of bit or phase errors with cyclic error correction. *Nature*, 595(7867):383–387, July 2021.

[3] Charles H. Bennett, Gilles Brassard, Sandu Popescu, Benjamin Schumacher, John A. Smolin, and William K. Wootters. Purification of noisy entanglement and faithful teleportation via noisy channels. *Physical Review Letters*, 76(5):722–725, jan 1996.

[4] Christopher D. B. Bentley, Harrison Ball, Michael J. Biercuk, Andre R. R. Carvalho, Michael R. Hush, and Harry J. Slatyer. Numeric optimization for configurable, parallel, error-robust entangling gates in large ion registers. *Advanced Quantum Technologies*, 3(11), July 2020.

[5] R. Bowler, J. Gaebler, Y. Lin, T. R. Tan, D. Hanneke, J. D. Jost, J. P. Home, D. Leibfried, and D. J. Wineland. Coherent diabatic ion transport and separation in a multizone trap array. *Phys. Rev. Lett.*, 109:080502, Aug 2012.

[6] Natalie C. Brown and Kenneth R. Brown. Leakage mitigation for quantum error correction using a mixed qubit scheme. *Phys. Rev. A*, 100:032325, Sep 2019.

[7] Colin D. Bruzewicz, John Chiaverini, Robert McConnell, and Jeremy M. Sage. Trapped-ion quantum computing: Progress and challenges. *Applied Physics Reviews*, 6(2):021314, Jun 2019.

[8] Colin D. Bruzewicz, John Chiaverini, Robert McConnell, and Jeremy M. Sage. Trapped-ion quantum computing: Progress and challenges. *Applied Physics Reviews*, 6(2):021314, 05 2019.

[9] Allison L. Carter, Jameson O'Reilly, George Toh, Sagnik Saha, Mikhail Shalaev, Isabella Goetting, and Christopher Monroe. Ion Trap with In-Vacuum High Numerical Aperture Imaging for a Dual-Species Modular Quantum Computer. *arXiv e-prints*, page arXiv:2310.07058, October 2023.

[10] M. Cetina, L.N. Egan, C. Noel, M.L. Goldman, D. Biswas, A.R. Risinger, D. Zhu, and C. Monroe. Control of transverse motion for quantum gates on individually addressed atomic qubits. *PRX Quantum*, 3:010334, Mar 2022.

[11] Jwo-Sy Chen, Erik Nielsen, Matthew Ebert, Volkan Inlek, Kenneth Wright, Vandiver Chaplin, Andrii Maksymov, Eduardo Páez, Amrit Poudel, Peter Maunz, and John Gamble. Benchmarking a trapped-ion quantum computer with 29 algorithmic qubits, 2023.

[12] Kevin S. Chou, Jacob Z. Blumoff, Christopher S. Wang, Philip C. Reinhold, Christopher J. Axline, Yvonne Y. Gao, L. Frunzio, M. H. Devoret, Liang Jiang, and R. J. Schoelkopf. Deterministic teleportation of a quantum gate between two logical qubits. *Nature*, 561(7723):368–373, September 2018.

[13] Stephen Crain, Clinton Cahall, Geert Vrijsen, Emma E. Wollman, Matthew D. Shaw, Varun B. Verma, Sae Woo Nam, and Jungsang Kim. High-speed low-crosstalk detection of a 171yb+ qubit using superconducting nanowire single photon detectors. *Communications Physics*, 2(1), August 2019.

[14] Dripto M Debroy, Muyuan Li, Shilin Huang, and Kenneth R Brown. Logical performance of 9 qubit compass codes in ion traps with crosstalk errors. *Quantum Science and Technology*, 5(3):034002, apr 2020.

[15] K. Duivenvoorden, N. P. Breuckmann, and B. M. Terhal. Renormalization group decoder for a four-dimensional toric code. *IEEE Transactions on Information Theory*, 65(4):2545–2562, April 2019.

[16] Laird Egan, Dripto M. Debroy, Crystal Noel, Andrew Risinger, Daiwei Zhu, Debopriyo Biswas, Michael Newman, Muyuan Li, Kenneth R. Brown, Marko Cetina, and Christopher Monroe. Fault-tolerant control of an error-corrected qubit. *Nature*, 598(7880):281–286, October 2021.

[17] Alexander Erhard, Hendrik Poulsen Nautrup, Michael Meth, Lukas Postler, Roman Stricker, Martin Stadler, Vlad Negnevitsky, Martin Ringbauer, Philipp Schindler, Hans J. Briegel, Rainer Blatt, Nicolai Friis, and Thomas Monz. Entangling logical qubits with lattice surgery. *Nature*, 589(7841):220–224, January 2021.

[18] Spencer D. Fallek, Vikram S. Sandhu, Ryan A. McGill, John M. Gray, Holly N. Tinkey, Craig R. Clark, and Kenton R. Brown. Rapid exchange cooling with trapped ions. *Nature Communications*, 15(1), February 2024.

[19] L. Feng, Y. Y Huang, Y. K. Wu, W. X. Guo, J. Y. Ma, H. X. Yang, L. Zhang, Y. Wang, C. X. Huang, C. Zhang, L. Yao, B. X. Qi, Y. F. Pu, Z. C. Zhou, and L. M. Duan. Realization of a crosstalk-avoided quantum network node with dual-type qubits by the same ion species, 2023.

[20] L. Feng, W. L. Tan, A. De, A. Menon, A. Chu, G. Pagano, and C. Monroe. Efficient ground-state cooling of large trapped-ion chains with an electromagnetically-induced-transparency tripod scheme. *Phys. Rev. Lett.*, 125:053001, Jul 2020.

[21] Austin G. Fowler, David S. Wang, Charles D. Hill, Thaddeus D. Ladd, Rodney Van Meter, and Lloyd C. L. Hollenberg. Surface code quantum communication. *Physical Review Letters*, 104(18), May 2010.

[22] J. P. Gaebler, C. H. Baldwin, S. A. Moses, J. M. Dreiling, C. Figgatt, M. Foss-Feig, D. Hayes, and J. M. Pino. Suppression of midcircuit measurement crosstalk errors with micromotion. *Phys. Rev. A*, 104:062440, Dec 2021.

[23] Kenneth Goodenough, Sébastian de Bone, Vaishnavi L. Addala, Stefan Krastanov, Sarah Jansen, Dion Gijswijt, and David Elkouss. Near-term n to k distillation protocols using graph codes, 2023.

[24] Nikodem Grzesiak, Reinhold Blümel, Kenneth Wright, Kristin M. Beck, Neal C. Pisenti, Ming Li, Vandiver Chaplin, Jason M. Amini, Shantanu Debnath, Jwo-Sy Chen, and Yunseong Nam. Efficient arbitrary simultaneously entangling gates on a trapped-ion quantum computer. *Nature Communications*, 11(1), June 2020.

[25] John Hannegan, James D. Siverns, Jake Cassell, and Qudsia Quraishi. Improving entanglement generation rates in trapped-ion quantum networks using nondestructive photon measurement and storage. *Phys. Rev. A*, 103:052433, May 2021.

[26] M K Joshi, A Fabre, C Maier, T Brydges, D Kiesenhofer, H Hainzer, R Blatt, and C F Roos. Polarization-gradient cooling of 1d and 2d ion coulomb crystals. *New Journal of Physics*, 22(10):103013, October 2020.

[27] Mingyu Kang, Qiyao Liang, Ming Li, and Yunseong Nam. Efficient motional-mode characterization for high-fidelity trapped-ion quantum computing. *Quantum Science and Technology*, 8(2):024002, jan 2023.

[28] V. Kaushal, B. Lekitsch, A. Stahl, J. Hilder, D. Pijn, C. Schmiegelow, A. Bermudez, M. Müller, F. Schmidt-Kaler, and U. Poschinger. Shuttling-based trapped-ion quantum information processing. *AVS Quantum Science*, 2(1):014101, 03 2020.

[29] Stefan Krastanov, Victor V. Albert, and Liang Jiang. Optimized Entanglement Purification. *Quantum*, 3:123, February 2019.

[30] Aleksander Kubica, Beni Yoshida, and Fernando Pastawski. Unfolding the color code. *New Journal of Physics*, 17(8):083026, August 2015.

[31] Tyler Leblond, Ryan S. Bennink, Justin G. Lietz, and Christopher M. Seck. In *TISCC: A Surface Code Compiler and Resource Estimator for Trapped-Ion Processors*, SC-W '23, page 1426–1435, New York, NY, USA, 2023. Association for Computing Machinery.

[32] Bjoern Lekitsch, Sebastian Weidt, Austin G. Fowler, Klaus Mølmer, Simon J. Devitt, Christof Wunderlich, and Winfried K. Hensinger. Blueprint for a microwave trapped ion quantum computer. *Science Advances*, 3(2):e1601540, 2017.

[33] Bjoern Lekitsch, Sebastian Weidt, Austin G. Fowler, Klaus Mølmer, Simon J. Devitt, Christof Wunderlich, and Winfried K. Hensinger. Blueprint for a microwave trapped ion quantum computer. *Science Advances*, 3(2):e1601540, 2017.

[34] Ying Li and Simon C. Benjamin. One-dimensional quantum computing with a 'segmented chain' is feasible with today's gate fidelities. *npj Quantum Information*, 4(1), May 2018.

[35] Guin-Dar Lin and L.-M. Duan. Sympathetic cooling in a large ion crystal. *Quantum Information Processing*, 15(12):5299–5313, November 2015.

[36] Y. Lin, J. P. Gaebler, T. R. Tan, R. Bowler, J. D. Jost, D. Leibfried, and D. J. Wineland. Sympathetic electromagnetically-induced-transparency laser cooling of motional modes in an ion chain. *Phys. Rev. Lett.*, 110:153002, Apr 2013.

[37] M. Malinowski, D.T.C. Allcock, and C.J. Ballance. How to wire a 1000-qubit trapped-ion quantum computer. *PRX Quantum*, 4:040313, Oct 2023.

[38] Z.-C. Mao, Y.-Z. Xu, Q.-X. Mei, W.-D. Zhao, Y. Jiang, Y. Wang, X.-Y. Chang, L. He, L. Yao, Z.-C. Zhou, Y.-K. Wu, and L.-M. Duan. Experimental realization of multi-ion sympathetic cooling on a trapped ion crystal. *Phys. Rev. Lett.*, 127:143201, Sep 2021.

[39] Zain Mehdi, Alexander K. Ratcliffe, and Joseph J. Hope. Scalable quantum computation with fast gates in two-dimensional microtrap arrays of trapped ions. *Phys. Rev. A*, 102:012618, Jul 2020.

[40] Karan K. Mehta, Chi Zhang, Maciej Malinowski, Thanh-Long Nguyen, Martin Stadler, and Jonathan P. Home. Integrated optical multi-ion quantum logic. *Nature*, 586(7830):533–537, October 2020.

[41] C. Monroe, R. Raussendorf, A. Ruthven, K. R. Brown, P. Maunz, L.-M. Duan, and J. Kim. Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects. *Phys. Rev. A*, 89:022317, Feb 2014.

[42] S. A. Moses, C. H. Baldwin, M. S. Allman, R. Ancona, L. Ascarrunz, C. Barnes, J. Bartolotta, B. Bjork, P. Blanchard, M. Bohn, J. G. Bohnet, N. C. Brown, N. Q. Burdick, W. C. Burton, S. L. Campbell, J. P. Campora, C. Carron, J. Chambers, J. W. Chan, Y. H. Chen, A. Chernoguzov, E. Chertkov, J. Colina, J. P. Curtis, R. Daniel, M. DeCross, D. Deen, C. Delaney, J. M. Dreiling, C. T. Ertsgaard, J. Esposito, B. Estey, M. Fabrikant, C. Figgatt, C. Foltz, M. Foss-Feig, D. Francois, J. P. Gaebler, T. M. Gatterman, C. N. Gilbreth, J. Giles, E. Glynn, A. Hall, A. M. Hankin, A. Hansen, D. Hayes, B. Higashi, I. M. Hoffman, B. Horning, J. J. Hout, R. Jacobs, J. Johansen, L. Jones, J. Karcz, T. Klein, P. Lauria, P. Lee, D. Liefer, S. T. Lu, D. Lucchetti, C. Lytle, A. Malm, M. Matheny, B. Mathewson, K. Mayer, D. B. Miller, M. Mills, B. Neyenhuis, L. Nugent, S. Olson, J. Parks, G. N. Price, Z. Price, M. Pugh, A. Ransford, A. P. Reed, C. Roman, M. Rowe, C. Ryan-Anderson, S. Sanders, J. Sedlacek, P. Shevchuk, P. Siegfried, T. Skripka, B. Spaun, R. T. Sprenkle, R. P. Stutz, M. Swallows, R. I. Tobey, A. Tran, T. Tran, E. Vogt, C. Volin,

J. Walker, A. M. Zolot, and J. M. Pino. A race-track trapped-ion quantum processor. *Phys. Rev. X*, 13:041052, Dec 2023.

[43] Prakash Murali, Dripto M. Debroy, Kenneth R. Brown, and Margaret Martonosi. Architecting noisy intermediate-scale trapped ion quantum computers. In *Proceedings of the ACM/IEEE 47th Annual International Symposium on Computer Architecture*, ISCA '20, page 529–542. IEEE Press, 2020.

[44] A. H. Myerson, D. J. Szwer, S. C. Webster, D. T. C. Allcock, M. J. Curtis, G. Imreh, J. A. Sherman, D. N. Stacey, A. M. Steane, and D. M. Lucas. High-fidelity readout of trapped-ion qubits. *Phys. Rev. Lett.*, 100:200502, May 2008.

[45] Kae Nemoto, Michael Trupke, Simon J. Devitt, Ashley M. Stephens, Burkhard Scharfenberger, Kathrin Buczak, Tobias Nöbauer, Mark S. Everitt, Jörg Schmiedmayer, and William J. Munro. Photonic architecture for scalable quantum information processing in diamond. *Phys. Rev. X*, 4:031022, Aug 2014.

[46] D. Nigg, M. Müller, E. A. Martinez, P. Schindler, M. Hennrich, T. Monz, M. A. Martin-Delgado, and R. Blatt. Quantum computations on a topologically encoded qubit. *Science*, 345(6194):302–305, 2014.

[47] J. M. Pino, J. M. Dreiling, C. Figgatt, J. P. Gaebler, S. A. Moses, M. S. Allman, C. H. Baldwin, M. Foss-Feig, D. Hayes, K. Mayer, C. Ryan-Anderson, and B. Neyenhuis. Demonstration of the trapped-ion quantum ccd computer architecture. *Nature*, 592(7853):209–213, April 2021.

[48] Joshua Ramette, Josiah Sinclair, Nikolas P. Breuckmann, and Vladan Vuletić. Fault-tolerant connection of error-corrected qubits with noisy links, 2023.

[49] Alexander K. Ratcliffe, Richard L. Taylor, Joseph J. Hope, and André R. R. Carvalho. Scaling trapped ion quantum computers using fast gates and microtraps. *Phys. Rev. Lett.*, 120:220501, May 2018.

[50] Joschka Roffe. Quantum error correction: an introductory guide. *Contemporary Physics*, 60(3):226–245, 2019.

[51] B. P. Ruzic, T. A. Barrick, J. D. Hunker, R. J. Law, B. K. McFarland, H. J. McGuinness, L. P. Parazzoli, J. D. Sterk, J. W. Van Der Wall, and D. Stick. Entangling-

gate error from coherently displaced motional modes of trapped ions. *Phys. Rev. A*, 105:052409, May 2022.

[52] C. Ryan-Anderson, N. C. Brown, M. S. Allman, B. Arkin, G. Asa-Attuah, C. Baldwin, J. Berg, J. G. Bohnet, S. Braxton, N. Burdick, J. P. Campora, A. Chernoguzov, J. Esposito, B. Evans, D. Francois, J. P. Gaebler, T. M. Gatterman, J. Gerber, K. Gilmore, D. Gresh, A. Hall, A. Hankin, J. Hostetter, D. Lucchetti, K. Mayer, J. Myers, B. Neyenhuis, J. Santiago, J. Sedlacek, T. Skripka, A. Slattery, R. P. Stutz, J. Tait, R. Tobey, G. Vittorini, J. Walker, and D. Hayes. Implementing fault-tolerant entangling gates on the five-qubit code and the color code, 2022.

[53] T Sägesser, R Matt, R Oswald, and J P Home. Robust dynamical exchange cooling with trapped ions. *New Journal of Physics*, 22(7):073069, July 2020.

[54] P. O. Schmidt, T. Rosenband, C. Langer, W. M. Itano, and et al. Spectroscopy using quantum logic. *Science*, 309(5735):749–52, Jul 29 2005. Copyright - Copyright American Association for the Advancement of Science Jul 29, 2005; Document feature - Equations; Diagrams; Graphs; ; Last updated - 2023-12-04; CODEN - SCIEAS.

[55] David Schwerdt, Lee Peleg, Yotam Shapira, Nadav Priel, Yanay Florshaim, Avram Gross, Ayelet Zalic, Gadi Afek, Nitzan Akerman, Ady Stern, Amit Ben Kish, and Roee Ozeri. Scalable architecture for trapped-ion quantum computing using rf traps and dynamic optical potentials, 2024.

[56] Yu-Ching Shen and Guin-Dar Lin. Scalable quantum computing stabilised by optical tweezers on an ion crystal. *New Journal of Physics*, 22(5):053032, May 2020.

[57] Anders Sørensen and Klaus Mølmer. Quantum computation with ions in thermal motion. *Phys. Rev. Lett.*, 82:1971–1974, Mar 1999.

[58] L. J. Stephenson, D. P. Nadlinger, B. C. Nichol, S. An, P. Drmota, T. G. Ballance, K. Thirumalai, J. F. Goodwin, D. M. Lucas, and C. J. Ballance. High-rate, high-fidelity entanglement of qubits across an elementary quantum network. *Phys. Rev. Lett.*, 124:110501, Mar 2020.

[59] J. D. Sterk, M. G. Blain, M. Delaney, R. Haltli, E. Heller, A. L. Holterhoff, T. Jennings, N. Jimenez, A. Kozhanov, Z. Meinelt, E. Ou, J. Van Der Wall, C. Noel, and D. Stick. Multi-junction surface ion trap for quantum computing, 2024.

[60] Jonathan D. Sterk, Henry Coakley, Joshua Goldberg, Vincent Hietala, Jason Lechtenberg, Hayden McGuinness, Daniel McMurtrey, L. Paul Parazzoli, Jay Van Der Wall, and Daniel Stick. Closed-loop optimization of fast trapped-ion shuttling with sub-quanta excitation. *npj Quantum Information*, 8(1):68, Jun 2022.

[61] Colin J Trout, Muyuan Li, Mauricio Gutiérrez, Yukai Wu, Sheng-Tao Wang, Luming Duan, and Kenneth R Brown. Simulating the performance of a distance-3 surface code in a linear ion trap. *New Journal of Physics*, 20(4):043038, April 2018.

[62] Marco Valentini, Martin W. van Mourik, Friederike Butt, Jakob Wahl, Matthias Dietl, Michael Pfeifer, Fabian Anmasser, Yves Colombe, Clemens Rössler, Philip Holz, Rainer Blatt, Markus Müller, Thomas Monz, and Philipp Schindler. Demonstration of two-dimensional connectivity for a scalable error-corrected ion-trap quantum processor architecture, 2024.

[63] S.-T. Wang, C. Shen, and L.-M. Duan. Quantum computation under micromotion in a planar ion crystal. *Scientific Reports*, 5(1), February 2015.

[64] Sabine Wolk, Ch Piltz, Theeraphot Sriarunothai, and Christof Wunderlich. State selective detection of hyperfine qubits. *Journal of Physics B: Atomic*, 48, 04 2015.

[65] K. Wright, K. M. Beck, S. Debnath, J. M. Amini, Y. Nam, N. Grzesiak, Chen J-S, N. C. Pisenti, M. Chmielewski, C. Collins, K. M. Hudek, J. Mizrahi, J. Wong-Campos, S. Allen, J. Apisdorf, P. Solomon, M. Williams, A. M. Ducore, A. Blinov, S. M. Kreikemeier, V. Chaplin, M. Keesan, C. Monroe, and J. Kim. Benchmarking an 11-qubit quantum computer. *Nature Communications*, 10:1–6, 11 2019.

[66] Y.-K. Wu, Z.-D. Liu, W.-D. Zhao, and L.-M. Duan. High-fidelity entangling gates in a three-dimensional ion crystal under micromotion. *Phys. Rev. A*, 103:022419, Feb 2021.

[67] H.-X. Yang, J.-Y. Ma, Y.-K. Wu, Y. Wang, M.-M. Cao, W.-X. Guo, Y.-Y. Huang,

L. Feng, Z.-C. Zhou, and L.-M. Duan. Realizing coherently convertible dual-type qubits with the same ion species. *Nature Physics*, 18(9):1058–1061, August 2022.

[68] Liudmila A. Zhukas, Peter Svihra, Andrei Nomerotski, and Boris B. Blinov. High-fidelity simultaneous detection of a trapped-ion qubit register. *Phys. Rev. A*, 103:062614, Jun 2021.

Chapter 5

Quantum circuits for simulating linear interferometers

> He sets his heart on finishing his handiwork, and he is careful to complete the decoration
>
> Sirach 38:28

5.1 Statement of work

Motivated by recent proposals for quantum proof of work protocols, I investigate approaches for simulating linear optical interferometers using digital quantum circuits. Section 5.2 is background work and (for the most part) does not represent original research. The possible exception is section 5.2.7 where I draw an analogy between the first and second quantization pictures in terms of quantum circuits. In section 5.3, I demonstrate how simulating linear optics in the first quantization requires the use of *symmetric qudit states* where the d in 'qudit' corresponds to the number of interferometer modes. Using this approach, I present a depth-four quantum circuit for simulating the *Hong-Ou-Mandel* effect. In section 5.4, I begin by outlining the basic principles of simulating in the second quantization picture. My research contributions begin in section 5.4.1 where I propose a polynomial-time circuit compilation algorithm that uses the principle of *divide-and-conquer*. I round off the discussion of the second quantization in section 5.4.2 by comparing this approach with *Trotterization*, which is the favored technique for Hamiltonian simulation. I present a software package `Aquinas` (**a qu**antum **in**terferometer

assembler) which uses the divide-and-conquer algorithm to generate circuits capable of simulating n-photon linear interferometers. This software is open-source, and is available through a GitHub repository [16]. To demonstrate the correctness of the software, I conclude the chapter by performing numerical simulations to verify that the expectation values of the circuit agree with theory. This research project was conducted under the supervision of Simon Devitt with extensive collaboration with Dr. Peter Turner. I thank both for their time and efforts.

5.2 Introduction

5.2.1 The linear interferometer

A linear interferometer is an optical device built from a network of beamsplitters and phase-shifters. It has m input ports and m output ports which are also known as *modes*. In a *sampling experiment*, an arbitrary number of indistinguishable photons simultaneously enter the interferometer according to some initial configuration. Multiple photons may occupy the same mode, and there is no limit to the number of photons that may enter at the start of the experiment. After the state is evolved through the interferometer, each mode is measured with a photo-detector that counts the number of photons present. In general, these measurements are non-deterministic since photons that enter at one mode may emerge in a coherent superposition of many modes. For simplicity, we assume that the total number of photons is conserved by the action of the interferometer.

While linear interferometers can be used to execute certain types of quantum operations, they themselves are not universal quantum computers. Nevertheless, there is significant interest in this computational model on account of the seminal work of Aaronson and Arkhipov who demonstrated that simulating boson sampling is classically intractable [1]. [1]

The reason for this intractability is because calculating (or even approximating) the output probabilities of a boson sampling experiment requires one to calculate the *permanent* of a $\mathbb{C}^{m\times m}$ matrix, which is a well known #P-complete problem. One implication of this is that linear interferometers (which are considerably easier to build than general purpose quantum computers) could in principle be used to demonstrate genuine (albeit

[1] Strictly speaking, if there are n photons entering an interferometer, the boson sampling problem is hard if the number of modes is $O(n^2)$ [1].

impractical) quantum supremacy. Significant work has therefore been done to engineer interferometers that are both large enough and reliable enough to demonstrate an incontestable quantum advantage (see Brod et. al. for a review of progress [6]). Of particular note is a recent paper from Madsen et. al. claiming supremacy with a variant of boson sampling that uses so-called *Gaussian states* [17].

Although the boson sampling model lacks the computational power of a general purpose quantum computer and is only "powerful" in a statistical sense, the proliferation of these experimental results has allowed a small body of research to develop around finding practical applications for boson sampling. Some computational examples include locating dense sub-graphs [3] and solving the graph isomorphism problem [7] (though whether such algorithms can yield a tangible advantage in practice remains to be seen). An arguably more promising application for boson sampling is in distributed consensus between quantum computers. Recent work has indicated the suitability of *coarse grained* boson sampling as a proof of work algorithm, which may see use in a future quantum blockchain [25].

One practical issue with developing boson samplers for commercial applications however is that *photon loss* limits quantum advantage. Though Aaronson et. al. proved that the hardness of boson sampling remains unchanged when a constant number of photons are lost [2], Oszmaniec et. al. later demonstrated that losing a *fraction* of the total number of photons can render the sampling problem classically tractable [22]. Since eliminating photon loss altogether is impractical, it would seem that boson sampling is not a promising model of computation long-term.

An option for scaling boson sampling beyond what is physically possible is to *simulate* the device on a conventional quantum computer. This unfortunately defeats the point of boson sampling for *computation* since any quantum computer that can emulate the interferometer will be far more powerful in general. Nevertheless, emulated boson sampling could conceivably be useful if the demand for a quantum proof-of-work algorithm exceeds what can be implemented with physical interferometers. Moreover, simulating linear interferometers beyond what is normally possible may be of use for research in fundamental physics.

Presently, the best reported algorithm for simulating linear interferometers is the classical method from Heurtel et. al. which, despite exponential savings over conventional

permanent based calculations, is still exponential in general (as expected) [12].

In this chapter, my main research contribution is a software package called `Aquinas` (**A Qu**antum **In**terferometer **As**sembler) [16] which transpiles a given linear interferometer into a quantum circuit that, when run, accurately simulates the behavior of a sampling experiment for up to a pre-specified number of photons. Broadly speaking, the algorithm works by first decomposing the initial interferometer into a grid of interlaced beamsplitters and waveplates. Following this, the Hamiltonian of each beamsplitter is calculated in terms of ladder operators over the respective modes. By truncating the ladder operators at a fixed photon depth (the total number of photons that enter the interferometer) and using some padding, we obtain a Hamiltonian that, when exponentiated, gives a relatively small unitary matrix that can be synthesised into a quantum circuit. These individual quantum circuits are then knitted together in the configuration of the original beamsplitters to form the interferometer circuit.

5.2.2 The symmetrization postulate

The *symmetrization postulate of quantum mechanics* [18] holds that a system of indistinguishable bosons is *invariant* under particle permutations. In other words, the corresponding state should be *unaffected* if two or more particles trace places. For example, if two identical photons p exist at positions a and b, then the state (p_a, p_b) would fail at symmetry since swapping the photons results in the equivalent but *differently labeled* state (p_b, p_b). We can resolve this issue by assuming these photons are in an *equal superposition* of the two possibilities.

$$\frac{1}{\sqrt{2}}\Big((p_a, p_b) + (p_b, p_a)\Big) \tag{5.1}$$

It is easily verified that this state is invariant under particle swapping since,

$$\frac{1}{\sqrt{2}}\Big((p_a, p_b) + (p_b, p_a)\Big) = \frac{1}{\sqrt{2}}\Big((p_b, p_a) + (p_a, p_b)\Big) \tag{5.2}$$

Making states symmetric via symmetrization

In general, any non-symmetric configuration of photons can be mapped to a symmetric state via *symmetrization*. Given some initial ensemble, the corresponding symmetric

state is found by taking the coherent superposition over all possible photon permutations. Let us visualise this with an example. Imagine we have three indistinguishable photons p that exist in modes a and b. The configuration (p_a, p_a, p_b) is easily seen to be a non-symmetric state since $(p_a, p_a, p_b) \neq (p_a, p_b, p_a)$. To find the corresponding symmetric state, we begin by enumerating all permutations of (p_a, p_a, p_b):

$$\{(p_a, p_a, p_b), (p_a, p_b, p_a), (p_b, p_a, p_a)\} \tag{5.3}$$

The resulting symmetric state is the equal superposition over all three of these possibilities.

$$\frac{1}{\sqrt{3}}\Big((p_a, p_a, p_b) + (p_a, p_b, p_a) + (p_b, p_a, p_a)\Big) \tag{5.4}$$

Notice that if any of the photons of this state are rearranged, all permutations are shuffled in the same way simultaneously. As a result, we see that reordering photons has no effect on the state, which by definition means that it is symmetric.

5.2.3 Fock space

In the previous sections, we presented the symmetrization postulate and described how we can map arbitrary photon configurations to symmetric states via symmetrization. Now we introduce the *Fock space* which is an extension of the Hilbert space that describes the possible states for an ensemble of arbitrarily many identical particles. As a motivation, let us first imagine a system where one photon can exist in a coherent superposition of m possible modes. By the postulates of quantum information theory laid out in the introductory chapter, the state space for this particle is the m-dimensional Hilbert space $\mathcal{H}_m$. If we introduce a second *identical* particle, we expect (by the product rule) for the state space to expand to $\mathcal{H}_m \otimes \mathcal{H}_m$. Because of the symmetrization postulate however, only the *symmetric states* $\mathrm{Sym}(\mathcal{H}_m \otimes \mathcal{H}_m)$ correspond to physical states. For systems with an arbitrary number of photons $n \neq 0$, it is straightforward to see that the corresponding state space is $\mathrm{Sym}(\mathcal{H}_m^{\otimes n})$. In the case where $n = 0$, the Hilbert space $\mathcal{H}_m^{\otimes 0}$ is *by definition* the complex field $\mathbb{C}$. Up to a phase, this field contains a single length one element which is called the *vacuum state* and denoted $|0\rangle$.

Formally, the bosonic Fock space is defined as the *direct sum* of the symmetric spaces

corresponding to every possible particle number.

$$\mathcal{F}_+(\mathcal{H}_m) = \bigoplus_{n=0}^{\infty} \mathrm{Sym}\left(\mathcal{H}_m^{\otimes n}\right) \tag{5.5}$$

The direct sum of two Hilbert spaces $\mathcal{H}_1, \mathcal{H}_2$ is denoted $\mathcal{H}_1 \oplus \mathcal{H}_2$ and can be interpreted as an ordered pair of elements from the respective spaces. For example, if $|\psi_1\rangle \in \mathcal{H}_a$ and $|\psi_2\rangle \in \mathcal{H}_b$ then the concatenated vector $\alpha|\psi_1\rangle \oplus \beta|\psi_2\rangle$ is a state of $\mathcal{H}_a \oplus \mathcal{H}_b$ where α and β are complex numbers satisfying the normalisation condition. The implication of the direct sum is that general photonic states can exist in a coherent superposition of states with *different numbers of photons*. For example, an arbitrary pure state of $\mathcal{F}_+(\mathcal{H}_m)$ has the form:

$$\alpha|0\rangle \oplus \sum_i \beta_i|\psi_i\rangle \oplus \mathrm{Sym}\left(\sum_{j,k} \gamma_{j,k}|\psi_j\rangle \otimes |\psi_k\rangle\right) \oplus \cdots \tag{5.6}$$

Where $|\psi_i\rangle, |\psi_j\rangle, |\psi_k\rangle, \cdots \in \mathcal{H}_m$ and where the complex coefficients $(\alpha, \beta_i, \gamma_{j,k}, \cdots)$ together satisfy the normalisation condition. For this chapter however, we are only really interested in Fock states with a fixed number of particles n in which case all amplitudes outside of the nth particle space are zero.

5.2.4 The second quantization

Interestingly, there is another equivalent way of formulating the Bosonic Fock space called the *second quantization* that is worth treating in detail. The primary advantage of this framework is that it is *considerably easier* to describe states than how we previously did in equation 5.6. In the last section, we saw that we could take a single particle Hilbert space $\mathcal{H}_m$ and extend it to a Fock space capable of supporting arbitrarily many particles $\mathcal{F}_+(\mathcal{H}_m)$. This representation is called the *first quantization*. In this section, we show how we can form the same space by starting with a *single mode* Fock space $\mathcal{F}_+(\mathcal{H}_1)$ and taking an m-fold Kronecker product of that space to then support m possible modes. In other words, the aim is to show that

$$\mathcal{F}_+(\mathcal{H}_m) = \mathcal{F}_+(\mathcal{H}_1)^{\otimes m} \tag{5.7}$$

Starting with the definition of the Fock space from eq. 5.5, we see that

$$\mathcal{F}_+(\mathcal{H}_1) = \alpha_0|0\rangle \oplus \mathcal{H}_1 \oplus \mathrm{Sym}(\mathcal{H}_1^{\otimes 2}) \oplus \cdots \tag{5.8}$$

The one dimensional Hilbert space $\mathcal{H}_1$ has a single length-one element (up to complex phase) which we denote $|1\rangle$. Consequently, $\mathrm{Sym}(\mathcal{H}_1 \otimes \mathcal{H}_1)$ is also a one dimensional space spanned by $|1\rangle \otimes |1\rangle$, and so on with $|1\rangle^{\otimes n}$ being the sole state of $\mathrm{Sym}(\mathcal{H}_1^{\otimes n})$. Using the relabeling $|1\rangle^{\otimes n} \to |n\rangle$ we find that

$$\mathcal{F}_+(\mathcal{H}_1) = \Big\{\alpha_0|0\rangle \oplus \alpha_1|1\rangle \oplus \alpha_2|2\rangle \oplus \cdots\Big\} \tag{5.9}$$

Where $(\alpha_0, \alpha_1, \cdots)$ are complex coefficients satisfying the normalisation condition. Elements of $\mathcal{F}_+(\mathcal{H}_1)$ are also called *number states* over a single mode since the unit vectors $(|0\rangle, |1\rangle, |2\rangle \cdots)$ specify the number of particles that are present. Number states over multiple modes are also possible. Let us now consider an m-fold Kronecker product of these spaces, where m is the number of modes we want to consider.

$$\mathcal{F}_+(\mathcal{H}_1)^{\otimes m} = \left\{ \bigotimes_{j=0}^{m-1} \left(\bigoplus_{n=0}^{\infty} \alpha_{n,j}|n\rangle \right) \right\} \tag{5.10}$$

We can simplify this by first expanding the Kronecker products to obtain a direct sum over all possible *multi-mode number states*, and then grouping terms together based on the total number of photons T_m over the m modes

$$= \left\{ \bigoplus_{n=0}^{\infty} T_m(n) \right\} \tag{5.11}$$

$T_m(0)$ for example is the direct sum of every possible m mode number state with 0 photons (which in this case is just $|0\rangle^{\otimes m}$ up to some complex phase). Similarly, $T_m(1)$ is the sum of states with one photon. Ignoring the wave amplitudes on each term, we see that

$$T_m(1) = |1_1, 0_2, \cdots, 0_m\rangle \oplus |0_1, 1_2, \cdots, 0_m\rangle \oplus \cdots \oplus |0_1, 0_2, \cdots, 1_m\rangle \tag{5.12}$$

In order to demonstrate that $\mathcal{F}_+(\mathcal{H}_1)^{\otimes m} = \mathcal{F}_+(\mathcal{H}_m)$, thereby showing that the second quantization picture is equivalent to the first, we first observe that equations 5.11 and 5.5 are both written as a direct sum over spaces corresponding to the possible photon

numbers:

$$\begin{aligned} \text{1st quantization}: &\quad \bigoplus_{n=0}^{\infty} \text{Sym}\left(\mathcal{H}_m^{\otimes n}\right) \\ \text{2nd quantization}: &\quad \bigoplus_{n=0}^{\infty} T_m(n) \end{aligned} \tag{5.13}$$

For these two spaces to be the same, it is therefore sufficient to show that for all $n, m \in \mathbb{Z}$:

$$\text{Dim}\left(\text{Sym}(\mathcal{H}_m^{\otimes n})\right) = \text{Dim}(T_m(n)) \tag{5.14}$$

The dimensionality of $T(m)$

Let us begin by determining how many dimensions there are in the space spanned by $T_m(n)$. Suppose that $|p\rangle$ is an n photon state over m modes such that

$$|p\rangle = |p_1\rangle \otimes |p_2\rangle \otimes \cdots \otimes |p_n\rangle \tag{5.15}$$

Where each $|p_i\rangle$ is a single mode number state such that $\sum_{i=1}^{m} p_i = n$. Observe that $|p_i\rangle \in \mathcal{H}_1$ when $p_i > 0$ and $|p_i\rangle \in \mathbb{C}$ when $p_i = 0$. In plain language, this means that each number state spans a one-dimensional Hilbert space, *except* for the vacuum state which spans the complex field. Consequently, the *kronecker product* of these number operators also spans $\mathcal{H}_1$, unless each p_i is zero.

$$\text{Dim}(|p_1\rangle \otimes |p_2\rangle \otimes \cdots \otimes |p_n\rangle) = \begin{cases} 1 \text{ if } \sum_{i=1}^{n} p_i \neq 0 \\ 0 \text{ else} \end{cases} \tag{5.16}$$

Thus, for $n \neq 0$, every element in the direct sum of $T_m(n)$ is a one-dimensional vector. Consequently, the dimension of $T_m(n)$ is equal to the number of terms in the direct sum. This is equal to the number of ways that n identical balls can be distributed over m baskets. Therefore:

$$\text{Dim}(T_m(n)) = \frac{(n+m-1)!}{n!(m-1)!} \tag{5.17}$$

The dimensionality of Sym$(\mathcal{H}_m^{\otimes n})$

Having determined the dimension of $T_m(n)$, we now show that the space $\mathrm{Sym}(\mathcal{H}_m^{\otimes n})$ has the same number of dimensions. First, observe that $\mathcal{H}_m^{\otimes n}$ is spanned by a basis of m^n elements. Not all of these vectors represent physical states however since not all are idempotent under photon permutation (for example, $|p_1, p_2\rangle \neq |p_2, p_1\rangle$). In section 5.2.2, we showed how we can map non-symmetric states to their symmetric counterparts by putting them in a coherent superposition of all possible permutations.

$$|p_1, p_2\rangle \xrightarrow{\text{Symmetrization}} \frac{1}{\sqrt{2}}(|p_1, p_2\rangle + |p_2, p_1\rangle) \tag{5.18}$$

Observe then that all states that are permutations of each other will map to the same symmetric state. This means that we can find the dimension of $\mathrm{Sym}(\mathcal{H}_m^{\otimes n})$ by counting the number of these perm-invariant collections. To see how this is done, first consider an example of two states that are not equivalent under photon permutation

$$|p_1, p_2, p_2\rangle \nLeftrightarrow |p_1, p_2, p_3\rangle \tag{5.19}$$

The reason why no permutation is possible here is because the two configurations represent *entirely different* distributions of photons. On the left hand side, there are no particles in mode three but on the right there is one particle there. Consequently, the number of perm-invariant collections is equal to the number of distinct photon configurations, or the number of ways that n identical balls can be distributed over m baskets. Clearly this is the same counting problem as the previous section, so we conclude that $\mathrm{Dim}(T_m(n)) = \mathrm{Dim}(\mathrm{Sym}(\mathcal{H}_m^{\otimes n}))$ and consequently $\mathrm{Dim}(\mathcal{F}_+(\mathcal{H}_m)) = \mathrm{Dim}(\mathcal{F}_+(\mathcal{H}_1)^{\otimes m})$.

5.2.5 Ladder operators

Creation and annihilation operators (also called ladder operators) are transformations of a Fock space that increase or decrease the number of photons in a particular mode by one. Though these ladder operators can (in theory) be described in either quantization, they are far easier to describe in the second quantization with number states, as I do here. The creation operator on mode j is denoted $\hat{a}_j^\dagger$. If $|n\rangle_j$ is a number state with n photons in mode j, the action of $a_j^\dagger$ is defined as follows:

$$\hat{a}_j^\dagger \, |n\rangle_j := \sqrt{n+1} \, |n+1\rangle_j \tag{5.20}$$

Likewise, for a state $|n\rangle_j$ (where $n \neq 0$) the effect of the annihilation operator $\hat{a}_j$ is

$$\hat{a}_j \, |n\rangle_j := \sqrt{n} \, |n-1\rangle_j \tag{5.21}$$

And if $n = 0$.

$$\hat{a}_j \, |0\rangle_j := 0 \tag{5.22}$$

This effect is referred to as the *annihilation of the vacuum state.* For all possible modes i and j, the following commutation relations hold for the ladder operators:

$$\begin{gathered} [a_i, a_j] = [a_i^\dagger, a_j^\dagger] = 0 \\ [a_i^\dagger, a_j] = \delta_{i,j} \end{gathered} \tag{5.23}$$

Assuming an m mode Fock space, it is possible then to construct linear operators for $\hat{a}_j^\dagger$ and $\hat{a}_j$ which are easily verified to satisfy these aforementioned properties:

$$\begin{aligned} \hat{a}_j^\dagger &= I_1 \otimes I_2 \otimes \cdots \otimes I_{j-1} \otimes \left(\textstyle\sum_{k=0}^{\infty} \sqrt{k+1} |k+1\rangle\langle k| \right) \otimes \cdots \otimes I_m \\ \hat{a}_j &= I_1 \otimes I_2 \otimes \cdots \otimes I_{j-1} \otimes \left(\textstyle\sum_{k=0}^{\infty} \sqrt{k+1} |k\rangle\langle k+1| \right) \otimes \cdots \otimes I_m \end{aligned} \tag{5.24}$$

Where each I_i is the identity operator on the ith mode. Though each mode can in theory support an unlimited number of photons, meaningful simulations only ever involve finitely many particles. What we do therefore is *truncate* the ladder operators by capping the summations of eq. 5.24 at some maximum photon number n. To signify that we've done this, we use the notation $\hat{a}_j^{\dagger(n)}$ and $\hat{a}_j^{(n)}$ respectively.

5.2.6 Characterising linear interferometers

Let $\mathcal{H}_{\hat{a}}$ be an m dimensional Hilbert space with basis vectors $\{\hat{a}_1^\dagger, \hat{a}_2^\dagger, \cdots, \hat{a}_m^\dagger\}$. The transformation induced by an m mode linear interferometer is described by a unitary operator U_{itf} over $\mathcal{H}_{\hat{a}}$ where the elements of U_{itf} are

$$[U_{\text{itf}}]_{\,i,j} = \lambda_{i,j}\ \hat{a}_i^\dagger \hat{a}_j \tag{5.25}$$

To illustrate, consider how the $\hat{a}_j^\dagger$ operator is evolved by U_{itf}

$$U_{\text{itf}}\ \hat{a}_j^\dagger = \sum_{i=1}^{m} \lambda_{i,j}\ \hat{a}_i^\dagger \tag{5.26}$$

Another way of interpreting U_{itf} is to think of it as the transformation of a single photon sent through the interferometer. To see why this is, let $|\psi\rangle \in \mathcal{H}_m$ be the state of a single particle over m modes. In the second quantization, this is expressed as

$$|\psi\rangle = \psi_1\ |1_1, 0_2, \cdots, 0_m\rangle + \psi_2\ |0_1, 1_2, \cdots, 0_m\rangle + \cdots + \psi_m\ |0_1, 0_2, \cdots, 1_m\rangle \tag{5.27}$$

The creation operator $\hat{a}_\psi^\dagger$ that generates this state from vacuum is easily seen to be $\sum_i \psi_j \hat{a}_j^\dagger$, which is an element of $\mathcal{H}_{\hat{a}}$. The evolved state is therefore $(U_{\text{itf}}\ \hat{a}_\psi^\dagger)|0\rangle^{\otimes m}$, which can be reframed as $U_{\text{itf}}|\psi\rangle$ by using the following change of basis

$$\hat{a}_j^\dagger \rightarrow |0_1, 0_2, \cdots, 1_j, \cdots 0_m\rangle \tag{5.28}$$

5.2.7 Circuit models for the first and second quantization picture

We saw in section 5.2.4 that we can describe the Fock space in either the first or second quantization picture, though it is not yet clear what the ramifications of these paradigms are for simulating linear interferometers with quantum circuits. Fortunately, there is a simple way of interpreting both the first and second quantizations in the circuit model that will aid us in our analysis.

Suppose we have n indistinguishable photons entering an interferometer where each particle can be found in one of m possible modes. In the first quantization, the circuit consists of n quantum registers of dimension m where each register represents a photon (see fig. 5.1). Measuring a register is therefore analogous to learning the position of a particular photon. Because these particles are indistinguishable however, we require by the *symmetrization postulate* (see section 5.2.2) for the input state to be invariant under photon permutation. Equivalently, the input state of our circuit must be invariant under

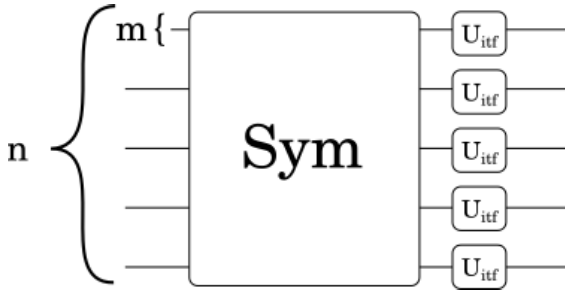

Figure 5.1: A basic circuit template for simulating a linear interferometer in the first quantization. The n registers of this circuit are m-dimensional qubits that represent indistinguishable photons. Each photon, when measured, can be found in one of m possible modes. By the symmetrization postulate of quantum mechanics, we require that the input state is *symmetric*. In the context of this circuit, this means that reordering the n registers doesn't change the underlying state. The first step therefore in simulating an interferometer in the first quantization is to *symmeterize* the initial state by mapping it to a coherent superposition of all possible permutations of itself. This is represented with the large *Sym* operation at the start of the circuit. Following this, each particle is evolved by the interferometer given by U_{itf} to complete the simulation.

register permutations. This means that the initial state must first be *symmetrized* which (as we will later see) is a difficult routine in general. After symmetrization however, simulating the interferometer becomes trivial since photon interactions are *extremely weak* and can be safely ignored for our purposes. Because of these negligible interactions, we can treat each photon of our symmetric ensemble as if it were the *only one* passing through the interferometer. Recall from the latter part of section 5.2.6 that U_{itf} can be interpreted as the transformation of a single photon. Therefore, the final step for simulating in the first quantization is to implement U_{itf} on each of the n registers. A basic schematic of this entire protocol is presented in Fig. 5.1.

In the second quantization the situation is flipped. Now, we have m registers of dimension n that represent each of the possible modes. Each mode can be measured to detect up to n possible photons with an additional promise that there are n *total* photons entering and exiting the interferometer. Unlike with the first quantization, there is no symmetrization step required here since the modes only need to encode the *number* of indistinguishable particles present. The hard part about simulating in the second quantization is constructing a circuit to implement the transformation $U_{\text{itf}}^{(n)}$, which is the unitary corresponding to the interferometer where each ladder operator is truncated at a maximum photon number n. In general, this $U_{\text{itf}}^{(n)}$ is exponentially larger than U_{itf}

with respect to both n and m, which makes it virtually impossible to directly synthesise an appropriate circuit. Nevertheless, we will later show how this can be done *indirectly*, either by Hamiltonian simulation, or unitary decomposition.

To summarise: Registers in the first quantization correspond to *indistinguishable photons* while registers in the second quantization correspond to *modes*. The advantage of the first quantization is that the simulation step is trivial, but the trade-off is that we need to prepare a complex symmetric state. On the other hand, the second quantization requires no state preparation but has difficulty implementing the truncated unitary $U_{\text{itf}}^{(n)}$ on account of its size.

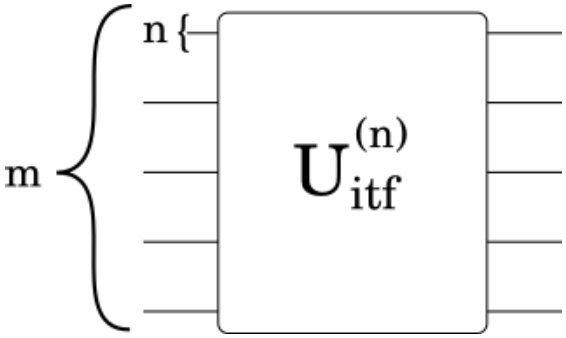

Figure 5.2: A basic template for simulating a linear interferometer in the second quantization picture. The m registers of the circuits are n-dimensional qubits that represent the m modes of an interferometer. Measuring a register reveals how many indistinguishable photons are present there. Unlike in the first quantization, no symmetrization is required over the initial state. The hard part of the second quantization is building a circuit to implement $U_{\text{itf}}^{(n)}$, which is the unitary matrix corresponding to the ladder unitary U_{itf} but where each ladder operator has been truncated to a depth of n photons.

5.3 Simulating in the first quantization

5.3.1 Two mode interferometers

To showcase the effectiveness of optical simulations in the first quantization picture, I now demonstrate how to construct efficient (in both size and depth) quantum circuits to simulate two-mode interferometers [2] for an arbitrary number of photons. In this scenario, we have n photons that are each entering the beamsplitter in one of two possible modes. Since $m = 2$, we see that registers of *one qubit each* are sufficient to encode the positions

[2]Physically, any two mode interferometer can be realised by a non-polarizing beamsplitter followed by a wave-plate on one of the modes to induce a relative phase shift. For convenience then, I will use the term 'beamsplitter' from this point on to refer to all two mode interferometers.

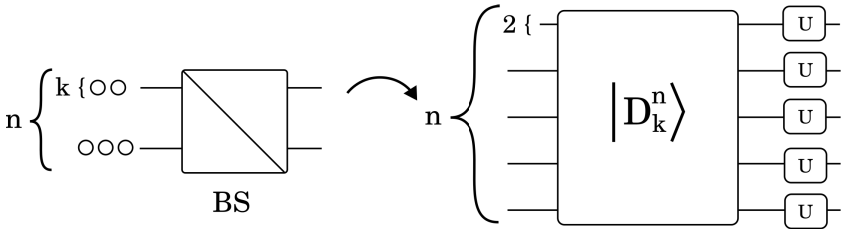

Figure 5.3: (left): N photons enter a beamsplitter described by the unitary U_{BS} with k of them going through the top. (right): The corresponding circuit in the first quantization. Begin by preparing the $|D_k^n\rangle$ Dicke state before applying U_{BS} to each qubit individually.

of each photon. Let a state of $|0\rangle$ denote occupation in the bottom mode and let $|1\rangle$ denote occupation in the top mode. For example, the state $|001\rangle$ indicates there are two photons in the bottom position and one photon up top. When symmetrized, this gives us the input state

$$\frac{1}{\sqrt{3}}(|001\rangle + |010\rangle + |100\rangle) \tag{5.29}$$

More generally, for a state of n identical photons where k of those photons are in the top mode, the resulting symmetric state is the $|D_k^n\rangle$ *Dicke state*

$$|D_k^n\rangle = \frac{1}{\sqrt{\binom{n}{2}}} \sum_{\substack{x\in\{0,1\}^n \\ w(x)=k}} |x\rangle \tag{5.30}$$

Where $w(x)$ is the Hamming weight (the number of ones) in the bitstring x. At the time of writing, the best known quantum algorithm for deterministic Dicke state preparation uses n qubits and requires a circuit depth of $\mathcal{O}(k \log \frac{n}{k})$ with little to no constant overhead [4]. The subsequent evolution of the state is performed by applying the 2×2 beamsplitter unitary U_{BS} to each mode in parallel, which is a constant depth operation. The overall depth of the protocol is therefore the same asymptotically as the Dicke state preparation.

Example: The Hong-Ou-Mandel effect

The Hong-Ou-Mandel (HOM) effect is perhaps the best known quantum-optical phenomena due to its counter-intuitive nature and because it is relatively easy to demonstrate in the lab. In a HOM experiment, we send two identical photons through the top and bottom modes of a 50:50 beamsplitter. Since this type of beamsplitter evenly splits

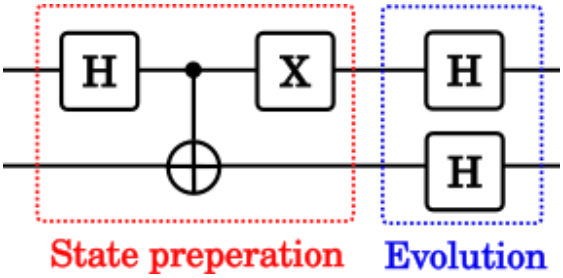

Figure 5.4: A circuit for simulating the *Hong-Ou-Mandel effect* in the first quantization picture. The first part of this circuit prepares the symmetric state $\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$ which signifies that there is one photon entering by the top mode and one photon by the bottom mode. After this, the simulated photons pass through a 50:50 beamsplitter whose transformation on the creation operators is identical to the Hadamard operation. The resulting state is $\frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$, meaning that (as expected) both photons are either in the top mode or in the bottom mode.

all incoming light, we naively expect to measure all possible two-photon configurations $\{\uparrow\uparrow, \uparrow\downarrow, \downarrow\uparrow, \downarrow\downarrow\}$ with equal probability. What happens in practice however is that we only ever record the *bunched* configurations $\{\uparrow\uparrow, \downarrow\downarrow\}$. To see why this is, we first note that the unitary transformation corresponding to the 50:50 beamsplitter is

$$U_{50:50} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{5.31}$$

Which (as an aside) is equivalent to the *Hadamard* operation H. Working in the second quantization picture, our initial state can be generated from vacuum with the operator $\hat{a}^\dagger_\uparrow + \hat{a}^\dagger_\downarrow$. Evolving this operator according to $U_{50:50}$ gives us

$$\begin{aligned} &\tfrac{1}{\sqrt{2}}(\hat{a}^\dagger_\uparrow + \hat{a}^\dagger_\downarrow) \times \tfrac{1}{\sqrt{2}}(\hat{a}^\dagger_\uparrow - \hat{a}^\dagger_\downarrow) \\ &= \tfrac{1}{2}((\hat{a}^\dagger_\uparrow)^2 - (\hat{a}^\dagger_\downarrow)^2) \end{aligned} \tag{5.32}$$

Which generates the number state $\frac{1}{\sqrt{2}}(|2,0\rangle + |0,2\rangle)$. A simple depth-four circuit for simulating a HOM experiment in the first quantization picture is presented in figure 5.4 together with a step-by-step explanation. Contrast this with the results of Mohan et. al. who only *approximate* HOM in the second quantization using a circuit with an (unoptimized) depth of 178 [19].

5.3.2 General interferometers

Having presented on beamsplitters, let us now consider the problem of simulating interferometers in the first quantization with $m > 2$ modes. Here, we see that we require one m dimensional quantum register per photon to encode its possible positions. As an example, let $m = 3$ and suppose we have four photons where two are in the bottom mode, one is in the middle, and one is on the top. One valid encoding of this state is $|0012\rangle$ which, when symmetrized, gives

$$\frac{1}{\sqrt{12}}\Big(|0012\rangle + |0021\rangle + |0102\rangle + |0120\rangle + |0201\rangle + |0210\rangle \\ + |1002\rangle + |1020\rangle + |1200\rangle + |2001\rangle + |2010\rangle + |2100\rangle\Big) \quad (5.33)$$

In general then, we see that the problem of preparing symmetric states for simulating m-mode interferometers boils down to the problem of preparing *symmetric qudit states* (SQDs), which are coherent superpositions over *all possible* permutations of the m dimensional registers. Nepomechie et. al. present an algorithm for constructing such states, though the asymptotic depth of these circuits is $\mathcal{O}(n^d)$ where n is the number of registers and d is the dimension of the register [20]. Having looked through the literature, I initially assumed that efficiently constructing SQDs was an open problem and proceeded to move on to the second quantization.

While writing this chapter however, I discovered that Berry et. al. had proposed two different methods for efficiently preparing *anti-symmetric qudit states* that could be trivially modified to create SQDs [5]. The more performant of these uses a *classical sorting network* to non-deterministically (but with high probability) prepare a resource state that can then be used to symmetrize arbitrary initial states. [3] The other (less efficient) approach uses a so-called *Fisher-Yates* shuffle to apply random permutations to the state until there is sufficient confidence the resulting state is symmetric. For an n particle, m mode system, the authors report an asymptotic circuit size [4] of $\mathcal{O}(m \log^c m \log n)$ and a circuit depth of $\mathcal{O}(\log^c m \log\log n)$ using a type of sorting network called called an

[3] For context, a sorting network is a type of sorting algorithm where the sequence of conditional-swap operations have been fixed *in advance*.

[4] i.e. the number of qubits

odd-even mergesort. [5]

We now consider the cost of evolving the symmetric state through the interferometer. Although this step is still constant depth with respect to n, it is not so for m however since the unitary being implemented on each mode is an $m \times m$ operator. Assuming there is no special structure to exploit within the operator, the cost of this step is equivalent to the cost of synthesising an arbitrary $\lceil \log_2(m) \rceil$ qubit unitary. The asymptotic number of qubits required for this operation is therefore $\mathcal{O}(\log(m))$, which I note is strictly less than the number of qubits needed for the state preparation step $\mathcal{O}(m \log^c m \log n)$. By the Solovay-Kitaev theorem, the number of *gates* required for a k qubit operator is [21]

$$\mathcal{O}\Big(4^k k^2 \log^c(4^k k^2/\epsilon)\Big) \tag{5.34}$$

Where ϵ is the desired accuracy. Substituting $k = \log_2(m)$ and using the fact that $2^{\log_2(m)} = m$ to simplify gives us an overall gate complexity of

$$\mathcal{O}\Big(m^2 \log^2(m) \log^c(m^2 \log^2(m)/\epsilon)\Big) \tag{5.35}$$

Which, for readability, we will write as

$$\mathcal{O}\Big(m^2 \operatorname{polylog}(m^2/\epsilon)\Big). \tag{5.36}$$

Assuming now that the gates of this decomposition are uniformly distributed across the qubits of the circuit, we therefore expect a circuit depth of around $\mathcal{O}(m^2 \operatorname{polylog}(m^2/\epsilon))/\log(m)$ which is asymptotically equal to equation 5.36. The combined depth of the state preparation and interferometer simulation is therefore approximately equal to

$$\mathcal{O}\Big(m \log^c m \log\log n + m^2 \operatorname{polylog}(m^2/\epsilon)\Big) \tag{5.37}$$

To conclude, the asymptotic number of qubits and circuit depth for simulating a general linear interferometer in the first quantization picture are as follows

Size	$\mathcal{O}(m \log^c m \log n)$
Depth	$\mathcal{O}\Big(m \log^c m \log\log n + m^2 \operatorname{polylog}(m^2/\epsilon)\Big)$

[5] Note that c is some constant power

5.4 Simulating in the second quantization

Let us now focus on simulating in the second quantization picture. Recall from Eq. 5.25 that an m mode interferometer can be represented as an $m \times m$ unitary matrix U_{itf} that describes how the creation operators of each mode are transformed. Given a maximum photon number n, our present objective is to transpile U_{itf} into a quantum circuit that implements $U_{\text{itf}}^{(n)}$ for some maximum photon number n (see fig. 5.2). Naively, we might try and calculate this unitary directly by substituting the ladder operators with their truncated counterparts:

$$\hat{a}_i^\dagger \hat{a}_j \to \hat{a}_i^{\dagger(n)} \hat{a}_j^{(n)} \tag{5.38}$$

The main problem with this approach is that it doesn't actually work. From eq. 5.24, we see that although $\hat{a}_i^\dagger \hat{a}$ is Hermitian, it is not unitary. The second problem is that $U_{\text{itf}}^{(n)}$ is an extremely large matrix and thus cannot be calculated directly. To see how large exactly, recall that a truncated creation operator takes the form

$$\hat{a}_j^{\dagger(n)} = I_1^{(n)} \otimes I_2^{(n)} \otimes \cdots \otimes I_{j-1}^{(n)} \otimes \left(\sum_{k=0}^{n} \sqrt{k+1} |k+1\rangle\langle k| \right) \otimes \cdots \otimes I_m^{(n)} \tag{5.39}$$

Since each of the constituent operators in this product is a $n \times n$ matrix, we see the entire operator acts over a m^n dimensional Hilbert space. Hypothetically though, if we were to calculate $U_{\text{itf}}^{(n)}$, we would begin by finding the *Hamiltonian operator* that generates U_{itf} under exponentiation.

$$H_{\text{itf}} = -i \log U_{\text{itf}} \tag{5.40}$$

Note that the log in the above expression refers to the matrix logarithm. *Now* we truncate the ladder operators

$$H_{\text{itf}} \xrightarrow{\hat{a}_i^\dagger \hat{a}_j \to a_i^{\dagger(n)} \hat{a}_j^{(n)}} H_{\text{itf}}^{(n)} \tag{5.41}$$

And exponentiate to obtain the truncated unitary.

$$U_{\text{itf}}^{(n)} = e^{iH_{\text{itf}}^{(n)}} \tag{5.42}$$

Worked example: Two mode interferometer

Let $\mathcal{H}_{\hat{a}(2)}$ be a two dimensional Hilbert space with basis vectors $\hat{a}_1^\dagger$ and $\hat{a}_2^\dagger$. The two photon interferometer (or beamsplitter for short) is described by the following unitary transformation over $\mathcal{H}_{\hat{a}(2)}$.

$$U_{BS} = \begin{pmatrix} \cos\theta & e^{i\phi}\sin\theta \\ -e^{-i\phi}\sin\theta & \cos\theta \end{pmatrix} \tag{5.43}$$

Here, θ is a parameter related to the reflectivity of the beamsplitter and ϕ is a relative phase introduced between the two output modes. The corresponding Hamiltonian is

$$H_{BS} = -i\log U_{BS} = \begin{pmatrix} 0 & -ie^{i\phi}\theta \\ ie^{-i\phi}\theta & 0 \end{pmatrix} \tag{5.44}$$

Which, expressed in terms of ladder operators, has the form

$$H = -ie^{i\phi}\theta\hat{a}_1^\dagger\hat{a}_2 + ie^{-i\phi}\theta\hat{a}_2^\dagger\hat{a}_1 \tag{5.45}$$

Say for example we want to consider the Hamiltonian with truncation depth $n = 2$. In this case we would perform the following substitutions

$$\hat{a}_1^\dagger\hat{a}_2 \to \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & \sqrt{2} & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & \sqrt{2} \\ 0 & 0 & 0 \end{pmatrix} \tag{5.46}$$

$$\hat{a}_2^\dagger\hat{a}_1 \to \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & \sqrt{2} \\ 0 & 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & \sqrt{2} & 0 \end{pmatrix} \tag{5.47}$$

The resulting Hamiltonian is

$$H_{BS}^{(2)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & ie^{-i\phi}\theta & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & i\sqrt{2}e^{-i\phi}\theta & 0 & 0 & 0 & 0 \\ 0 & -ie^{i\phi}\theta & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -i\sqrt{2}e^{i\phi}\theta & 0 & 0 & 0 & i\sqrt{2}e^{-i\phi}\theta & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2ie^{-i\phi}\theta & 0 \\ 0 & 0 & 0 & 0 & -i\sqrt{2}e^{i\phi}\theta & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2ie^{i\phi}\theta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{5.48}$$

Which, exponentiated gives the unitary $U_{BS}^{(2)}$:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \text{Cos}[\theta] & 0 & -e^{-i\phi}\text{Sin}[\theta] & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \text{Cos}[\theta]^2 & 0 & -\sqrt{2}e^{-i\phi}\text{Cos}[\theta]\text{Sin}[\theta] & 0 & e^{-2i\phi}\text{Sin}[\theta]^2 & 0 & 0 \\ 0 & e^{i\phi}\text{Sin}[\theta] & 0 & \text{Cos}[\theta] & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{2}e^{i\phi}\text{Cos}[\theta]\text{Sin}[\theta] & 0 & \text{Cos}[2\theta] & 0 & -\sqrt{2}e^{-i\phi}\text{Cos}[\theta]\text{Sin}[\theta] & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \text{Cos}[2\theta] & 0 & -2e^{-i\phi}\text{Cos}[\theta]\text{Sin}[\theta] & 0 \\ 0 & 0 & e^{2i\phi}\text{Sin}[\theta]^2 & 0 & \sqrt{2}e^{i\phi}\text{Cos}[\theta]\text{Sin}[\theta] & 0 & \text{Cos}[\theta]^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{i\phi}\text{Sin}[2\theta] & 0 & \text{Cos}[2\theta] & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.49}$$

Notice however that the size of this matrix is not equal to a power of two since the creation and annihilation operators we substituted in equations 5.46 and 5.47 were not shaped as powers of two themselves. We have two options for expanding into a qubit-friendly Hilbert space. The first and most obvious choice is to pick the next highest truncation depth that has a power-two dimension. In this example, that depth would be $n = 3$

$$\hat{a}^{\dagger(3)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & \sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{3} & 0 \end{pmatrix} \tag{5.50}$$

Alternatively, if one favours a slightly more sparse matrix, one can simply *pad* the creation, annihilation, and identity operators with zeros until the next power of two is reached. For example,

$$\hat{a}^{\dagger(2)}_{\text{padded}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & \sqrt{2} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{5.51}$$

To see the effect of this padding, consider a padded operator of the form $\hat{a}_i^\dagger \hat{a}_j$. When exponentiated, the action of the resulting unitary is seen with some effort to be idempotent on states with a photon number greater than the truncation depth.

5.4.1 Divide and conquer

We've seen in the previous sections that calculating $U^{(n)}_{\text{itf}}$ is a computationally intractable problem due to its size. An intuitive approach for working around this constraint is to *divide and conquer* by decomposing U_{itf} into a product of 2×2 unitaries (which can equivalently be thought of as *beamsplitters*). We then calculate the truncated versions of these smaller unitaries to a number depth n (where n is the *total number* of photons in the interferometer) and stitch them back in their original configuration to obtain a construction for $U^{(n)}_{\text{itf}}$ (See fig 5.5 for a schematic). This approach is somewhat more manageable than a naive synthesis of $U^{(n)}_{\text{itf}}$ since the beamsplitters can be implemented with fewer qubits. To see how many exactly, first note that the minimum number of qubits needed to encode up to n photons in a single mode is $\lceil \log_2(n) \rceil$. A two mode unitary truncated at depth n therefore acts on $2\lceil \log_2(n) \rceil$ qubits Using the same reasoning as before, we can substitute $k = 2\lceil \log_2(n) \rceil$ into eq. 5.34 and then divide by $\lceil 2 \log_2(n) \rceil$ to obtain the asymptotic circuit depth for a single beamsplitter[6] . Simplifying, this comes out to:

$$\mathcal{O}\Big(n^4 \log_2(n) \log^c(n^4 \log_2(n)/\epsilon) \Big) \tag{5.52}$$

Having established the size and depth of a *single* beamsplitter circuit we now consider

[6]For the sake of transparency, I'd like to note that I initially did this calculation wrong. Consequently, I thought the circuit depth of the beamsplitter was roughly *quadratic* with respect to n instead of roughly quartic. Had I known this, I likely wouldn't have pursued the "divide-and-conquer" approach, since the previous section indicates the SQD algorithm is far more efficient. I comment on this again in the conclusion of this chapter.

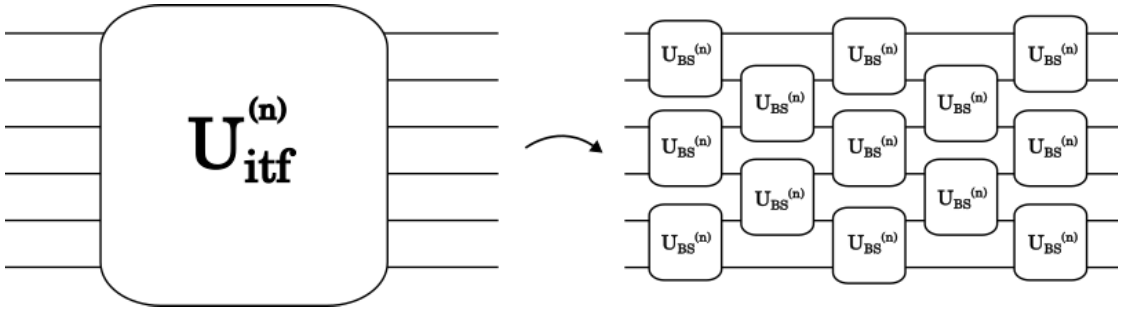

Figure 5.5: Given a linear interferometer U_{itf}, it is not computationally tractable to generate a circuit for the truncated operator $U_{\text{itf}}^{(n)}$. In the divide and conquer approach, I use the Clements decomposition [10] to break U_{itf} into a grid of two-by-two unitaries that are significantly easier to calculate (but still have exponential size with respect to n). As such, this method is best suited for simulating sparse linear interferometers where $n \ll m$

the number of beamsplitters that are required to decompose the interferometer. As an historical aside, Adolf Hurwitz was the first to study 2×2 decompositions of *orthogonal matrices* in the late nineteenth century [13] [7]. At the close of the twentieth century, Reck et. al. independently discovered a similar method for *unitary* matrices using a triangular decomposition [23]. Recently however, an improvement to the algorithm was discovered by Clements et. al. that requires half the depth of the Reck decomposition and is provably optimal [10]. With the Clements decomposition, a grid of $m(m-1)/2$ two-mode interferometers are required to reconstruct the original interferometer, which immediately implies a depth of $\mathcal{O}(m)$ beamsplitter elements and therefore an overall depth of $\mathcal{O}\big(mn^4 \log_2(n) \log^c(n^4 \log_2(n)/\epsilon)\big)$.

As with the section on the first quantization, we conclude by presenting the asymptotic size and depth requirements for this divide-and-conquer method:

Size	$\mathcal{O}(m \log(n))$
Depth	$\mathcal{O}\big(mn^4 \log_2(n) \log^c(n^4 \log_2(n)/\epsilon)\big)$

5.4.2 Approximate Hamiltonian simulation with Trotterization

Hamiltonian simulation is a rich and active field of research where the objective is to build quantum circuits that closely approximate the *time-evolution* of a system. Specifically, if we are given a Hamiltonian H, our goal is to implement the operator

[7] Thanks to Dr. Peter Turner for this observation

$$U(t) = e^{-iHt} \tag{5.53}$$

For some time t. We recall from equation 5.40 that we can calculate the Hamiltonian corresponding to an interferometer H_{itf} by taking the matrix logarithm of U_{itf}. By simulating the Hamiltonian, it is possible to *efficiently* implement $U_{\text{itf}}^{(n)}$ without needing to perform computationally intractable truncations and exponentiations of H_{itf}. A full review of the field is beyond the scope of this chapter[8], so for the sake of brevity, I limit myself to a type of Hamiltonian simulation called *first-order Trotterization* (FOT). Although this is by no means the most efficient method *asymptotically*, Sawaya et. al. [24] point out that FOT is more likely to be realised on near-term hardware since it is expected to have lower error rates over other simulation techniques [8] [9]. The FOT method is predicated on the fact that for two identically sized and non-commuting Hermitian operators A, B we have that [21]

$$\lim_{n\to\infty} (e^{iAt/n} e^{iBt/n})^n = e^{i(A+B)t} \tag{5.54}$$

The significance of this equality is that if a truncated Hamiltonian $H_{\text{itf}}^{(n)}$ can be written as a sum of *easily implemented* Hermitian operators $H_{\text{itf}}^{(n)} = \sum_j H_j$, then by eq. 5.54 we have that

$$\lim_{n\to\infty} \left(\prod_j e^{iH_j t/n} \right)^n = e^{iH_{\text{itf}}^{(n)} t} \tag{5.55}$$

From which it follows that if $\prod_j e^{iH_j t/n}$ is repeatedly applied, the resulting operator is a good approximation of $e^{iH_{\text{itf}}^{(n)} t}$ provided that n is sufficiently large. An obvious disadvantage of FOT (and Hamiltonian simulation more broadly) is that there is intrinsic *algorithmic error*. This is unlike the other simulation techniques I have discussed whose circuits (at least in theory) exactly implement the target unitary. [9] Consequently, it is more difficult to conduct resource analysis for FOT than other approaches in the first and second quantization. A further complication (though a blessing in disguise) is that error bounds for Hamiltonian simulation often *overestimate* the true error. Whittling

[8] See Childs et. al. for a fairly modern review [8]

[9] This is never true in practice though since unitary-to-circuit synthesis generally introduces a small amount of error.

down these bounds proves challenging however – even for special cases of FOT [15].

In the following section, I present a standard technique for FOT that relates a generic truncated Hamiltonian to a sum of *Pauli strings* for which efficient circuit constructions are known. I discuss the problem of *binary encoding* which complicates analysis *even more* and point out some recent progress that has been made in optimising FOT circuits. Although I am unable at this stage to directly compare the performance of using FOT to simulate linear interferometers with the other techniques I've developed, I nevertheless hope that this introduction may serve as a helpful starting point for future work.

Truncated Hamiltonians as sums of Pauli strings

We recall from section 5.2.4 that number states are labeled with integers describing the quantity of photons present in a particular mode. Observe then that a generic operator acting on a single mode with up to n photons [10] can be expressed as

$$A = \sum_{i,j=0}^{n} \alpha_{i,j} |i\rangle\langle j| \tag{5.56}$$

Where the coefficients $[\alpha_{i,j}]$ form a unitary matrix. To translate this operator into a transformation on *qubits*, we choose a binary encoding $\mathcal{B} : \mathcal{Z} \to \{0,1\}^{\oplus k}$ which maps an integer (within the range of the encoding) to a constant-length string of binary digits. Consider for example the *standard binary encoding* for $k = 3$

$$0 \to 000, \qquad 1 \to 001, \qquad 2 \to 010, \quad \cdots \quad 7 \to 111 \tag{5.57}$$

With this encoding, a number operator like $|1\rangle\langle 3|$ is converted to $|001\rangle\langle 011|$. Expanding this out gives us

$$|0\rangle\langle 0| \otimes |0\rangle\langle 1| \otimes |1\rangle\langle 1| \tag{5.58}$$

It is easily verified that these *binary operators* can related to Pauli matrices in the following way

[10] i.e. an operator over the truncated Fock space $\mathcal{F}_+(\mathcal{H}_1)^{(n)}$

$$|0\rangle\langle 1| = \tfrac{1}{2}(X + iY) \qquad |1\rangle\langle 0| = \tfrac{1}{2}(X - iY)$$
$$|0\rangle\langle 0| = \tfrac{1}{2}(I + Z) \qquad |1\rangle\langle 1| = \tfrac{1}{2}(I - Z) \tag{5.59}$$

Substituting the identities of equation 5.59 into 5.58 and expanding gives us an operator that encodes $|1\rangle\langle 3|$ as a sum of Pauli strings.

$$\frac{1}{8}(ZXI - IXZ + IXI + iZYI - iIYZ + iIYI - ZXZ - iZYZ) \tag{5.60}$$

This approach easily generalises to multi-mode number operators. To briefly illustrate, consider the operator $|1\rangle\langle 3| \otimes |3\rangle\langle 1|$. Using the standard binary encoding, this transformation is mapped to $|001\rangle\langle 011| \otimes |011\rangle\langle 001|$ which can then be expanded and substituted in the same way as we did in equation 5.60.

The main challenge of FOT is to choose a binary encoding $\mathcal{B}$ that *minimizes* the number of Pauli strings in the Hamiltonian encoding $H_{\text{itf}}^{(n)} = \sum_j P_j$. The reason why this is desirable is because each Pauli string P_j requires its own circuit in order for the product $\prod_j e^{iP_j t/n}$ to be implemented. Recent research by Sawaya et. al. [24] however indicates that choosing optimal encodings is likely a hard problem for one of two reasons. The first is that there are exponentially many binary encodings to choose from. The second is that the performance of a particular encoding depends *significantly* on the Hamiltonian being simulated.

Another important strategy for optimising FOT is to reorder the operations of $\prod_j e^{iP_j t/n}$ in such a way as to *maximize* the number of gate cancellations. Because of the regular structure of the circuits implementing $e^{iP_j t/n}$, this seemingly pedantic optimization can actually result in fairly significant savings. Tomesh et. al. study this *ordering problem* and use classical optimization techniques that (in part) reduces the depth of an FOT circuit to one-third of its naive implementation [26].

In summary, the main selling point of FOT compared to other methods of Hamiltonian simulation is the fact that it is more suitable for near term quantum hardware. On the other hand, its primary drawback is its relative complexity compared with the other methods previously discussed, which makes judging its performance difficult at face value. We have seen that it is difficult to accurately gauge the minimum number of 'steps' required in equation 5.55 for the simulation to be within error tolerance, and

we have pointed out the difficulty of choosing good encodings and orderings within the product formulae. It is therefore unclear at the present moment whether there is any advantage of using FOT to simulate linear optics compared with other methods in the first and second quantizations, though this seems to be a promising direction for further research.

5.5 Results

The `Aquinas` software package takes an $m \times m$ unitary matrix representing an m mode linear interferometer and uses the *divide and conquer* method described in section 5.4.1 to synthesise a quantum circuit that simulates the interferometer in the second quantization picture. To compensate for the fact that creation and annihilation operators truncated at depth n generally aren't sized as a power of two, I use the operator padding described in the worked example of section 5.4 in order to ensure that $U_{\text{itf}}^{(n)}$ can be implemented as a quantum circuit.

To demonstrate that the transpiled circuits are working as intended, I present two numerically simulated interferometer experiments. In the first experiment, I randomly selected a three mode interferometer $U_{\text{rand}} \in \text{SU}(3)$ and began with three indistinguishable photons in the arbitrarily chosen configuration $(2, 1, 0)$ (which is to say there are two photons in the first mode and one in the second mode). Using `Aquinas`, I then constructed a quantum circuit that implements the transformation $U_{\text{rand}}^{(3)}$ using 6 qubits with a depth of 567. I simulated this circuit noiselessly using the `Qiskit` [14] software package to obtain 10,000 measurement samples which were then converted into estimates for the probabilities of observing each of the 10 possible output configurations. In order to validate these probabilities, I used the permanent-based method of Aaronson and Arkhipov to the calculate the precise expectation values for each outcome [1]. I compare the estimated probabilities with the exact expectation values for this experiment using the histogram in 5.6. From inspection, it appears that our estimates are convincingly converging on the genuine expectation values with minor variations that may possibly be attributed to sampling uncertainty. To verify that this convergence was genuine, I began by using `qiskit`'s built-in `Operator` class to transform the circuit into a $2^6 \times 2^6$ matrix. After then, I calculated the expectation values of a random assortment of output

Number of photons	Circuit depth
1	5
2	197
4	3653
8	60677

Table 5.1: Circuit depths for a two-mode interferometer for various numbers of photons.

probabilities by 'looking up' the appropriate matrix element:

$$p_{\text{out}} = \langle \psi_{\text{out}} | U^{(3)}_{\text{rand}} | \psi_{\text{out}} \rangle \tag{5.61}$$

With this, I was able to quickly verify that the expectation values of the circuit were within machine point precision of the expectation values calculated via matrix permanent[11].

The second experiment was virtually identical to the first, with the only difference being that I instead simulated a five mode interferometer with two photons in the starting configuration $(2, 0, 0, 0, 0)$. In this case, the resulting 10 qubit circuit has a depth of 1972. The estimated probabilities of the 15 possible output distributions from $10,000$ circuit samples are plotted in figure 5.7 together with the analytical expectation values. As before, we see convincing evidence of genuine convergence that I confirmed by manually checking the expectation values of the operator corresponding to the circuit.

In order to calculate the depth of `Aquinas`'s circuits with respect to m and n, it is *sufficient* to calculate the depths of the beamsplitter circuits with respect to n. This is because every circuit with $m > 2$ modes is composed of multiple beamsplitter circuits arranged in a predictable pattern (Recall figure 5.5). A selection of these depths are presented in the above table.

5.6 Conclusion and outlook

In this chapter, I compared and contrasted a variety of different approaches for simulating linear optics in the first and second quantization pictures. I proposed a method for simulating in the first quantization based on the preparation of *symmetric qudit states* (SQDs), which had an asymptotic qubit count of $\mathcal{O}(m \log^c m \log n)$ and an asymptotic

[11] For larger circuits where calculating these expectations is intractable, *statistical bootstrapping* [11] may be used instead to test for convergence.

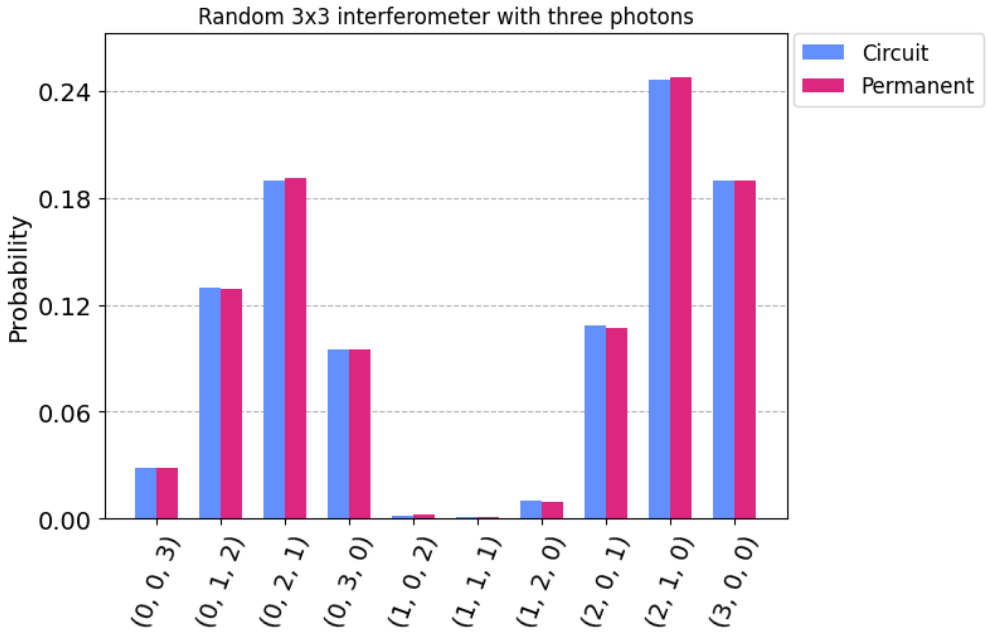

Figure 5.6: An arbitrary 3×3 interferometer with up to three photons is simulated as a 6 qubit, depth 567 circuit. The initial photon configuration (2,1,0) is scrambled into one of 10 possible output distributions.

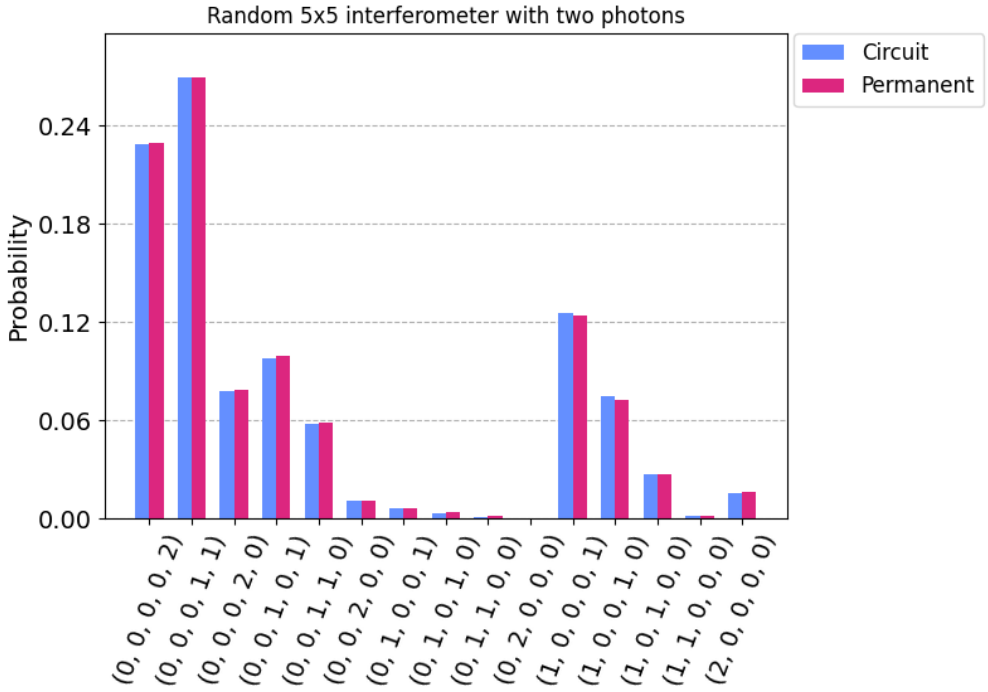

Figure 5.7: An arbitrary 5×5 interferometer with up to two photons is simulated as a 10 qubit, depth 1972 circuit. The initial photon configuration (2,0,0,0,0) is scrambled into 14 possible output distributions.

depth of $\mathcal{O}(m \log^c m \log\log n + m^2 \text{polylog}(m^2/\epsilon))$. In the second quantization picture, I presented a *divide and conquer* strategy and wrote software to construct circuits according to this method. I found that the asymptotic circuit size and depth were $\mathcal{O}(m \log(n))$ and $\mathcal{O}\left(n^4 \log_2(n) \log^c(n^4 \log_2(n)/\epsilon)\right)$ respectively. As I previously noted in the footnote of section 5.4.1, an initial oversight led me to believe that the asymptotic circuit depths for the *SQD* and *divide-and-conquer* algorithms were roughly comparable (with some minor trade-offs in performance between m and n). In actuality, the circuit depth for SQD is *considerably* better in terms of the photon number n ($\log\log$ vs. quartic!). Had I spotted this mistake earlier, I would have almost certainly developed my software around the SQD approach instead. Although we initially hoped that the *divide-and-conquer* method would allow us to construct interferometer circuits capable of simulating *non-trivial instances* of boson sampling (upwards of $n \geq 30$), compiling these circuits at scale quickly became a challenge (as indicated by the beamsplitter circuit depths of table 5.1). Given that 8-photon beamsplitter requires a depth of around 60,000, it is unlikely that our *divide-and-conquer* approach will see any practical use, even with further optimization. Nevertheless, this is an encouraging first step towards optical simulations on digital quantum computers. Based on these results, it seems that the most promising candidates for quantum optical simulations are the SQD and FOT methods. Future work should therefore focus on comparing these two to determine which is better in practice.

Bibliography

[1] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, page 333–342, New York, NY, USA, 2011. Association for Computing Machinery.

[2] Scott Aaronson and Daniel J. Brod. Bosonsampling with lost photons. *Phys. Rev. A*, 93:012335, Jan 2016.

[3] Juan Miguel Arrazola and Thomas R. Bromley. Using gaussian boson sampling to find dense subgraphs. *Phys. Rev. Lett.*, 121:030503, Jul 2018.

[4] Andreas Bartschi and Stephan Eidenbenz. Short-depth circuits for dicke state prepa-

ration. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 87–96, 2022.

[5] Dominic W. Berry, Mária Kieferová, Artur Scherer, Yuval R. Sanders, Guang Hao Low, Nathan Wiebe, Craig Gidney, and Ryan Babbush. Improved techniques for preparing eigenstates of fermionic hamiltonians. *npj Quantum Information*, 4(1), May 2018.

[6] Daniel J. Brod, Ernesto F. Galvão, Andrea Crespi, Roberto Osellame, Nicolò Spagnolo, and Fabio Sciarrino. Photonic implementation of boson sampling: a review. *Advanced Photonics*, 1(11):034001, 3 2019.

[7] Kamil Brádler, Shmuel Friedland, Josh Izaac, Nathan Killoran, and Daiqin Su. Graph isomorphism and gaussian boson sampling. *Special Matrices*, 9(1):166–196, 2021.

[8] Andrew M. Childs, Dmitri Maslov, Yunseong Nam, Neil J. Ross, and Yuan Su. Toward the first quantum simulation with quantum speedup. *Proceedings of the National Academy of Sciences*, 115(38):9456–9461, September 2018.

[9] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu. Theory of trotter error with commutator scaling. *Physical Review X*, 11(1), February 2021.

[10] William R. Clements, Peter C. Humphreys, Benjamin J. Metcalf, W. Steven Kolthammer, and Ian A. Walmsley. Optimal design for universal multiport interferometers. *Optica*, 3(12):1460–1465, Dec 2016.

[11] A. Ralph Henderson. The bootstrap: A technique for data-driven statistics. using computer-intensive analyses to explore experimental data. *Clinica Chimica Acta*, 359(1–2):1–26, September 2005.

[12] Nicolas Heurtel, Shane Mansfield, Jean Senellart, and Benoît Valiron. Strong simulation of linear optical processes. *Computer Physics Communications*, 291:108848, October 2023.

[13] A. Hurwitz. über die erzeugung der invarianten durch integration. *Nachrichten*

von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse, 1897:71–2, 1897.

[14] Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J. Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D. Nation, Lev S. Bishop, Andrew W. Cross, Blake R. Johnson, and Jay M. Gambetta. Quantum computing with Qiskit, 2024.

[15] David Layden. First-order trotter error from a second-order perspective. *Phys. Rev. Lett.*, 128:210501, May 2022.

[16] Hudson Leone. Aquinas. `https://github.com/FalafelGood/Aquinas`. Version: 1.0.0, Accessed: 2024-09-12.

[17] Lars S. Madsen, Fabian Laudenbach, Mohsen Falamarzi. Askarani, Fabien Rortais, Trevor Vincent, Jacob F. F. Bulmer, Filippo M. Miatto, Leonhard Neuhaus, Lukas G. Helt, Matthew J. Collins, Adriana E. Lita, Thomas Gerrits, Sae Woo Nam, Varun D. Vaidya, Matteo Menotti, Ish Dhand, Zachary Vernon, Nicolás Quesada, and Jonathan Lavoie. Quantum computational advantage with a programmable photonic processor. *Nature*, 606(7912):75–81, June 2022.

[18] A. M. L. Messiah and O. W. Greenberg. Symmetrization postulate and its experimental foundation. *Phys. Rev.*, 136:B248–B267, Oct 1964.

[19] Navaneeth Krishnan Mohan, Rikteem Bhowmick, Devesh Kumar, and Rohit Chaurasiya. Digital quantum simulations of hong-ou-mandel interference, 2024.

[20] Rafael I. Nepomechie and David Raveh. Qudit dicke state preparation, 2023.

[21] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[22] Michał Oszmaniec and Daniel J Brod. Classical simulation of photonic linear optics with lost particles. *New Journal of Physics*, 20(9):092002, September 2018.

[23] Michael Reck, Anton Zeilinger, Herbert J. Bernstein, and Philip Bertani. Experimental realization of any discrete unitary operator. *Phys. Rev. Lett.*, 73:58–61, Jul 1994.

[24] Nicolas P. D. Sawaya, Tim Menke, Thi Ha Kyaw, Sonika Johri, Alán Aspuru-Guzik, and Gian Giacomo Guerreschi. Resource-efficient digital quantum simulation of d-level systems for photonic, vibrational, and spin-s hamiltonians. *npj Quantum Information*, 6(1), June 2020.

[25] Deepesh Singh, Gopikrishnan Muraleedharan, Boxiang Fu, Chen-Mou Cheng, Nicolas Roussy Newton, Peter P. Rohde, and Gavin K. Brennen. Proof-of-work consensus by quantum sampling, 2024.

[26] T. Tomesh, K. Gui, P. Gokhale, Y. Shi, F. T. Chong, M. Martonosi, and M. Suchara. Optimized quantum program execution ordering to mitigate errors in simulations of quantum systems. In *2021 International Conference on Rebooting Computing (ICRC)*, pages 1–13, Los Alamitos, CA, USA, dec 2021. IEEE Computer Society.

Preprints and Publications

[1] Leone, H., Le, T., Srikara, S. and Devitt, S. [2024]. Resource overheads and attainable rates for trapped-ion lattice surgery.
URL: *https://arxiv.org/abs/2406.18764*

[2] Leone, H., Miller, N. R., Singh, D., Langford, N. K. and Rohde, P. P. [2021]. Cost vector analysis & multi-path entanglement routing in quantum networks.
URL: *https://arxiv.org/abs/2105.00418*

[3] Leone, H., Srikara, S., Rohde, P. P. and Devitt, S. [2023]. Upper bounds for the clock speeds of fault-tolerant distributed quantum computation using satellites to supply entangled photon pairs, *Phys. Rev. Res.* **5**: 043302.
URL: *https://link.aps.org/doi/10.1103/PhysRevResearch.5.043302*

The author wishes to note as well that there is a preprint ready (but not yet publicly available) for the material covered in the final chapter of this thesis.

In Thanksgiving:

Te Deum laudamus, te Dominum confitemur

Te aeternum Patrem omnis terra veneratur

Tibi omnes Angeli, tibi Caeli et universae Potestates

Tibi Cherubim et Seraphim incessabili voce proclamant

Sanctus, Sanctus, Sanctus, Dominus Deus Sabaoth

Pleni sunt caeli et terra maiestatis gloriae tuae.

Te gloriosus Apostolorum chorus

Te Prophetarum laudabilis numerus

Te Prophetarum laudabilis numerus

Te Martyrum candidatus laudat exercitus

Te per orbem terrarum sancta confitetur Ecclesia

Patrem immensae maiestatis

Venerandum tuum verum et unicum Filium

Sanctum quoque Paraclitum Spiritum

Tu Rex gloriae, Christe

Tu Patris sempiternus es Filius

Tu ad liberandum suscepturus hominem, non horruisti Virginis uterum

Tu, devicto mortis aculeo, aperuisti credentibus regna caelorum

Tu ad dexteram Dei sedes, in gloria Patris

Iudex crederis esse venturus

Te ergo quaesumus, tuis famulis subveni: quos pretioso sanguine redemisti

Aeterna fac cum sanctis tuis in gloria numerari

Amen