

Resource-efficient quantum circuits in the context of near-term devices

Kyriakos Georgiades

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of
University College London.

Department of Physics and Astronomy
University College London

June 16, 2024

I, Kyriakos Georgiades, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

The one-way model in measurement-based quantum computing is equivalent to the circuit model and allows for depth reduction at the cost of adaptive measurements and ancilla overheads. The one-way model is also less compatible with devices of low qubit connectivity and has mainly been experimentally realised on photonics devices in the past. In this thesis, I introduce a tuneable hybrid model which allows for more precise control of the amount of depth reduction and width overheads achieved. This is done by splitting a circuit into multiple segments of equal depth and converting each segment into a separate measurement pattern. Finally, the patterns are linked in sequence. This approach is designed to make efficient use of the limited resources available in the NISQ era. I also introduce algorithms using graph theoretic techniques such as local complementation and pivoting to reduce the degree of graph states, for easier implementation on restricted architectures. Finally, I show the first experimental realization of the one-way model of a non-trivial quantum circuit on an ion trap device, also demonstrating the use of the hybrid model and local complementation algorithms in practice.

Impact Statement

The work done, techniques developed and results produced by this project are useful both within and outside academia. The hybrid model of quantum computing which has been developed and is presented in this work is an entirely new technique which provides a different perspective on the one-way model and allows some measurement patterns to be implemented on devices that would otherwise not have a sufficient capacity of physical qubits to enable this. This technique can be of interest to researchers working on near-term quantum devices or applications of measurement-based quantum computing, as well as individuals and companies working in the fast-developing industry of quantum computers in general. A second technique developed in this project is a family of algorithms which take advantage of the graph theoretic properties of local complementation and pivoting to reduce the entanglement depth of measurement patterns. These algorithms can be of great benefit to any implementation of the one-way model because they are computationally cheap to execute and can reduce the number of gates required to implement a measurement pattern with no downside. During this project, I have demonstrated not only the aforementioned techniques, but also one of the first ever implementations of the one-way model of quantum computing (with adaptive measurements) on a real ion trap device. Consequently, in addition to researchers working on near-term devices or measurement-based quantum computing, this project is also of interest to those working on the ion trap branch of quantum computers in particular. In fact, this project has been co-funded by Quantinuum precisely because of its industrial relevance and its overlap in the fields of NISQ computing, circuit compilation and ion trap hardware.

Acknowledgements

This work was supported and funded by the Engineering and Physical Sciences Research Council [EP/L015242/1] and Quantinuum (formerly Cambridge Quantum Computing). I want to thank the authors of the *PyZX* [1] library for the variety of tools they have developed which have been used throughout this project. I also want to thank my supervisors, Professor Dan Browne of UCL and Dr. Ross Duncan of Quantinuum. Finally, I want to thank the authors of the Pytket library (within Quantinuum) which I have also used extensively [2].

Contents

1	Introduction	8
2	Background Material	18
2.1	Introduction	18
2.2	Circuit model	19
2.3	Definitions of width and depth	22
2.4	Measurement-based quantum computation	23
2.5	Graph states	26
2.6	ZX-calculus	32
2.7	Upper bound for graph width	37
2.8	Circuit generation procedure	40
2.9	Determinism in MBQC and Pauli flow	43
2.10	The need for a hybrid model	44
3	Hybrid model of quantum computation	46
3.1	Introduction	46
3.2	Definition of the hybrid model	47
3.3	Hybrid model with one MBQC segment	49
3.4	Hybrid model with two MBQC segments	52
3.5	Hybrid model with n MBQC segments	52
3.6	Resource comparison in the hybrid model	57
3.7	Routing for restricted architectures	62
3.8	Summary of the hybrid model	67

4	Local complementation passes	69
4.1	Introduction	69
4.2	Local complementation algorithm	70
4.3	Pivoting algorithm	73
4.4	Composite algorithms	78
4.5	Result comparison for degree reduction algorithms	81
5	Ion trap implementation	88
5.1	Introduction	88
5.2	Motivation for the experiment	89
5.3	Quantinum H1-1 device specifications	90
5.4	Experiment overview	91
5.5	Result discussion	92
5.6	Experiment summary	99
6	Conclusions and future work	101
	Bibliography	105

Chapter 1

Introduction

The idea of quantum computing has been around since the early 1980s when multiple authors first discussed the potential of computers that use quantum mechanics to simulate physics [3, 4, 5]. In 1980, Russian mathematician Yuri Manin first mentioned the idea of a quantum automaton that used superposition and entanglement [3]. In the same year, Paul Benioff talked about a quantum mechanical Hamiltonian functioning as a model of Turing machines [4]. More famously, the beginning of quantum computing is often credited to Richard Feynman for his 1982 journal article titled “Simulating Physics with Computers” [5].

In the following decade, multiple algorithms were published which, although not having direct applications, demonstrated that more information can be obtained by querying a black box in a quantum superposition [6, 7, 8]. In 1994, mathematician Peter Shor’s quantum algorithm for factorization sparked widespread interest and excitement in the field, as it simultaneously represented both a concrete application of quantum computing in breaking widely used classical encryption strategies and even more importantly, an area where quantum computers have the potential to outperform classical machines [9]. In 1996, computer scientist Lov Grover published a quantum algorithm which provides a quadratic speedup in the unstructured search problem [10].

As early as 1995, a fundamental quantum logic gate - the controlled-NOT gate - was first demonstrated on a system of trapped ions [11]. In 1996, David DiVincenzo proposed a list of minimal requirements for the realization of a quantum computer

[12]:

- Multiple qubits are needed, each of which is a 2-level system with an energy gap that will always have a state in the subspace of these 2 levels.
- The qubits must be initialised in a known state.
- The qubits must have a long enough coherence time to allow all the quantum gates and measurements to be performed during runtime.
- A universal set of quantum gates must be available (explained below).
- The qubits must be able to be measured.

Expanding on the fourth requirement, a universal set of quantum gates is one that can reproduce any unitary operation to arbitrary accuracy using a finite sequence of gates [13, 14].

Because chapter 5 of this thesis discusses an implementation of the techniques introduced in earlier chapters on an ion trap device, I will elaborate a bit further on how such devices commonly meet the DiVincenzo criteria. Typically, ions of various molecules are confined and suspended through the use of an electromagnetic field and the two-level systems required for the initialisation of qubits are achieved using the electrons of the suspended ions [14]. Specifically, the electrons are excited through a technique known as *optical pumping* which dates back to the 1960s and which involves the electrons being raised to a higher energy level through the use of a laser, before being allowed to decay to a lower state that is no longer coupled with the laser, forming the ground state of the 2-level system [15].

Ion trap qubits are usually measured using lasers that are coupled to only one of the two possible measurement outcomes [16]. If the superposition collapses to that outcome, the laser will continuously excite the electron, which then proceeds to continuously decay, emitting photons that can be collected through a photomultiplier tube or charge-coupled device camera [16]. Conversely, if the superposition collapses to the alternative state, no photons are detected [16].

Single-qubit gates in ion trap devices are often accomplished using magnetic dipole transitions, by exposing the ions to an external electromagnetic field with a controlled frequency [17]. As for CNOT gates, they are typically implemented using the Cirac-Zoller scheme proposed in 1995 where the interaction between a pair of qubits is mediated by the joint motion of the involved ions [18]. It should also be noted that ion trap qubits are inherently robust, exhibiting longer coherence times compared to other technologies which is one of the reasons they were chosen for our experiment [19].

The first physical realisation of a quantum computer dates back to the 2-qubit experiment in 1998 by Chuang et al. which demonstrated Grover's algorithm on a system with four states, using nuclear magnetic resonance techniques on a solution of chloroform molecules [20]. In 2001, an alternative technology for implementing quantum computing using single-photon sources, linear optical elements and single-photon detectors was shown to be possible, laying the foundation for the field of linear optical quantum computing [21].

The first experimental demonstration of entanglement for a system of five photons was achieved in 2004 [22]. A 12-qubit machine using nuclear magnetic resonance was benchmarked in 2006, marking the first time a quantum computer with a number of qubits in the double digits was shown [23]. The first quantum controlled-NOT gate acting on a pair of superconducting qubits was shown in 2007 [24] - an important milestone considering the relative success achieved by superconducting quantum devices in later years. In 2015, a 9-qubit superconducting quantum computer was used to simulate a hydrogen atom [25]. This was followed by rapid developments in 2017, when IBM revealed 17-qubit [26] and 50-qubit [27] superconducting devices, advancing the technology to a more central role in the field. Google broke this record in 2018 with their 72-qubit superconducting device "Bristlecone" [28].

In 2018, the first ion trap device with a double-digit number of qubits was introduced by IonQ [29, 30]. Specifically, this device can host 160 qubits, although a string of quantum operations was achieved on 79 qubits at a time with only 11

qubits being fully connected and subjected to full quantum operations [29, 30]. Nevertheless, this showed that ion trap devices are competing with superconducting devices as potential candidates for a fault-tolerant machine in the future.

The year 2019 marked an extremely important milestone in the field, as Google published a paper claiming quantum supremacy was achieved on a superconducting quantum computer with 53 qubits [31]. Specifically, they demonstrated that the aforementioned device could complete a task which cannot be replicated by a classical computer in the same timeframe. In 2023, IBM announced a 433-qubit superconducting device [32].

In the late 20th century, quantum computing focused on the circuit model which is structured - perhaps out of convenience - in a way analogous to how classical computers work: A number of reversible quantum logic gates are sequentially applied on quantum bits, much like classical computers rely on classical logic gates acting on classical bits [13]. However, in 2001, Robert Raussendorf and Hans J. Briegel introduced an alternative model of quantum computation, by showing how rather than reversible quantum gates it is possible to instead process the input state through a series of destructive measurements, thus introducing measurement-based quantum computing which is more suitable for linear optical devices [33]. In 2007, the measurement calculus was introduced by Danos et al. which further streamlined measurement-based quantum computing by introducing measurement patterns - a more intuitive way to view measurement-based computations [34].

Parallel to these developments, the field of categorical quantum mechanics was publicized in 2004 by Samson Abramsky and Bob Coecke as a way to apply paradigms from mathematics and classical computer science to describe the physical processes that take place in quantum computing [35]. In this setting, sequential processes are described by morphisms while parallel processes are described by tensor products [35]. While wires in these diagrams can be bent, they are always two-ended as a way to capture the no-cloning theorem [36, 37, 38] and no-deletion theorems [39, 40] found in quantum information in contrast to classical information [41].

In 2008, categorical quantum mechanics were extended to ZX-calculus by Bob Coecke and Ross Duncan, which is a rigorous graphical reasoning language used to describe linear maps between qubits [42]. Many of the rewrite rules used today were present in the original paper, with the addition of the Euler decomposition of the Hadamard gate introduced in 2009 by Ross Duncan and Simon Perdrix [43]. These rewrite rules form a complete set for diagrams with phases up to $\frac{\pi}{2}$ as shown in 2013 by Miriam Backens [44]. This result was later extended to include completeness for the approximately universal case including phases up to $\frac{\pi}{4}$ [45] and the universal case where phases can take any arbitrary value in 2017 [46, 47]. This means that any linear map between qubits can be represented with a ZX-diagram, which encompasses all quantum circuits and measurement patterns. Consequently, ZX-calculus is a convenient tool for bridging the standard circuit model and MBQC representations as well as an intuitive way to describe measurement patterns in particular with some adjustments.

At this point, a distinction must be made between ‘logical’ qubits needed in quantum algorithms and ‘physical’ qubits - the actual implementations of quantum bits of information in some real 2-level system, whether that is accomplished via trapped ions, superconducting circuits, entangled photons or any other technology. While experimentalists strive to match the properties needed for quantum computation (as described by the DiVincenzo criteria) as closely as possible, physical systems are imperfect and prone to various types of quantum errors. Any non-isolated quantum system - which by necessity includes any possible quantum computer - is subject to *decoherence* due to contact with its environment which is a memory issue and puts a time limit to the quantum computation that is determined by the properties of the underlying technology [48, 49].

Other sources of quantum errors include initialisation errors, gate errors and measurement errors which are caused by quantum operations which do not perfectly conform with the underlying mathematics 100% of the time [14]. While the effect of an individual faulty operation might be relatively small, a deep circuit can accumulate errors from multiple such operations until the real state of the system

deviates substantially from its intended state [14]. Such errors accumulating from multiple imperfect operations are often referred to as *depolarizing noise* and modelled using a depolarizing channel [14]. When the depolarizing channel is strong enough, the state of the output tends to the maximally mixed state regardless of the state of the input [14].

Due to the presence of various sources of quantum errors, the notion of *state fidelity* [50, 51] is often used in the field which is, in general, a measure of how “close” one quantum state is to another. State fidelity is especially useful in quantifying how close the quantum state achieved by an experimental realisation of a quantum computation is to the *theoretical* or *intended* state. Improving the state fidelity of quantum computations is necessary to make quantum computers practically useful and can be achieved through *error suppression*, *error mitigation* or *error correction* [52, 14, 53, 54]. While these terms sound similar, they differ in that they refer to tackling quantum errors at different levels and to different extents.

Error suppression generally involves minimizing the occurrence of quantum errors and can be achieved by relatively simpler strategies such as reducing the depth of a quantum circuit to minimize the effect of decoherence or reducing the number of quantum gates involved to minimize depolarization. Error mitigation often refers to strategies that eliminate the impact of errors on the measurement outcomes by making clever use of the underlying mathematics. Quantum error correction is much more costly to use as it involves using many more ancillary qubits to spread out quantum information and uses non-destructive measurements (such as parity checks on multiple qubits) to detect and remove quantum errors before they become a problem, in a way inspired by how redundancy is used in classical error correction [52, 14, 53, 54].

Quantum error correction is typically viewed as an important long-term goal in the field, as when physical error rates become lower than a certain threshold, it will allow for arbitrarily deep circuits to be executed with high output fidelity through the correct application of error correcting codes [55, 56, 57]. This will enable fault-tolerant quantum computing in a manner similar to fault-tolerance in

classical computers [58].

As I write this thesis, despite the rapid progress in recent years, a large-scale universal quantum computer remains a distant goal. There are a lot of challenges that must be overcome before such machines can be realised, some of the most serious ones being physical scalability and practical error correction. Consequently, some of the more powerful quantum algorithms, such as Shor's algorithm or Grover's algorithm, are more difficult to implement in the short-term.

However, smaller machines that support under a hundred physical qubits have been available for some time [31] and others that support over a hundred qubits are already under development [32]. Such devices are prone to various types of quantum noise in their gates and measurements and because of that, they are often referred to as noisy intermediate-scale quantum (NISQ) devices [59]. It is conceivable that the first useful applications of quantum computing will emerge among classes of quantum algorithms with less steep requirements, such as the variational quantum eigensolver (VQE) [60, 61, 62], the quantum approximate optimisation algorithm (QAOA) [63], or perhaps other approaches in the fields of quantum simulation and quantum chemistry [5, 64]. It is also possible that the first examples of useful applications could be realisable on one of the NISQ devices already in existence or ones that may emerge in the near-term. The primary focus of this thesis is to investigate how to optimise quantum circuits so that they can leverage the available resources as efficiently as possible in this NISQ setting. Near-term quantum devices are characterised by limited numbers of physical qubits prone to noise [59], creating a need for circuits to be compiled in a smart way to take advantage of the available resources.

Much like compilers for classical computers are used to convert programs written in a high-level language into machine code that is executable on a device [65], there exist compilers such as Qiskit [66] and Pytket [2] which do the same for quantum computers. In fact, heavy emphasis is given to the Pytket library in this thesis, as the tools included in the library are very compatible with several of the theoretical concepts discussed and used throughout the thesis, such as measurement-based

quantum computing, ZX-calculus and Pauli flow.

This thesis explores the relationship between the circuit model [12], measurement-based quantum computing [67, 34] and ZX calculus [42]. Most of the topics discussed in this thesis involve the conversion of quantum circuits into ZX diagrams which allow for convenient simplifications and further transformations into measurement patterns or back into circuits.

Circuit width and depth are not fixed, but depend heavily on the specific device or platform on which a quantum circuit is to be implemented. Consider the following examples:

- A quantum circuit of width 20 is to be implemented on a device with a capacity for 10 qubits. In this situation, there are simply not enough physical qubits to fit the circuit on the device, so the priority is to try and reduce the width of the circuit at any cost, otherwise the other resources become completely irrelevant.
- A simulation where the ground state of a molecule can be estimated with insufficient accuracy using a circuit of width 30, on a device with a capacity for 70 qubits. In this case, the available capacity is more than twice what is required, so it could be useful to convert the circuit to an equivalent circuit which requires more qubits, but fewer 2-qubit gates. In this way, the excess width available on the device is used to achieve an improvement in the accuracy of the simulation.

The technology on which a quantum device is based also affects how we evaluate these resources. For ion trap devices, depolarizing noise is less prominent than it is for, say, superconducting devices because ions have longer coherence times. However, the runtime of a circuit implemented on an ion trap device is also much longer. Consequently, depth and entanglement depth are important for both technologies, but for different reasons.

While this thesis explores various techniques to convert or otherwise manipulate the resources required by a quantum circuit, it is always important to contrast

them against device specifications to ultimately decide which form of the circuit is most suitable for implementation, on a case by case basis.

Chapter 2 focuses on the background material required to understand the techniques presented in this thesis. This background material includes an introduction to the circuit model [12], an introduction to the one-way model of measurement-based quantum computation [67, 34], some relevant graph theoretic techniques such as local complementation and pivoting [68, 69, 70] and an introduction to ZX-calculus [42]. Additionally, notions of causal flow [71] and Pauli flow [72] which are conditions sufficient for determinism in the one-way model are also introduced.

Chapter 3 focuses on the trade-off between the number of qubits (width) and the number of timesteps (depth) required to execute a given circuit or an equivalent measurement pattern. This chapter introduces the hybrid model which allows circuits to be converted into a sequence of small measurement patterns to fine-tune the circuit width/depth trade-off to fit on a specified platform. In general, measurement patterns require more physical qubits but exhibit a lower depth than their equivalent circuits, with the hybrid model implementation existing in a spectrum bounded by these two extremes. In addition to discussing the development process and the advantages/disadvantages of the hybrid model, the same chapter explores some approaches for routing circuits in the hybrid model on devices with restricted qubit connectivity as well as the impact of these routing approaches on the resulting depth. We find that the one-way model is less compatible with routing than the circuit model because converting a circuit to a measurement pattern often increases the number of neighbouring qubits that each physical qubit has to interact with, which translates to more SWAP gates being necessary during runtime. I have worked on two separate solutions to this incompatibility between the hybrid model and routing which are the main focus of chapters 4 and 5.

Chapter 4 focuses on some strategies for reducing the entanglement depth of measurement patterns by borrowing graph theoretic techniques such as local complementation and pivoting. By specifically targeting high-degree vertices in the graph state, it becomes possible to rewrite the corresponding ZX-diagram into one

of lower degree - reducing the entanglement depth of the resulting measurement pattern in the process. An important benefit of this approach is that it can potentially reduce the depth overhead introduced by routing.

In chapter 5, I describe an experiment involving the physical realisation of the topics discussed in chapters 3 and 4 on a 20-qubit ion trap device. Specifically, a 3-qubit circuit is split into two subcircuits, each of which is then converted to a measurement pattern. Subsequently, each of the measurement patterns is subjected to the depth reduction techniques involving local complementation before ultimately being implemented on one of Quantinuum's H1-series ion trap devices. The output distribution is then compared against that of the original circuit. One of the reasons for choosing an ion trap device for this experiment is due to the higher qubit connectivity compared to an alternative such as a superconducting device. This indirectly serves as another solution to avoid adding routing overheads to the one-way model because all the qubits are fully connected to one another in this setting and no SWAP gates need to be added. Using an error mitigation technique to remove depolarizing noise from the experimental results [73], it was possible to obtain an output distribution for the hybrid model which conforms better to the output distribution of the noise-free benchmark when compared to the circuit model.

Chapter 2

Background Material

2.1 Introduction

In this chapter I introduce the technical background required by the later chapters of the thesis. Specifically, I introduce the circuit model first introduced in the 1980s [6], which was the first formal description of quantum computations and was designed in a way analogous to classical circuits used by modern computers. Furthermore, I introduce the notions of circuit width and depth (which I also apply to measurement patterns later on) as limited resources needed by quantum computations and which have to be managed efficiently.

Subsequently, I introduce measurement-based quantum computation, a different yet equivalent approach to the circuit model where quantum operations are performed via irreversible measurements instead of reversible quantum gates [33]. Measurement-based quantum computation allows quantum operations to be implemented at a lower depth and requires a resource known as a graph state, the properties of which allow for some optimization strategies that become more important in chapter 4.

ZX calculus is a diagrammatic reasoning language that is also introduced in this chapter and serves as a convenient tool for transpiling quantum circuits from the circuit model to the one-way model [42]. Additionally, it contains a number of rewrite rules that can be used to simplify circuits and measurement patterns by removing redundant elements from their description [42].

The width of a graph has an upper bound determined by the width of the quantum circuit it has originated from as well as the number of non-Clifford gates involved in that circuit [74]. In this chapter, I include a derivation of this upper bound for graphs with measurements restricted to the XY plane.

Additionally, two separate types of circuit samples have been used in this thesis. The first set of samples is randomly-generated Clifford+T circuits which has been chosen due to being a general class which uses a universal gate set. The second sample class are phase polynomial circuits which have applications in quantum chemistry and VQE algorithms [75, 76]. In this chapter, I include a section with more details on the specifics of the circuit generation procedures for these sample classes.

Quantum circuits presented in the circuit model are inherently deterministic, but that is not true for a general ZX diagram. One of the requirements for implementing a ZX diagram as a measurement pattern is to ensure that it is deterministic, which leads to a discussion about Pauli flow, a condition necessary and sufficient for determinism in the one-way model [72], found in section 2.9 of this chapter.

2.2 Circuit model

The most commonly used resource in quantum computing are quantum bits of information or *qubits* (a term first coined in 1995 [77]) which are typically realised using quantum mechanical 2-level systems [12]. While a qubit in general can exist as a coherent superposition of two orthonormal basis states, upon measurement it collapses to one of the two possible states with a probability dependent on the amplitudes in the superposition [14].

The *Dirac notation* or *bra-ket notation* is commonly used to represent the state vectors of qubits with pure states $|0\rangle$, $|1\rangle$ and arbitrary superposition state $|\psi\rangle$ as shown in equations 2.1, 2.2 and 2.3 [78]. Note that the probability amplitudes α and β are complex numbers:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad (2.1)$$

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad (2.2)$$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}. \quad (2.3)$$

While a single qubit is said to span a 2-dimensional Hilbert space, a set of n qubits forms an n -qubit register which is represented by the tensor product of the individual qubit state vectors and spans a 2^n -dimensional Hilbert space [14].

Quantum logic gates are reversible logic gates that form a parallel to classical logic gates (such as AND, OR, NOT, NAND, NOR, XOR) used in conventional computers [79]. Quantum logic gates act on state vectors and are commonly represented with $2^n \times 2^n$ unitary matrices U such that $UU^\dagger = U^\dagger U = I$. The group of all such gates joined via the group operation of matrix multiplication is known as the symmetry group $U(2^n)$ [79].

A *quantum circuit* is a computation that consists of a sequence of quantum gates and measurements acting on a number of input qubits which are previously initialised in some known quantum state(s) [12]. The *circuit model* generally refers to the approach of viewing quantum computations in terms of quantum circuits, in a way analogous to classical computers, which in many cases is more convenient than using Hamiltonians [13].

More informally, the circuit model can be thought of as a 2-dimensional coordinate system made of a spatial dimension spanning all the qubits involved in the quantum circuit and a temporal dimension spanning all the timesteps during which quantum operations can be executed [6]. Quantum logic gates are thus placed in this coordinate system such that only one gate can act on any qubit during a single timestep, but multiple gates can be executed in parallel during the same timestep as long as they act on different sets of qubits.

A universal set of quantum gates and the one which is used predominantly throughout this thesis is the Clifford+T set which consists of two single-qubit gates (Hadamard and T gates) and the CNOT gate acting on two qubits, which are defined

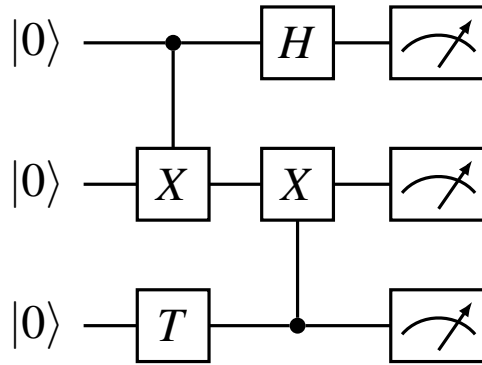


Figure 2.1: An example of a simple quantum circuit acting on three qubits. This circuit uses the Hadamard, T and CNOT gates which form a universal gateset. Due to some operations being performed in parallel, only 3 timesteps are needed to run this circuit even though it involves 7 different operations.

in equations 2.4, 2.5 and 2.6 respectively. The Clifford group can be generated using Hadamard, CNOT and S gates (with S gates being equivalent to a sequence of two T gates) and is a group of gates which maps tensor products of Pauli matrices to other tensor products of Pauli matrices through conjugation [80].

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad (2.4)$$

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}, \quad (2.5)$$

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (2.6)$$

The Clifford group is of particular importance in quantum computing, because according to the theorem by Daniel Gottesman and Emanuel Knill, any circuit comprised entirely of Clifford gates can be efficiently simulated on a classical device [80]. Yet, with just the addition of any non-Clifford gate (such as the T gate), uni-

versality can be achieved [13, 14]. An example of a quantum circuit acting on three qubits can be seen in figure 2.1.

It should be noted that in this thesis, the definition of the CM does not include/allow the existence of Quantum Fan-out gates [81]. Such circuits can potentially use ancilla qubits to achieve depth reduction, much like what is possible through MBQC. However, the implementation of such Fan-out gates is also non-trivial [81].

2.3 Definitions of width and depth

To discuss how to optimise the use of the limited resources available, it is important to first define what these resources are. I define *width* as the number of physical qubits required to implement a quantum circuit. *Depth* is the number of timesteps required to execute a quantum circuit. *2-qubit depth* or *entanglement depth* is the number of timesteps, which involve at least one 2-qubit gate, required to execute the circuit.

A quantum circuit of width w_C can only be executed on a device with a qubit capacity of w_D iif $w_C \leq w_D$. Circuit depth is different from the number of required operations because quantum operations can be performed in parallel as long as they act on different qubits, as shown in figure 2.1. A circuit with an excessively high depth comes at the cost of additional exposure to errors such as depolarizing noise [14]. Therefore, it is ideal to minimize the depth of a circuit when possible. However, some sources of errors (such as gate errors, measurement errors and cross-talk) have less to do with the runtime of the circuit and more to do with the actual number of operations involved. The gate fidelity of 2-qubit gates is often at least an order of magnitude lower than the fidelity of single qubit gates, for many devices currently available (e.g. *ibm.algiers* [82], *Quantinuum H1-1* [23]), so the total number of 2-qubit gates involved in the circuit is also a variable that must be taken under consideration and minimized where possible.

2.4 Measurement-based quantum computation

Measurement-based quantum computation (MBQC) is an alternative but equivalent approach to the more commonly used circuit model. In the circuit model, state processing is accomplished exclusively via quantum logic gates while measurements are reserved until the very end, with the sole purpose of reading out the final state of the outputs. In MBQC, ancilla qubits are initialised and entangled with the inputs and measurements are used throughout the computation to process the state of the inputs by carefully and irreversibly collapsing parts of the superposition. A final round of measurements is still necessary at the end to read out the state of the output qubits [67, 34].

Due to the non-deterministic nature of these destructive measurements used for processing, it is necessary to apply Pauli corrections on the remaining qubits following each such measurement, to ensure that the overall computation remains deterministic until the end [67, 34]. However, instead of having to apply these corrections explicitly as additional operations, it is possible to instead absorb them into subsequent measurements by shifting the measurement basis accordingly, which creates a new requirement for efficient MBQC in the form of *adaptive measurements* [67, 34].

Despite the added cost of ancilla qubits and adaptive measurements, MBQC is also universal and it allows circuits to be implemented at a lower depth (and lower entanglement depth) [83, 34]. An example of a simple MBQC circuit that implements a Hadamard gate can be seen in figure 2.2. In this example, the first qubit is initialised in an arbitrary state $|\psi\rangle$ while a second qubit is an ancilla initialised in the $|+\rangle$ state. The qubits are entangled via a CZ gate and after a local rotation, the first qubit is measured. This partially collapses the superposition such that the state $|\psi\rangle$ is teleported to the ancilla qubit. Depending on the classical outcome of the measurement, the state of the second qubit branches out to one of two possibilities, thus resulting in a non-deterministic outcome. To obtain a deterministic outcome, a conditional X operation is applied on the second qubit and is dependent on the outcome of the measurement of the first. Similarly, an MBQC circuit that teleports

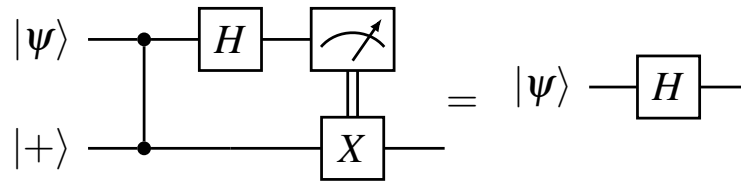


Figure 2.2: A simple measurement-based quantum computation circuit showing how a Hadamard gate can be applied via teleportation. The first qubit is an input in an arbitrary state $|\psi\rangle$ while the second qubit is an ancilla initialised in the $|+\rangle$ state. After the MBQC circuit is performed, the second qubit becomes an output with the state of $H|\psi\rangle$.

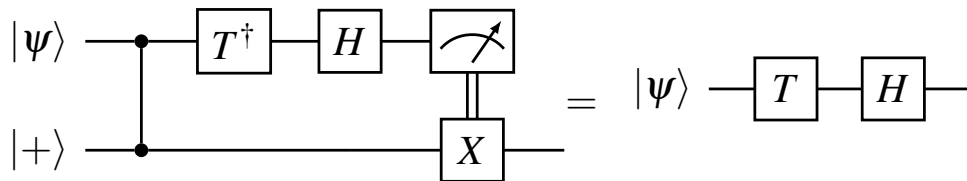


Figure 2.3: A simple MBQC circuit showing how a T gate can be applied via teleportation. The first qubit is an input in an arbitrary state $|\psi\rangle$ while the second qubit is an ancilla initialised in the $|+\rangle$ state. After the circuit is performed, the second qubit becomes an output with the state of $HT|\psi\rangle$.

a Z-rotation such as T gate can be seen in figure 2.3.

As seen in figures 2.2 and 2.3, MBQC circuits appear very similar to regular quantum circuits. The main differences between the two is that adaptive measurements and ancilla qubits are required in the MBQC setting. It is worth noting that all the components for universality are present in the above examples. Specifically, both T gates and Hadamard gates can be achieved via teleportation and both of these patterns utilise a CZ gate which can be used along with Hadamard gates and T gates to complete the Clifford+T gate set required for universality. Consequently, since both the circuit model and MBQC are universal, they are equivalent and it is possible to rewrite any quantum circuit into an MBQC circuit by simply substituting all the operation with the corresponding MBQC components.

However, this is not necessarily the most efficient way to convert circuits to MBQC and thus, we make use of the *measurement calculus* formalism which was

first introduced in 2007 by Danos et al. [34]. Under this formalism, we move away from MBQC circuits to more specialized resources known as *measurement patterns* as described by definition 2.4.1.

Definition 2.4.1 *In general, a measurement pattern [34] \mathbb{P} acts on a set of qubits $i \in V$ of which the subsets $I, I^C, O, O^C \subseteq V$ represent the inputs, prepared qubits, outputs and measured qubits respectively. The operations that make up the pattern are separated into the following four types:*

- *Preparations are denoted by N_u and represent the initialisation of qubit $u \in I^C$ in the $|+\rangle$ state.*
- *Entanglement operators are denoted by E_{ij} and represent CZ gates between pairs of qubits $i, j \in V$.*
- *Destructive measurements are denoted by $M_v^{\lambda, \alpha}$ and represent a measurement of qubit $v \in O^C$ in the plane $\lambda \in \{XY, XZ, YZ\}$ rotated by phase α .*
- *Corrections are denoted by $P_k^{s_v}$ and represent Pauli rotations $P \in \{X, Z\}$ acting on qubit $k \in V$ and conditional on the signal $s_v \in \{0, 1\}$ representing the classical outcome of a measurement on qubit $v \in O^C$.*

A general measurement pattern takes the form shown by equation 2.7,

$$\mathbb{P} = \prod_{v \in O^C} (P_k^{s_v} M_v^{\lambda, \alpha}) E_{ij} N_u. \quad (2.7)$$

It is possible to concatenate multiple measurement patterns as the preparation step commutes with all the other operators and the entanglement operators also commute with the Pauli corrections up to a local rotation as shown in equations 2.8 and 2.9. Consequently, regardless of the number of measurement patterns concatenated, the resulting pattern can always be rewritten in the form shown by equation 2.7 [34].

$$Z_i^{s_u} E_{ij} = E_{ij} Z_i^{s_u}, \quad (2.8)$$

$$Z_j^{s_u} X_i^{s_u} E_{ij} = E_{ij} X_i^{s_u}. \quad (2.9)$$

As multiple measurement patterns are concatenated, their entanglement operators combine into the term E_{ij} , which now represents a network of connections between the qubits involved in the measurement pattern. This network can be visually represented as a diagram of nodes interconnected by edges. Conveniently, CZ gates are not only symmetrical but also commute with one another and thus the diagram can remain very simple, without the need for any special ordering or directionality to be added to the edges. This diagram is commonly referred to as a *graph state* [84].

The main advantage of viewing these entanglement operators as graph states is that this allows some useful techniques to be borrowed from graph theory as expanded upon in section 2.5. It should also be noted that the more crude version of MBQC which achieved computation from the perspective of state teleportation introduced in the earlier years is equivalent to the graph state representation that became popular more recently [85].

2.5 Graph states

As explained in section 2.4, it is possible to concatenate multiple measurement patterns and to merge their entanglement operators and preparatory steps into a broad term of form $E_{ij}Ni$. Conveniently, this term has the property of being easily represented using a graph of nodes joined together with directionless edges. The result is known as a graph state and is the primary resource required for the implementation of a measurement pattern. More formally, a graph state is defined as follows:

Definition 2.5.1 An undirected graph [86] $G = (V, E)$ is defined by a set of vertices $i \in V$ as well as a set of edges $(i, j) \in E$ between pairs of vertices $i, j \in V$.

Definition 2.5.2 An open graph [86] $G = (V, E, I, O)$ is an undirected graph with some additional defined subsets of vertices, $I, O \subseteq V$ representing inputs and outputs respectively.

Definition 2.5.3 A graph state [84] $|G\rangle$ based on graph G is defined as shown in

equation 2.10 and is the resource state required for a measurement pattern.

$$|G\rangle = \left(\prod_{(i,j) \in E} CZ_{ij} \right) |+\rangle_i^{\otimes i} \quad (2.10)$$

A graph state consists of the preparation and entanglement steps of the measurement pattern. The sets of qubits, entanglement operators, inputs and outputs of the graph state correspond to the sets of vertices, edges, inputs and outputs of the underlying simple graph G . The edges of G are unordered and there are no self-loops.

By mapping the graph states used in measurement patterns to graphs, it becomes possible to take advantage of some graph theoretic properties and techniques.

Definition 2.5.4 *The neighbourhood [87] of u , $N_G(u)$ is the set of vertices $v \in V$ where $(u, v) \in E$.*

Definition 2.5.5 *The degree of a vertex [87] is equal to the number of its neighbours, $\text{deg}(u) = |N_G(u)|$*

Definition 2.5.6 *The degree of a graph [87] is denoted by Δ_G and is the maximum degree of any vertex in the graph.*

It is possible to leverage the graph-like structure of the qubits in the graph state by making use of the stabilizer formalism [80]. An arbitrary error U can be represented by a discrete superposition of Pauli operators [80]. The stabilizer formalism allows for a way to detect these Pauli errors when they act on a quantum state, using only measurements which commute with the observables of interest, so as to avoid collapsing the superposition in a way that is detrimental to the computation taking place [80]. This is accomplished by defining a group of operators known as *stabilizers*, S , which typically consist of parity checks across multiple qubits [80]. The stabilizers are constructed carefully so that they anti-commute with individual Pauli errors so that the errors can be spotted when a stabilizer is measured.

This becomes relevant in the construction of graph states because their structure allows for the existence of intuitive stabilizers known as *graph stabilizers* [88]. Because the graph state is constructed using a network of CZ gates, it is possible to

use the commutation relationship of equation 2.9 to correct a Pauli-X error acting on a qubit i by adding a Pauli-Z correction on each of its neighbours, $j \in N_G(i)$. This leads to the definition of the graph stabilizer group shown in equation 2.11,

$$S_i = X_i \prod_{j \in N_G(i)} Z_j. \quad (2.11)$$

The product of the graph stabilizers given by equation 2.11 can be used to define the graph state and is in fact an equivalent but alternative definition to that of equation 2.10 [88]. The graph state is invariant under the operation defined by the graph stabilizers and we can take advantage of this property to correct Pauli errors and measurement errors.

Constructing a graph state requires that the physical qubits involved are entangled in a certain way that is indicated by operator E_{ij} . However, implementing this operator on a real device can often depend on what 2-qubit gates are allowed by its architecture as the relevant pairs of qubits are not necessarily physically adjacent.

Due to this limitation, it is useful to have tools which allow for an equivalent measurement pattern with a modified edge set E'_{ij} to be executed, such that it can more easily fit on the restricted architecture. This is where we turn to an interesting property of graph states which emerges from the interaction between graph stabilizers and unitary operations [88]. Consider an arbitrary unitary operation U acting on the graph state as shown in equation 2.12,

$$\begin{aligned} S_i |G\rangle &= |G\rangle, \\ US_i |G\rangle &= U |G\rangle, \\ US_i U^\dagger |G\rangle &= U |G\rangle. \end{aligned} \quad (2.12)$$

It is important to note that the expression $US_i U^\dagger$ is functionally a new graph stabilizer which leaves the new state $U |G\rangle$ invariant. Therefore, it suffices to observe the new stabilizer group $US_i U^\dagger$ in order to understand the effect of the operator U on the edge set of the graph. Equivalently, we can use local rotations on the vertices of the graph state to change the edge set to something more convenient.

Consider the graph shown in figure 2.4, with graph stabilizer generators $S_g = \{XZZ, ZXZ, ZZX\}$, with the full stabilizer group $S = \{III, XZZ, ZXZ, ZZX, -YYI, -YIY, -IYY, -XXX\}$, under the effect of the local Clifford operator L_C , given in equation 2.13 [84, 89],

$$L_C = \sqrt{X_u}^\dagger \prod_{v \in N_G(u)} \sqrt{Z_v}. \quad (2.13)$$

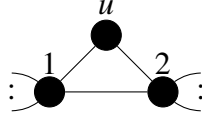


Figure 2.4: Small graph with 3 vertices shown before the application of local Clifford operator L_C defined in equation 2.13, on vertex u .

Under this operation, the new graph is defined by the new stabilizer group $L_C S L_C^\dagger$. The graph stabilizer S_u acting on vertex u , about which we are complementing, is a special case and remains unchanged, indicating that the neighbourhood $N_G(u)$ is not affected, as shown in equation 2.14,

$$\begin{aligned} L_C S_u L_C^\dagger &= (\sqrt{X_u}^\dagger \prod_{v \in N_G(u)} \sqrt{Z_v}) (X_u \prod_{v \in N_G(u)} Z_v) (\sqrt{X_u} \prod_{v \in N_G(u)} \sqrt{Z_v}^\dagger), \\ &= \sqrt{X_u}^\dagger X_u \sqrt{X_u} \prod_{v \in N_G(u)} (\sqrt{Z_v} Z_v \sqrt{Z_v}^\dagger), \\ &= X_u \prod_{v \in N_G(u)} Z_v, \\ &= S_u. \end{aligned} \quad (2.14)$$

As for the entire stabilizer group, it is mapped to the new group $S' = \{III, XZZ, YYZ, YZY, -ZXI, -ZIX, -IXX, -XYY\}$, with the new generators being $S'_g = \{XZZ, -ZXI, -ZIX\}$, indicating that the graph has been transformed to that of figure 2.5, removing the edge between the shared neighbours of u , $E_{1,2}$. In general, the application of the local Clifford L_C on a vertex u brings about a transformation known as local complementation which is one of the tools that allow us to modify the edge set of a graph state and is described in definition 2.5.7 [68, 69, 70].

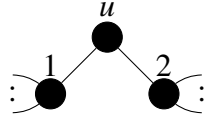


Figure 2.5: Small graph with 3 vertices shown after the application of local Clifford operator L_C defined in equation 2.13, on vertex u .

Definition 2.5.7 Given a graph $G = (V, E)$, local complementation (LC) [68, 69, 70] about vertex $u \in V$ is defined as $G \star u = (V, E')$. For every pair of vertices $v, w \in N_G(u)$, $(v, w) \in E'$ iif $(v, w) \notin E$. All other edges remain unchanged.

A visual demonstration of LC can be found in figure 2.6. The relationship between

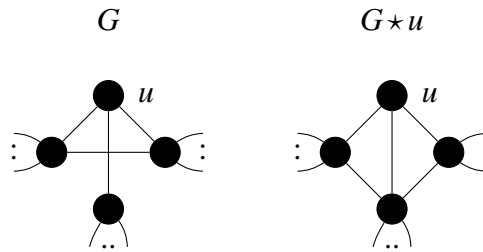


Figure 2.6: Example showing how local complementation of graph G about vertex u affects the neighbourhood of u . Left: $G = (V, E)$. Right: $G \star u = (V, E')$.

graph states before and after the transformation is given by equation 2.15,

$$|G\rangle = L_C |G \star u\rangle. \quad (2.15)$$

While the neighbours of the affected vertex $v \in N_G(u)$ remain unchanged, each vertex v is disconnected from its previously shared neighbours with u and gains a new connection to the remaining neighbours of u [84, 89]. It should be noted that when LC is applied on an open graph, it cannot be applied to the input vertices. By carefully selecting what vertices to perform LC on, it is possible to reduce the entanglement depth of the measurement pattern. More details as to how this is done are found in chapter 4. LC has a convenient representation in ZX calculus which are discussed in section 2.6. Another graph transformation which is derived from LC, called edge local complementation (ELC) is described in definition 2.5.8 [68, 69].

Definition 2.5.8 The edge local complementation (ELC) (also known as pivoting) [68, 69] about edge $(v, w) \in E$ is given by $G^{(vw)} = G \star v \star w \star v = G \star w \star v \star w = (V, E')$.

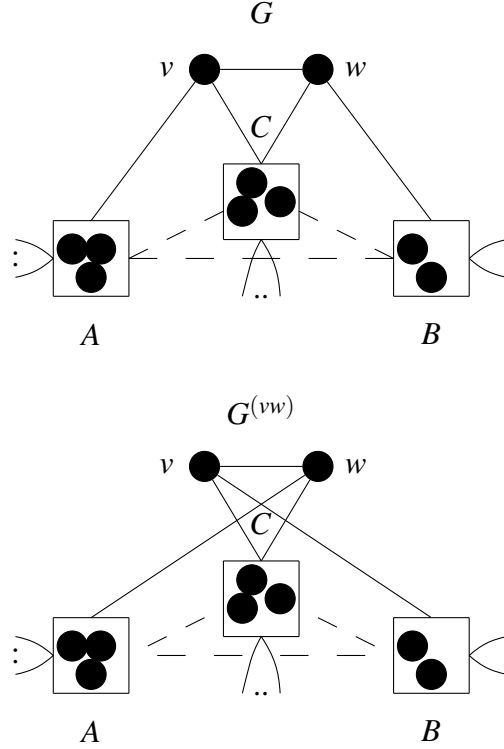


Figure 2.7: Example showing how pivoting graph G about edge (v, w) affects the neighbourhood of vertices v and w . Top: $G = (V, E)$. Bottom: $G^{(vw)} = (V, E')$.

While ELC is equivalent to a sequence of three LC steps on a pair of neighbouring vertices, we give it special importance because of its effect on the neighbourhoods of the affected vertices. We can define subsets $A(G, v, w), B(G, v, w), C(G, v, w) \subseteq V$ where $A(G, v, w) = N_G(v) - N_G(w) - \{w\}$, $B(G, v, w) = N_G(w) - N_G(v) - \{v\}$, $C(G, v, w) = N_G(v) \cap N_G(w)$. Now consider the ELC $G^{(vw)}$. For every pair of sets $X, Y \in \{A, B, C\}$ and for every pair of vertices $x \in X$ and $y \in Y$, $(x, y) \in E'$ iif $(x, y) \notin E$. Furthermore, $A(G', v, w) = B(G, v, w)$ and $B(G', v, w) = A(G, v, w)$. For every other unaccounted pair of vertices, $(x, y) \in E'$ iif $(x, y) \in E$.

A visual demonstration of how the edges of graph G change under pivoting is shown in figure 2.7. It should be noted that the entanglement depth of a measurement pattern is equal to degree of the underlying graph as the edges are imple-

mented via CZ gates which can be parallelised as long as no two gates are acting on the same qubit during the same timestep.

As is the case with LC, ELC can also be used to reduce the entanglement depth of a measurement pattern and in some cases, it can even lead to higher depth reduction than what can be achieved with just LC. Specific algorithms designed to reduce entanglement depth with either LC and/or ELC are introduced and compared in chapter 4.

While the measurement calculus and graph states serve as intuitive representations of MBQC, it is equally important to have a convenient tool for converting quantum circuits to measurement patterns. This role is filled by ZX-calculus which is discussed in section 2.6.

2.6 ZX-calculus

The ZX-calculus is a diagrammatic reasoning language used for representing linear maps and it was first proposed in 2008 as an extension of categorical quantum mechanics [42]. It is convenient in that it is easy to convert both quantum circuits and measurement patterns to and from ZX-calculus, thus serving as a bridge between the two models. This is accomplished through a number of rewrite rules [42] which can be used to transform and/or simplify a ZX diagram into an equivalent diagram. The Euler decomposition of a Hadamard gate using ZX-calculus was also added to the set of rewrite rules in 2009 [43] and in 2014 this set of rewrite rules was proven to be complete [44].

In this section I provide an overview of ZX-calculus including a complete set of rewrite rules which is sufficient for the purpose of this thesis. For a larger assortment of rewrite rules and a more in-depth explanation please see the review paper by van de Wetering [90] but do note that all the additional rewrite rules can be reproduced from the ones present here. Also, note that ZX diagrams in this thesis are defined up to a global phase.

ZX diagrams are composed of two types of elements: *spiders* and *wires*. Spiders can have either the Z basis (coloured green) or the X basis (coloured red) and

or its shape is distorted. Furthermore, all the rewrite rules hold if the colours of all the spiders in the diagram are inverted.

To more easily convert ZX-diagrams into measurement patterns it is necessary to first normalise ZX-diagrams into a form that is more adjacent to that of a graph state. Specifically, a graph state only contains one type of nodes whereas a general ZX-diagram contains both red and green spiders. Similarly, a graph state only contains one type of edges whereas a general ZX-diagram contains both simple wires and Hadamard wires. Luckily, there exist rewrite rules for changing one color of spiders to the other, as well as for merging two spiders of the same color as long as they are connected by a simple wire (thus eliminating the wire in the process). This allows us to transform the diagrams into forms more suitable for MBQC, namely diagrams in *graph state form* and diagrams in *MBQC form* [92], described in definitions 2.6.1 and 2.6.2 respectively.

Definition 2.6.1 *A ZX diagram is in graph state form [92] iif:*

- *There are no input wires.*
- *Every spider in the diagram is a green spider.*
- *Every spider in the diagram has a phase of 0.*
- *Every wire between two spiders is a blue wire (Hadamard edge).*
- *Every spider in the diagram is connected to exactly one output wire.*

A ZX-diagram in graph state form is one that is perfect for representing the entanglement operator E_{ij} of a measurement pattern (i.e. the graph state, hence the name) as it contains no information outside of which qubits are connected. By allowing each of the spiders in graph state form to be further connected to arbitrary inputs or measurement gadgets, it is possible to account for the other components of the measurement pattern such as the preparation operator N_i or the measurement operator $M_v^{\lambda, \alpha}$). With these additions we move from the graph state form to the MBQC form described in definition 2.6.2.

Definition 2.6.2 A ZX diagram is in MBQC form [92] iif:

- Each spider in the graph state diagram may be connected to either one or zero input wires.
- Every spider in the diagram is a green spider unless it is degree-1.
- Every spider in the diagram has a phase of 0 or $\frac{\pi}{2}$ unless it is degree-1.
- Every wire between two spiders is a blue wire (Hadamard edge) unless one of them is degree-1.
- Every spider in the diagram is connected to either one output wire or a degree-1 spider, unless it is itself a degree-1 spider.

Every ZX diagram which represents a quantum circuit or a measurement pattern can be transformed into *MBQC-form* [92]. An example of a ZX diagram in MBQC-form can be seen in figure 2.11. ZX diagrams in MBQC-form can be di-

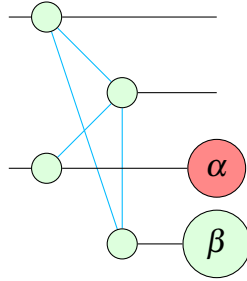


Figure 2.11: Example of a ZX diagram in MBQC-form. This diagram represents a measurement pattern with four qubits, two inputs and two outputs. Of the two non-output qubits, one is to be measured in the YZ plane (rotated by an angle α) and the other in the XY plane (rotated by an angle β).

rectly implemented as measurement patterns. In this case, all the non-input qubits are typically prepared in the $|+\rangle$ state. Subsequently, all the qubits are entangled via CZ gates according to the graph, $G = (V, E)$ defined by the underlying graph state diagram.

It is possible to use LC and ELC on ZX diagrams in MBQC-form [93]. Doing so changes the edges in the graph state diagram in exactly the same way discussed

in definition 2.5.7. However, to preserve the linear map represented by the ZX diagram we need to implement the local Clifford as described in equations 2.13 and 2.15.

Specifically, after performing LC about vertex $u \in V$, we have to add a red spider with a phase of $-\frac{\pi}{2}$ to the output of the spider representing u . Similarly, we have to add a green spider with a phase of $\frac{\pi}{2}$ to the outputs of the spiders representing all the neighbours $v \in N_G(u)$. An example of an application of LC on a ZX diagram in MBQC-form is shown in figure 2.12. ELC can also be applied to ZX diagrams in MBQC-form by simply breaking it into three steps of LC as described in definition 2.5.8. For non-output nodes (i.e. ancilla qubits which are

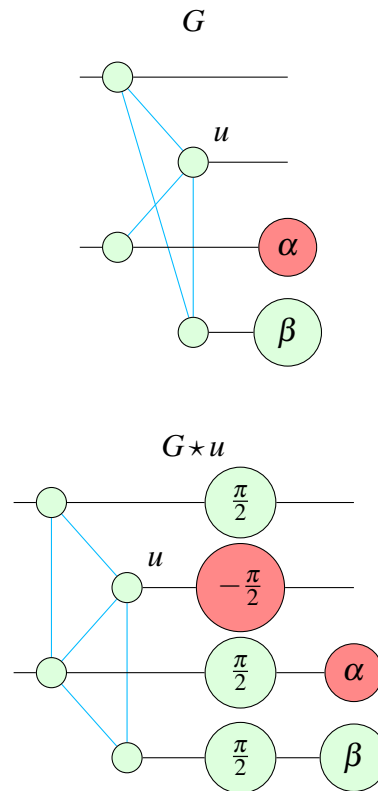


Figure 2.12: Example of local complementation on a ZX diagram in MBQC-form. Top: $G = (V, E)$. Bottom: $G \star u = (V, E')$.

measured during processing) we can remove the added spiders by simply absorbing them into the measurement effects and adjusting the measurement plane/angle accordingly, while for the output qubits these phases can be grouped together and simplified into a small number of single qubit rotations at the end. As a result we

can largely disregard these spiders when considering the transformations caused by local complementation on a ZX diagram. The effect of the accumulated phases on some common measurement effects is shown in figure 2.13.

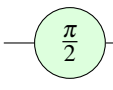
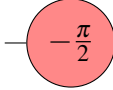
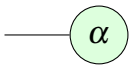
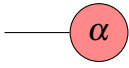
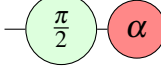
<i>Measurement effect</i>	<i>Measurement plane/angle</i>	<i>Shifted due to:</i> 	<i>Shifted due to:</i> 
	$\langle XY, \alpha $	$\langle XY, \alpha + \frac{\pi}{2} $	$\langle XZ, \alpha + \frac{\pi}{2} $
	$\langle YZ, \alpha $	$\langle XZ, \alpha $	$\langle YZ, \alpha - \frac{\pi}{2} $
	$\langle XZ, \alpha $	$\langle YZ, -\alpha $	$\langle XY, \frac{\pi}{2} - \alpha $

Figure 2.13: Table showing how the phases accumulated through local complementation on a ZX diagram in MBQC-form affect the measurement planes and angles of the non-output vertices.

2.7 Upper bound for graph width

In this thesis, I also use the term "width" to describe the number of nodes in a graph. I derive the upper bound for the graph width, w_g , with respect to the width of the underlying circuit, w_c , and the number of irreducible non-Clifford gates, g_{nc} . The derivation assumes an understanding of ZX calculus [42].

Any quantum circuit can be reduced to a diagram containing three types of green spiders, connected via Hadamard edges: i) exterior/boundary spiders, ii) interior non-Clifford spiders and iii) interior Pauli spiders which are only connected to non-Clifford spiders [74]. Anything else can be removed using ZX rewrite rules. A diagram of this form can be read as a measurement pattern with single-plane measurements, with each spider corresponding to one qubit.

The boundary spiders in the diagram depend on the width of the original circuit. If measurements in multiple planes are allowed, there can be up to $2w_c$ boundary spiders. Here, only XY-plane measurements are allowed, which require up to $4w_c$ boundary spiders. There can only be as many non-Clifford spiders in the diagram as there are non-Clifford gates in the circuit.

To understand the upper bound for the third type of spiders we look at how they appear. Consider the 2-qubit circuit shown in figure 2.14 and the resulting ZX diagram in figure 2.15. This is an example of a *trapped Pauli* spider. While deformations are allowed in ZX calculus, this diagram is kept in a circuit-like form to illustrate that there are two types of non-Clifford spiders contributing to the entrapment of the Pauli. The two non-Clifford spiders on the horizontal line correspond to T gates acting on the same qubit at different times in the original circuit. I refer to these as *temporal connections*. The third non-Clifford spider is a T gate acting on a different qubit which contributes to the entrapment of the Pauli via a CNOT gate. I refer to this as a *spatial connection*.

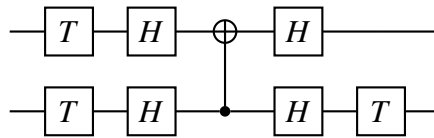


Figure 2.14: A simple 2-qubit circuit that leads to a trapped Pauli spider when converted to a ZX diagram.

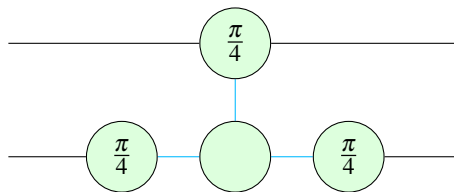


Figure 2.15: The circuit from figure 2.14 converted to a ZX diagram demonstrating a degree-3 trapped Pauli spider.

Because the graph is derived from a circuit, every interior spider must have at least 2 edges, which are the *temporal connections* of the gate in circuit form. If a trapped Pauli only has temporal connections and no spatial connections it is possible to extract the phase to the boundaries, leaving only the identity behind which leads to simplifications. This means that at least one spatial connection is required in order to turn the spider into degree-3 or higher, to prevent it from being simplified. This means that a minimum of 3 non-Clifford spiders are required to contain a trapped Pauli spider.

However, it is also possible for the non-Clifford spiders to contribute to multiple trapped Paulis. Consider the circuit shown in figure 2.16 and the corresponding ZX diagram shown in figure 2.17. It is possible for a non-Clifford spider to make multiple spatial connections, either to the same qubit in different instances in time or multiple qubits. A non-Clifford spider with a temporal connection to trapped Pauli may also form a temporal connection with a second trapped Pauli (one in the past and one in the future).

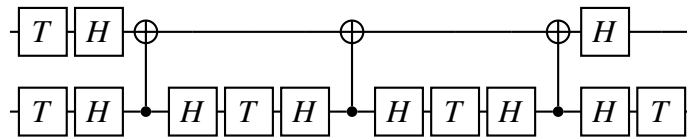


Figure 2.16: A 2-qubit circuit which leads to multiple trapped Paulis in ZX calculus.

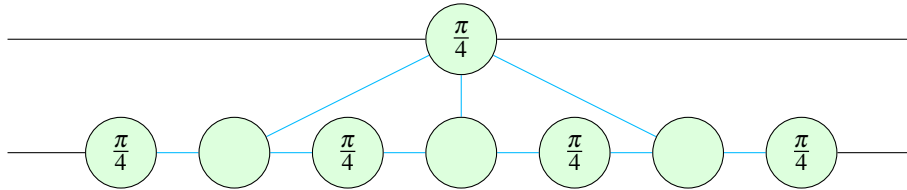


Figure 2.17: The circuit from figure 2.16 converted to a ZX diagram demonstrating multiple degree-3 trapped Pauli spider with the minimum required number of non-Clifford gates.

In the worst case scenario (which looks like the one in figure 2.17), each *trapped Pauli* is connected to exactly 3 non-Clifford spiders, a single non-Clifford spider provides the spatial connection required for every trapped Pauli and all the trapped Paulis are located on the same qubit (in the circuit model) allowing for maximum reusability of the non-Clifford spiders on the temporal axis. In such a scenario, the maximum number of *trapped Paulis* in the diagram is equal to $g_{nc} - 2$. The upper bound for graph width is thus given by equation 2.16,

$$w_g \leq 4w_c + 2g_{nc} - 2. \tag{2.16}$$

2.8 Circuit generation procedure

Two classes of sample circuits have been used in this work. The first are random Clifford+T circuits. These are generated using single-qubit Hadamard and T gates as well as two-qubit CNOT gates. Each of these building blocks is assigned a ratio with the sum of the three ratios being 1. For the samples in this work, the ratios for Hadamard, CNOT and T gates are 0.49, 0.49 and 0.02 respectively. The process for the random circuit generation is outlined in algorithm 1. The resulting circuits may contain a number of redundancies such as two Hadamard gates acting on the same qubit in a row. To make the samples more meaningful, they are simplified using routines of the *PyZX* [1] and *pytket* [2] libraries. Clifford+T circuits are interesting samples because the set of gates they employ is sufficient for universality.

Algorithm 1: The process for building random Clifford+T circuits

Input: Number of qubits, $n \in \mathbb{N}$

Total gates, $g_t \in \mathbb{N}$

Gate ratios, $r_i \in [0, 1]$ where $\sum_{i=1}^3 r_i = 1$

Output: Circuit, C_n

begin

 Initialise C_n with n qubits in the state $|0\rangle$;

for each gate g up to g_t **do**

 Randomly pick a gate type according to weights r_i ;

 Randomly pick a target qubit;

if g is a CNOT **then**

 Randomly pick a control qubit from the remaining qubits;

 Add g to C_n ;

end

end

The second class of circuits used are phase polynomials [75, 76]. These are defined using an arbitrary number of Pauli gadgets. Each of the gadgets is defined by an n -qubit Pauli operator as well as a phase implemented by a Z rotation. An example showing a Pauli gadget with the operator $IXZZX$ and a phase of $\frac{\pi}{4}$ is shown

in figure 2.18. The algorithm for generating phase polynomial circuits is outlined in algorithm 2. Phase polynomials are related to quantum chemistry applications such as VQE algorithms which is why they were chosen as samples for this thesis.

Algorithm 2: The process for building random phase polynomials**Input:** Number of qubits, $q_n \in \mathbb{N}$ Number of polynomials, $p_n \in \mathbb{N}$ **Output:** Circuit, C_{q_n} **begin**Initialise C_{q_n} with q_n qubits in the state $|0\rangle$;**for** each phase polynomial p_i up to p_n **do**Initialise subcircuit, C'_{q_n} , with q_n qubits;Define set of qubits, $L = \{q_1, \dots, q_n\}$;Generate a random integer, $m \in [1, 4^{q_n}]$;Construct a q_n -qubit Pauli, P , from m ;Generate a random phase, $\theta \in [0, 2\pi]$;**for** each single-qubit Pauli component P_i in P **do****if** $P_i = I_i$ **then**| $L = L \setminus \{q_i\}$;**else if** $P_i = X_i$ **then**| Add a $H(q_i)$ gate to C'_n ;**else if** $P_i = Y_i$ **then**| Add $H(q_i)$, $S(q_i)$, $H(q_i)$ gates to C'_n ;**end****while** size of $L > 1$ **do**| Split L into pairs (q_a, q_b) leaving any odd qubit out;**for** each pair (q_a, q_b) **do**| Add a $CNOT(q_a, q_b)$ gate to C'_n ;| $L = L \setminus \{q_a\}$;**end****end**Define a circuit C'_{n^\dagger} which is the reverse of C'_n ;Add C'_n to C_n ;Define a ZPhase gate with a phase of θ , acting on the only qubitleft in L and add it to C_n ;Add C'_{n^\dagger} to C_n ;**end****end**

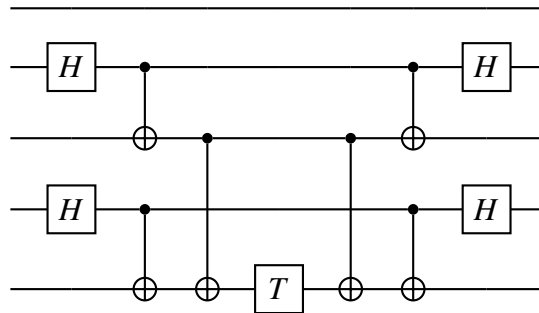


Figure 2.18: A 5-qubit circuit implementing a Pauli gadget defined by the Pauli operator $IXZZZ$ and a phase of $\frac{\pi}{4}$.

2.9 Determinism in MBQC and Pauli flow

As explained in section 2.6, a ZX-diagram in MBQC-form sufficiently describes the preparation, entanglement and measurement components of a measurement pattern defined in equation 2.7. Specifically, the diagram contains information about the number of physical qubits needed, what state they have to be prepared in, how they are meant to be entangled to form the graph state and in what measurement planes/angles they are to be measured. However, there is a piece of the puzzle missing: how to order the measurements and how to apply the corresponding corrections needed to ensure a deterministic output.

To accomplish this, the non-deterministic measurements that occur during the execution of a measurement pattern are considered to be affected by a Pauli error iff the negative eigenstate is observed. Whenever a computational basis measurement returns the negative eigenstate, that is equivalent to the result of the positive eigenstate flipped under the effect of a Pauli-X error. The Pauli errors can then be corrected by taking advantage of the graph stabilizers defined in equation 2.11 [71, 72]. However, qubits that have already been measured can no longer be corrected, which imposes a dependency order on the measurements/corrections to make the measurement pattern *runnable* [71, 72, 92].

In 2006, the notion of *causal flow* was introduced by Danos and Kashefi, which is a condition sufficient but not necessary for determinism in the one-way model

[71]. This means that if an open graph is found to have causal flow, it is possible to order its measurements and corrections such that the pattern can be run in a deterministic way. Furthermore, an algorithm for finding causal flow (if a graph has one) was later publicized by Mehdi Mhalla and Simon Perdrix [94]. However, otherwise deterministic graphs might not necessarily exhibit causal flow. Additionally, the definition of causal flow restricts measurements to just the XY plane.

A more general condition called *Pauli flow* was found in 2007 by Browne et al [72]. Pauli flow is both a necessary and sufficient condition for determinism in the one-way model and has loosened the restriction to allow for Pauli measurements as well as measurements about an arbitrary angle in the XY, XZ and YZ planes [72]. More recently, an algorithm was produced by Will Simmons which can be used to find Pauli flow for an arbitrary ZX-diagram in MBQC-form iff one exists [95], with a pythonic implementation of this algorithm available in the Pytket library [2] used in this thesis. It should also be noted that both LC and ELC are operations which preserve Pauli flow [95, 93].

2.10 The need for a hybrid model

By using the various tools and formalisms introduced in this chapter, quantum circuits can be rewritten as ZX-diagrams and then normalised to their MBQC form so that they can be implemented as measurement patterns. When rewriting the circuit model into ZX-calculus, all the operations are converted to ZX spiders and after all the simplifications and rewrite rules are applied, whatever spiders remain in the diagram constitute the qubits involved in the graph state (with the spiders corresponding to measurement effects being an exception).

Therefore, it is not surprising that measurement patterns typically display lower depth at the cost of additional ancilla qubits because if one carefully observes the conversion process, they may notice that quantum gates in the circuit model are effectively converted into qubits in the graph state - a quite literal conversion of depth to width.

However, this process is not very flexible as it does not give us the choice

of how much of the circuit's depth to convert and it can easily lead to situations where the measurement pattern requires too many qubits to be implementable on a device. This lack of flexibility is precisely the motivation for developing a hybrid between the circuit model and MBQC so that the previously inaccessible depth-width spectrum which exists between the two pure models can be navigated. This hybrid model is the focus of chapter 3.

Chapter 3

Hybrid model of quantum computation

3.1 Introduction

In this chapter, I introduce a *hybrid model of quantum computation (HMQC)* which serves as a tuneable mid-point between the circuit model and MBQC, such that the width-depth tradeoff between the two approaches can be controlled and tailored for implementation on a known platform. Specifically, converting a circuit into a measurement pattern can drastically reduce the depth and 2-qubit depth at the cost of increasing width, as multiple ancilla qubits are required. However, both the circuit model and MBQC can be viewed as two fixed points in what is a spectrum of depth-to-width conversion. The purpose of our description of the hybrid model is to tap into this spectrum for more flexible control of the resource trade-off.

In section 3.2, I provide a definition of the hybrid model before providing an example of how HMQC can be constructed. In section 3.3, I provide an example of a HMQC with one MBQC segment, demonstrating how the width and depth of the HMQC forms a spectrum bounded by the circuit model and MBQC, depending on the subcircuit selected for conversion. In section 3.4, I expand this analysis to include up to two MBQC segments, showing that more than one MBQC segment opens the door for ancilla reusability from one measurement pattern to the next, which in turn leads to a better width-depth tradeoff.

As the number of MBQC layers increases, it becomes impractical to fully enumerate all the possible HMQC combinations. Consequently, I provide a more systematic approach which leads to the *targetted hybrid model* which accounts for n -layers of MBQC, defined in section 3.5. The goal of our implementation of the hybrid model is to optimise the ratio of depth reduction to ancilla overhead needed. The targetted hybrid model we introduce in section 3.5 is the iteration that we find to be best at accomplishing this goal, as it is designed to maximize the reusability of ancilla qubits between segments, minimising the qubit overhead required.

A comparison between the width and depth of the circuit model, HMQC and MBQC can be found in section 3.6. HMQC is primarily aimed towards depth reduction and achieves a higher amount of depth reduction when applied to deeper circuits. In section 3.6, I also introduce a method of determining whether this technique is suitable for a specific circuit (or circuit class) as well as approximately predicting the performance of the converted circuit. Finally, I discuss an attempt to route hybrid model circuits for devices with limited qubit connectivity and the challenges involved in that process, in section 3.7.

3.2 Definition of the hybrid model

In this section, I provide a formal definition for the hybrid model while also producing an example, demonstrating how the hybrid model is constructed. First, let's consider the general case of HMQC that we describe in definition 3.2.1:

Definition 3.2.1 *The hybrid model is the sequential composition of n -layers of computation, with each individual layer being described in either the circuit model or in MBQC. The inputs of a layer are mapped to the corresponding outputs of the preceding layer. Classical signals can be communicated between layers.*

It follows from definition 3.2.1 that if every layer is in circuit model form, then the entire computation is equivalent to the circuit model. Likewise, if every layer is a measurement pattern then the entire computation is a form of MBQC, albeit not in a standard form.

Here, I describe the compilation process used to produce a HMQC. This representation is constructed by splitting a quantum circuit into two or more subcircuits along its depth. Any number of the resulting segments are converted to measurement patterns which introduce ancilla qubits but do not affect the number of input and output qubits. For a measurement pattern to be implemented deterministically, it is necessary to use classical feedforward in the form of Pauli corrections on the remaining qubits following each measurement. This feedforward can either be carried out in the same layer as the measurement pattern or it can be passed on to subsequent layers and grouped together with classical information emerging from those layers.

Consider the circuit c_0 shown in figure 3.1. This circuit can be converted entirely to a measurement pattern as shown in figure 3.2. Alternatively, it can be split down the middle into two subcircuits, c_1 and c_2 , performed in sequence. We can convert c_1 and c_2 into separate graphs as shown in figure 3.3. The outputs of the graph derived from c_1 are feeding into the inputs of the graph derived from c_2 . Performing the two segments in sequence results in the same number of outputs as the original circuit but the width and the depth of the overall operation have changed. In this case we do not observe significant reduction of depth because the original

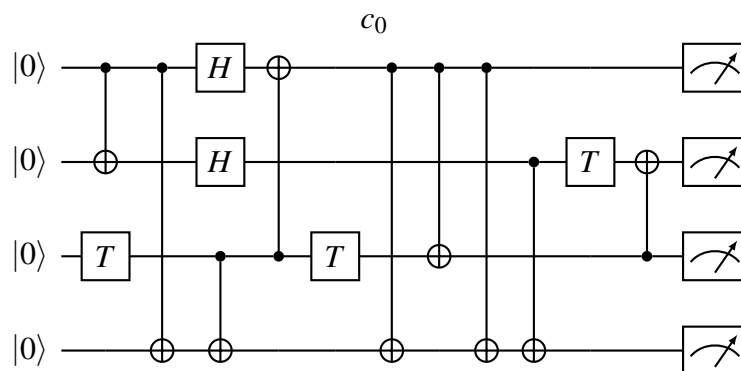


Figure 3.1: A randomly-generated 4-qubit circuit of depth 10, composed of CNOT, Hadamard and T gates.

circuit is too shallow. However, the same process may be applied to deeper circuits for more interesting results.

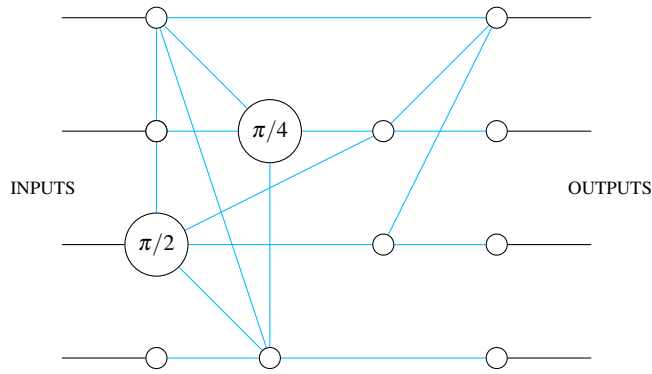


Figure 3.2: c_0 converted to a single graph state. Each vertex represents a different qubit, so 12 qubits are needed to execute the corresponding measurement pattern.

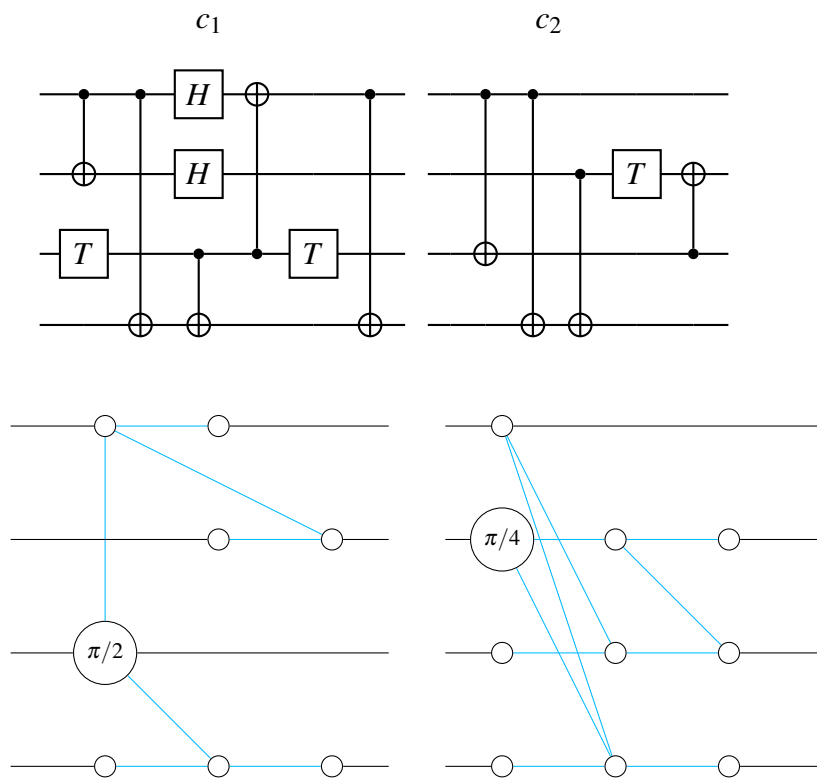


Figure 3.3: Two graph states obtained by splitting c_0 into two subcircuits and converting each one to MBQC separately. Because the ancilla qubits used for the first graph state can be reused for the second graph state, only 10 qubits are needed to execute the computation in this form.

3.3 Hybrid model with one MBQC segment

Since transpiling an entire circuit to MBQC converts a certain amount of depth into width, it is reasonable to expect that transpiling only a subcircuit of the original

circuit would likewise result in only a partial conversion of depth to width. Consider a circuit c , of depth d . Furthermore, we define an arbitrary subcircuit c_i of c , where parameter i is an index used to enumerate all the possible unique subcircuits of c . Consequently, subcircuit c_i can be fully defined with two parameters: Its depth, $d_i \in [1, d]$ and its offset from the beginning of the circuit, $k_i \in [0, d - d_i]$. Since both d_i and k_i scale linearly with d , the number of possible subcircuits i scales quadratically with d .

To illustrate the properties of the hybrid model with one MBQC segment, we use a numerical example, the results of which are shown in figure 3.4. For each case i we convert *only* subcircuit c_i to MBQC while leaving the rest of circuit c in the circuit model. The overall width and depth of the HMQC implementation of c is marked as an individual datapoint in figure 3.4. The circuit used to produce figure 3.4 is a randomly-generated 5-qubit Clifford+T circuit of depth=90. There are $i + 1$ datapoints in the figure and they form part of the depth-width spectrum bounded by the circuit model and MBQC, indicating the part of the spectrum which is achievable by only converting a single subcircuit to MBQC. The dashed box contains the cases where the width is lower than full MBQC but higher than the circuit model and the depth is lower than the circuit model but higher than full MBQC. The dashed diagonal line represents the rate by which depth is converted to width in the full MBQC case. Consequently, any datapoint which exists within the bounded box demonstrates that it is possible to obtain a version of the circuit which has lower depth than the circuit model form and lower width than the pure MBQC form.

Additionally, the datapoints which exist below the dashed diagonal are even more interesting as they represent a case where the ratio of reduced depth divided by increased width is lower than the full MBQC conversion. However, in the case where only a single subcircuit is converted to MBQC there are very few datapoints below the dashed diagonal. These datapoints are artifacts of the ZX-diagram rewrite operations and are not significant (or scalable). In cases with more than one MBQC segment ancilla qubits can be recycled from one measurement pattern to the next so

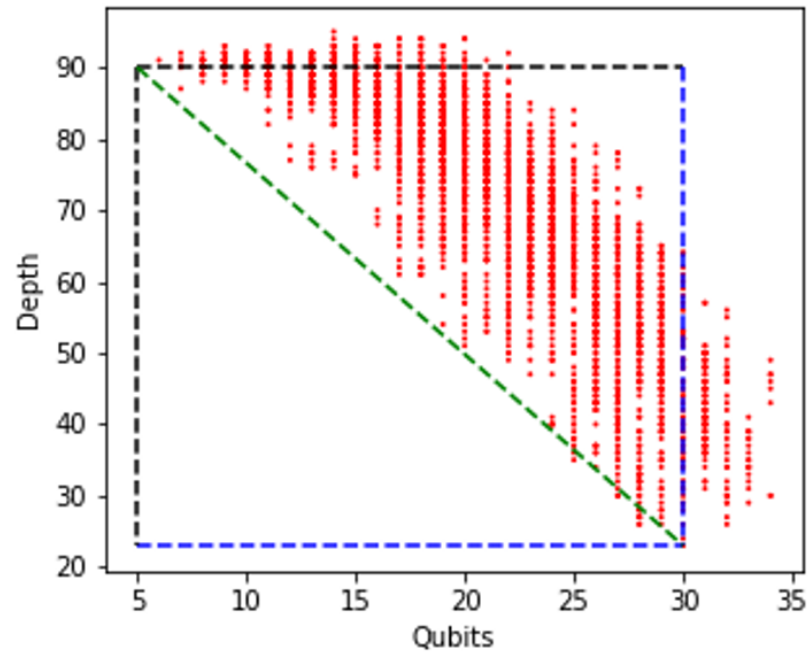


Figure 3.4: The figure shows the width-depth tradeoff of a randomly-generated 5-qubit Clifford+T circuit with an original depth=90 after converting a single, arbitrarily selected subcircuit to MBQC. The top-left corner of the bounded box indicates the circuit model. The bottom-right corner indicates a full conversion to MBQC. Every other datapoint represents the case where only subcircuit c_i is converted to MBQC. The dashed diagonal represents the depth-width conversion rate when converting the entire circuit to MBQC.

we expect more datapoints below the dashed diagonal to appear.

3.4 Hybrid model with two MBQC segments

We expect to see higher depth reduction in the case where **two** subcircuits are converted to MBQC at the same time instead of one, as this allows ancilla qubits to be recycled. Consider the case where the circuit is split down the middle, such that the first segment has a depth $d_1 = d/2$ and offset $k_1 = 0$ and the second segment has a depth $d_2 = d/2$ and offset $k_2 = d/2$. In this situation, each segment is converted to MBQC separately. As a result, the entire depth of the original circuit is converted, but there are two measurement patterns instead of one. This process can be expanded to include all possible combinations of two MBQC segments. To better illustrate this, we perform a numerical analysis, the results of which are shown in figure 3.5. The circuit used for figure 3.5 is the same randomly-generated 5-qubit Clifford+T circuit that was used for figure 3.4.

By converting two subcircuits to MBQC, more options are introduced and a broader slice of the width-depth trade-off spectrum becomes accessible. There exist many more instances of the hybrid model where the datapoints are located below the dashed diagonal, thus indicating a lower ratio of reduced depth divided by increased width than a full conversion to MBQC. This happens because the recycled ancillas lower the overall requirement for physical qubits in the entire HMQC. Since an arbitrary subcircuit can be defined using two parameters, the depth d_i and the offset k_i , each of which scales linearly with the depth of the original circuit d , the number of possible subcircuits scales quadratically with d .

3.5 Hybrid model with n MBQC segments

When we allow for n subcircuits to be converted to MBQC, the number of possible combinations scales exponentially with the depth of the original circuit. To see that, consider that the original circuit has enough space for at most d subcircuits of depth $d_i = 1$ and each of those subcircuits can exist in 2 forms: in the circuit model or converted into a measurement pattern. Once we allow for an arbitrary combination of segments in the hybrid model, the number of possible combinations is at least 2^d . In fact, it's higher than that because two adjacent slices of the circuit could

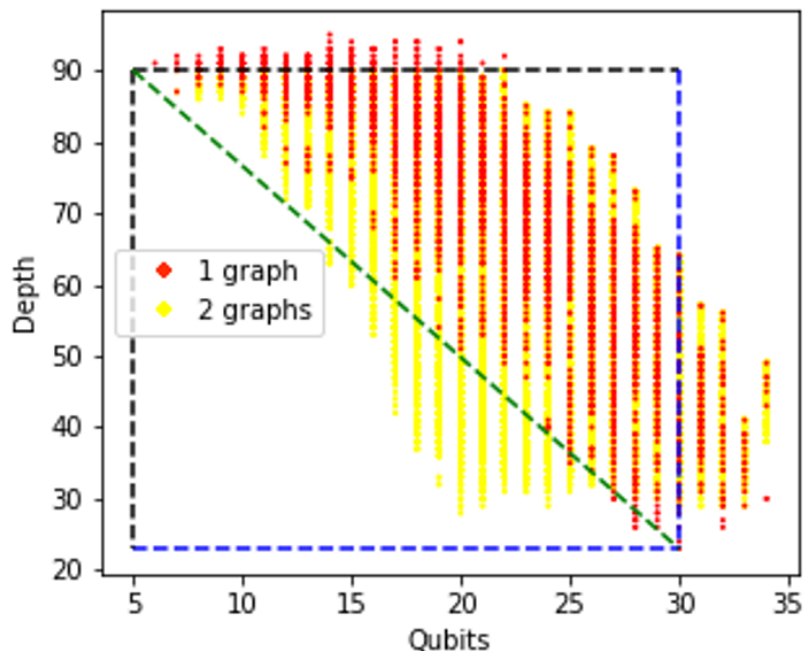


Figure 3.5: The figure shows the width-depth tradeoff of a randomly-generated 5-qubit Clifford+T circuit with an original depth=90 after converting two, arbitrarily selected subcircuits to MBQC. The top-left corner of the bounded box indicates the circuit model. The bottom-right corner indicates a full conversion to MBQC. Every red datapoint represents the case where only subcircuit c_i is converted to MBQC. Every yellow point represents the case where non-overlapping subcircuits c_i and c_j are simultaneously converted to MBQC. The dashed diagonal represents the depth-width conversion rate when converting the entire circuit to MBQC.

be converted individually into two separate measurement patterns, or merged and converted together into a larger one.

Therefore, it is not realistic to test out every possible conversion and pick the best one. Instead, it is preferable to have a more systematic approach to splitting and converting the circuit. Converting a single subcircuit to MBQC adds a number of ancilla qubits. If another subcircuit is converted, it is possible to recycle these ancillas by re-initialising them after the first MBQC segment is completed so that they can be reused in the second segment. This means that the total number of qubits required will not increase if the second segment doesn't involve more ancillas than the first.

Furthermore, the relationship between the width in MBQC w_g , the width in the circuit model w_c and the number of non-Clifford gates in the circuit model g_{nc} is discussed in section 2.7. The upper bound for graph width was derived in the same section and was shown in equation 2.16, which we reproduce here for convenience,

$$w_g \leq 4w_c + 2g_{nc} - 2. \quad (3.1)$$

Consequently, as long as a single subcircuit is converted to MBQC, then all the remaining subcircuits should also be converted to MBQC because this results in depth reduction at no added cost, provided that no segment requires more ancillas than the first. From equation 3.1, the width of a segment is known to depend on the width of the original circuit (which is not affected by the choice of subcircuit) as well as the number of non-Clifford gates in the corresponding subcircuit.

To maximize ancilla reusability, the circuit should be split up into segments containing the same number of non-Clifford gates. In this thesis, we are primarily interested in Clifford+T circuits and phase polynomial circuits. The reasons for these classes of circuits being important as well as the process in which they are being generated are discussed in section 2.8. For these classes of circuits, the distribution of non-Clifford gates can be approximated to be uniform along the depth of the circuit. This leads to definition 3.5.1.

Definition 3.5.1 *The targeted hybrid model is a special case of the generalised hybrid model, where the following conditions apply:*

- *Every subcircuit is converted to MBQC as long as it results in a depth reduction.*
- *Every subcircuit has the same depth before conversion to MBQC (up to -1 timesteps).*

Targeted HMQC involves splitting a circuit into segments of equal depth and converting each one to MBQC. Refer to figure 3.6 for a visual demonstration of how targeted HMQC results in a better utilisation of resources. Again, we turn

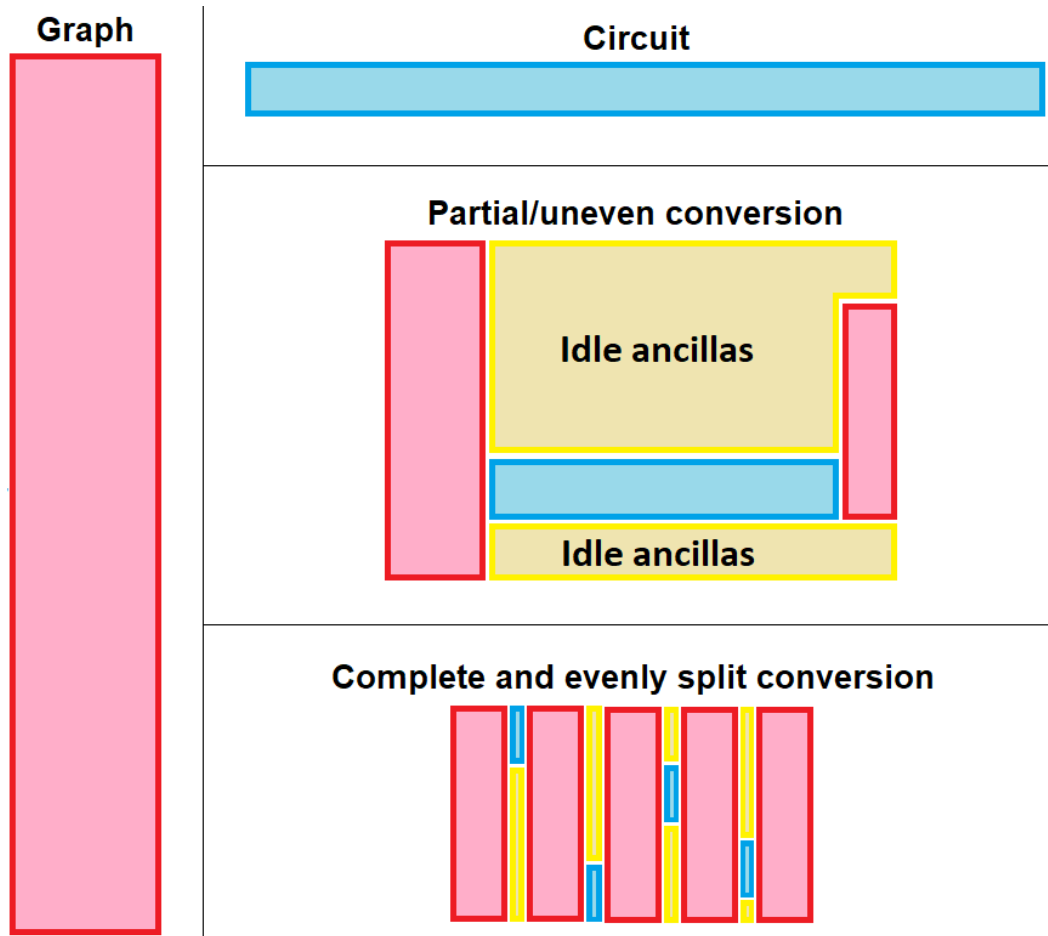


Figure 3.6: Abstract visual comparison of different implementations of the same quantum circuit. The horizontal dimension represents depth, while the vertical dimension represents width. Blue sections represent a circuit or subcircuit in the circuit model while red sections represent measurement patterns. Yellow sections represent resources that would be taken up but not utilised during implementation. The shape on the left represents a full MBQC implementation. The shape at the top-right represents the full circuit model implementation. The shape at the middle-right represents an arbitrarily selected version of the hybrid model while the shape at the bottom-right represents the version of the targeted hybrid model involving 5 MBQC segments.

to a numerical example to better demonstrate the properties of targeted HMQC. These results are shown in figure 3.7 which involve a case of targeted HMQC for varying numbers of segments of equal depth, applied to a randomly-generated 10-qubit Clifford+T circuit. For each datapoint, the base circuit is the same, but it is split into a different number of subcircuits which are then individually converted first to ZX-diagrams and then to measurement patterns. The resulting depth and width for the measurement patterns in sequence are then used to mark the points

within the tradeoff spectrum. Note that not only are there many data points that correspond to ‘cheaper’ depth reduction than MBQC but it is now much easier to compute these cases because the number of segments i now scales linearly with the depth of the circuit d under this new setting. For the remainder of this thesis, the term ‘HMQC’ will refer to the targeted hybrid model unless otherwise stated.

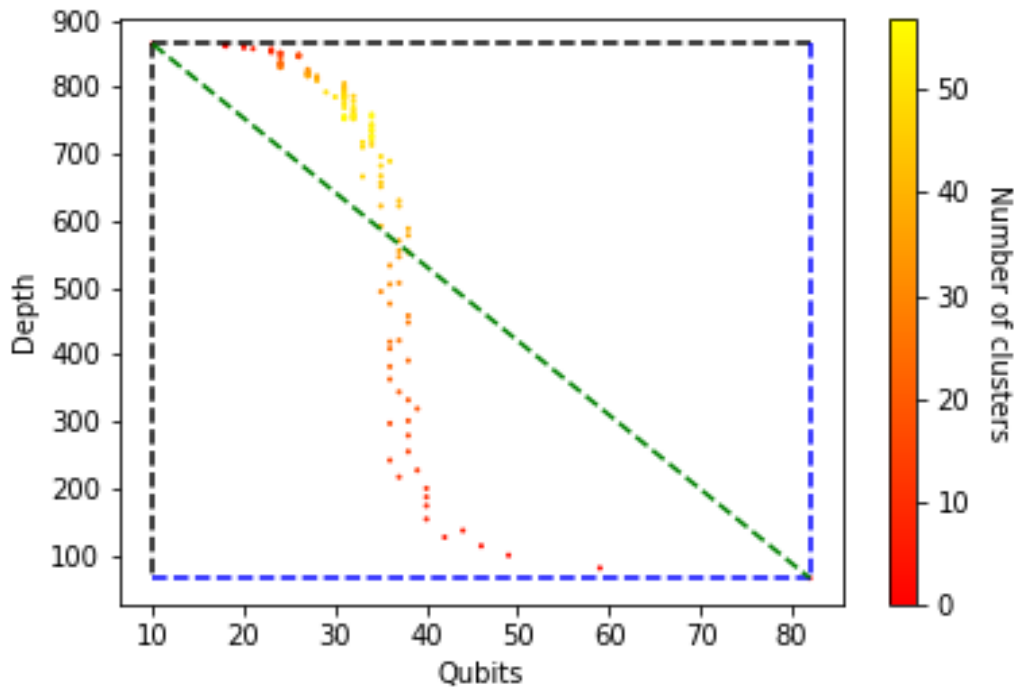


Figure 3.7: Width-depth tradeoff of a randomly-generated 10-qubit Clifford+T circuit after converting to the targeted hybrid model. Each datapoint represents the case where the circuit is split into N subcircuits of equal depth and each subcircuit is converted into a measurement pattern. The color of the datapoints indicates the value of N . The top-left corner of the bounded box indicates the circuit model. The bottom-right corner indicates a full conversion to MBQC. The dashed diagonal represents the depth-width conversion rate when converting the entire circuit to MBQC.

3.6 Resource comparison in the hybrid model

To better evaluate whether the technique discussed in this chapter is worth using for a specific circuit or circuit class, I developed a model to estimate the width and depth of HMQC. While the relationship between circuit width and MBQC width is largely understood and governed by equation 3.1, the relationship between the parameters of the original circuit and the depth of the corresponding measurement pattern is still unknown and varies on a case by case basis. Due to this, it was not possible to produce a fully analytical model.

However, the depth of a measurement pattern was empirically found to be half of its width for both the randomly-generated Clifford+T circuits and the phase polynomial circuits. Therefore, it was possible to incorporate this approximation and the corresponding margins of error into a numerical predictive model that serves as an informative reference point to determine whether a circuit would be worth converting to the hybrid model. The procedure I followed to accomplish this is explained in this section.

For reasons discussed in section 2.8, two separate sample sets of 1000 random Clifford+T circuits and 1000 phase polynomials of widths $w_c \in [5,25]$ and depths $d_c \in [50,2500]$ were used. Each sample was converted to a graph and the resulting graph properties were compared to their circuit properties. Figures 3.8 and 3.9 demonstrate a strong linear dependence of the graph width, w_g , to both the width of the original circuit, w_c , and the number of non-Clifford gates in the circuit, g_{nc} . The numerical relationship extracted from the samples is of a similar form to the theoretical upper bound for graph width, given in equation 3.1.

I have chosen to develop and use heuristics to represent the relationship between circuit and graph width in the predictive model. This is done because the coefficients extracted numerically from the samples better capture the behaviour of the *average case* circuit, as opposed to the *worst case* circuit described by the theoretical upper bound. These heuristics were obtained by using linear regression techniques on the data extracted from both the random Clifford+T samples and the phase polynomial samples as shown in figures 3.8 and 3.9 respectively. Specifically,

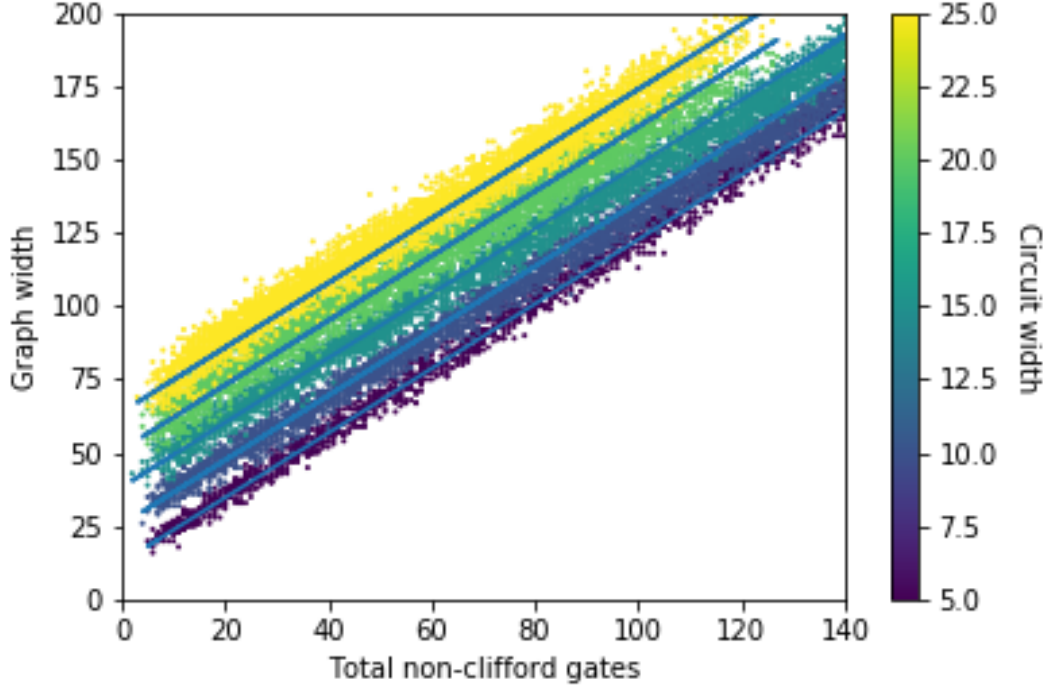


Figure 3.8: Relationship between the graph width, circuit width and number of non-Clifford gates in a sample set of random Clifford+T circuits.

w_g is treated as the dependent variable of a linear model while w_c and g_{nc} are treated as the two regressors of the model. The approximation of w_g takes the form shown in equation 3.2,

$$w_g = \alpha w_c + \beta g_{nc}. \quad (3.2)$$

For the random Clifford+T samples, the graph width is denoted by $w_{g,r}$ with the coefficients determined via regression to be $\alpha_r = 2.56$ and $\beta_r = 1.10$. Similarly, the graph width for the phase polynomial samples is denoted by $w_{g,p}$ and the coefficients are $\alpha_p = 2.83$ and $\beta_p = 1.02$. The relative standard deviation for $w_{g,r}$ is $\pm 3.8\%$ while the relative standard deviation for $w_{g,p}$ is $\pm 3.9\%$. In order to verify the robustness of the heuristics in the face of different datasets, I have investigated the effects of applying the approximation for $w_{g,p}$ on the random circuit samples and found the new relative standard deviation to be equal to $\pm 6.1\%$. Similarly, I have taken the heuristic derived from the random circuits and applied it to the phase polynomial samples to find a relative standard deviation of $\pm 6.8\%$. As expected,

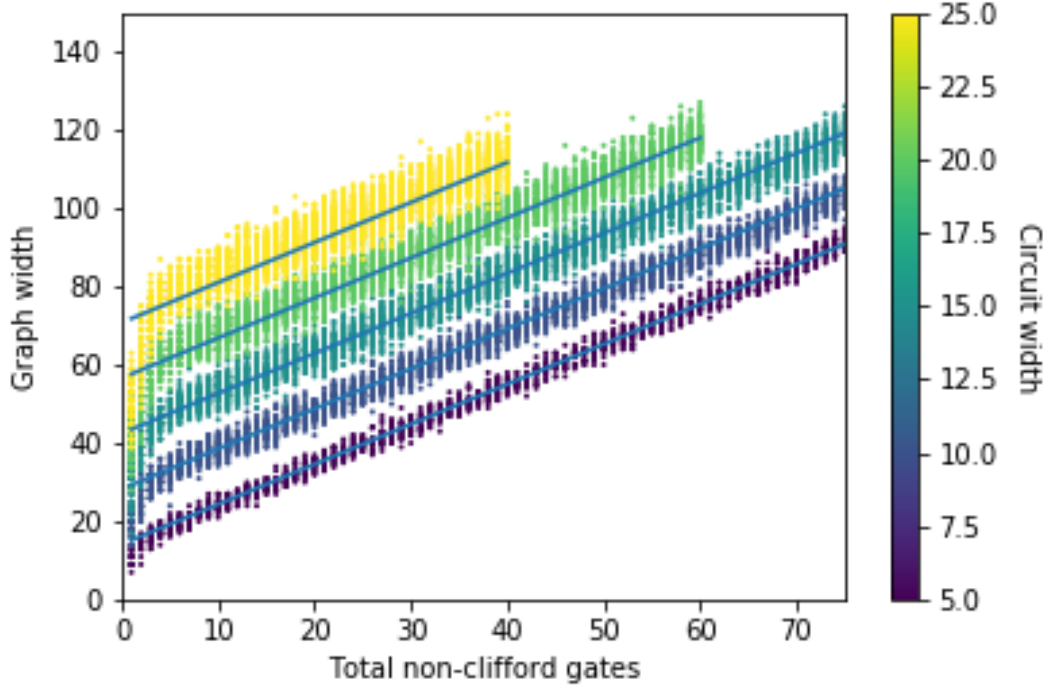


Figure 3.9: Relationship between the graph width, circuit width and number of non-Clifford gates in a sample set of phase polynomial circuits.

using the approximation derived from one sample set to another lowers the accuracy but the difference is small which indicates that the relationship of w_g to w_c and g_{nc} is very consistent between circuit classes.

The depth in the graph state, d_g , was found to be dependent on both w_g and w_c , taking the form shown in equation 3.3. The graph depth for the random circuit samples is denoted by, $d_{g,r}$, with the coefficients found via regression to be $\gamma_r = 13.9$, $\delta_r = 0.51$ and $\epsilon_r = 0.37$. The graph depth for the phase polynomial samples is denoted by $d_{g,p}$ with the coefficients found to be $\gamma_p = 17.1$, $\delta_p = 0.84$ and $\epsilon_p = 1.39$.

$$d_g = \gamma + \delta w_g - \epsilon w_c \quad (3.3)$$

Note that the heuristics which describe the graph depth are less accurate than those for the graph width, with $d_{g,r}$ exhibiting a relative standard deviation of $\pm 13.1\%$ when applied to the random circuit samples and $\pm 36.2\%$ when applied to the phase polynomials. Similarly, $d_{g,p}$ has a relative standard deviation of $\pm 13.4\%$

when applied to phase polynomials and $\pm 51.8\%$ when applied to random circuits. In essence, this means that the heuristics for graph depth which are extracted from one circuit class are not accurate enough to be used on the other. This happens because each version of the predictive model is based on a different sample class and subject to different circuit properties. Consequently, if accuracy is required, it would be best to use approximations developed specifically for the circuit class under consideration.

For the case where a circuit is split into N subcircuits and then converted into a sequence of graphs, the final width w_h and depth d_h , are given by the equations 3.4 and 3.5 respectively. These approximations are based on equations 3.2 and 3.3 respectively since the hybrid model still involves graph states. Note that for classes of circuits where non-Clifford gates are not concentrated on a specific part of the circuit but can instead be assumed to be distributed uniformly (which is true for both random circuits and phase polynomials), the number of such gates in each subcircuit is $\approx \frac{g_{nc}}{N}$. Also note that the overall depth of HMQC is equal to the sum of the depth of all the segments but the same is not true for the width due to how qubits are reused between segments. As a result, only the depth equation is multiplied by the number of segments, N ,

$$w_h = \alpha w_c + \beta \frac{g_{nc}}{N}, \quad (3.4)$$

$$d_h = (\gamma + \delta w_h - \epsilon w_c)N. \quad (3.5)$$

By knowing the number of non-Clifford gates and the number of qubits in a quantum circuit and by referring to equations 3.4 and 3.5 as well as the empirically obtained margins of error found from the samples it is possible to develop a predictive model which determines whether the methods presented in this paper are suitable to be used for a specific circuit or a class of circuits. Examples of such a predictive model in use can be seen in figures 3.10 and 3.11.

The margin of error for this predictive model reaches up to $\pm 20\%$ as it is based on a number of approximations which are affected by the large variance seen between different circuits of the same width/depth class. The purpose of these heuris-

tics is simply to be used as tools to approximately predict the behaviour of a circuit converted to HMQC, mainly to determine whether such a conversion is worthwhile in the first place. Despite the large margin of error, it is still possible to use these approximations to predict whether HMQC can be used to navigate the width-depth resource spectrum for a given application. Once this is determined, there is no further need to rely on the approximations as the actual circuit can simply be converted and investigated in more detail.

The predictive models are more likely to succeed when used for circuits in the same class for which they were developed. For example, the accuracy of the model derived from the Clifford+T samples is lower when used on the phase polynomial samples. Therefore, if a similar model is needed for a circuit class outside of those considered in this thesis, it is recommended that a new specialised model is derived for the circuit class in question.

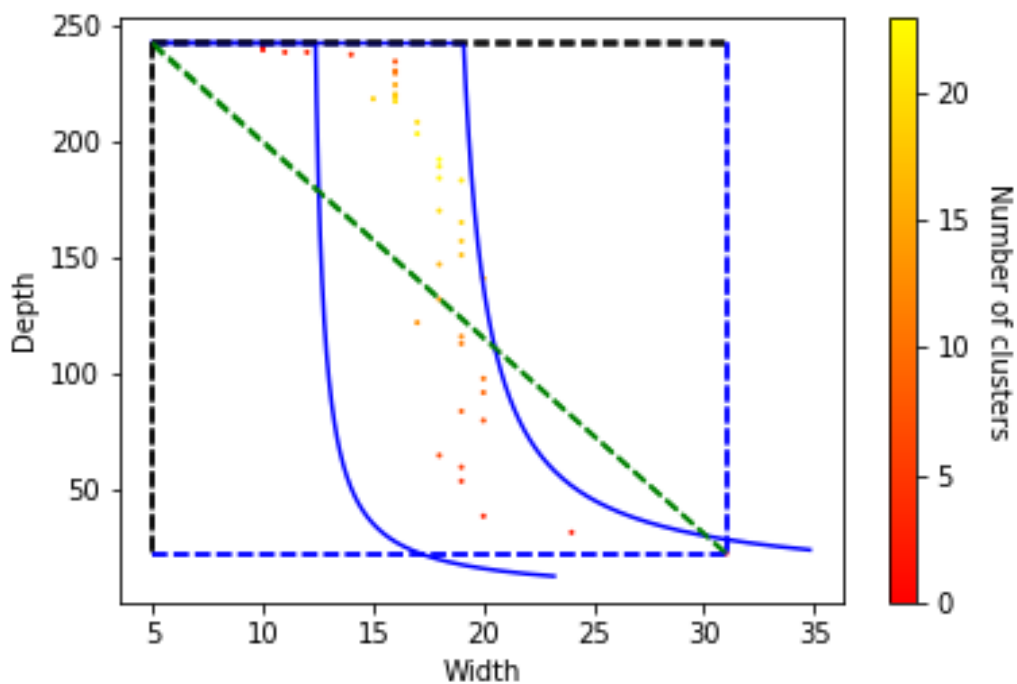


Figure 3.10: Example of the predictive model (blue curved lines) determining ahead of time the region where the depth and width of HMQC will lie in when applied to a randomly-generated 5-qubit Clifford+T circuit. The dotted lines maintain the same meaning they had in figures 3.4 and 3.5.

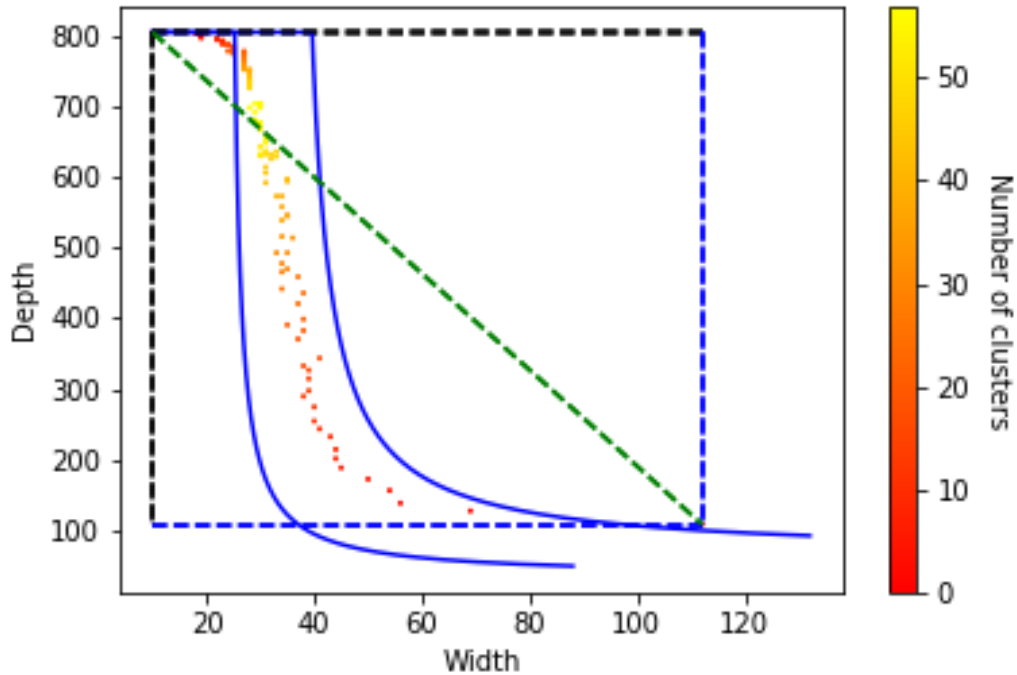


Figure 3.11: Example of the predictive model (blue curved lines) determining ahead of time the region where the depth and width of HMQC will lie in when applied to a randomly-generated 5-qubit phase polynomial circuit. The dotted lines maintain the same meaning they had in figures 3.4 and 3.5.

Despite the differences between MBQC and HMQC introduced in this thesis, the hybrid model inherits many properties from MBQC since it still relies on graph states, adaptive measurements and ancilla qubits. When converting a quantum circuit to a graph state, high-degree graphs can be produced which can lead to difficulties implementing the graphs on devices with limited qubit connectivity. To fit the graphs on such architectures, it is often necessary to route the circuit, which can lead to a large number of SWAP gates being added. As such, it is important to assess how the hybrid model performs in such a setting, which is the focus of section 3.7.

3.7 Routing for restricted architectures

In this section, three different strategies for routing a hybrid model circuit on a device with limited connectivity are compared. The architecture under consideration

is assumed to be a 2D square lattice (degree-4 connectivity). This architecture is chosen primarily for its simplicity and for the fact that it is sufficient to demonstrate the negative impacts of the routing process on the depth of the final circuit. In real applications, the available connectivity is often even sparser than the one used here but that can only increase the impact of routing even further.

The mapping of the circuits onto the grid is achieved by using the methods native to Pytket [2]. This is done by iterating through the two-qubit gates in the temporal order in which they appear, assigning each qubit a temporary placement on the architecture, until a two-qubit gate is found which does not conform to the current placement. When that happens, the algorithm aims to move one of the qubits to a new location using SWAP gates, so as to make the two-qubit gate executable. Candidate placements are scored, selecting the one which requires the fewest SWAP gates to be achieved. This is repeated until the entire circuit object is placed on the device [2].

Some special considerations are made for each strategy to accommodate for the properties unique to the hybrid model, as opposed to a simple measurement pattern in general. Specifically, a measurement pattern on its own can be read as a circuit object and mapped on a device in a straightforward way, such that logical qubits that have to be entangled are either arranged to be physical neighbours on the device from the beginning or have to be moved closer to one another via SWAP gates during runtime. When considering the hybrid model, one has to remember that multiple measurement patterns are involved and so deciding how they are to be mapped on the device and interconnected becomes a bit more complicated.

The three approaches under consideration are the following: *Naive* routing, *separate* routing and *sequential* routing. In the case of naive routing, instead of treating the different measurement patterns as separate circuit objects, they are all interconnected into a single circuit object and handed to the routing algorithm together. In the case of separate routing, each measurement pattern is mapped on the architecture one at a time. Consequently, the outputs of one segment do not automatically align with the corresponding inputs of the next segment and thus an inter-

mediary step is required where the appropriate network of SWAP gates is generated between segments. Pseudo-code outlining the three different routing strategies is found in algorithm 3.

In the case of sequential routing, the segments are still mapped individually and one at a time on the architecture, but the inputs of each segment are locked in the location of the outputs of the preceding segment. This means that the routing method has less freedom when mapping all but the first segment but this removes the overhead caused by having to add extra SWAP gates between segments.

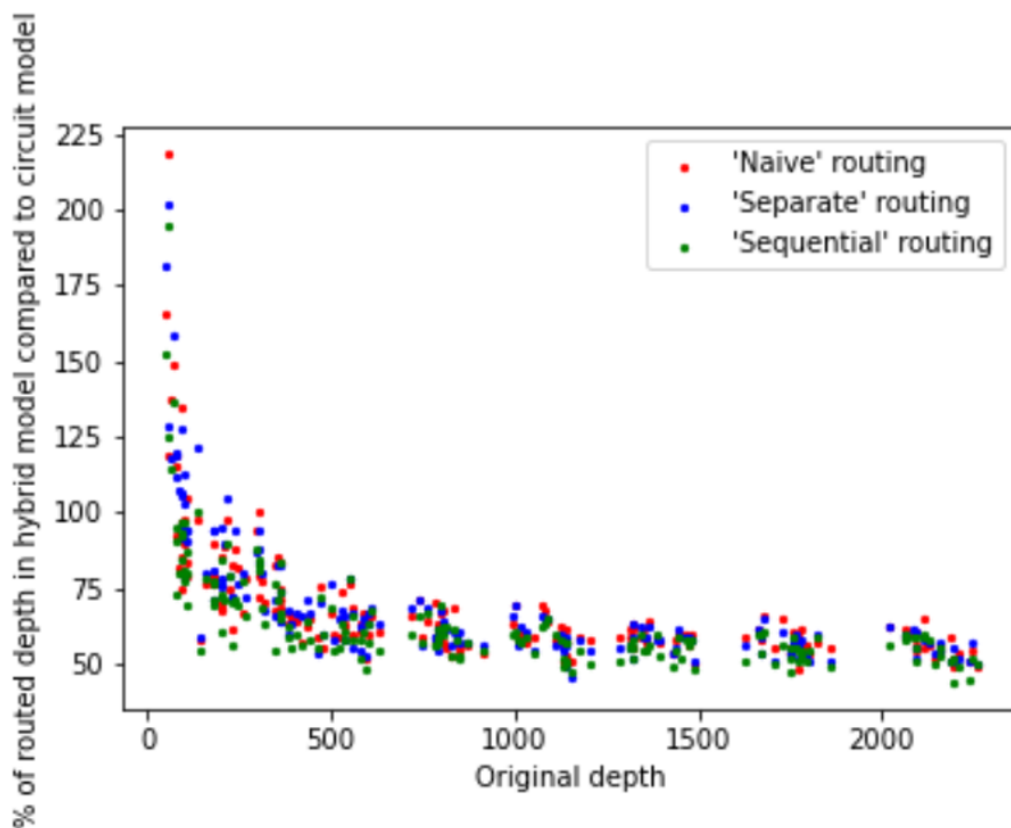


Figure 3.12: Comparison of the three routing approaches. Each data point represents a different randomly-generated Clifford+T circuit, of a width in the range $w_c \in [2, 15]$ and depth in the range of $d \in [100, 2300]$ and each color represents a different routing approach. The x-axis represents the depth of the original circuit before routing. The y-axis represents the percentage of the routed depth of the hybrid model compared to the routed depth of the circuit model.

Once again, we turn to a numerical analysis to illustrate the comparison of the three routing strategies, the results of which are shown in figure 3.12. Here we have

applied each of the routing strategies on multiple circuits of varying sizes. All the samples used are randomly-generated Clifford+T circuits with a variety of widths in the range $w_c \in [2, 15]$ and depths in the range of $d \in [100, 2300]$. Furthermore, in all the cases, the circuits were split into 3 segments, each subcircuit was converted to MBQC and the resulting measurement patterns were linked together with each of the three routing strategies. It should be noted that when the circuits under consideration are shallow to begin with, the depth overhead added due to routing is dominant and therefore, completely invalidates any potential depth reduction expected from MBQC. However, for deeper circuits it is possible to reach a depth reduction of up to 50% even after routing, which is still much lower than the 90% possible without architecture restrictions for the same cases but still potentially worthwhile.

Looking more closely at the results revealed that routing separately results in the lowest overall depth 30.7% of the time, the lowest 2-qubit depth 29.2% of the time and the lowest 2-qubit gate count 23.4% of the time for the collected samples. Routing sequentially results in the lowest overall depth 74.1% of the time and lowest 2-qubit depth 75.5% of the time and the lowest 2-qubit gate count 90.1% of the time. As seen in figure 3.12 the difference between the three approaches in practice is negligible and hard to spot, so each approach could be used in a real application. However, the exact statistics do seem to suggest that the sequential routing approach is slightly preferable which is to be expected because it is the more nuanced of the three.

Algorithm 3: Routing strategies.

Input: List of subcircuits, $c_i \in C$ Architecture object, α **Output:** Routed circuit, c' **NAIVE** (C, α)**begin**| $c' = []$ | **for** $c_i \in C$ **do**| | **if** $i < N$ **then**| | | Map inputs of c_{i+1} to outputs of c_i | | $c' += c_i$ | **end**| Route c' on α | **return** c' **end****SEQUENTIAL** (C, α)**begin**| $c' = []$ | **for** $c_i \in C$ **do**| | **if** $i = 0$ **then**| | | Route c_i on α | | **else**| | | Route c_i on α with locked inputs| | **if** $i < N$ **then**| | | Map inputs of c_{i+1} to outputs of c_i | | $c' += c_i$ | **end**| **return** c' **end****SEPARATE** (C, α)**begin**| $c' = []$ | **for** $c_i \in C$ **do**| | Route c_i on α | | **if** $i > 0$ **then**| | | Generate swap circuit, c_S moving the outputs of c_{i-1} to the inputs of c_i | | | $c' += c_S$ | | $c' += c_i$ | **end**| **return** c' **end**

3.8 Summary of the hybrid model

In this chapter, I have introduced the hybrid model of quantum computation which is a sequential layering of the circuit model and the one-way model. The goal of our version of the hybrid model is to achieve a high fraction of the depth reduction achieved by the one-way model at a lower fraction of the qubit overhead required by pure MBQC. It succeeds in doing so by maximizing reuse of ancilla qubits from one measurement pattern to the next, thus minimising the total number of qubits needed for the entire HMQC. This is accomplished by splitting the original circuit into N subcircuits of equal depth before converting each one to a measurement pattern. The inputs of each segment are then mapped to the outputs of the preceding segment. As for the classical information that emerges during each MBQC segment, such as the outcomes of the mid-circuit measurements and the corresponding conditional corrections, these can be handled either in the same layer by absorbing them as rotations to the basis of future measurements, or passed to subsequent layers.

It should be noted that HMQC is not a universally suitable approach. While it can be used to reduce the depth of a quantum circuit, thus reducing the effect of decoherence, it does so at the cost of adding more measurements and qubit re-initialisation. In platforms where measurements are costly, this can lead to higher amounts of depolarizing noise.

Additionally, while the results are promising in the fully connected setting, our attempts at routing the measurement patterns for restricted architectures have shown that graph states usually have a high vertex degree and contain more interconnected qubits when compared to the quantum circuits from which they are derived. As a result, more SWAP gates are needed to place the measurement patterns on a restricted architecture and much of the depth reduction achieved through the conversion to MBQC (or HMQC) is lost to this overhead.

In the next chapters, I provide two separate solutions to deal with the routing overhead. First, I introduce algorithms based on local complementation and pivoting in chapter 4 which aim to reduce the degree of a graph/ZX-diagram so as to directly reduce the number of SWAP gates needed to place the graphs. A second

solution is to consider devices with all-to-all connectivity, such as the Quantinuum H1-1 ion trap device [23]. An experimental implementation of the hybrid model on this device is discussed in chapter 5.

Chapter 4

Local complementation passes

4.1 Introduction

Implementations of the one-way model or the hybrid model introduced in chapter 3 can involve high-degree graphs which are difficult to place on devices with restricted qubit architectures because physical qubits can only interact with a small number of neighbouring qubits. Consequently, attempting to use any form of MBQC on such a platform can result in large overheads of SWAP gates, as discussed in section 3.7. However, tools such as *local complementation (LC)* and *edge local complementation (ELC)* or *pivoting*, discussed in section 2.5, allow for graph transformations which change the set of edges E_{ij} of a graph G , using local Clifford operations. By leveraging these techniques it is possible to reduce the entanglement depth of the measurement patterns. In this chapter, I introduce and discuss the implementation of specialised algorithms that LC and ELC to reduce entanglement depth.

Both LC and ELC can change the degree of various vertices in a graph. It follows that these transformations can be exploited to obtain a new graph, equivalent to the original, with a lower maximum degree. However, the number of graphs equivalent under LC is intractable for larger graphs [96, 97]. An exhaustive approach where every possible case in the LC equivalent class would be investigated and the most suitable candidate would be selected is not feasible. Consequently, a more careful and computationally cheaper approach must be followed.

Specifically, I introduce four separate algorithms that can be used to find a se-

quence of LC and/or ELC operations on carefully selected vertices to reduce the maximum degree of the graph. The first algorithm relies exclusively on LC operations and is discussed in section 4.2. The second uses only ELC operations and is discussed in section 4.3. The third algorithm uses both techniques but prioritises LC operations which are cheaper. The fourth algorithm expands upon the third, by selecting the best of a finite number of possible solutions each separated by a certain degree of randomisation. Since both the third and fourth algorithms are composite approaches that build up on the first two, they are jointly discussed in section 4.4. Finally, the results of these algorithms in practice are shown and compared in section 4.5.

4.2 Local complementation algorithm

The first algorithm I introduce is one which only relies on LC operations. Consider a ZX diagram in MBQC-form, with the underlying graph $G = (V, E, I, O)$ where $I \subseteq V$ is the set of inputs and $O \subseteq V$ is the set of outputs. It is possible to reduce $\Delta(G)$ by identifying all the vertices where $\deg(v) = \Delta(G)$ and applying a transformation such that $\deg'(v) < \Delta(G)$ (where \deg' represents the degree of a vertex after the transformation). Applying LC about v does not change any of the edges of the vertex v itself, but it does affect the edges of its neighbours, $N_G(v)$. In order to reduce $\Delta(G)$ it is necessary to find a vertex $u \in N_G(v)$ where $G \star u$ results in $\deg'(v) < \deg(v)$.

Additionally, it is important to ensure that for every vertex $w \in N_G(u)$, $\deg'(w) < \Delta(G)$. By understanding the effect of LC on the neighbourhood $N_G(u)$, the new

degree of vertex w , $deg'(w)$ can be derived as shown in equation 4.1,

$$deg'(w) = \text{New degree of } w,$$

$$deg(w) = \text{Previous degree of } w,$$

$$|N_G(u) - N_G(w) - \{w\}| = \text{Neighbours added to } w \text{ due to } G \star u,$$

$$|N_G(u) \cap N_G(w)| = \text{Neighbours removed from } w \text{ due to } G \star u,$$

$$deg'(w) = deg(w) + |N_G(u) - N_G(w) - \{w\}| - |N_G(u) \cap N_G(w)|$$

$$= |N_G(w)| + |N_G(u) - N_G(w) - \{w\}| - |N_G(u) \cap N_G(w)|$$

$$= |N_G(w)| + |N_G(u)| - 2|N_G(u) \cap N_G(w)| - 1.$$

(4.1)

In the pseudocode below, I present algorithm 4 which iteratively identifies the neighbours of maximum degree vertices in a graph and attempts to find valid instances of LC to affect them. The algorithm terminates when no further improvements can be found for the graph. The resulting graph has an equal or lower maximum degree than the original and thus, the underlying measurement pattern also has a lower entanglement depth.

Algorithm 4: Reduces the degree of a graph using LC.

Input: The original graph, $G = (V, E, I, O)$
Output: The improved graph, $G' = (V, E', I, O)$

```

ALG4 ( $G$ )
begin
   $G' = G$ 
   $apply = True$ 
  while  $apply$  do
    | ( $G', apply$ ) = LC_PASS( $G'$ )
  end
  return  $G'$ 
end

```

LC_PASS (G) Reduces $\Delta(G)$ by 1 using LC.

```

begin
   $G' = G$ 
   $U = \{\}$ 
  for  $u \in V$  do
    | if  $deg(u) = \Delta(G')$  then
    | |  $U = U \cup \{u\}$ 
  end
   $successes = 0$ 
  for  $u \in U$  do
    | if  $deg(u) = \Delta(G')$  then
    | | for  $v \in N_{G'}(u) - I$  do
    | | | if LC_CHECK( $G', v$ ) then
    | | | |  $G' = G' * v$ 
    | | | |  $successes = successes + 1$ 
    | | | | break
    | | | end
    | | else
    | | |  $successes = successes + 1$ 
    | | end
  end
  return ( $G', successes == |U|$ )
end

```

LC_CHECK (G, v) Checks if LC reduces degree.

```

begin
   $B = N_G(v)$ 
  for  $w \in B$  do
    | Degree of  $w$  after the edges change.
    |  $deg'(w) = |N_G(w)| + |B| - 2|B \cap N_G(w)| - 1$ 
    | if  $deg'(w) \geq \Delta(G)$  then
    | | return False
  end
  return True
end

```

Algorithm 4 was tested on degree-4 square lattices of width $w_g \in [3, 10]$ which had been previously distorted with random applications of LC. Figure 4.1 illustrates the partial recovery of the distorted graphs that was possible using algorithm 4. For each separate grid size, this procedure was repeated 20 times, with figure 4 illustrating the average values.

It should be noted that the scaling of algorithm 4 is polynomial in the size of the graph and can be estimated by observing the factors proportional to the number of vertices n that emerge at different points in the algorithm. The algorithm is applied iteratively while reducing the entanglement depth of the graph by 1 (at worst) during each iteration. As a vertex can, at most, be entangled with every other vertex, the number of iterations cannot exceed $n - 1$. During each iteration, the algorithm has to locate all the (current) maximum degree vertices, $u \in V$ in the graph which adds another factor proportional to and upper-bounded by n . For each maximum degree vertex u , the algorithm has to consider all of its neighbours $v \in N_G(u)$ as potential candidates for local complementation, adding another factor of $n - 1$ to the scaling. Finally, for each candidate vertex v , we have to consider the impact of the local complementation $G \star v$ on its own neighbourhood, $w \in N_G(v)$. So the overall scaling of algorithm 4 is upper-bounded by $O(n^4)$ where n is the number of vertices in the graph.

There is still room for improvement between the output of algorithm 4 and the benchmark indicated by the dashed horizontal line, which motivated the development of other more effective algorithms discussed in sections 4.3 and 4.4.

4.3 Pivoting algorithm

ELC can also reduce the maximum degree of a graph, and here I propose a second algorithm, algorithm 5, which exclusively relies on ELC to accomplish this. ELC is a more complex and more computationally expensive operation than LC but it is expected to allow us to find some potential improvements that algorithm 4 may have missed.

To reduce $\text{deg}(u)$ using ELC it is necessary to identify the neighbours $v \in$

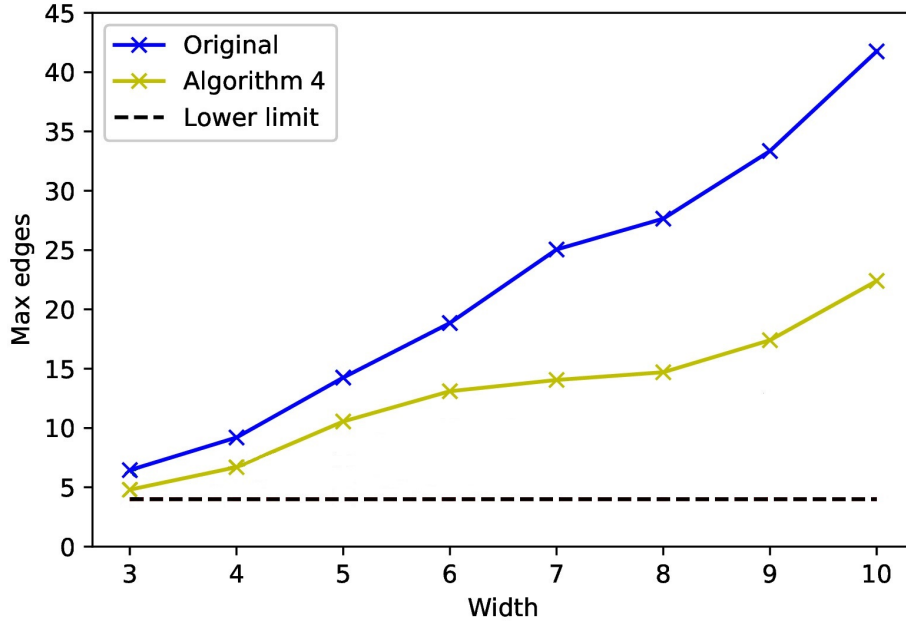


Figure 4.1: Evaluating the effectiveness of algorithm 4 in lowering the maximum degree of graphs of width $w_g \in [3, 10]$. The graphs are constructed by distorting a degree-4 square lattice using random applications of LC. The blue datapoints indicate the number of maximum edges after distortion and before recovery. The yellow datapoints indicate maximum edges after using algorithm 4 which relies on LC. The dashed horizontal line indicates the original degree of the lattice before the distortions and the minimum achievable after recovery. This is done 20 times for each width value and the datapoints indicate the average max edges before and after the application of algorithm 4.

$N_G(u)$ and for each v , to identify its own neighbours $w \in N_G(v) - \{u\}$. The focus is on three subsets, $A(G, v, w), B(G, v, w), C(G, v, w) \subseteq V$ as previously discussed in section 2.5. I define the vertices in these sets as follows: $a \in A, b \in B, c \in C$. The new degrees of these vertices in $G' = G^{(vw)}$ are given by equation 4.2 which is derived in a way similar to 4.1,

$$\begin{aligned}
 deg'(a) &= |N_G(a)| + |B| + |C| - 2|B \cap N_G(a)| - 2|C \cap N_G(a)|, \\
 deg'(b) &= |N_G(b)| + |A| + |C| - 2|A \cap N_G(b)| - 2|C \cap N_G(b)|, \\
 deg'(c) &= |N_G(c)| + |B| + |A| - 2|B \cap N_G(c)| - 2|A \cap N_G(c)|.
 \end{aligned} \tag{4.2}$$

As such, I present algorithm 5 which iteratively identifies the neighbours of maximum degree vertices in a graph and attempts to find neighbouring connected pairs

of vertices v and w where $\Delta(G^{(vw)}) \leq \Delta(G)$. The algorithm terminates when no further improvements can be found for the graph.

The scaling of algorithm 5 can be derived similarly to algorithm 4 with the difference that pivoting is a rotation about a *pair* of vertices rather than a single vertex. This means that in the worst case scenario where every vertex is entangled with every other vertex, the number of possible pivoting operations is proportional to n^2 instead of n (which is the case for local complementation). This raises the overall scaling of algorithm 5 to $O(n^5)$. This is slightly more expensive than algorithm 4 because pivoting is a higher order operation when compared to LC. However, it is still polynomial scaling and it can easily be run on a classical machine even for larger graphs.

Algorithm 5: Reduces the degree of a graph using ELC.

Input: The original graph, $G = (V, E, I, O)$ **Output:** The improved graph, $G' = (V, E', I, O)$

```

ALG5 ( $G$ )
begin
   $G' = G$ 
   $apply = True$ 
  while  $apply$  do
    |  $(G', apply) = ELC\_PASS(G')$ 
  end
  return  $G'$ 
end

ELC_PASS ( $G$ ) Reduces  $\Delta(G)$  by 1 using ELC.
begin
   $G' = G$ 
   $U = \{\}$ 
  for  $u \in V$  do
    | if  $deg(u) = \Delta(G')$  then
    | |  $U = U \cup \{u\}$ 
  end
   $successes = 0$ 
  for  $u \in U$  do
    | if  $deg(u) = \Delta(G')$  then
    | |  $E_P = \{\}$ 
    | | for  $v \in N_{G'}(u) - I$  do
    | | | for  $w \in N_{G'}(v) - I - \{u\}$  do
    | | | | if  $(w, v) \notin E_P$  then
    | | | | |  $E_P = E_P \cup \{(v, w)\}$ 
    | | | | end
    | | | end
    | | | for  $(v, w) \in E_P$  do
    | | | | if ELC_CHECK( $G', v, w$ ) then
    | | | | |  $G' = G'^{(vw)}$ 
    | | | | |  $successes = successes + 1$ 
    | | | | | break
    | | | | end
    | | | else
    | | | |  $successes = successes + 1$ 
    | | end
  end
  return ( $G', successes == |U|$ )
end

ELC_CHECK ( $G, v, w$ ) Checks if ELC reduces degree.
begin
   $A = N_G(v) - N_G(w) - \{w\}$ 
   $B = N_G(w) - N_G(v) - \{v\}$ 
   $C = N_G(v) \cap N_G(w)$ 
  for  $X \in \{A, B, C\}$  do
    |  $Y, Z \in \{A, B, C\} - \{X\}$ 
    | for  $x \in X$  do
    | | Degree of  $x$  after the edges change.
    | |  $deg'(x) = |N_G(x)| + |Y| + |Z| - 2|Y \cap N_G(x)| - 2|Z \cap N_G(x)|$ 
    | | if  $deg'(x) \geq \Delta(G)$  then
    | | | return False
    | | end
  end
  return True
end

```

Algorithm 5 was tested on degree-4 square lattices of width $w_g \in [3, 10]$ which had been previously distorted with random applications of LC. Figure 4.2 illustrates the partial recovery of the distorted graphs that was possible using only ELC. Again, this procedure was repeated 20 times for each separate grid size and the figure shows the averaged resulting depth before and after the algorithm is applied. Algorithm 5 performs better than algorithm 4 on the same set of distorted grid samples which was analysed in figure 4.1, as pivoting is a higher order operation than LC. However, algorithm 5 still does not provide full recovery of the original grid and so a new algorithm that uses both local complementation and pivoting is discussed in section 4.4.

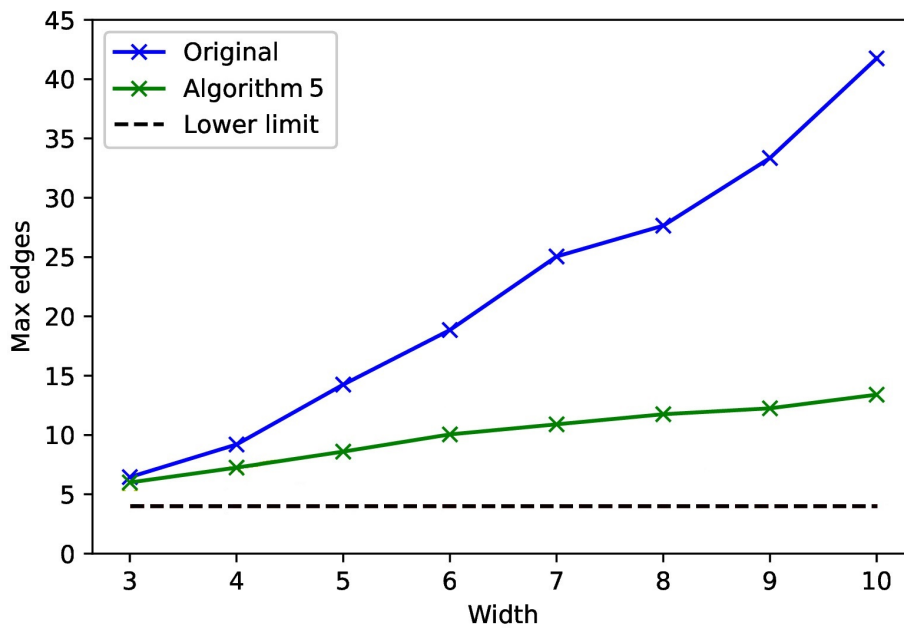


Figure 4.2: Evaluating the effectiveness of algorithm 5 in lowering the maximum degree of graphs of width $w_g \in [3, 10]$. The graphs are constructed by distorting a degree-4 square lattice using random applications of LC. This is done 20 times for each separate value of width with the datapoints in the figure representing the average number of maximum edges. The blue datapoints indicate the number of maximum edges after distortion and before recovery. The green datapoints indicate maximum edges after using algorithm 5 which relies on pivoting. The dashed horizontal line indicates the original degree of the lattice before the distortions and the minimum achievable after recovery.

4.4 Composite algorithms

Algorithms 4 and 5 may each find a number of possible improvements that do not necessarily overlap. As such, I expect the best approach to be a combination of the two. Thus, I present algorithm 6 which prioritises LC operations where possible (as they are cheaper) but attempts to find a further depth reduction using ELC when LC reaches a dead end.

Algorithm 6: Reduces the degree of a graph using both LC and ELC.

Input: The original graph, $G = (V, E, I, O)$

Output: The improved graph, $G' = (V, E', I, O)$

ALG6 (G)

begin

$G' = G$

$apply = True$

while $apply$ **do**

$(G', apply) = LC_PASS(G')$

if not $apply$ **then**

$(G', apply) = ELC_PASS(G')$

end

return G'

end

Furthermore, I have noticed that the results of algorithm 6 are somewhat dependent on the form of the original graph it is given as an input. If the solution space of interest contains all the graphs equivalent to G under LC, algorithms 4, 5 and 6 follow a trajectory of minimising $\Delta(G)$ towards a local minimum in this solution space.

The scaling of algorithm 6 is $O(n^5)$ where n is the number of vertices in the graph. This is the same cost as algorithm 5 since this algorithm also uses pivoting but in practice it may prove slightly cheaper than algorithm 5 because algorithm 6 prioritises LC over pivoting whenever possible.

Therefore, I introduce algorithm 7, which expands upon algorithm 6 by selecting the best of a number of solutions with each solution being obtained from a different starting point in the solution space. This is achieved by first creating multiple

copies of the original graph state, each of which represents a different optimisation ‘attempt’. Each copy is subjected to different number of random LC operations, called ‘distortions’ to ensure that each attempt has a different starting point. Algorithm 6 is then applied to each copy separately and the results are compared to pick the graph with the lowest degree. This process is equivalent to exploring multiple local minima in the solution space.

The scaling of algorithm 7 is $O(n^5 + c)$ where n is the number of vertices in the graph and c is a constant factor (related to the number of attempts and distortions chosen).

Algorithm 7: Reduces the degree of a graph using LC, ELC and some randomisation.

Input: The original graph, $G = (V, E, I, O)$

Output: The improved graph, $G' = (V, E', I, O)$

ALG7 ($G, attempts, distortions$)

```

begin
   $G' = G$ 
   $I' = V - I$ 
  for  $i \in [1, attempts]$  do
     $G'' = G$ 
    for  $k \in [1, distortions]$  do
       $u \in_{RAND} I'$ 
       $G'' = G'' \star u$ 
    end
     $G'' = \text{ALG6}(G'')$ 
    if  $\Delta(G'') < \Delta(G')$  then
       $G' = G''$ 
    end
  end
  return  $G'$ 
end

```

Algorithms 6 and 7 were tested on degree-4 square lattices of width $w_g \in [3, 10]$ which had been previously distorted with random applications of LC. This was done 20 times for each value of width. Figure 4.3 illustrates the partial recovery of the distorted graphs that was possible using both LC and pivoting. Algorithm 6 performs better than algorithms 4 and 5 on the same set of distorted grid samples which were analysed in figures 4.1 and 4.2, which is expected as there are now

two operations available that can be used to reduce the graph. In cases where LC cannot provide any further reduction, pivoting might be able to and vice versa. As a result, algorithm 6 outperforms both algorithms 4 and 5 which only rely on a single operation each. Algorithm 7 is another step-up from algorithm 6 which finally fully recovers up to 10x10 grids. This is achieved by widening the solution subspace investigated through multiple attempts.

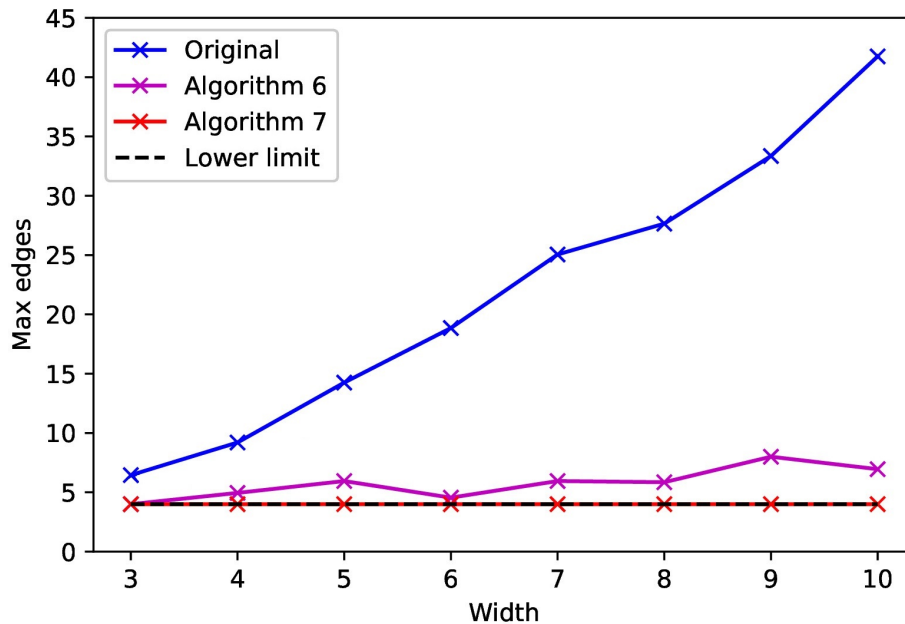


Figure 4.3: Evaluating the effectiveness of algorithms 6 and 7 in lowering the maximum degree of graphs of width $w_g \in [3, 10]$. The graphs are constructed by distorting a degree-4 square lattice using random applications of LC. The blue datapoints indicate the number of maximum edges after distortion and before recovery. The purple datapoints indicate maximum edges after using algorithm 6 which uses Lc and pivoting but prioritizes the former. The red datapoints indicate maximum edges after using algorithm 7 which is a variant of algorithm 6 which uses multiple attempts to search through a broader solution subspace. The dashed horizontal line indicates the original degree of the lattice before the distortions and the minimum achievable after recovery.

4.5 Result comparison for degree reduction algorithms

In this section, I discuss the results of applying algorithms 4, 5, 6 and 7 presented in sections 4.2-4.4 on a variety of sample graphs. I have chosen to investigate three different classes of graphs for this analysis and four different approaches at reducing $\Delta(G)$ for each of the three sample classes.

The first type of samples were obtained by distorting an $n \times n$ square lattice through a series of LC operations on randomly chosen vertices. The original graphs have a low degree, $\Delta(G) = 4$, which increases after the distortions. The reason for choosing this sample class is to evaluate how well the algorithms manage to recover the original grid. The minimum degree achievable is already known so there is a target result to use as a benchmark for the algorithms. As previously mentioned, the algorithms are applied 20 times to each separate grid size and the results presented in this section are the averages.

The second type of samples were obtained by generating random Clifford+T circuits. Specifically, I initialised quantum circuits with n qubits (20 separate circuits for each value of $n \in [3, 10]$), then added a total of 500 gates out of which 20% were T gates, 40% were Hadamard gates and 40% were CNOT gates. These gates were applied to random qubits/pairs of qubits. The resulting circuits were then converted into a ZX diagram in MBQC-form. This sample class was chosen because the Clifford+T gate set is universal making this a useful benchmark for the algorithms. The specific numbers of gates were chosen to make the resulting measurement patterns have width and depth similar to the corresponding square lattice samples.

The third type of samples are phase polynomials of width n containing a number of randomly-generated Pauli gadgets [75, 76]. These samples are interesting because phase polynomials are used in quantum chemistry applications such as VQE algorithms. The number of Pauli gadgets was chosen so that these graphs had a similar original width and depth to the corresponding square lattices. As before, 20

separate phase polynomial circuits were considered for each value of $n \in [3, 10]$ and the results in this section are averages.

I have compared the four different algorithms on each of these sample sets. Algorithm 4 only uses LC operations making its scaling $O(n^4)$. As a result this is the cheapest of the algorithms in terms of computational power. Algorithm 5 exclusively uses ELC operations making its scaling $O(n^5)$. Algorithm 6 uses both LC and ELC thus also giving it a scaling of $O(n^5)$. However, it prioritises LC operations whenever they lead to an improvement which means that its effective scaling will be lower than algorithm 5. Algorithm 7 is essentially an expansion of algorithm 6 with the addition of a constant factor.

Figure 4.4 compares the results of each algorithm on distorted square lattices of varying sizes. All four algorithms succeed in reducing the maximum degree of the grid. The approach using ELC as outlined in algorithm 5 is more effective than the approach relying purely on LC. This is to be expected since ELC can be viewed as a higher order operation. However, algorithm 6 produces even better results showing that the combination of both LC and ELC is more effective than the exclusive use of either one. The addition of some randomisation (algorithm 7) leads to the complete recovery of the original grid structure for sizes up to 10x10. Table 4.1 shows the numerical values of figure 4.4 in more detail.

Figure 4.5 compares the results on randomly-generated Clifford+T circuits of varying width. Table 4.2 lists the numerical values of figure 4.5. Figure 4.6 compares the results on randomly-generated phase polynomials of varying width. Table 4.3 shows the numerical values of figure 4.6 in more detail. In both cases, the outcomes follow the same hierarchy seen in the grid samples, with algorithm 7 resulting in the lowest entanglement depth. Regardless of the circuit class chosen as the samples, the results show that the same hierarchy exists in the effectiveness of the four algorithms as what would be expected. Specifically, algorithm 7 is a direct improvement over algorithm 6, therefore, it performs better in every scenario, while algorithm 6 is itself a direct improvement over algorithms 4 and 5 as it incorporates and consequently outperforms both. The only connection that is less direct is that

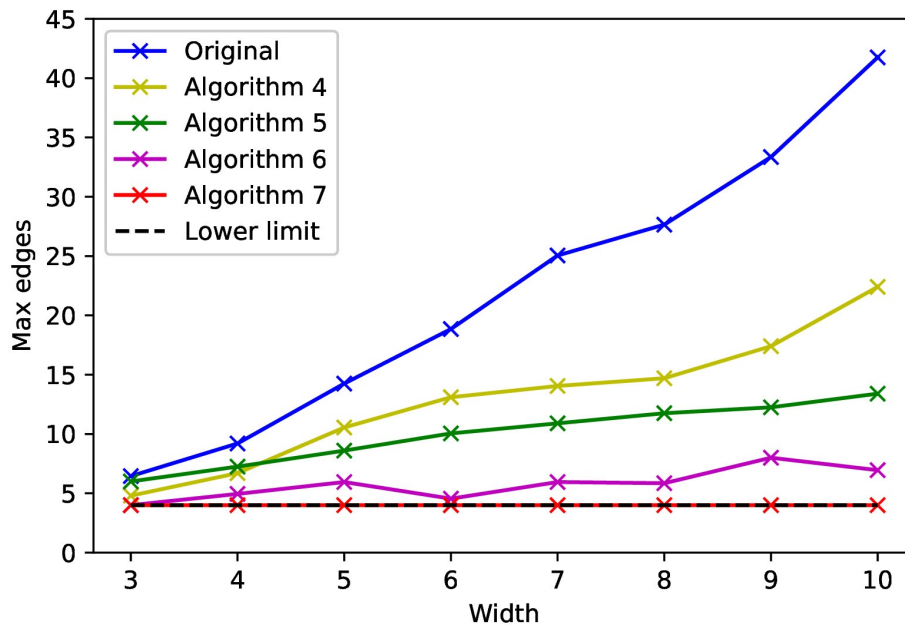


Figure 4.4: Comparing the effectiveness of four algorithms to lower the maximum degree of graphs of $width \in [3, 10]$. The graphs are constructed by distorting a degree-4 square lattice using random applications of LC. The blue datapoints indicate the number of maximum edges after distortion and before recovery. The yellow datapoints indicate maximum edges after using algorithm 4 which relies on LC. The green datapoints indicate maximum edges after using algorithm 5 which relies on pivoting. The purple datapoints indicate maximum edges after using algorithm 6 which uses LC and pivoting but prioritizes the former. The red datapoints indicate maximum edges after using algorithm 7 which is a variant of algorithm 6 applied over several attempts. The dashed horizontal line indicates the original degree of the lattice before the distortions and the minimum achievable after recovery.

between algorithms 4 and 5 as they are based on different operations, but the latter outperforming the former is within expectations because pivoting is in many ways a higher order operation than LC: Not only is one step of pivoting equivalent to three steps of LC but pivoting is performed about *pairs of vertices* rather than individual vertices, giving algorithm 5 more choices at every iteration, which is also reflected in the slightly steeper scaling. However, considering that all four algorithms are computationally cheap to run on a classical machine due to their low polynomial scaling - with only minor differences between the cheaper and most expensive of the four - in most cases, algorithm 7 should be used as it is the most comprehensive

n	Original	Algorithm 4	Algorithm 5	Algorithm 6	Algorithm 7
3	6.5	4.8	6.0	4.0	4.0
4	9.2	6.7	7.2	5.0	4.0
5	14.2	10.6	8.6	6.0	4.0
6	18.9	13.1	10.1	4.6	4.0
7	25.1	14.1	10.9	5.9	4.0
8	27.7	14.7	11.7	5.8	4.0
9	33.4	17.4	12.3	8.0	4.0
10	41.8	22.4	13.4	6.9	4.0

Table 4.1: Table showing the numerical values of the datapoints in figure 4.4. Grids of dimensions $n \times n$ are distorted and then subjected to each of the four algorithms for a maximum edge reduction. This is repeated 20 times for each size $n \in [3, 10]$. The entries in the table represent the average maximum edges for each sample size after each algorithm is applied.

n	Original	Algorithm 4	Algorithm 5	Algorithm 6	Algorithm 7
3	14.6	10.6	9.1	8.7	7.8
4	17.7	13.7	12.1	11.5	10.5
5	22.1	17.3	14.7	14.2	12.9
6	24.7	19.8	17.2	16.6	15.2
7	29.7	23.5	20.0	19.6	18.5
8	32.3	25.9	22.5	22.4	21.0
9	35.1	28.7	24.5	23.9	23.0
10	38.1	30.7	27.5	26.9	26.3

Table 4.2: Table showing the numerical values of the datapoints in figure 4.5. 20 randomly-generated Clifford+T circuits for each width $n \in [3, 10]$ are converted to ZX diagrams in MBQC-form. They are then distorted and then subjected to each of the four algorithms for a maximum edge reduction. The entries in the table represent the average maximum edges for each sample size after each algorithm is applied.

and best performing option.

In general, the algorithms introduced in this chapter can be used to reduce entanglement depth of measurement patterns both in the pure one-way model and the hybrid model introduced in chapter 3. Specifically, reduced entanglement depth can help make any version of MBQC more compatible with devices that exhibit low connectivity between physical qubits as less interconnected graphs are easier to fit onto chips with restricted architectures and require fewer SWAP gates to do so - a topic that was discussed more extensively in section 3.7. As previously mentioned, an alternative solution is to consider devices with full connectivity, to bypass the

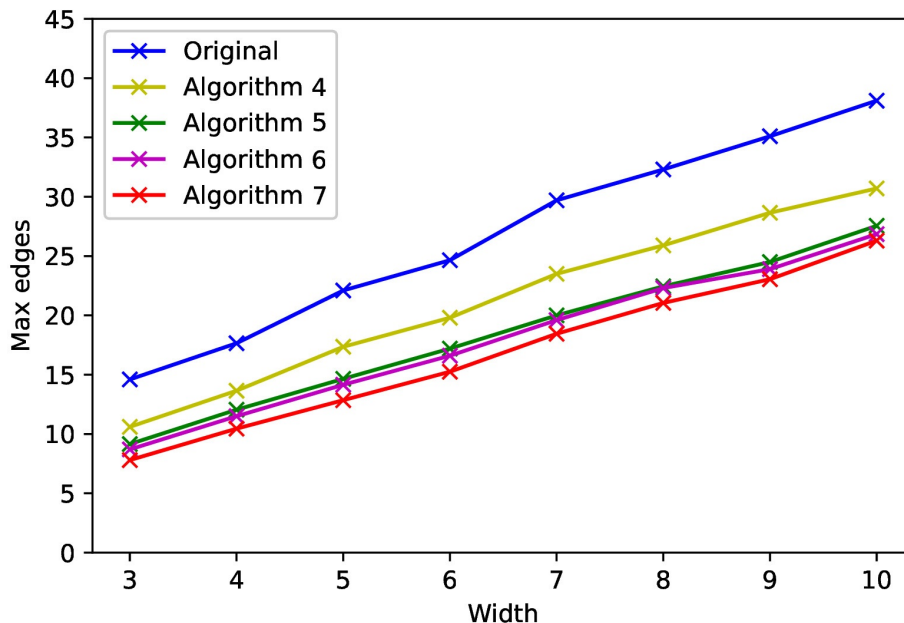


Figure 4.5: Comparing the effectiveness of four algorithms to lower the maximum degree of graphs of $width \in [3, 10]$. The graphs are constructed by converting randomly-generated Clifford+T circuits to measurement patterns. The blue datapoints indicate the number of maximum edges after distortion and before recovery. The yellow datapoints indicate maximum edges after using algorithm 4 which relies on LC. The green datapoints indicate maximum edges after using algorithm 5 which relies on pivoting. The purple datapoints indicate maximum edges after using algorithm 6 which uses LC and pivoting but prioritizes the former. The red datapoints indicate maximum edges after using algorithm 7 which is a variant of algorithm 6 applied over several attempts.

routing problem entirely, such as the Quantinuum H1-1 ion trap device [23]. Additionally, having a measurement pattern with lower entanglement depth means that, by definition, fewer timesteps are required to prepare the underlying graph state. As the preparation of the graph state is constructed mainly by CZ gates, a lower entanglement depth directly results in fewer serialised CZ gates, which is equivalent to a lower execution time for the measurement pattern. Consequently, the algorithms introduced in this chapter remain beneficial and relevant even in the absence of routing and on platforms with full connectivity.

In chapter 5, I discuss an implementation of the hybrid model and the depth reduction algorithms on a quantum circuit executed on the H1-1 device. The goals

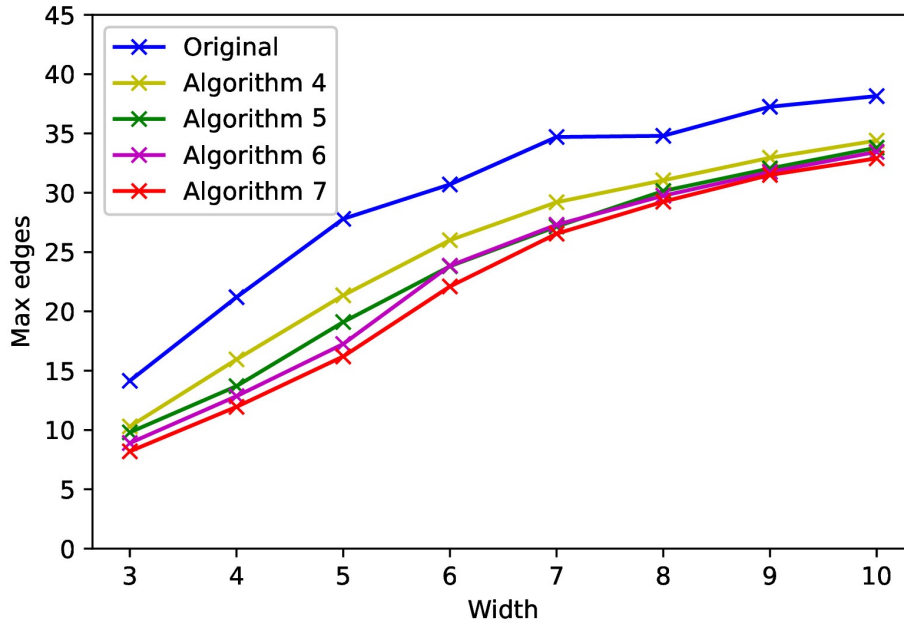


Figure 4.6: Comparing the effectiveness of four algorithms to lower the maximum degree of graphs of $width \in [3, 10]$. The graphs are constructed by converting randomly-generated phase polynomial circuits to measurement patterns. The blue datapoints indicate the number of maximum edges after distortion and before recovery. The yellow datapoints indicate maximum edges after using algorithm 4 which relies on LC. The green datapoints indicate maximum edges after using algorithm 5 which relies on pivoting. The purple datapoints indicate maximum edges after using algorithm 6 which uses LC and pivoting but prioritizes the former. The red datapoints indicate maximum edges after using algorithm 7 which is a variant of algorithm 6 applied over several attempts.

of this experiment are many: It serves as a proof-of-concept of the techniques introduced in this thesis and a practical demonstration of their use in practice. It also serves as one of the first implementations of the one-way model on a device which supports adaptive measurements. Finally, it allows for a more meaningful evaluation of both the hybrid model and the algorithms introduced in this chapter.

n	Original	Algorithm 4	Algorithm 5	Algorithm 6	Algorithm 7
3	14.2	10.3	9.8	8.9	8.2
4	21.2	16.0	13.7	12.9	12.0
5	27.8	21.3	19.1	17.3	16.2
6	30.7	26.0	23.8	23.8	22.1
7	34.9	29.2	27.2	27.3	26.6
8	34.7	31.0	30.2	29.8	29.1
9	37.2	33.0	32.1	31.7	31.5
10	38.2	34.4	33.8	33.4	32.9

Table 4.3: Table showing the numerical values of the datapoints in figure 4.6. 20 randomly-generated phase polynomial circuits for each width $n \in [3, 10]$ are converted to ZX diagrams in MBQC-form. They are then distorted and then subjected to each of the four algorithms for a maximum edge reduction. The entries in the table represent the average maximum edges for each sample size after each algorithm is applied.

Chapter 5

Ion trap implementation

5.1 Introduction

While the one-way model of quantum computation is equivalent to the circuit model and carries its own set of advantages and disadvantages, experimental realizations of MBQC have been scarcer, mainly due to the additional requirements such as adaptive measurements and higher numbers of physical qubits needed. Still, there have been experimental demonstrations of feed-forward and non-destructive CNOT gates acting on independent photon qubits since the early 2000s [98, 99].

Additionally, this was more recently expanded to a demonstration of MBQC on a photonics device containing up to 5 qubits [100]. Despite these promising results in the field of optics, there have not been any notable realizations of the one-way model on other platforms, such as superconducting qubits or ion traps. One of the goals of this thesis was to change this, by achieving a realization of not only the one-way model but also the techniques developed earlier in the thesis, on an ion trap device. The motivation for this experiment was partly to produce a proof-of-concept for the one-way model in a platform other than photonics, but also to more specifically observe how our techniques work in practice, using non-trivial measurement patterns using up to 18 physical qubits.

This chapter focuses on the experimental implementation of the hybrid model and the local complementation pass technique on the Quantinuum H1-1 [23], achieved over the months of March & April 2023.

In section 5.2, the motivation for an experimental implementation of the techniques presented in this work are discussed. In the same section, the reasons for choosing this particular hardware over alternatives are also given. The device specifications of the H1-1 device used for the experiment are listed in section 5.3. Section 5.4 gives a more detailed description of the experimental setup, including information about the methods and sample circuits used. The results of the hybrid model implementation are compared to both a noiseless benchmark and a circuit model implementation on the same device in the presence of noise. This comparison is visually presented in section 5.5.

5.2 Motivation for the experiment

While measurement patterns in the one-way model can be read and executed in a manner very similar to quantum circuits, they exhibit some additional properties which make the requirements for an experimental realisation steeper than that of an equivalent circuit.

The two most important such properties are the requirement for adaptive measurements and the fact that converting a circuit to a measurement pattern requires a large qubit overhead. The Quantinuum H1 series ion trap devices [23] are among the first cases of quantum hardware to support adaptive measurements. The H1-1 device used in this thesis allows up to 20 physical qubits to be stored at once which is also sufficiently large to accommodate the qubit overhead introduced when transpiling our 3-qubit circuit sample to a version of the hybrid model introduced in chapter 3.

It is important to note that the specific sample circuit used in this experiment was explicitly chosen to have a high enough circuit depth, such that converting it to a single measurement pattern requires more qubits than what the device can accommodate, yet under 20 qubits when the hybrid model is used instead. An additional reason for choosing this specific device is the fact that this hardware does not require routing during compilation which, as discussed in section 3.7, would result in large depth overheads when combined with MBQC.

Finally, this machine is compatible with the Pytket library which is used extensively throughout this thesis and includes useful tools such as the conversion of quantum circuits to ZX-diagrams and measurement patterns. Therefore, this particular experimental setup not only meets the requirements for an implementation of the one-way in general, but is also extremely compatible with the specific tools and requirements related to this thesis.

5.3 Quantinuum H1-1 device specifications

This section contains information about the H1-1 device. All the data comes from the "System Model H1 Product Data Sheet" from the official Quantinuum website [23].

The H1-series devices (two at the time of writing) feature 20 physical qubits with all-to-all connectivity and quantum gates implemented via lasers. The devices use the *quantum charged-coupled device (QCCD)* architecture with five zones for parallel gates [101]. The devices also allow mid-circuit measurements and qubit reuse after the mid-circuit measurements which meet the key requirements for the execution of MBQC and HMQC. The native gate set involves the single-qubit rotations and two-qubit ZZ gates defined by equations 5.1-5.3,

$$U_{1q}(\theta, \phi) = e^{-i(\cos\phi X + \sin\phi Y)\frac{\theta}{2}} = \begin{bmatrix} \cos\frac{\theta}{2} & -ie^{-i\phi}\sin\frac{\theta}{2} \\ -ie^{i\phi}\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}, \quad (5.1)$$

$$R_z(\lambda) = e^{-iZ\frac{\lambda}{2}} = \begin{bmatrix} e^{-i\frac{\lambda}{2}} & 0 \\ 0 & e^{i\frac{\lambda}{2}} \end{bmatrix}, \quad (5.2)$$

$$RZZ(\theta) = e^{-i\frac{\theta}{2}Z\otimes Z} = e^{-i\frac{\theta}{2}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\theta} & 0 & 0 \\ 0 & 0 & e^{i\theta} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5.3)$$

The conversion from the Clifford+T gateset we use to the native gateset used by the H1-1 device is handled automatically by the TKET compiler when a circuit object

is passed for compilation [2, 23]. Table 5.1 lists the error rates of various operations for the H1-1 device.

Error cause	Minimum val.	Typical val.	Maximum val.
Single-qubit gate	1×10^{-5}	4×10^{-5}	3×10^{-4}
Two-qubit gate	1.7×10^{-3}	2×10^{-3}	5×10^{-3}
State preparation & measurement	2×10^{-3}	3×10^{-3}	5×10^{-3}
Memory error/qubit for $d = 1$	1×10^{-4}	2×10^{-4}	1×10^{-3}
Measurement cross-talk	5×10^{-6}	1×10^{-5}	2×10^{-4}

Table 5.1: Known error rates of various operations of the Quantinuum H1-1 ion trap device. Minimum, typical and maximum values are listed for each type.

5.4 Experiment overview

The circuit sample used in this experiment is a randomly generated 3-qubit phase polynomial circuit with 20 phase gadgets which is produced using the algorithm described in section 2.8. The reason for choosing a sample in the class of phase polynomial circuits is due to their relevance in NISQ applications, such as quantum chemistry where the VQE algorithm is often used [60, 61, 62].

The sample was chosen to contain no fewer than 20 phase gadgets because it has to be sufficiently deep to demonstrate the benefits of depth reduction that would not be present when transpiling a shallower circuit to MBQC or the hybrid model. It also does not contain any *more* than 20 phase gadgets or 3 qubits because that would exceed the qubit capacity of the H1-1 device using the hybrid model even after splitting it into two segments. If the sample was any wider or deeper, it would require at least three segments to be implemented via the hybrid model. This specific sample cannot be implemented in a single segment, in a pure MBQC form, as the corresponding pattern requires more than 20 qubits and thus exceeds the capacity of the device.

The sample circuit is then simplified, removing redundant gate pairs (such as a sequence of two Hadamard gates acting on the same qubit in a row or similar), by using some of the methods available in the Pytket library. The next step is to split the circuit into two subcircuits of equal depth as described in section 3.2 and to convert each one of them into a ZX-diagram in MBQC-form using the corresponding

methods from the Pytket library. On each of the resulting measurement patterns, a Python implementation of algorithm 6 discussed in section 4.2 is applied, in order to reduce the entanglement depth of the patterns as much as possible.

It is also important to note that because the original circuit contains 3 qubits, the resulting measurement patterns in the hybrid model also contain 3 logical qubits each, but many more physical qubits (close to 20). Therefore, input/output maps are generated during the conversion to measurement patterns and the physical qubits of the second pattern have to be re-labelled so that the 3 output qubits of the first pattern are fed into the corresponding input qubits of the second pattern.

Subsequently, the measurement patterns are compiled together into a new circuit object (which also contains adaptive measurements) and then executed on the H1-1 device. The original circuit is also executed on the device directly as a point of comparison, but also run on the H1-1E emulator with quantum noise disabled to serve as a benchmark. The results of the benchmark, the circuit model and the hybrid model are then compared in section 5.5.

5.5 Result discussion

The original sample circuit used in this experiment is a 3-qubit circuit which can be seen in figure 5.1. When transpiled into a single measurement pattern, the entanglement depth decreases but the width increases to 28 which means that this measurement pattern exceeds the qubit capacity of the H1-1 device. However, when a 2-segment version of the hybrid model is used the width decreases to 18, allowing the new circuit to be executed on the device. Table 5.2 shows a comparison of width, 2-qubit depth and total number of 2-qubit gates between the 3 implementations.

Resource	Circuit Model	One-way Model	Hybrid Model ($N = 2$)
Qubits	3	28	18
2-qubit Depth	36	12	13
2-qubit Gates	36	77	68

Table 5.2: Comparison of width, 2-qubit depth and total number of 2-qubit gates for a quantum circuit implemented in the circuit model, one-way model and the hybrid model with 2 segments. Both the one-way model and hybrid model implementations have been subjected to the depth reduction algorithm from chapter 4.

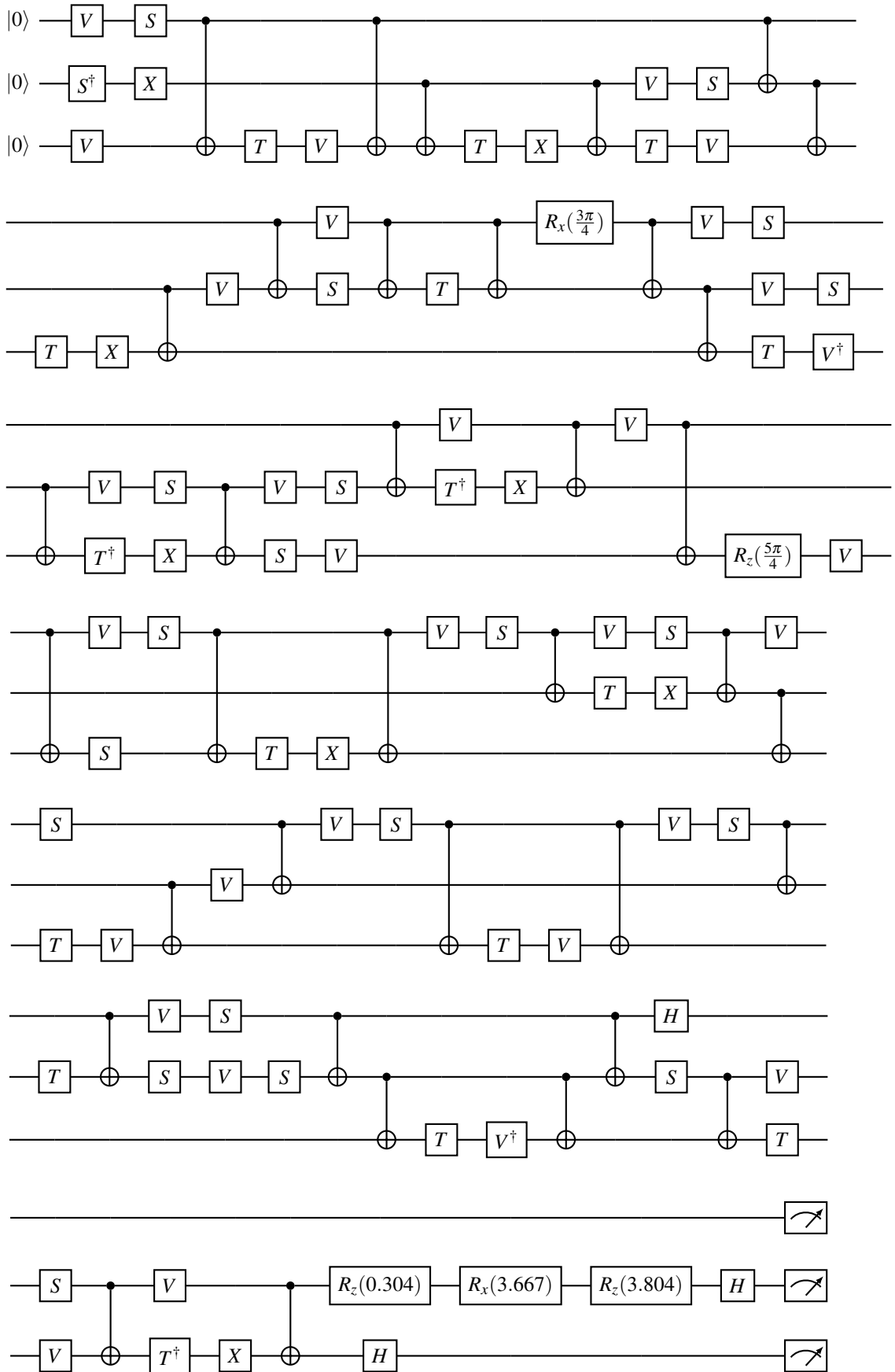


Figure 5.1: The original 3-qubit sample circuit implemented on the device.

These metrics are also illustrated in figure 5.2 which compares the same resources while placing into perspective the effectiveness of the hybrid model compared to the normal version of the one-way model. The black dotted line connects the circuit model and MBQC data points and is there to highlight that the hybrid model results in ‘cheaper’ depth reduction than pure MBQC. The red vertical dashed line indicates the qubit capacity of the H1-1 device. The MBQC implemen-

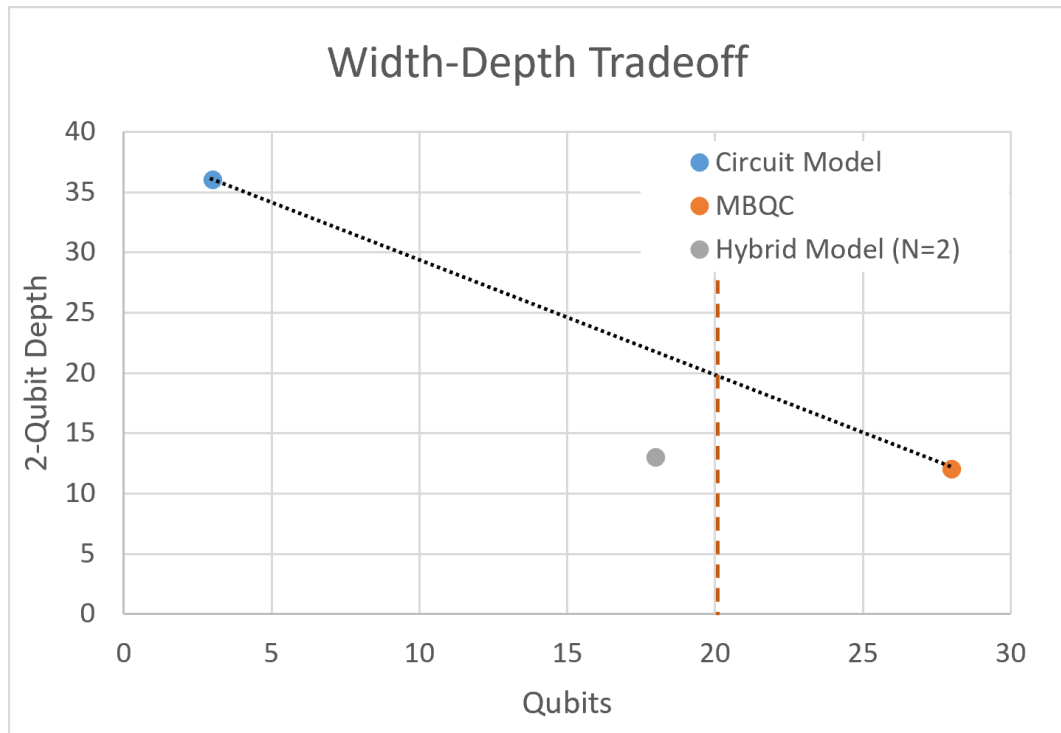


Figure 5.2: Comparison of the resource requirements of three different implementations for the sample circuit shown in figure 5.1. The x-axis indicates the qubit requirement while the y-axis indicates the 2-qubit depth requirement. The three implementations are as follows: Blue = Circuit model, Orange = MBQC, Grey = Hybrid model (with 2 segments). The vertical dotted line represents the qubit capacity of the device.

tation cannot be executed on the device due to its qubit requirement but the circuit model implementation and the hybrid model implementation are both run on the device for comparison. The circuit model is first executed on the H1-1E emulator, which is designed to perform like the real device but provides the option for quantum noise to be turned off, to serve as a benchmark for the other two results. Each of the three instances is run and measured 100000 times to minimize any statistical variation. The output distributions are plotted side-by-side in figure 5.3. To more

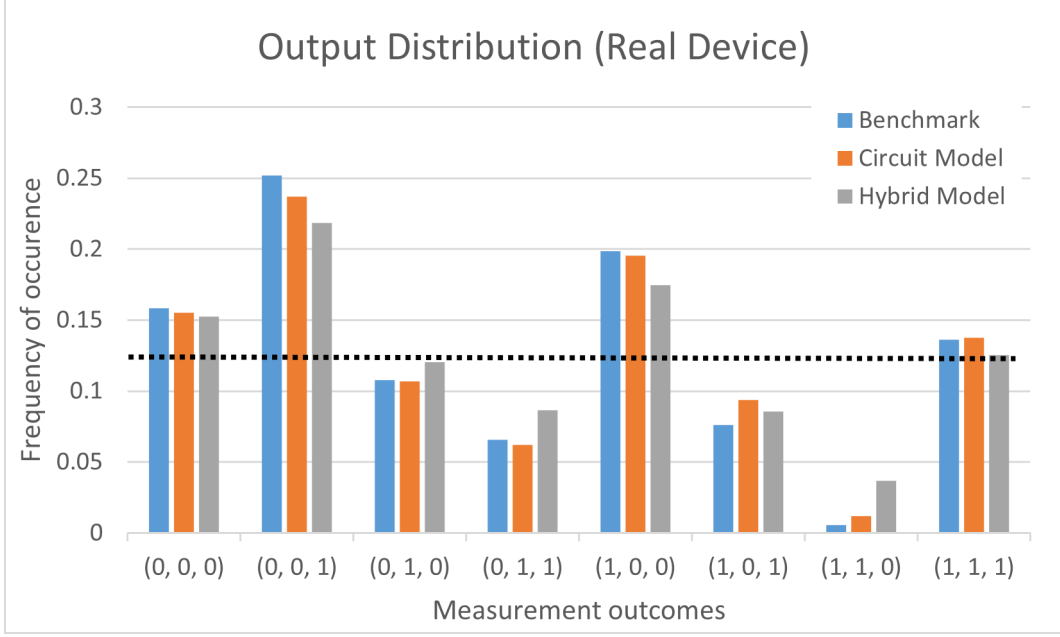


Figure 5.3: Comparison of the benchmark (blue), circuit model (orange) and hybrid model (grey) output distributions for 100000 shots. A horizontal dashed line is placed at a frequency of 0.125 which indicates what a uniform distribution would look like for a the 3-qubit circuit shown in figure 5.1.

accurately compare two output distributions D_i, D_j , a metric called *total variation distance* [102] is used, which is defined in equation 5.4,

$$d(D_i, D_j) = \text{Max}(|D_i(u) - D_j(u)| : u \in D_i, D_j). \quad (5.4)$$

Both the circuit model and the hybrid model are compared against the benchmark distribution. For the circuit model, the total variation distance is 0.017 and for the hybrid model it is 0.034 indicating that the latter is further off from the error-free values. As shown in figure 5.3, for most measurement outcomes the hybrid model results tend closer to the uniform distribution than the circuit model.

Consider the single-qubit depolarizing channel $\Delta_{1,k}(\rho)$ given in equations 5.5 and 5.6 and parameterised by scalar k [14],

$$\Delta_{1,k}(\rho) = \sum_{i=0}^3 K_i \rho K_i^\dagger, \quad (5.5)$$

$$K_0 = \sqrt{1 - \frac{3k}{4}}I, K_i = \sqrt{\frac{k}{4}}P_i, i \in [1, 3]. \quad (5.6)$$

In this case, when the scalar k is large, the strength of the channel is maximized and the measured state of the system tends to the maximally-mixed state, which for the single qubit case returns each outcome with a 50% probability [14]. This can be generalised to the n -qubit case with a depolarising channel $\Delta_{n,k}(\rho)$ given in equations 5.7 and 5.8,

$$\Delta_{n,k}(\rho) = \sum_{i=0}^{4^n-1} K_i \rho K_i^\dagger, \quad (5.7)$$

$$K_0 = \sqrt{1 - \frac{(4^n-1)k}{4^n}}I, K_i = \sqrt{\frac{k}{4^n}}P_i, i \in [1, 4^n-1]. \quad (5.8)$$

A simplified but equivalent definition of the n -qubit depolarizing channel is given in equation 5.9 [14],

$$\Delta_{n,k}(\rho) = \rho_k = (1-k)\rho + k\frac{I}{2^n}. \quad (5.9)$$

When we consider the 3-qubit case (as is the sample circuit executed on the ion trap device) given by equation 5.10, the maximally-mixed state would look like a uniform distribution over the eight possible measurement outcomes, each occurring with a probability of 12.5%. Therefore, by observing the measurement results on the device, given in figure 5.3, we notice that for the hybrid model case, the measurements are closer to the maximally-mixed state than the circuit model case which indicates the presence of strong depolarizing noise.

$$\rho_k = (1-k)\rho + k\frac{I}{8} \quad (5.10)$$

From the definition of the depolarizing channel, when the scalar parameter k tends to 0 the effect of the depolarizing channel disappears and the result tends to the error-free value unaffected by depolarizing noise while when k tends to 1 the result instead tends to the maximally-mixed state - a uniform distribution, as indicated in

equations 5.11,

$$\begin{aligned}\lim_{k \rightarrow 0} \rho_k &= \rho, \\ \lim_{k \rightarrow 1} \rho_k &= \frac{I}{8}.\end{aligned}\tag{5.11}$$

It is then reasonable to make the assumption that the experimental results on the H1-1 device are subject to different levels of depolarizing noise. This assumption introduces the possibility of correcting for the effects of the noise with simple error mitigation techniques [73].

It should be noted that each qubit is normally subject to its separate, local depolarizing channel, while the error mitigation technique we use would only be useful in reducing the effect of global depolarizing noise acting on the entire system [73]. In practice, we notice that the error mitigation technique seems to work better in the hybrid model case than it does in the circuit model case, which would indicate that converting the circuit to a graph state has the welcome side-effect of moving the depolarising noise acting on the qubits individually into something closer to global depolarising noise, making it easier to remove.

We define the probability of observing the measurement outcome b as p^b for the noise-free case and as p_k^b in the presence of depolarizing noise. To calculate the latter, we use equation 5.12,

$$\begin{aligned}p_k^b &= \text{Tr}[[b] \langle b| U \rho_k U^\dagger], \\ &= \text{Tr}[[b] \langle b| U ((1-k)\rho + k\frac{I}{8}) U^\dagger], \\ &= (1-k)\text{Tr}[[b] \langle b| U \rho U^\dagger] + k\text{Tr}[[b] \langle b| U \frac{I}{8} U^\dagger], \\ &= (1-k)p^b + k\text{Tr}[[b] \langle b| \frac{I}{8}], \\ &= (1-k)p^b + \frac{k}{8}, \\ &= (1-k)p^b + \frac{k}{8} + \frac{1-k}{8} - \frac{1-k}{8}, \\ &= (1-k)(p^b - \frac{1}{8}) + \frac{1}{8}.\end{aligned}\tag{5.12}$$

The effect of the scalar parameter k is to move the measurement outcomes from their real value closer to the uniform distribution. Therefore, the noisy measurement outcomes p_k^b can be scaled back to the real outcomes p^b using k . Figure 5.4 is produced by varying an estimate of k for each model and adjusting the real results accordingly while computing the new total variation distance from the benchmark for each case. By choosing the points where the total variation distance is at its min-

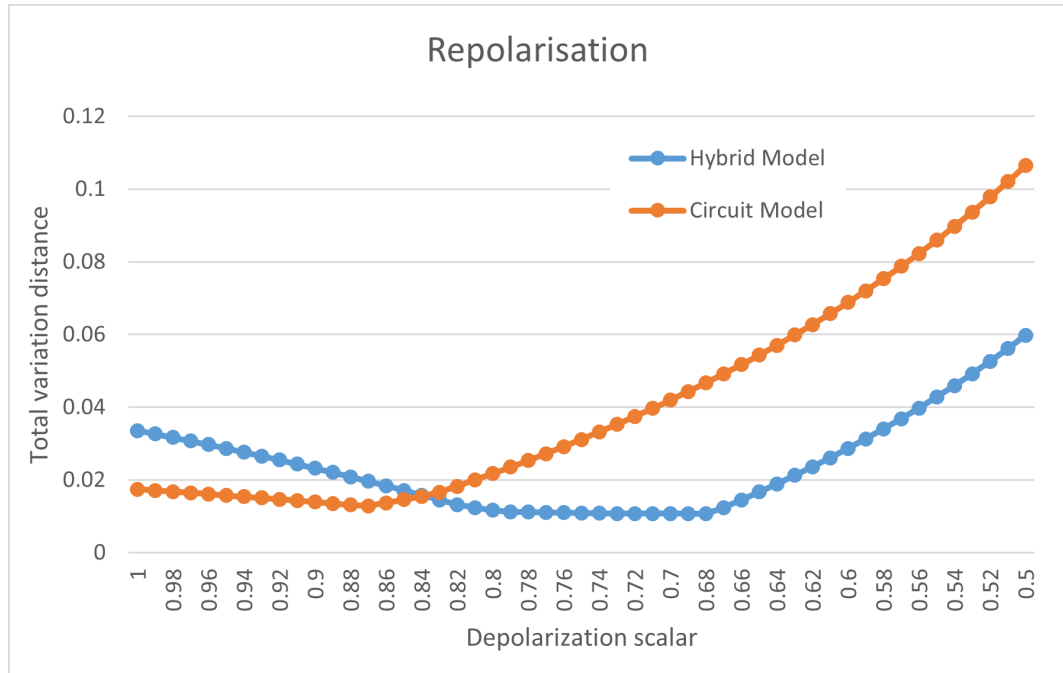


Figure 5.4: Readjusting the experimental outcomes for the circuit (orange) and hybrid models (blue) by using different estimates of the depolarisation parameter, k (x-axis) and then comparing the modified distributions against the benchmark to obtain a new total variation distance (y-axis).

imum it is possible to obtain good estimates of the depolarization parameter. For the circuit model, when $k = 0.87$ the total variation distance decreases to 0.013 (previously 0.017) and for the hybrid model, when $k = 0.68$ the total variation distance decreases to 0.011 (previously 0.034). The repolarised distributions are illustrated in figure 5.5. While this re-polarization approach does largely correct the discrepancy between the experimental output distribution of the hybrid model and that of the benchmark in this case, in order for this technique to be practically useful it is important to understand the factors which contribute to the higher depolarization parameter in the case of the hybrid model and to develop an alternative way of

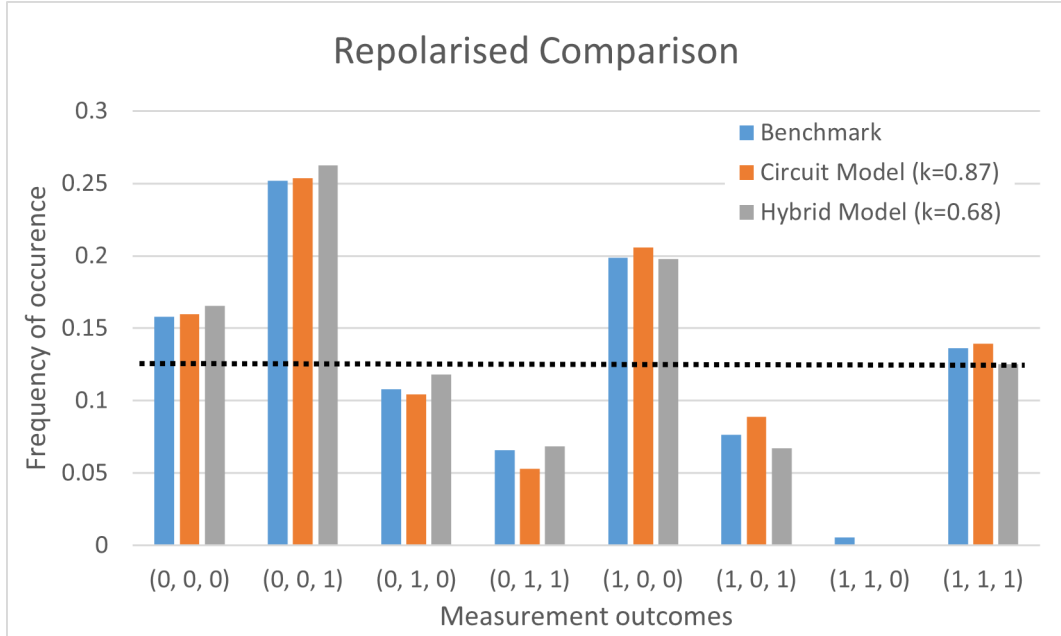


Figure 5.5: Comparison of the benchmark (blue), circuit model (orange) and hybrid model (grey) output distributions for 100000 shots after adjustments for depolarisation. A horizontal dashed line is placed at a frequency of 0.125 which indicates what a uniform distribution would look like for a 3-qubit circuit.

computing k in the absence of a benchmark distribution as that will no longer be available when considering industrial applications. Perhaps the increased noise is caused by the higher total number of 2-qubit gates in the hybrid model case and that can be verified in the future through further research.

5.6 Experiment summary

The experiment on the H1-1 device has produced many meaningful results. While demonstrations of the one-way model have been achieved in the past in optics, in this thesis I have realised the first implementation of the one-way model of a measurement pattern derived from a non-trivial phase polynomial circuit on an ion trap device. In addition to the general feasibility of the one-way model, this experiment has provided proof-of-concept for the techniques introduced in earlier chapters of this thesis.

Specifically, the circuit which was tested on the device was chosen to have dimensions that would result in a measurement pattern too large to fit within the 20-qubit capacity of the H1-1 device if converted to a single measurement pattern.

It was only by taking advantage of the hybrid model technique that an MBQC implementation of the circuit could be accomplished, by splitting the circuit into two measurement patterns rather than one. This solidifies the position of the hybrid model as a customization approach which allows quantum circuits to be converted to a tuneable version of the one-way model, taking into account device limitation.

Furthermore, by using algorithm 6 introduced in chapter 4 it was possible to achieve a level of depth reduction in the hybrid model that is very close to that of the standard one-way model. Consequently, we eliminated the requirement for 10 additional ancilla qubits for little to no compromise in terms of a reduction to entanglement depth, as indicated in table 5.2.

The experimental results are also cause for optimism when compared to the circuit model. While the unprocessed outcome distribution for the hybrid model exhibited a higher total variation distance than the circuit model, the opposite was true after using an error mitigation strategy to reduce the effects of depolarizing noise on the outputs. Further research is needed to explore this topic, but it may be possible to refine the strategy to obtain even better results from the one-way model which would be in line with the original motivations for considering MBQC in the first place.

Chapter 6

Conclusions and future work

While there remain some challenges to overcome before large-scale, fault-tolerant quantum computers become available, we have already entered an era where quantum devices with qubit capacities in the triple digits are on the horizon. These NISQ devices might not be sufficient for practical implementations of Shor's algorithm or Grover's algorithm but by carefully leveraging the available resources it may be possible to realise useful applications in quantum simulations of chemistry, biology and material science.

The circuit model and one-way model of quantum computation serve as two equivalent but distinct approaches to compiling quantum circuits on a given platform. While the former is focused on using a smaller number of physical qubits at the cost of a higher depth in the time-domain, the latter includes more ancilla qubits with the goal of minimizing the depth of measurement patterns.

As these two options are quite rigid and not necessarily conducive to the efficient use of the limited resources characteristic of NISQ devices, I have introduced a third approach - a hybrid model of quantum computation - which serves to bridge the spectrum spanned by the two extremes and which functions as a tuneable version of MBQC that can be modified to fit hardware requirements with higher precision.

The hybrid model was introduced and evaluated in chapter 3. It was compared to the circuit and one-way models, showing how most of the depth reduction achieved by the one-way model can be reached with only a fraction of the qubit overhead. However, I have also found that any form of MBQC - whether the origi-

nal one-way model or the hybrid model - is less compatible with routing strategies than the circuit model. I attribute this to the high-degree of graph states which necessitates that more SWAP gates are used to fit the graphs on restricted architectures.

To work around the routing incompatibility, I have introduced multiple algorithms which build upon one another and use local complementation and pivoting as tools to reduce the entanglement depth of measurement patterns. These algorithms were introduced in chapter 4. While this solution was motivated by the large depth overheads added during routing, reducing the entanglement depth of a measurement pattern is beneficial even in the fully connected setting.

Combined with the fact that these algorithms are computationally cheap to run on a classical machine, exhibiting low-polynomial scaling of, at most, $O(n^5)$ where n is the number of vertices in the graph, these algorithms can be widely useful for any implementation of the one-way model. This remains true even outside of routing or the hybrid model implementations discussed in this thesis.

In chapter 5, I have discussed the experimental realisation of the hybrid model and the local complementation algorithms on the Quantinuum H1-1 ion trap device. This is the first experimental realisation of the one-way model on an ion trap device, only made possible by the fact that the H1 series devices support mid-circuit measurements, reuse of qubits after the mid-circuit measurements and operations conditional on these measurements. Furthermore, the H1 series devices have a large enough capacity for physical qubits to allow non-trivial quantum circuits to be run after conversion into measurement patterns.

Additionally, this experiment showcased the primary advantage of the hybrid model as the sample circuit was too large to be run on the device in the form of a single measurement pattern, yet became implementable when separated into two subcircuits and executed as a sequence of the two resulting measurement patterns.

The unprocessed output of the hybrid model was subject to higher levels of depolarizing noise than the circuit model, likely due to the higher total number of 2-qubit gates and measurements involved, but it was possible to obtain better results (a

lower total variation distance) from the hybrid model after using an error mitigation strategy to reduce the effect of the depolarizing channel. This is a cause of optimism in line with the original motivation of using MBQC as it indicates that the one-way model might have the side effect of moving local depolarizing noise closer to global depolarizing noise, which is more easily removed using error mitigation techniques.

Between the depth reduction algorithms and the focus on platforms with all-to-all connectivity such as ion traps, we have provided two potential solutions to deal with the depth overheads added due to routing. It should be noted that recent work by Grzesiak et al. indicates that it may be possible to achieve even more efficient implementations of graph states on ion trap devices, by simultaneously applying multiple CZ gates acting on the same set of qubits, thus lowering entanglement depth even more [103]. Due to a lack of time, I have not had the opportunity to test how well the LC algorithms reduce the routing overhead on superconducting devices with low connectivity and this is something that I hope to test in the future.

While the algorithms introduced in chapter 4 prioritised entanglement depth reduction, they can also be changed to prioritise reducing the total number of 2-qubit gates, with very few minor modifications. In the future, I can test how this alternative strategy performs in practice and whether reducing the total number of 2-qubit gates helps improve the accuracy of the simulation even more than a lower entanglement depth.

When designing the hybrid model and LC algorithms, I used sample circuits of two main classes: Randomly-generated Clifford+T circuits which are a broad circuit class constructed with a universal gate-set and randomly-generated phase polynomial circuits which constitute a more specialized class of circuits, adjacent to those needed in VQE applications.

The specific circuit-generation processes used were discussed in section 2.8. A phase polynomial circuit from the same sample set was also chosen for the experimental implementation on the H1-1 device. In the future, I plan to perform a similar experiment on circuits taken directly from quantum simulation applications which should lead to even better insight as to the usefulness of the techniques discussed in

this thesis, when applied in practice.

Bibliography

- [1] Kissinger A, van de Wetering J. PyZX: Large Scale Automated Diagrammatic Reasoning. *Electronic Proceedings in Theoretical Computer Science*. 2020 May;318:229-41. ArXiv: 1904.04735. Available from: <http://arxiv.org/abs/1904.04735>.
- [2] Sivarajah S, Dilkes S, Cowtan A, Simmons W, Edgington A, Duncan R. $t|ket\rangle$: a retargetable compiler for NISQ devices. *Quantum Science and Technology*. 2020 Nov;6(1):014003. Publisher: IOP Publishing. Available from: <https://doi.org/10.1088/2058-9565/ab8e92>.
- [3] Manin Y. Mathematics as Metaphor: Selected Essays of Yuri I. Manin. In: *Philosophia Mathematica*. vol. 17; 2008. p. 122-3. ISSN: 0031-8019, 1744-6406 Issue: 1 Journal Abbreviation: *Philosophia Mathematica*. Available from: <https://academic.oup.com/philmat/article-lookup/doi/10.1093/philmat/nkn013>.
- [4] Benioff P. The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. *Journal of Statistical Physics*. 1980 May;22(5):563-91. Available from: <https://doi.org/10.1007/BF01011339>.
- [5] Feynman RP. Simulating Physics with Computers. *International Journal of Theoretical Physics*. 1982 Jun;21:467-88. Available from: <http://adsabs.harvard.edu/abs/1982IJTP...21..467F>.

- [6] Deutsch D, Jozsa R. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London Series A: Mathematical and Physical Sciences*. 1997 Jan;439(1907):553-8. Publisher: Royal Society. Available from: <https://royalsocietypublishing.org/doi/10.1098/rspa.1992.0167>.
- [7] Bernstein E, Vazirani U. Quantum Complexity Theory. *SIAM Journal on Computing*. 1997 Oct;26(5):1411-73. Publisher: Society for Industrial and Applied Mathematics. Available from: <https://epubs.siam.org/doi/10.1137/S0097539796300921>.
- [8] Simon DR. On the Power of Quantum Computation. *SIAM Journal on Computing*. 1997 Oct;26(5):1474-83. Publisher: Society for Industrial and Applied Mathematics. Available from: <https://epubs.siam.org/doi/10.1137/S0097539796298637>.
- [9] Shor PW. Algorithms for quantum computation: discrete logarithms and factoring. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*; 1994. p. 124-34.
- [10] Grover LK. A fast quantum mechanical algorithm for database search. *arXiv:quant-ph/9605043*. 1996 Nov. *ArXiv: quant-ph/9605043*. Available from: <http://arxiv.org/abs/quant-ph/9605043>.
- [11] Monroe C, Meekhof DM, King BE, Itano WM, Wineland DJ. Demonstration of a Fundamental Quantum Logic Gate. *Physical Review Letters*. 1995 Dec;75(25):4714-7. Available from: <https://link.aps.org/doi/10.1103/PhysRevLett.75.4714>.
- [12] DiVincenzo DP, IBM. The Physical Implementation of Quantum Computation. *Fortschritte der Physik*. 2000 Sep;48(9-11):771-83. *ArXiv:quant-ph/0002077*. Available from: <http://arxiv.org/abs/quant-ph/0002077>.

- [13] Kitaev AY. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*. 1997 Dec;52:1191-249. Available from: <http://adsabs.harvard.edu/abs/1997RuMaS..52.1191K>.
- [14] Nielsen MA, Chuang IL. *Quantum computation and quantum information*. 10th ed. Cambridge ; New York: Cambridge University Press; 2010.
- [15] Taylor N. *Laser: The Inventor, the Nobel Laureate, and the Thirty-Year Patent War*. 0th ed. Backinprint.com; 2007.
- [16] An FA, Ransford A, Schaffer A, Sletten LR, Gaebler J, Hostetter J, et al. High Fidelity State Preparation and Measurement of Ion Hyperfine Qubits with $I > \frac{1}{2}$. *Physical Review Letters*. 2022 Sep;129(13):130501. Publisher: American Physical Society. Available from: <https://link.aps.org/doi/10.1103/PhysRevLett.129.130501>.
- [17] Cohen-Tannoudji C, Diu B, Laloë F. *Quantum Mechanics, Volume 1: Basic Concepts, Tools, and Applications*. 2nd ed. Weinheim: Wiley-VCH; 2019.
- [18] Cirac JI, Zoller P. Quantum Computations with Cold Trapped Ions. *Physical Review Letters*. 1995 May;74(20):4091-4. Publisher: American Physical Society. Available from: <https://link.aps.org/doi/10.1103/PhysRevLett.74.4091>.
- [19] Blinov BB, Leibfried D, Monroe C, Wineland DJ. Quantum Computing with Trapped Ion Hyperfine Qubits. *Quantum Information Processing*. 2004 Oct;3(1):45-59. Available from: <https://doi.org/10.1007/s11128-004-9417-3>.
- [20] Chuang IL, Gershenfeld N, Kubinec M. Experimental Implementation of Fast Quantum Searching. *Physical Review Letters*. 1998 Apr;80(15):3408-11. Publisher: American Physical Society. Available from: <https://link.aps.org/doi/10.1103/PhysRevLett.80.3408>.

- [21] Knill E, Laflamme R, Milburn GJ. A scheme for efficient quantum computation with linear optics. *Nature*. 2001 Jan;409(6816):46-52. Number: 6816 Publisher: Nature Publishing Group. Available from: <https://www.nature.com/articles/35051009>.
- [22] Zhao Z, Chen YA, Zhang AN, Yang T, Briegel H, Pan JW. Experimental Demonstration of Five-photon Entanglement and Open-destination Teleportation. *Nature*. 2004 Jul;430(6995):54-8. ArXiv:quant-ph/0402096. Available from: <http://arxiv.org/abs/quant-ph/0402096>.
- [23] Accelerating Quantum Computing | Quantinuum;. Available from: <https://www.quantinuum.com/>.
- [24] Plantenberg JH, de Groot PC, Harmans CJPM, Mooij JE. Demonstration of controlled-NOT quantum gates on a pair of superconducting quantum bits. *Nature*. 2007 Jun;447(7146):836-9. Number: 7146 Publisher: Nature Publishing Group. Available from: <https://www.nature.com/articles/nature05896>.
- [25] O'Malley PJJ, Babbush R, Kivlichan ID, Romero J, McClean JR, Barends R, et al. Scalable Quantum Simulation of Molecular Energies. *Physical Review X*. 2016 Jul;6(3):031007. ArXiv:1512.06860 [physics, physics:quant-ph]. Available from: <http://arxiv.org/abs/1512.06860>.
- [26] Bauer MR. IBM Just Made a 17 Qubit Quantum Processor, Its Most Powerful One Yet; 2017. Available from: <https://www.vice.com/en/article/wnwk5w/ibm-17-qubit-quantum-processor-computer-google>.
- [27] IBM Raises the Bar with a 50-Qubit Quantum Computer;. Available from: <https://www.technologyreview.com/2017/11/10/147728/ibm-raises-the-bar-with-a-50-qubit-quantum-computer/>.

- [28] Google moves toward quantum supremacy with 72-qubit computer; 2018. Section: Quantum Physics. Available from: <https://www.sciencenews.org/article/google-moves-toward-quantum-supremacy-72-qubit-computer>.
- [29] Ion-based commercial quantum computer is a first; 2018. Available from: <https://physicsworld.com/a/ion-based-commercial-quantum-computer-is-a-first/>.
- [30] IonQ | Trapped Ion Quantum Computing;. Available from: <https://ionq.com/>.
- [31] Arute F, Arya K, Babbush R, Bacon D, Bardin JC, Barends R, et al. Quantum supremacy using a programmable superconducting processor. *Nature*. 2019 Oct;574(7779):505-10. Number: 7779 Publisher: Nature Publishing Group. Available from: <https://www.nature.com/articles/s41586-019-1666-5>.
- [32] Börner SD, Berke C, DiVincenzo DP, Trebst S, Altland A. Classical Chaos in Quantum Computers. *arXiv*; 2023. ArXiv:2304.14435 [cond-mat, physics:quant-ph]. Available from: <http://arxiv.org/abs/2304.14435>.
- [33] Raussendorf R, Briegel HJ. A One-Way Quantum Computer. *Physical Review Letters*. 2001 May;86(22):5188-91. Publisher: American Physical Society. Available from: <https://link.aps.org/doi/10.1103/PhysRevLett.86.5188>.
- [34] Danos V, Kashefi E, Panangaden P. The Measurement Calculus. *arXiv*:07041263 [quant-ph]. 2007 Apr. ArXiv: 0704.1263. Available from: <http://arxiv.org/abs/0704.1263>.
- [35] Abramsky S, Coecke B. A categorical semantics of quantum protocols. *arXiv*; 2007. ArXiv:quant-ph/0402130. Available from: <http://arxiv.org/abs/quant-ph/0402130>.

- [36] Park JL. The concept of transition in quantum mechanics. *Foundations of Physics*. 1970 Mar;1(1):23-33. Available from: <https://doi.org/10.1007/BF00708652>.
- [37] Wootters WK, Zurek WH. A single quantum cannot be cloned. *Nature*. 1982 Oct;299(5886):802-3. Number: 5886 Publisher: Nature Publishing Group. Available from: <https://www.nature.com/articles/299802a0>.
- [38] Dieks D. Communication by EPR devices. *Physics Letters A*. 1982 Nov;92(6):271-2. Available from: <https://www.sciencedirect.com/science/article/pii/0375960182900846>.
- [39] Coecke B. Quantum Picturalism. *Contemporary Physics*. 2010 Jan;51(1):59-83. ArXiv:0908.1787 [math-ph, physics:quant-ph]. Available from: <http://arxiv.org/abs/0908.1787>.
- [40] Baez J, Stay M. Physics, Topology, Logic and Computation: A Rosetta Stone. In: Coecke B, editor. *New Structures for Physics*. vol. 813. Berlin, Heidelberg: Springer Berlin Heidelberg; 2010. p. 95-172. Series Title: *Lecture Notes in Physics*. Available from: http://link.springer.com/10.1007/978-3-642-12821-9_2.
- [41] Coecke B, Pavlovic D. Quantum measurements without sums. arXiv; 2008. ArXiv:quant-ph/0608035. Available from: <http://arxiv.org/abs/quant-ph/0608035>.
- [42] Coecke B, Duncan R. Interacting Quantum Observables. In: Aceto L, Damgård I, Goldberg LA, Halldórsson MM, Ingólfssdóttir A, Walukiewicz I, editors. *Automata, Languages and Programming*. *Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer; 2008. p. 298-310.
- [43] Duncan R, Perdrix S. Graphs States and the necessity of Euler Decomposition. vol. 5635; 2009. ArXiv:0902.0500 [quant-ph]. Available from: <http://arxiv.org/abs/0902.0500>.

- [44] Backens M. The ZX-calculus is complete for stabilizer quantum mechanics. *New Journal of Physics*. 2014 Sep;16(9):093021. Available from: <https://iopscience.iop.org/article/10.1088/1367-2630/16/9/093021>.
- [45] Jeandel E, Perdrix S, Vilmart R. A Complete Axiomatisation of the ZX-Calculus for Clifford+T Quantum Mechanics. In: *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*. Oxford United Kingdom: ACM; 2018. p. 559-68. Available from: <https://dl.acm.org/doi/10.1145/3209108.3209131>.
- [46] Jeandel E, Perdrix S, Vilmart R. Diagrammatic Reasoning beyond Clifford+T Quantum Mechanics. *arXiv*; 2018. ArXiv:1801.10142 [quant-ph]. Available from: <http://arxiv.org/abs/1801.10142>.
- [47] Hadzihasanovic A, Ng KF, Wang Q. Two complete axiomatisations of pure-state qubit quantum computing. In: *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science. LICS '18*. New York, NY, USA: Association for Computing Machinery; 2018. p. 502-11. Available from: <https://doi.org/10.1145/3209108.3209128>.
- [48] Zeh HD. On the interpretation of measurement in quantum theory. *Foundations of Physics*. 1970 Mar;1(1):69-76. Available from: <https://doi.org/10.1007/BF00708656>.
- [49] Schlosshauer M. Decoherence, the measurement problem, and interpretations of quantum mechanics. *Reviews of Modern Physics*. 2005 Feb;76(4):1267-305. ArXiv: quant-ph/0312059. Available from: <http://arxiv.org/abs/quant-ph/0312059>.
- [50] Fuchs CA, Caves CM. Ensemble-Dependent Bounds for Accessible Information in Quantum Mechanics. *Physical Review Letters*. 1994 Dec;73(23):3047-50. Publisher: American Physical Society. Available

from: <https://link.aps.org/doi/10.1103/PhysRevLett.73.3047>.

- [51] Jozsa R. Fidelity for Mixed Quantum States. *Journal of Modern Optics*. 1994 Dec;41(12):2315-23. Publisher: Taylor & Francis eprint: <https://doi.org/10.1080/09500349414552171>. Available from: <https://doi.org/10.1080/09500349414552171>.
- [52] Peres A. Reversible logic and quantum computers. *Physical Review A*. 1985 Dec;32(6):3266-76. Publisher: American Physical Society. Available from: <https://link.aps.org/doi/10.1103/PhysRevA.32.3266>.
- [53] Shor PW. Scheme for reducing decoherence in quantum computer memory. *Physical Review A*. 1995 Oct;52:R2493-6. Available from: <http://adsabs.harvard.edu/abs/1995PhRvA..52.2493S>.
- [54] Devitt SJ, Nemoto K, Munro WJ. Quantum Error Correction for Beginners. *Reports on Progress in Physics*. 2013 Jul;76(7):076001. ArXiv:0905.2794 [quant-ph]. Available from: <http://arxiv.org/abs/0905.2794>.
- [55] Aharonov D, Ben-Or M. Fault-Tolerant Quantum Computation With Constant Error Rate. arXiv; 1999. ArXiv:quant-ph/9906129. Available from: <http://arxiv.org/abs/quant-ph/9906129>.
- [56] Knill E, Laflamme R, Zurek WH. Resilient Quantum Computation: Error Models and Thresholds. *Proceedings of the Royal Society of London Series A: Mathematical, Physical and Engineering Sciences*. 1998 Jan;454(1969):365-84. ArXiv:quant-ph/9702058. Available from: <http://arxiv.org/abs/quant-ph/9702058>.
- [57] Kitaev AY. Fault-tolerant quantum computation by anyons. *Annals of Physics*. 2003 Jan;303(1):2-30. ArXiv:quant-ph/9707021. Available from: <http://arxiv.org/abs/quant-ph/9707021>.

- [58] Neumann Jv. Probabilistic Logics and the Synthesis of Reliable Organisms From Unreliable Components. In: Probabilistic Logics and the Synthesis of Reliable Organisms From Unreliable Components. Princeton University Press; 2016. p. 43-98. Available from: <https://www.degruyter.com/document/doi/10.1515/9781400882618-003/html>.
- [59] Preskill J. Quantum Computing in the NISQ era and beyond. *Quantum*. 2018 Aug;2:79. ArXiv: 1801.00862. Available from: <http://arxiv.org/abs/1801.00862>.
- [60] Peruzzo A, McClean J, Shadbolt P, Yung MH, Zhou XQ, Love PJ, et al. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*. 2014 Jul;5(1):4213. Number: 1 Publisher: Nature Publishing Group. Available from: <https://www.nature.com/articles/ncomms5213>.
- [61] Wecker D, Hastings MB, Troyer M. Progress towards practical quantum variational algorithms. *Physical Review A*. 2015 Oct;92(4):042303. Publisher: American Physical Society. Available from: <https://link.aps.org/doi/10.1103/PhysRevA.92.042303>.
- [62] McClean JR, Romero J, Babbush R, Aspuru-Guzik A. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*. 2016 Feb;18(2):023023. Publisher: IOP Publishing. Available from: <https://dx.doi.org/10.1088/1367-2630/18/2/023023>.
- [63] Farhi E, Goldstone J, Gutmann S. A Quantum Approximate Optimization Algorithm. arXiv; 2014. ArXiv:1411.4028 [quant-ph]. Available from: <http://arxiv.org/abs/1411.4028>.
- [64] Lu D, Xu B, Xu N, Li Z, Chen H, Peng X, et al. Quantum chemistry simulation on quantum computers: theories and experiments. *Physical Chemistry Chemical Physics*. 2012 Jun;14(26):9411-20. Publisher: The Royal

- Society of Chemistry. Available from: <https://pubs.rsc.org/en/content/articlelanding/2012/cp/c2cp23700h>.
- [65] BÖHM C. On encoding logical-mathematical formulas using the machine itself during program conception; 1954. Available from: <https://www.itu.dk/~sestoft/boehmthesis/boehm.pdf>.
- [66] Qiskit contributors. Qiskit: An Open-source Framework for Quantum Computing; 2023.
- [67] Raussendorf R, Browne DE, Briegel HJ. Measurement-based quantum computation with cluster states. *Physical Review A*. 2003 Aug;68(2):022312. ArXiv: quant-ph/0301052. Available from: <http://arxiv.org/abs/quant-ph/0301052>.
- [68] de Fraysseix H. Local complementation and interlacement graphs. *Discrete Mathematics*. 1981 Jan;33(1):29-35. Available from: <https://www.sciencedirect.com/science/article/pii/0012365X81902557>.
- [69] Bouchet A. Graphic presentations of isotropic systems. *Journal of Combinatorial Theory, Series B*. 1988 Aug;45(1):58-76. Available from: <https://www.sciencedirect.com/science/article/pii/009589568890055X>.
- [70] Bouchet A. Recognizing locally equivalent graphs. *Discrete Mathematics*. 1993 Apr;114(1):75-86. Available from: <https://www.sciencedirect.com/science/article/pii/0012365X9390357Y>.
- [71] Danos V, Kashefi E. Determinism in the one-way model. *Physical Review A*. 2006 Nov;74(5):052310. ArXiv: quant-ph/0506062. Available from: <http://arxiv.org/abs/quant-ph/0506062>.

- [72] Browne DE, Kashefi E, Mhalla M, Perdrix S. Generalized Flow and Determinism in Measurement-based Quantum Computation. *New Journal of Physics*. 2007 Aug;9(8):250-0. ArXiv: quant-ph/0702212. Available from: <http://arxiv.org/abs/quant-ph/0702212>.
- [73] Vovrosh J, Khosla KE, Greenaway S, Self C, Kim M, Knolle J. Simple Mitigation of Global Depolarizing Errors in Quantum Simulations. *Physical Review E*. 2021 Sep;104(3):035309. ArXiv:2101.01690 [cond-mat, physics:quant-ph]. Available from: <http://arxiv.org/abs/2101.01690>.
- [74] Duncan R, Kissinger A, Perdrix S, van de Wetering J. Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus. *Quantum*. 2020 Jun;4:279. ArXiv: 1902.03178. Available from: <http://arxiv.org/abs/1902.03178>.
- [75] Amy M, Maslov D, Mosca M. Polynomial-Time T-Depth Optimization of Clifford+T Circuits Via Matroid Partitioning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2014 Oct;33(10):1476-89. Conference Name: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.
- [76] de Griend AMv, Duncan R. Architecture-aware synthesis of phase polynomials for NISQ devices. arXiv:200406052 [quant-ph]. 2020 Apr. ArXiv: 2004.06052. Available from: <http://arxiv.org/abs/2004.06052>.
- [77] Schumacher B. Quantum coding. *Physical Review A*. 1995 Apr;51(4):2738-47. Publisher: American Physical Society. Available from: <https://link.aps.org/doi/10.1103/PhysRevA.51.2738>.
- [78] Dirac PaM. A new notation for quantum mechanics. *Mathematical Proceedings of the Cambridge Philosophical Society*. 1939

- Jul;35(3):416-8. Publisher: Cambridge University Press. Available from: <https://www.cambridge.org/core/journals/mathematical-proceedings-of-the-cambridge-philosophical-society/article/abs/new-notation-for-quantum-mechanics/4631DB9213D680D6332BA11799D76AFB>.
- [79] Barenco A, Bennett CH, Cleve R, DiVincenzo DP, Margolus N, Shor P, et al. Elementary gates for quantum computation. *Physical Review A*. 1995 Nov;52(5):3457-67. ArXiv:quant-ph/9503016. Available from: <http://arxiv.org/abs/quant-ph/9503016>.
- [80] Gottesman D. Stabilizer Codes and Quantum Error Correction. arXiv:quant-ph/9705052. 1997 May. ArXiv: quant-ph/9705052. Available from: <http://arxiv.org/abs/quant-ph/9705052>.
- [81] Hoyer P, Spalek R. Quantum Circuits with Unbounded Fan-out. *Theory of Computing*. 2005;1(1):81-103. ArXiv: quant-ph/0208043. Available from: <http://arxiv.org/abs/quant-ph/0208043>.
- [82] IBM Quantum;. Available from: <https://quantum-computing.ibm.com/>.
- [83] Høyer P, Mhalla M, Perdrix S. Resources required for preparing graph states. In: *Algorithms and Computation: 17th International Symposium, ISAAC 2006, Kolkata, India, December 18-20, 2006. Proceedings 17*. Springer; 2006. p. 638-49.
- [84] Hein M, Eisert J, Briegel HJ. Multi-party entanglement in graph states. *Physical Review A*. 2004 Jun;69(6):062311. ArXiv:quant-ph/0307130. Available from: <http://arxiv.org/abs/quant-ph/0307130>.
- [85] Jorrand P, Perdrix S. Unifying Quantum Computation with Projective Measurements only and One-Way Quantum Computation; 2005. p. 44. ArXiv:quant-ph/0404125. Available from: <http://arxiv.org/abs/quant-ph/0404125>.

- [86] Williamson EABSG. Lists, decisions and graphs. S. Gill Williamson;. Available from: https://books.google.co.uk/books?id=vaXv_yhefG8C.
- [87] Diestel R. Graph Theory. vol. 173 of Graduate Texts in Mathematics. Berlin, Heidelberg: Springer; 2017. Available from: <http://link.springer.com/10.1007/978-3-662-53622-3>.
- [88] Hein M, Dür W, Eisert J, Raussendorf R, Nest MVd, Briegel HJ. Entanglement in Graph States and its Applications. arXiv; 2006. ArXiv:quant-ph/0602096. Available from: <http://arxiv.org/abs/quant-ph/0602096>.
- [89] Nest MVd, Dehaene J, De Moor B. Graphical description of the action of local Clifford transformations on graph states. Physical Review A. 2004 Feb;69(2):022316. ArXiv:quant-ph/0308151. Available from: <http://arxiv.org/abs/quant-ph/0308151>.
- [90] van de Wetering J. ZX-calculus for the working quantum computer scientist. arXiv; 2020. ArXiv:2012.13966 [quant-ph]. Available from: <http://arxiv.org/abs/2012.13966>.
- [91] Ng KF, Wang Q. A universal completion of the ZX-calculus. arXiv; 2017. arXiv:1706.09877. ArXiv:1706.09877 [quant-ph] type: article. Available from: <http://arxiv.org/abs/1706.09877>.
- [92] Backens M, Miller-Bakewell H, de Felice G, Lobski L, van de Wetering J. There and back again: A circuit extraction tale. arXiv:200301664 [quant-ph]. 2020 Mar. ArXiv: 2003.01664. Available from: <http://arxiv.org/abs/2003.01664>.
- [93] McElvanney T, Backens M. Complete flow-preserving rewrite rules for MBQC patterns with Pauli measurements. arXiv; 2022. arXiv:2205.02009. ArXiv:2205.02009 [quant-ph] type: article. Available from: <http://arxiv.org/abs/2205.02009>.

- [94] Mhalla M, Perdrix S. Finding Optimal Flows Efficiently. In: Aceto L, Damgård I, Goldberg LA, Halldórsson MM, Ingólfssdóttir A, Walukiewicz I, editors. Automata, Languages and Programming. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer; 2008. p. 857-68.
- [95] Simmons W. Relating Measurement Patterns to Circuits via Pauli Flow. *Electronic Proceedings in Theoretical Computer Science*. 2021 Sep;343:50-101. Available from: <http://arxiv.org/abs/2109.05654v1>.
- [96] Dahlberg A, Helsen J, Wehner S. Counting single-qubit Clifford equivalent graph states is #P-complete. *Journal of Mathematical Physics*. 2020 Feb;61(2):022202. Publisher: American Institute of Physics. Available from: <https://aip.scitation.org/doi/full/10.1063/1.5120591>.
- [97] Adcock JC, Morley-Short S, Dahlberg A, Silverstone JW. Mapping graph state orbits under local complementation. *Quantum*. 2020 Aug;4:305. ArXiv:1910.03969 [math-ph, physics:physics, physics:quant-ph]. Available from: <http://arxiv.org/abs/1910.03969>.
- [98] Gasparoni S, Pan JW, Walther P, Rudolph T, Zeilinger A. Realization of a Photonic Controlled-NOT Gate Sufficient for Quantum Computation. *Physical Review Letters*. 2004 Jul;93(2):020504. Publisher: American Physical Society. Available from: <https://link.aps.org/doi/10.1103/PhysRevLett.93.020504>.
- [99] Zhao Z, Zhang AN, Chen YA, Zhang H, Du JF, Yang T, et al. Experimental demonstration of a nondestructive controlled-NOT quantum gate for two independent photon qubits. *Physical review letters*. 2005;94(3):030501. Publisher: APS.
- [100] Vigliar C, Paesani S, Ding Y, Adcock JC, Wang J, Morley-Short S, et al. Error-protected qubits in a silicon photonic chip. *Nature Physics*. 2021 Oct;17(10):1137-43. Number: 10 Publisher: Nature Publishing

- Group. Available from: <https://www.nature.com/articles/s41567-021-01333-w>.
- [101] Pino JM, Dreiling JM, Figgatt C, Gaebler JP, Moses SA, Allman MS, et al. Demonstration of the trapped-ion quantum-CCD computer architecture. *Nature*. 2021 Apr;592(7853):209-13. ArXiv:2003.01293 [quant-ph]. Available from: <http://arxiv.org/abs/2003.01293>.
- [102] Chatterjee S. Wayback Machine; 2008. Available from: <https://web.archive.org/web/20080708205758/http://www.stat.berkeley.edu/%7Esourav/Lecture2.pdf>.
- [103] Grzesiak N, Blümel R, Beck K, Wright K, Chaplin V, Amini JM, et al. Efficient Arbitrary Simultaneously Entangling Gates on a trapped-ion quantum computer. *Nature Communications*. 2020 Jun;11(1):2963. ArXiv:1905.09294 [quant-ph]. Available from: <http://arxiv.org/abs/1905.09294>.