

**Imperial College  
London**

**Event Generation on Quantum  
Computers**

Simon Jonathan Williams

Department of Physics, Imperial College London

A thesis submitted to Imperial College London  
for the degree of Doctor of Philosophy



# Abstract

The synthesis of high quality simulated data from event generators is essential in the search for new physics at collider experiments. Modern event generator algorithms use Monte Carlo processes to simulate the evolution of an event from the collision of high energy particles to the formation of long-lived particles. One of the major building blocks of the event generation process is the QCD parton shower. However, despite being a key aspect of modern event generation, the core algorithms which simulate the showering process have remained unchanged since the 1980s, and will become a limiting factor as we move to an era of higher energy and higher luminosity experiments.

With the rapid development of quantum computation, dedicated algorithms are required which exploit the potential that quantum computing provides to address problems in high energy physics. In this thesis, we present three novel quantum algorithms for the simulation of a QCD parton shower. The first algorithm provides a proof-of-principle, classical Monte Carlo inspired approach with the ability to simulate two shower steps of a collinear QCD model. By exploiting the compact circuit architecture of the quantum walk, one can drastically reduce the quantum resources required to simulate a shower step. The second algorithm shows that, in this framework, the quantum parton shower can be extended to simulate realistic shower depths whilst using fewer quantum resources. Finally, the third algorithm utilises a discrete QCD approach to parton showering to include kinematics in the shower, simulating a dipole cascade. In this construction, the algorithm has achieved the first data comparison between

synthetic data produced using a Noisy Intermediate-Scale Quantum (NISQ) device, and “real-life” archival collider data from the Large Electron Positron collider. The three algorithms represent the development of quantum algorithms for the simulation of parton showers, acting as a first step towards a fully quantum simulation of a high energy collision event.

# Declaration

I, the author, hereby declare the work contained in this thesis to be my own, except where otherwise stated. The research has been carried out at Imperial College, London in collaboration with members of the Institute for Particle Physics Phenomenology at Durham University, and the Department of Physics and Astronomy, University College London. Parts of this thesis are based on published work:

- [1] K. Beperi, S. Malik, M. Spannowsky and S. Williams, “Towards a quantum computing algorithm for helicity amplitudes and parton showers,” [Physical Review D \*\*103\*\* \(2021\) no.7, 076020](#).
- [2] K. Beperi, S. Malik, M. Spannowsky and S. Williams, “Quantum walk approach to simulating parton showers,” [Physical Review D \*\*106\*\* \(2022\) no.5, 056002](#).
- [3] G. Gustafson, S. Prestel, M. Spannowsky and S. Williams, “Collider events on a quantum computer,” [Journal of High Energy Physics \*\*11\*\* \(2022\), 035](#).

Specifically, Chapter 4 is based on Reference [1], Chapter 5 is based on Reference [2] and Chapter 6 is based on Reference [3].

Simon Jonathan Williams

# Copyright

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC). Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose. When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes. Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

# Acknowledgements

Firstly, I would like to thank the High Energy Physics group at Imperial College London for providing an inspiring and exciting environment for me to complete my doctoral studies. Furthermore, I would especially like to thank the group for allowing me to carry out my non-conformist, and quite different, research from within the group. Not only that, I would like to thank everyone for encouraging me so much to pursue this new and exciting field of quantum computation.

Throughout my Ph.D. I have had the privilege to meet and work with many people who have helped to shape my academic experience, and I would like to highlight a particular few who have made an immeasurable impact on my experience and have helped me to thrive in an academic environment. Firstly, I would like to thank my supervisor Sarah Malik from University College London for first introducing me to the exciting field of quantum computation and to many brilliant researchers who are working in this accelerating field. I would like to thank Michael Spannowsky from the Institute for Particle Physics Phenomenology at the Durham University for being so supportive throughout my Ph.D. and for directing me towards what, in my opinion, have been very interesting projects in collider phenomenology and theoretical particle physics. Finally, I would like to thank Gavin Davies from Imperial College London for co-supervising me and for providing invaluable guidance during my studies. Without the support of all of you, I do not think that I would have been able to achieve the work that is presented in this thesis. I am very grateful for the opportunity to have worked closely with all of you, and I hope that I will continue to do so in the future.

I would like to thank the whole postgraduate cohort of the High Energy Physics group at Imperial College London for allowing me and Timothy Marley the pleasure of representing them at a group and department level for over two years. Thank you also goes to Timothy Marley for being so patient with me as a fellow postgraduate representative, always taking my suggestion of curling as a postgraduate social seriously! Being so integrated into the social aspect of the group has been a highlight of my experience at Imperial, and I have memories that I will never forget! I wish every one of the Ph.D. students the best of successes in the future, and I hope that many of you will stay in touch (and perhaps even move into the quantum computing field!).

Funding is of course of paramount importance in this field, and I would like to thank the Royal Society for funding the majority of my Ph.D. I would also like to thank the High Energy Physics group at Imperial College London for providing me with interim funding in the middle of my third year, allowing me to pursue a project which ultimately lead to the first data description of “real-life” archival collider data using a quantum computer [3]. I would like to acknowledge the use of IBM Quantum services for the work presented in this thesis, and to advanced services provided by the IBM Quantum Researchers Program. The views expressed are entirely the views of myself, and do not reflect the official policy or position of IBM or the IBM Quantum team. I would like to specifically thank the IBM Support Centre for their excellent service and expert help with running on the `ibm_cloud`.

As with any large body of work, there have been many typographical and general errors in each of the drafts. I would therefore like to thank everyone who proof read all or parts of my thesis, namely Gavin Davies, Michael Spannowsky, Stefan Prestel, Elliott Leader, Sarah Malik, my father Darren Williams and my partner Jessica Portwood. Your feedback has been invaluable, and has helped me to produce this thesis to the quality that it deserves.

Finally I would like to thank my family for their love and endless support during my studies. Specifically, I would like to thank my parents, Amanda and Darren, for supporting me both



emotionally and financially to complete my eight years at university. Without you I would not have been able to achieve any of the work presented in this thesis, and I would not be in the position I am, or the person I am today. I want to thank you from the bottom of my heart, and I love you very much.

*And now for something completely different...*



*“Nature isn’t classical, dammit, and if you want to make a simulation of nature, you’d better make it quantum mechanical, and by golly it’s a wonderful problem, because it doesn’t look so easy.”*

- Richard P. Feynman, 1981 [\[4\]](#)

# Preface

The focus of this thesis outlines the development of parton shower algorithms which can be simulated using a quantum device. The shower algorithms have progressed from a simple, classically inspired, quantum Monte Carlo parton shower with the ability to simulate a simplified QCD model, to a sophisticated dipole shower, implemented using an elegant method to discretise QCD. The thesis takes a phenomenological approach to simulating the evolution of a particle cascade, as a proof of principle and the first step in creating a fully quantum event generator algorithm in the future.

In addition to the work presented in this thesis, the author would like to highlight previous work which has contributed to the development of the parton shower algorithms. Reference [1] presents a quantum computing approach to calculating the hard process by exploiting the remarkable equivalence between a helicity spinor  $|p\rangle^{\dot{a}}$ ,

$$|p\rangle^{\dot{a}} = \sqrt{2E} \begin{pmatrix} \cos \frac{\theta}{2} \\ e^{i\phi} \sin \frac{\theta}{2} \end{pmatrix},$$

representing a massless state with momentum  $p^\mu$  and energy  $E$ , and a qubit

$$|\psi\rangle = \begin{pmatrix} \cos \frac{\theta}{2} \\ e^{i\phi} \sin \frac{\theta}{2} \end{pmatrix}.$$

parameterised by  $\theta$  and  $\phi^*$ . The two objects are identical, modulo an overall normalisation factor  $\sqrt{2E}$ . Furthermore, helicity amplitude calculations proceed in the same way as a quantum algorithm: unitary operations applied to a set of initialised states, culminating in a measurement which projects information onto a classical basis. Therefore, as a faithful representation of a helicity spinor, the qubit model is a natural platform to calculate helicity amplitudes. In Reference [1], this has been demonstrated by exploring two simple examples: a  $1 \rightarrow 2$  process corresponding to a  $gq\bar{q}$  process, and a  $2 \rightarrow 2$  scattering process, corresponding to  $q\bar{q} \rightarrow q\bar{q}$ . The two examples were shown to have good agreement with classically calculated amplitudes, and a discussion about extending the approach to a  $2 \rightarrow n$  process is given in Reference [1].

---

\*This form is described fully in Chapter 3.

# Contents

<b>1</b>	<b>Introduction</b>	<b>18</b>
<b>2</b>	<b>Theory and Motivations</b>	<b>21</b>
2.1	Gauge Theory . . . . .	24
2.1.1	Abelian Gauge Symmetries . . . . .	24
2.1.2	Non-Abelian Gauge Symmetries . . . . .	30
2.2	The Standard Model of Particle Physics . . . . .	35
2.2.1	The Gauge Sector . . . . .	38
2.2.2	The Fermionic Sector . . . . .	39
2.2.3	The Higgs Sector . . . . .	41
2.2.4	The Yukawa Sector . . . . .	47
2.3	The Parton Shower . . . . .	51
2.3.1	Collinear Factorisation . . . . .	51
2.3.2	Monte Carlo Method . . . . .	56
2.3.3	The Dipole Shower . . . . .	58
<b>3</b>	<b>Introduction to Quantum Computing</b>	<b>61</b>
3.1	Quantum Computation . . . . .	62
3.1.1	The Qubit . . . . .	62

---

3.1.2	Multiple Qubits . . . . .	64
3.1.3	Quantum Algorithms and Advantage . . . . .	66
3.2	The Quantum Walk . . . . .	67
3.2.1	Discrete-Time Coined Quantum Walk . . . . .	68
3.2.2	Quantum Walks with Memory . . . . .	77
3.2.3	Quantum Walks for Simulation . . . . .	78
<b>4</b>	<b>Quantum Monte Carlo Parton Shower</b>	<b>79</b>
4.1	Collinear Parton Shower on a Quantum Device . . . . .	80
4.2	Quantum Circuit Implementation . . . . .	82
4.2.1	Counting Operation . . . . .	84
4.2.2	Emission Operation . . . . .	86
4.2.3	History Operation . . . . .	88
4.2.4	Resetting the Emission and Counting Registers . . . . .	91
4.2.5	Update Operation . . . . .	91
4.3	Results . . . . .	94
<b>5</b>	<b>Quantum Walk Approach to Shower Algorithms</b>	<b>97</b>
5.1	The Parton Shower as a Quantum Walk . . . . .	98
5.2	Scalar Quantum Walk Shower . . . . .	99
5.2.1	Coin Operation . . . . .	102
5.2.2	Shift Operation . . . . .	103
5.2.3	Memory Operation . . . . .	105
5.3	Quantum Walk Parton Shower . . . . .	106
5.3.1	Coin Operation . . . . .	107
5.3.2	Shift Operation . . . . .	110
5.3.3	Memory Operation . . . . .	113

---

5.4	Results . . . . .	114
<b>6</b>	<b>Discrete QCD Parton Shower</b>	<b>117</b>
6.1	Discretising QCD . . . . .	118
6.1.1	Generating Scattering Events from Groves . . . . .	125
6.2	Quantum Circuit Implementation . . . . .	126
6.2.1	Coin operation . . . . .	128
6.2.2	Shift Operation . . . . .	129
6.2.3	Memory Operation . . . . .	131
6.3	Generating Collider Events on a Quantum Computer . . . . .	131
6.3.1	Tailoring the Quantum Circuit . . . . .	133
6.3.2	Generating Grove Structures on a Quantum Device . . . . .	134
6.3.3	Event Generation using the Quantum Device . . . . .	137
6.3.4	Tuning Simulation Parameters . . . . .	140
6.4	Results and Overview . . . . .	142
<b>7</b>	<b>Quantum Advantage and the Future of Quantum Parton Showers</b>	<b>145</b>
7.1	Speedup via Quantum Walks . . . . .	146
7.1.1	Quantum Walk Parton Shower . . . . .	147
7.2	Redundancy of Quantum Error Correction . . . . .	148
7.3	Towards a Fully Quantum Event Generator . . . . .	149
<b>8</b>	<b>Conclusions and Outlook</b>	<b>151</b>
<b>A</b>	<b>Quantum Circuit Diagrams</b>	<b>170</b>
A.1	Glossary of Quantum Gate Operations . . . . .	171



---

<b>B Quantum Measurement</b>	<b>174</b>
B.1 Computational Basis . . . . .	175
<b>C Quantum Errors</b>	<b>178</b>

# Chapter 1

## Introduction

Particle physics is the study of the fundamental building blocks of the Universe, and has culminated in the formulation of the Standard Model of Particle Physics, our best description of the elementary constituents of Nature and their interactions. Despite its success, it is known that the Standard Model is an incomplete theory of Nature, and is missing vital building blocks of the universe. For that reason, there is a world-wide research initiative to search for new physics beyond the Standard Model and to build a description of the Universe, both in the experimental and theoretical communities. This is not an easy task, and experimentally has seen some of the largest experiments ever built, reaching unprecedented energies and producing unprecedented amounts of data.

With the High-Luminosity Large Hadron Collider around the corner, and new colliders such as the Future Circular Collider on the horizon, particle physics is on the brink of a new computational frontier. To deal with the sheer amount of data produced by these machines, and to accurately generate simulations of high energy processes which occur inside the collider experiments, new computational techniques must be explored. With its rapid and continuous development, quantum computation offers a paradigm shift in information science, and promises to revolutionise modern computational techniques. Particle physics is set to benefit not only

from any speedup provided by quantum computing, but also, and perhaps more fundamentally, from the devices' ability to form highly entangled quantum systems, performing calculations in a regime never before accessible. Furthermore, all processes in high energy physics are inherently quantum mechanical, thus particle physics is an excellent test bed for the quantum algorithmic design.

Already, there has been a quickly developing research effort which is beginning to use quantum devices for applications in particle physics, ranging from the simulation of quantum field theories [5–9], and collision events [1–3,10,11], to event classification and analysis [12–16]. So far these algorithms have been proof-of-principle, demonstrating that practical quantum algorithms are within reach. The aim is to test the limits of the rapidly progressing Noisy Intermediate Scale Quantum (NISQ) devices. Although quantum supremacy is currently over the horizon, now is the time to begin to understand the potential that these devices provide, and learn how best to deploy the resources on offer to tackle troubling problems in fundamental physics.

In this thesis, we take a phenomenological approach, presenting novel quantum computing algorithms to combat the growing challenge of producing synthetic data for high energy collisions. One of the major building blocks of data synthesis in event generator programs is the parton shower. Here we concentrate on quantum computing approaches for the simulation of parton showers in high energy hadronic colliders. Chapters 2 and 3 provide the theoretical groundings in both particle physics and quantum computation needed for the reading of this thesis. Chapter 4 outlines the implementation of a simple, two-step collinear parton shower using a Monte Carlo inspired quantum algorithm. A quantum walk framework is used in Chapter 5 to dramatically increase the efficiency of the quantum parton shower, allowing for 31 shower steps to be simulated on a quantum simulator. These simple shower algorithms are entirely collinear and do not include kinematics. Chapter 6 shows that kinematics can be included in the shower by using a discrete QCD model, leading to the first data comparison between synthetic scattering data produced by

a quantum device and “real-life” archival collider data from experiments on the Large Electron Positron collider. Finally, Chapter 7 discusses the future of quantum parton shower algorithms and outlines steps towards a fully quantum simulation of the event generation process.

## Chapter 2

# Theory and Motivations

For over half a century the Standard Model of particle physics has been subject to rigorous experimental investigation, and so far has continued to be robust against scrutiny, remaining consistent with all experimental observations. Augmented by the neutrino masses, the Standard Model provides a remarkably successful description of the elementary particles and their interactions. However, in spite of its success, the Standard Model is not a complete theory of Nature and has several important missing features. The model does not have a mechanism for gravity and it is not compatible with the classical field theory of General Relativity. Nor does it provide a candidate for Dark Matter or explain the apparent presence of Dark Energy, both of which have been experimentally inferred from the rotation curves of galaxies and the Cosmic Microwave Background. The Standard Model does not explain the excess of matter over anti-matter in the Universe, and it does not explain why Quantum Chromodynamics preserves charge-parity (CP), the so-called *strong-CP problem*.

It is therefore expected that the Standard Model is a low-energy effective field theory, and that new physics will exist at a higher energy scale. One can postulate that new physics must be present at the reduced Planck scale,  $M_P = 2.4 \times 10^{18}$  GeV, where quantum gravitational effects will equal the strength of gauge interactions. However, the Standard Model Higgs boson has a

troubling dependence on new physics at large energy scales, which is the root of the so-called *gauge hierarchy problem*. Extending the Standard Model is therefore a major research effort, and many potential extensions have been suggested. References [17, 18] provide a review of potential Beyond the Standard Model physics. A popular extension of the Standard Model is *supersymmetry*, which predicts each particle in the Standard Model has a *superpartner* particle. This method provides an efficient mechanism to solve the hierarchy problem and has been the subject of numerous experimental searches. Reference [19] provides a detailed review of the current status of supersymmetry.

Modern collider experiments, such as the Large Hadron Collider (LHC) at CERN, are pushing the energy frontier in search for physics beyond the Standard Model. In order to examine physics processes in high energy experiments, accurate modelling of particle collisions and simulations of the subsequent detector response are needed. These are then used in the analysis of data from the detector to extract information about the underlying physics process. Consequently, accurate generation of synthetic data is crucial to modern particle physics, and efficient event generator programs are being actively developed. Typically these event generators are based on Monte Carlo approaches, such as PYTHIA [20], HERWIG [21] and SHERPA [22].

The LHC collides protons at a centre of mass energy up to  $E_{\text{CM}} = 14$  TeV, producing hundreds of subsequent particles as a result of the evolution from the proton-proton collision to the formation of long lived hadrons, leptons and photons, which are captured by the detector. The corresponding theoretical description can therefore be highly complex. This process can be broken down into several main parts. A proton consists of many partons, each of which carries a fraction of the proton's total energy. The momentum distribution of partons within the proton is conventionally described by the *Parton Distribution Function* (PDF). Colliding protons can result in two partons interacting with a large momentum transfer in the *hard process*. At high energy, this process has the ability to probe new physics. Colour-charged particles produced

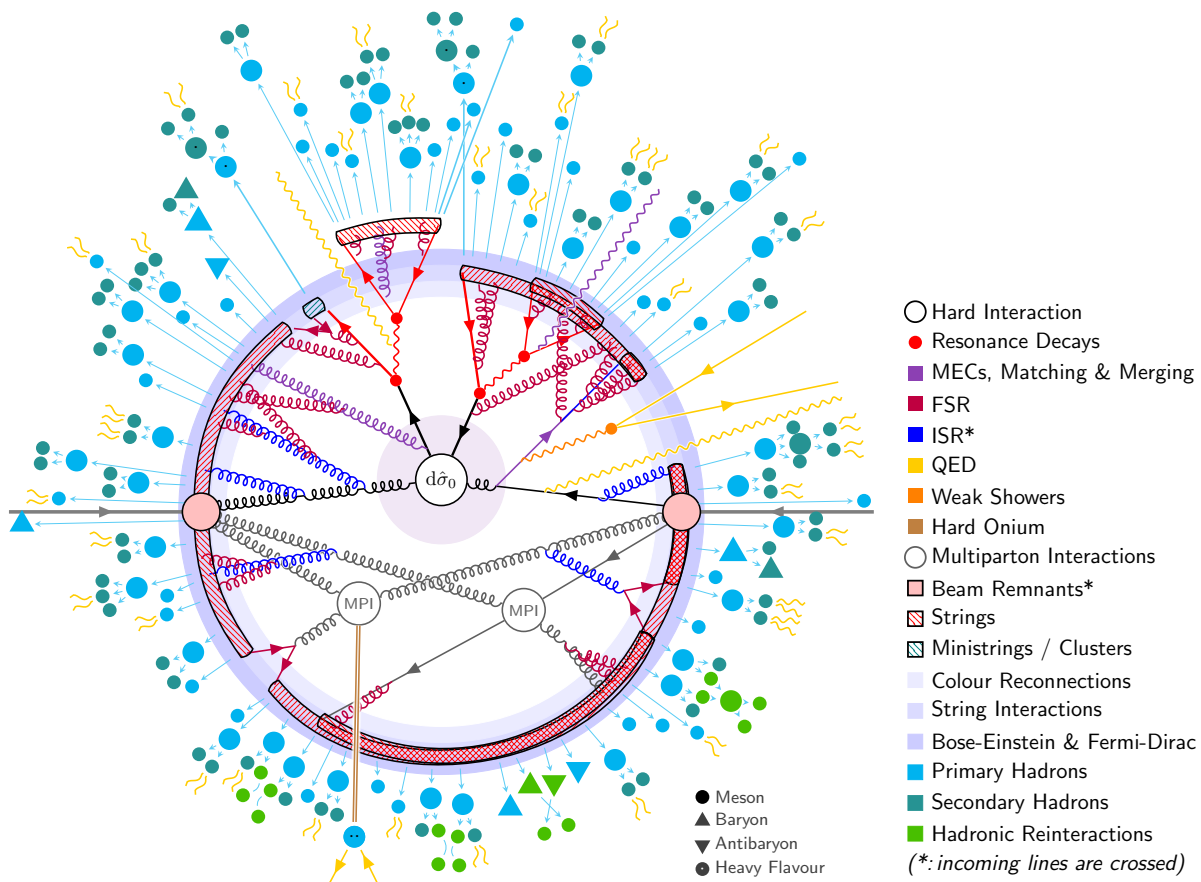


Figure 2.1: Schematic of a  $pp \rightarrow t\bar{t}$  event from Reference [20]. The schematic has been simplified from a real PYTHIA event, such that 1) shower branchings and final-state hadrons are less numerous, 2) the effect of recoil is not correctly illustrated, 3) weak decays of light hadrons are not included, and 4) incoming momenta are depicted as crossed ( $p \rightarrow -p$ ).

as a result of the hard process will likely radiate other colour-charged particles in the so-called *parton shower*. The parton shower process evolves the system from the high energy process of the hard interaction, down to the relatively low energy *hadronisation scale*,  $\mathcal{O}(\Lambda_{\text{QCD}})$ . During hadronisation, non-perturbative processes rearrange partons into colour-neutral hadrons. These hadrons are then measured in the detector. Figure 2.1 shows a schematic of a PYTHIA event for a  $pp \rightarrow t\bar{t}$  process, highlighting the complexity of event generation.

Due to the factorisation theorem [23–25], which separates the short and long-distance physics in QCD, the parton shower is an entirely perturbative process. Therefore, parton shower cal-

culations can be computationally expensive. Nevertheless, parton showers are a major building block of event generators and therefore there is much appetite for efficient parton shower algorithms. It is possible that quantum computers can provide an advantage for the calculation of parton showers, and this is discussed in detail in Chapters 4, 5 and 6. In this chapter, we lay the theoretical groundwork for the rest of the thesis. In Section 2.1 we discuss Abelian and non-Abelian gauge theories as a preliminary for understanding the Standard Model, which is reviewed in Section 2.2, leading to a review of the parton shower in Section 2.3. A detailed review of QFT and the Standard Model is available in References [26–28], and References [29–33] provide an overview of modern event generation methods and parton showers.

## 2.1 Gauge Theory

The theoretical understanding of modern particle physics is founded on gauge field theories: a type of field theory for which the Lagrangian is invariant under a set of local transformations described by a Lie group. In fact, these invariances, often called *gauge symmetries*, have fundamental implications on the form of the Lagrangian. In this section we briefly discuss the two types of gauge theories, Abelian and non-Abelian field theories. Here we take a more modern geometric approach, which is discussed in detail in Reference [27, 28].

### 2.1.1 Abelian Gauge Symmetries

The principal step when building a Lagrangian is to enforce invariance under a set of transformations, regulating the redundant degrees of freedom in the Lagrangian. Consider the simple case where we demand that the theory is invariant under a local transformation of the Dirac field,

$$\psi(x) \rightarrow e^{i\alpha(x)}\psi(x), \tag{2.1.1}$$



where  $\psi(x)$  is rotated through an angle  $\alpha(x)$ , which depends on the position  $x$ . This is a simple phase transformation called a local transformation, as  $\alpha(x)$  can differ arbitrarily from point to point. For this transformation to be a symmetry of the theory, all terms in the Lagrangian must be invariant under this transformation. Renormalisable terms which can be present in the Lagrangian include bilinear terms of  $\psi$  and derivatives of  $\psi$ . It is clear that bilinear terms,  $\bar{\psi}\psi$ , are allowed under the local transformation, as

$$\bar{\psi}(x) \rightarrow \bar{\psi}(x)e^{-i\alpha(x)}, \quad (2.1.2)$$

where  $\bar{\psi} = \psi^\dagger \gamma^0$  is the Dirac adjoint of the field  $\psi$ . The effect is equivalent to the global transformation case [27, 28].

However, in the local case, terms involving a derivative of  $\psi$  are not invariant under the transformation\*. To illustrate this, consider the derivative of  $\psi$  in the direction of the vector  $n^\mu$ ,

$$n^\mu \partial_\mu \psi(x) = \lim_{\epsilon \rightarrow 0} \left[ \frac{1}{\epsilon} \left( \psi(x + \epsilon n) - \psi(x) \right) \right], \quad (2.1.3)$$

where the field at position  $x$  is subtracted from the field at position  $x + \epsilon n$ . The two fields  $\psi(x)$  and  $\psi(x + \epsilon n)$  transform differently under the local transformation from Equation 2.1.1. To correctly define the derivative, these two fields must be subtracted from each other. Consequently, the derivative  $\partial_\mu \psi(x)$  does not have a simple transformation law and thus has no useful geometric interpretation [28].

Therefore, to construct a local transformation for Equation 2.1.3, we must first define a scalar quantity which accounts for the difference in local transformations from one point to the next, ensuring that the subtraction of the two fields  $\psi(x)$  and  $\psi(x + \epsilon n)$  can be performed. Following the procedure from Reference [28], we define a pure phase,

---

\*In the global case, derivatives are straightforward as  $\alpha$  does not depend on  $x$ .

$$U(y, x) = e^{i\phi(y, x)}, \quad (2.1.4)$$

which depends on two points,  $y$  and  $x$ . This object is often called a *comparator* or a *Wilson Line* [27, 28]. We construct the comparator such that its transformation takes the form

$$U(y, x) \rightarrow e^{i\alpha(y)}U(y, x)e^{-i\alpha(x)}. \quad (2.1.5)$$

Consequently, from Equation 2.1.1,  $\psi(x)$  and  $U(y, x)\psi(x)$  now have the same form under the local transformation, if we stipulate that the scalar value defined at the same position,  $U(x, x) =$

1. Using this result, we can now define the derivative

$$n^\mu D_\mu \psi(x) = \lim_{\epsilon \rightarrow 0} \left[ \frac{1}{\epsilon} \left( \psi(x + \epsilon n) - U(x + \epsilon n, x)\psi(x) \right) \right], \quad (2.1.6)$$

which is called the *covariant derivative*. Assuming that  $U(y, x)$  is a continuous function, we can expand the comparator in  $\epsilon n$ , the separation between two points, such that

$$\begin{aligned} U(x + \epsilon n, x) &= U(x, x) + \epsilon n^\mu \partial_\mu U(x, x) + \frac{\epsilon^2 n^\mu n^\nu}{2!} \partial_\mu \partial_\nu U(x, x) + \mathcal{O}(\epsilon^3), \\ &= 1 - i\epsilon n^\mu A_\mu(x) - ie \frac{\epsilon^2 n^\mu n^\nu}{2!} \partial_\mu A_\nu(x) + \mathcal{O}(\epsilon^3), \end{aligned} \quad (2.1.7)$$

where, as per convention,  $e$  has been extracted as an arbitrary constant and the coefficient of displacement has been identified as a new vector field,  $A_\mu(x) = \frac{i}{e} \partial_\mu U(x, x)$ . This new vector field is called the *connection* and occurs as the infinitesimal limit of the comparator [27, 28].

Using this expansion, we can now examine the derivative in Equation 2.1.6,

$$n^\mu D_\mu \psi(x) = \lim_{\epsilon \rightarrow 0} \left[ \frac{1}{\epsilon} \left( \psi(x + \epsilon n) - (1 - ie\epsilon n^\mu A_\mu(x))\psi(x) + \mathcal{O}(\epsilon^2) \right) \right], \quad (2.1.8)$$

and therefore one can define the covariant derivative,

$$D_\mu \psi(x) = \partial_\mu \psi(x) + ieA_\mu \psi(x), \quad (2.1.9)$$

which now has a term involving the connection.

To determine how the covariant derivative transforms under the local transformation, we must examine the effect of substituting Equation 2.1.7 into Equation 2.1.5. The transformation of  $A_\mu$  is then found to be

$$A_\mu(x) \rightarrow A_\mu(x) - \frac{1}{e} \partial_\mu \alpha(x). \quad (2.1.10)$$

From this transformation, we can now examine how the covariant derivative transforms,

$$\begin{aligned} D_\mu \psi(x) &\rightarrow \partial_\mu (e^{i\alpha(x)} \psi(x)) + ie(A_\mu(x) - \frac{1}{e} \partial_\mu \alpha(x)) e^{i\alpha(x)} \psi(x), \\ &= e^{i\alpha(x)} (\partial_\mu \psi(x) + ieA_\mu \psi(x)). \end{aligned} \quad (2.1.11)$$

The covariant derivative therefore transforms as desired, ensuring locally invariant kinetic terms for  $\psi$  in the Lagrangian. Consequently, demanding that the Lagrangian is locally invariant comes at the price of introducing a new vector field into the theory. The final step in building the most general Lagrangian that is invariant under the local transformation is to construct a kinetic term for the new field  $A_\mu$ . This can be done by inspecting the commutator of the covariant derivative, and this method is discussed in Section 2.1.2 for the non-Abelian case, and also in

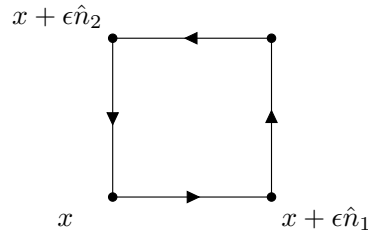


Figure 2.2: Pictorial representation of the construction of the field tensor from four consecutive comparators forming an infinitesimal square loop in the  $(\hat{n}_1, \hat{n}_2)$  plane. Figure adapted from Reference [28].

Reference [27]. To fully highlight the geometric origin of the kinetic terms for  $A_\mu$ , we follow the approach outlined in Reference [28] and construct a series of four comparators defining a square loop in space time,

$$\begin{aligned} \mathbf{U}(x) &= U(x, x + \epsilon \hat{n}_2) U(x + \epsilon \hat{n}_2, x + \epsilon \hat{n}_1 + \epsilon \hat{n}_2) \\ &\quad \times U(x + \epsilon \hat{n}_1 + \epsilon \hat{n}_2, x + \epsilon \hat{n}_1) U(x + \epsilon \hat{n}_1, x). \end{aligned} \quad (2.1.12)$$

This loop is shown pictorially in Figure 2.2. From Equation 2.1.5, it is clear that  $\mathbf{U}(x)$  is invariant under the local transformation.

To proceed, we revisit the expansion from Equation 2.1.7. If we consider the terms up to and including factors of  $\epsilon^2$ , we can rewrite the comparator as

$$U(x + \epsilon n, x) = \exp \left[ -i \epsilon n^\mu A_\mu \left( x + \frac{\epsilon}{2} n \right) + \mathcal{O}(\epsilon^3) \right], \quad (2.1.13)$$

under the restriction that  $(U(x, y))^\dagger = U(y, x)$  [28]. Using this expression in Equation 2.1.12, we obtain

$$\mathbf{U}(x) = \exp \left[ -i\epsilon e \left( -A_2 \left( x + \frac{\epsilon}{2} \hat{n}_2 \right) - A_1 \left( x + \frac{\epsilon}{2} \hat{n}_1 + \epsilon \hat{n}_2 \right) \right. \right. \\ \left. \left. + A_2 \left( x + \epsilon \hat{n}_1 + \frac{\epsilon}{2} \hat{n}_2 \right) + A_1 \left( x + \frac{\epsilon}{2} \hat{n}_1 \right) \right) + \mathcal{O}(\epsilon^3) \right], \quad (2.1.14)$$

where we have used  $A_{1,2} = \hat{n}_{1,2} A_\mu$ , and the sign of each term depends on the direction the comparator is pointing. If the  $A_\mu$  fields are then expanded in powers of  $\epsilon$ , we find

$$\mathbf{U}(x) = \exp \left[ -i\epsilon e \left( \epsilon \hat{n}_1 \partial_\mu A_2(x) - \epsilon \hat{n}_2 \partial_\mu A_1(x) \right) + \mathcal{O}(\epsilon^3) \right], \\ = 1 - i\epsilon^2 e \left( \partial_1 A_2(x) - \partial_2 A_1(x) \right) + \mathcal{O}(\epsilon^3), \\ = 1 - i\epsilon^2 e n_1^\mu n_2^\nu \left( \partial_\mu A_\nu(x) - \partial_\nu A_\mu(x) \right) + \mathcal{O}(\epsilon^3). \quad (2.1.15)$$

Therefore, as  $\mathbf{U}(x)$  is locally invariant, then the form

$$F_{\mu\nu} = \partial_\mu A_\nu(x) - \partial_\nu A_\mu(x), \quad (2.1.16)$$

is also invariant under the local transformation defined in Equation 2.1.1. We can identify this as the field tensor for electrodynamics, and the connection  $A_\mu$  as the photon field. We can now write down the most general, renormalisable Lagrangian that is invariant under a local phase transformation,

$$\mathcal{L} = -\frac{1}{4} F^{\mu\nu} F_{\mu\nu} + \bar{\psi} (i\not{D} - m)\psi, \quad (2.1.17)$$

where standard normalisations have been used, and terms that violate the simultaneous discrete

symmetries of charge (C), parity (P) and time reversal (T) have been omitted<sup>†</sup>. This Lagrangian is the standard Lagrangian for Quantum Electrodynamics (QED). Therefore, and remarkably, using the geometric approach and demanding that a Lagrangian is invariant under a local phase transformation, we have recovered QED by obtaining the field tensor, ultimately showing that the presence of an additional vector field  $A_\mu$  is a direct consequence of the phase transformation. This transformation is defined by the U(1) group, which is an *Abelian* group, sometimes called a commutative group. In addition to the Abelian case, the geometric approach outlined here allows for the construction of a general Lagrangian for more complicated geometric transformations. Such transformations will be the subject of Section 2.1.2.

### 2.1.2 Non-Abelian Gauge Symmetries

In the previous section, the physical implications of demanding invariance under the local phase transformation defined by the U(1) Abelian group was examined, and QED was recovered. In this section, we extend the argument to include continuous Lie groups. This was first proposed by Yang and Mills for the special unitary group SU( $N$ ) [34]. Here we will consider a general symmetry group with the local transformation

$$\psi(x) \rightarrow e^{i\alpha^a(x)t^a} \psi(x), \quad (2.1.18)$$

where  $\psi(x)$  is a Dirac field,  $\alpha^a(x)$  is an arbitrary function of  $x$ , and  $t^a$  are the generators of the group<sup>‡</sup>. The procedure to construct the Lagrangian, under the stipulation that it is invariant under the local transformation, is the same as for the Abelian case, however we now have orthogonal symmetry motions which do not commute with one another [28]. Therefore, these symmetry groups gain the name *non-Abelian*, or non-commuting groups. It will be shown that

<sup>†</sup>For a detailed discussion of such terms, *exempli gratia*  $c\epsilon^{\alpha\beta\mu\nu}F_{\alpha\beta}F_{\mu\nu}$ , see References [26–28].

<sup>‡</sup>Note that for different symmetry groups,  $\psi$  and  $t^a$  take on different forms. For example, for SU(2),  $\psi$  is a doublet of Dirac fields,  $\psi = (\psi_1(x), \psi_2(x))^T$  and the generators are the Pauli matrices,  $t^a = \sigma^a/2$ .

the non-commuting aspects of the theory lead to additional terms in the field strength tensor.

Analogously to the Abelian case, bilinear terms,  $\bar{\psi}\psi$ , are clearly invariant for non-Abelian symmetries, however terms containing derivatives do not have a straightforward transformation law. Following the procedure outlined in Section 2.1.1 and References [27, 28], we start by defining a comparator  $U(y, x)$  and construct a covariant derivative of the form in Equation 2.1.6. Unlike the Abelian case,  $\psi$  is now an  $N$  component vector in the fundamental representation, and thus the comparator must be a  $N \times N$  matrix, with the transformation law

$$\begin{aligned} U(y, x) &\rightarrow e^{i\alpha^a(y)t^a} U(y, x) e^{-i\alpha^a(x)t^a}, \\ &\equiv V(y)U(y, x)V^\dagger(x), \end{aligned} \tag{2.1.19}$$

where, for clarity in later calculations, we define  $V(x) = \exp(i\alpha^a(x)t^a)$ . Once again, by requiring  $U(x, x) = 1$  and restricting the comparator to be unitary, we can expand in separation such that

$$U(x + \epsilon n, x) = 1 + ig\epsilon n^\mu A_\mu^a t^a + \mathcal{O}(\epsilon^2), \tag{2.1.20}$$

where, like the Abelian case, we extract the arbitrary constant  $g$  and use the convention that the expansion has positive terms. Following the steps outlined in Equations 2.1.7 to 2.1.9, we define the covariant derivative for the non-Abelian case,

$$D_\mu\psi(x) = \partial_\mu\psi(x) - igA_\mu^a t^a\psi(x). \tag{2.1.21}$$

As the index,  $a$ , is summed over, it is clear that for the non-Abelian case we require  $a$  additional vector fields to make the derivative invariant under the local gauge transformation. For example, for  $SU(2)$ , there are three generators and therefore three additional vector fields are introduced

into the theory.

The next step is to define the transformation of the additional vector fields. It is possible to follow the procedure prescribed by the Abelian case, expanding the transformation of the comparator in Equation 2.1.19, and this is discussed in Reference [28]. Instead, here we use the fact that we require the covariant derivative to transform in the same way as the Dirac field, such that  $D_\mu\psi \rightarrow V(x)D_\mu\psi$ . Therefore,

$$A_\mu^a(x)t^a \rightarrow V(x)\left(A_\mu^a(x)t^a + \frac{i}{g}\partial_\mu\right)V^\dagger(x), \quad (2.1.22)$$

where the derivative now acts on  $V^\dagger$ . This is not straightforward, as the derivative does not necessarily commute with the generators in the exponent. Expanding to first order in  $\alpha$ , the infinitesimal transformation has the form

$$A_\mu^a(x)t^a \rightarrow A_\mu^a(x)t^a + \frac{1}{g}\partial_\mu(\alpha^a(x))t^a + i\left[\alpha^a(x)t^a, A_\mu^b(x)t^b\right] + \mathcal{O}(\alpha^2). \quad (2.1.23)$$

Comparing to the Abelian case in Equation 2.1.10, in the non-Abelian case there is an extra term. This additional commutator is a consequence of the non-commutativity of the local transformation.

To finish the construction of the most general Lagrangian that is invariant under a non-Abelian symmetry, we must now build the field tensor. The approach is analogous to the construction of the Abelian field tensor shown in Equation 2.1.16, and therefore allows for the inclusion of a kinetic term for the additional vector fields. It is possible to construct the field tensor for the non-Abelian case by considering a loop of comparators, as shown in Figure 2.2. This approach is discussed in detail in Reference [35]. For brevity, here we can make an argument to show the invariance of the field tensor by using the commutator of two covariant derivatives,



as outlined in References [27,28]. Since we have shown that the covariant derivative transforms in the same way as the Dirac field, then the second covariant derivative will also transform in this way. Therefore, one can define the commutator of covariant derivatives as

$$[D_\mu, D_\nu]\psi(x) \rightarrow V(x)[D_\mu, D_\nu]\psi(x). \quad (2.1.24)$$

Note that this is equivalent to the geometric approach using comparators, outlined in Section 2.1.1, and this equivalence is discussed in detail in References [27,28]. If we now write the commutator explicitly,

$$\begin{aligned} [D_\mu, D_\nu]\psi &= [\partial_\mu, \partial_\nu]\psi - ig\left([\partial_\mu, A_\nu^b t^b] - [\partial_\nu, A_\mu^a t^a]\right)\psi - g^2[A_\mu^a t^a, A_\nu^b t^b]\psi, \\ &= -ig\left(\partial_\mu A_\nu^a t^a - \partial_\nu A_\mu^a t^a - ig[A_\mu^a t^a, A_\nu^b t^b]\right)\psi, \end{aligned} \quad (2.1.25)$$

we see that the commutator  $[D_\mu, D_\nu]$  is a multiplicative factor acting on  $\psi$ . In the Abelian case, since  $\psi$  accounts for the entire transformation in Equation 2.1.24, then the commutator of covariant derivatives must be invariant under the local transformation. However, in the non-Abelian case, there remains an extra term in the field tensor, such that

$$F_{\mu\nu}^a t^a = \partial_\mu A_\nu^a t^a - \partial_\nu A_\mu^a t^a - ig[A_\mu^a t^a, A_\nu^b t^b]. \quad (2.1.26)$$

This can be simplified using the conventional form for the generators,

$$[t^a, t^b] = if^{abc}t^c, \quad (2.1.27)$$

where  $f^{abc}$  represents the structure constants, such that  $f^{abc}$  is completely asymmetric. Using

this definition, Equation 2.1.26 simplifies to the standard form

$$F_{\mu\nu}^a = \partial_\mu A_\nu^a - \partial_\nu A_\mu^a + gf^{abc} A_\mu^b A_\nu^c. \quad (2.1.28)$$

Examining the infinitesimal transformation of  $F_{\mu\nu}^a$ ,

$$F_{\mu\nu}^a \rightarrow F_{\mu\nu}^a - f^{abc} \alpha^b F_{\mu\nu}^c, \quad (2.1.29)$$

we see that the non-Abelian field tensor is not a gauge-invariant, as it contains  $a$  fields, which are associated with  $a$  different transformations, and therefore  $F_{\mu\nu}^a$  does not have a trivial transformation law. However, it is possible to write an invariant form by combining the  $a$  field strength tensors in the trace

$$-\frac{1}{2} \text{Tr} \left[ (F_{\mu\nu}^a t^a)^2 \right] = -\frac{1}{4} F_{\mu\nu}^a F^{a,\mu\nu}. \quad (2.1.30)$$

We can now write down the general, renormalisable Lagrangian for a non-Abelian gauge symmetry, including all terms that do not violate  $C$ ,  $P$  and  $T$  symmetries,

$$\mathcal{L} = -\frac{1}{4} F_{\mu\nu}^a F^{a,\mu\nu} + \bar{\psi}(i\not{D} - m)\psi. \quad (2.1.31)$$

This Lagrangian looks strikingly similar to the Abelian case in Equation 2.1.17, however there are now non-linear terms present in the field tensor, proportional to the structure constants,  $f^{abc}$ . As a consequence, the non-Abelian symmetry allows for interactions between gauge bosons. In the Standard Model, there are two non-Abelian gauge groups and one Abelian group. These can all be derived from the general approach defined above, with the Abelian case being recovered if  $f^{abc} = 0$ .

## 2.2 The Standard Model of Particle Physics

The Standard Model of particle physics is a Quantum Field Theory (QFT) with the gauge structure

$$\mathrm{SU}(3)_c \times \mathrm{SU}(2)_L \times \mathrm{U}(1)_Y, \quad (2.2.1)$$

where the subscripts  $c$ ,  $L$  and  $Y$  correspond to colour, left-chirality and hypercharge respectively and can be split into two main parts: the Quantum Chromodynamic (QCD) sector and the Electroweak (EW) sector. The  $\mathrm{SU}(3)_c$  group is the gauge group of QCD [36–38] with a corresponding coupling  $g_s$  and eight gauge bosons,  $G^a$  where  $a = 1, \dots, 8$ , called *gluons*, which mediate the strong force. Quantum Chromodynamics is not chiral and acts on colour-charged particles, the quarks,  $q_c^i$ , where  $i$  denotes the flavour and  $c = 1, 2, 3$  is the colour index. The gauge group  $\mathrm{SU}(3)_c$  is a non-Abelian group and therefore, as shown in Section 2.1.2 and discussed further in Section 2.2.1, this leads to self interactions between the gluons.

The  $\mathrm{SU}(2)_L \times \mathrm{U}(1)_Y$  part is the gauge group of the EW sector [39–41]. The  $\mathrm{SU}(2)_L$  group has coupling  $g$  and gauge bosons,  $W^i$  where  $i = 1, 2, 3$ . The  $\mathrm{U}(1)_Y$  group has corresponding coupling  $g'$  and has a single gauge boson,  $B$ . The  $\mathrm{SU}(2)_L \times \mathrm{U}(1)_Y$  is a chiral theory, and therefore acts on left and right-chiral particles differently:  $\mathrm{SU}(2)_L$  acts upon flavour indices of left-chiral particles only, and  $\mathrm{U}(1)_Y$  acts on left and right-chiral particles with different charges. Similarly to QCD,  $\mathrm{SU}(2)_L$  is a non-Abelian group and therefore includes self interactions between gauge bosons, however,  $\mathrm{U}(1)_Y$  is an Abelian group and thus we do not have self-interaction terms from  $\mathrm{U}(1)_Y$  in the Lagrangian.

Under the gauge structure from Equation 2.2.1, the matter content of the Standard Model, the fermions, take the form

$$\begin{aligned}
q_L^i &= \begin{pmatrix} u_L \\ d_L \end{pmatrix}, \begin{pmatrix} c_L \\ s_L \end{pmatrix}, \begin{pmatrix} t_L \\ b_L \end{pmatrix}, \\
u_R^i &= u_R, c_R, t_R, \\
d_R^i &= d_R, s_R, b_R, \\
l_L^i &= \begin{pmatrix} \nu_{eL} \\ e_L \end{pmatrix}, \begin{pmatrix} \nu_{\mu L} \\ \mu_L \end{pmatrix}, \begin{pmatrix} \nu_{\tau L} \\ \tau_L \end{pmatrix}, \\
e_R^i &= e_R, \mu_R, \tau_R.
\end{aligned} \tag{2.2.2}$$

where the index  $i$  represents the flavour of the fermion. Due to the chiral nature of the  $SU(2)_L$  gauge group, the left-chiral states (subscript  $L$ ) transform as doublets under  $SU(2)_L$ , and right-chiral states (subscript  $R$ ) transform as singlets. In Section 2.2.3, we will see that the symmetry group of the Standard Model spontaneously breaks to

$$SU(3)_c \times SU(2)_L \times U(1)_Y \rightarrow SU(3)_c \times U(1)_Q, \tag{2.2.3}$$

and the photon,  $Z$  and  $W^\pm$  bosons will be resolved from the gauge bosons  $W^i$  and  $B$ . Here, the EW sector,  $SU(2)_L \times U(1)_Y$ , breaks to  $U(1)_Q$ , the gauge group of Quantum Electrodynamics (QED). Quantum Chromodynamics is not spontaneously broken. This spontaneous symmetry breaking gives rise to a massive scalar boson, the Higgs boson, which in turn provides a mechanism that gives mass to the  $W^\pm$  and  $Z$  bosons and the fermions.

With the discovery of the Higgs boson in 2012 [44, 45], the particle content of the Standard Model has been experimentally confirmed and the latest observations from the Particle Data Group (PDG) [43] are displayed pictorially in Figure 2.3. The dynamics of any QFT are encoded into the Lagrangian density. Imposing Poincaré and the gauge symmetries from Equation 2.2.1,

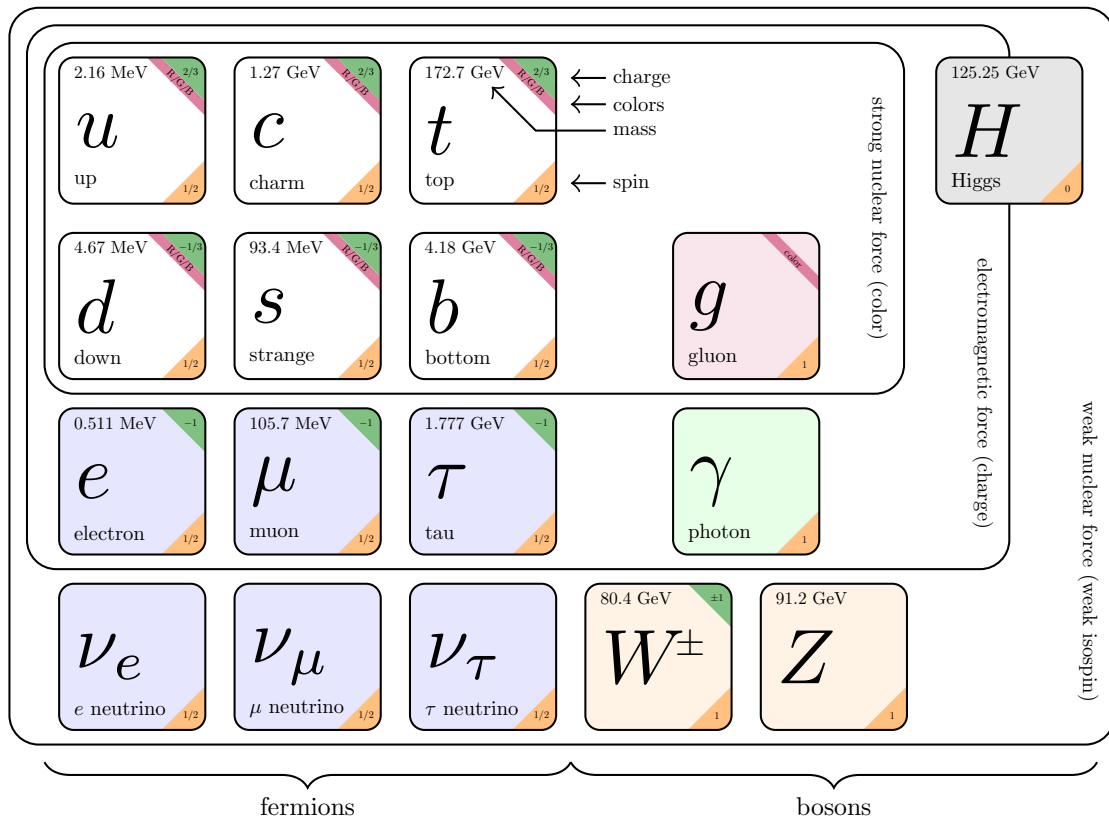


Figure 2.3: The Standard Model of particle physics consists of three generations of fermions and the gauge bosons. Displayed in the white coloured boxes are the six quarks, which interact via the strong and electroweak forces. In the light blue boxes are the leptons, the first row includes the charged leptons which interact via the electroweak force, and the second row shows the neutrinos, which interact via the weak force only. The gauge bosons are the gluons (red), which mediate the strong force, the photon (green), which mediates the electromagnetic force, and the weak bosons (orange), which mediate the weak force. Finally, the Higgs boson, which provides a mechanism to generate the masses of the particles, is shown in grey. The Figure has been adapted from Reference [42] to include the latest observed values from the Particle Data Group [43].

we can write the Standard Model Lagrangian in four main terms,

$$\mathcal{L} = \mathcal{L}_{\text{gauge}} + \mathcal{L}_{\text{fermion}} + \mathcal{L}_{\text{Higgs}} + \mathcal{L}_{\text{Yukawa}}. \quad (2.2.4)$$

To fully understand the dynamics of the Standard Model, in this section, we review each part of the Lagrangian in the following sections.

### 2.2.1 The Gauge Sector

The first part of the Standard Model is the gauge sector, which describes the dynamics of the gauge fields. The most general formulation<sup>§</sup> of the gauge sector is

$$\mathcal{L}_{\text{gauge}} = -\frac{1}{4}G_{\mu\nu}^a G^{a,\mu\nu} - \frac{1}{4}W_{\mu\nu}^i W^{i,\mu\nu} - \frac{1}{4}B_{\mu\nu} B^{\mu\nu}, \quad (2.2.5)$$

where  $G_{\mu\nu}^a$  and  $W_{\mu\nu}^i$  are the field tensors of the non-Abelian gauge groups  $\text{SU}(3)_c$  and  $\text{SU}(2)_L$  respectively, and  $B_{\mu\nu}$  is the field tensor of the Abelian gauge group  $\text{U}(1)_Y$ . The field tensors are defined in the standard form

$$G_{\mu\nu}^a = \partial_\mu G_\nu^a - \partial_\nu G_\mu^a - g_s f^{abc} G_\mu^b G_\nu^c, \quad a, b, c = 1, \dots, 8, \quad (2.2.6)$$

$$W_{\mu\nu}^i = \partial_\mu W_\nu^i - \partial_\nu W_\mu^i - g \epsilon^{ijk} W_\mu^j W_\nu^k, \quad i, j, k = 1, 2, 3, \quad (2.2.7)$$

$$B_{\mu\nu} = \partial_\mu B_\nu - \partial_\nu B_\mu, \quad (2.2.8)$$

where  $f^{abc}$  and  $\epsilon^{ijk}$  are the structure constants of  $\text{SU}(3)_c$  and  $\text{SU}(2)_L$  respectively, and the indices of  $\text{SU}(N)$  groups grow as  $N^2 - 1$ . As discussed in Section 2.1, the non-commutative nature of

---

<sup>§</sup>Note that terms which violate discrete symmetries such as  $T$  and  $P$  are not included. These are discussed in detail in References [26–28].

non-Abelian symmetries leads to the non-linear terms in the field tensors. From Equation 2.2.5, we see that this leads to three-point and four-point self-interactions between the gauge bosons for the non-Abelian groups, whereas there are no self-interactions for the Abelian group.

Performing an infinitesimal gauge transformation of the respective gauge groups on the field tensors, we find

$$G_\mu^a \rightarrow G_\mu^a - \frac{1}{g_s} \partial_\mu \alpha^a + f^{abc} \alpha^b G_\mu^c, \quad (2.2.9)$$

$$W_\mu^i \rightarrow W_\mu^i - \frac{1}{g} \partial_\mu \alpha^i + \epsilon^{ijk} \alpha^j W_\mu^k, \quad (2.2.10)$$

$$B_\mu \rightarrow B_\mu - \frac{1}{g'} \partial_\mu \alpha, \quad (2.2.11)$$

where the fields have associated couplings  $g_s$ ,  $g$  and  $g'$ . These transformations follow from the discussion of gauge invariance in Section 2.1 and are required if we demand the Lagrangian to be invariant under the gauge structure in Equation 2.2.1. Furthermore, from Equations 2.2.9 to 2.2.11 we can see that mass terms of the form  $G_\mu^a G^{a,\mu}$ ,  $W_\mu^i W^{i,\mu}$  and  $B_\mu B^\mu$  are not gauge invariant, and without a symmetry breaking mechanism, the gauge bosons remain massless.

## 2.2.2 The Fermionic Sector

The fermionic sector of the Standard Model considers the interactions of the fermions, spin- $\frac{1}{2}$  fields which make up the matter content of the Standard Model. As shown in Equation 2.2.2, the fermionic sector is constructed from three families of quarks and leptons, which couple to the gauge fields through the covariant derivative

$$D_\mu = \partial_\mu + ig' B_\mu Y + ig W_\mu^j I^j + ig_s G_\mu^a T^a, \quad (2.2.12)$$

where  $Y$ ,  $I^i$ , and  $T^a$  are the generators of the  $U(1)_Y$ ,  $SU(2)_L$ , and  $SU(3)_c$  groups respectively. The form of the generators depends on which field the covariant derivative is acting on. The generators of  $SU(3)_c$  act only on colour-charged particles, the quarks, and take the form  $T^a = \lambda^a/2$ , where  $\lambda^a$  are the Gell-Mann matrices. The quarks transform as  $SU(3)_c$  triplets, such that the up quark has the form

$$u_L = \begin{pmatrix} u_L^r \\ u_L^b \\ u_L^g \end{pmatrix}, \quad (2.2.13)$$

where  $r$ ,  $g$  and  $b$  are the colours of QCD.

The  $SU(2)_L$  group is chiral and therefore is dependent on the chirality of the fermion. When acting on left-chiral particles, the generators  $I^i = \sigma^i/2$ , where  $\sigma^i$  are the Pauli matrices, and when acting on right-chiral particles the generators  $I^i = 0$ . Therefore, as shown in Equation 2.2.2, left-chiral fields transform as  $SU(2)_L$  doublets, and right-chiral fields transform as singlets, introducing parity breaking in  $SU(2)_L$ . As  $SU(2)_L$  does not commute with  $SU(3)_c$ , there are no flavour-changing interactions in QCD.

For  $U(1)_Y$ , the weak hypercharge  $Y$  is related to the electric charge of the fermion via the Gell-Mann-Nishijima relation,

$$Q = I^3 + \frac{1}{2}Y, \quad (2.2.14)$$

where  $I^3$  is the third generator of the  $SU(2)_L$  group. The  $U(1)_Y$  group commutes with  $SU(3)_c$  and  $SU(2)_L$ , and therefore each component of the  $SU(3)_c \times SU(2)_L$  multiplets has the same weak hypercharge. Table 2.1 shows the relative charges of the particles under the Standard Model gauge group.



Names		Spin	(SU(3) <sub>c</sub> , SU(2) <sub>L</sub> , U(1) <sub>Y</sub> )
Quarks,	$q_L^i$	$\frac{1}{2}$	$(\mathbf{3}, \mathbf{2}, \frac{1}{6})$
	$u_R^i$	$\frac{1}{2}$	$(\mathbf{3}, \mathbf{1}, \frac{2}{3})$
	$d_R^i$	$\frac{1}{2}$	$(\mathbf{3}, \mathbf{1}, -\frac{1}{3})$
Leptons,	$l_L^i$	$\frac{1}{2}$	$(\mathbf{1}, \mathbf{2}, -\frac{1}{2})$
	$e_R^i$	$\frac{1}{2}$	$(\mathbf{1}, \mathbf{1}, -1)$
Gluon	$g$	1	$(\mathbf{8}, \mathbf{1}, 0)$
W bosons	$W^i$	1	$(\mathbf{1}, \mathbf{3}, 0)$
B boson	$B$	1	$(\mathbf{1}, \mathbf{1}, 0)$
Higgs	$\phi$	0	$(\mathbf{1}, \mathbf{2}, \frac{1}{2})$

Table 2.1: Transformation properties of the chiral multiplets of the Standard Model in the fundamental representation, under the gauge group SU(3)<sub>c</sub>×SU(2)<sub>L</sub>×U(1)<sub>Y</sub>.

Using the fields defined in Equation 2.2.2 and the form of the covariant derivative in Equation 2.2.12, we can write down the Lagrangian for the fermionic sector of the Standard Model,

$$\begin{aligned} \mathcal{L}_{\text{fermion}} = & \bar{q}_L^i i\gamma^\mu D_\mu q_L^i + \bar{u}_R^i i\gamma^\mu D_\mu u_R^i \\ & + \bar{d}_R^i i\gamma^\mu D_\mu d_R^i + \bar{l}_L^i i\gamma^\mu D_\mu l_L^i + \bar{e}_R^i i\gamma^\mu D_\mu e_R^i, \end{aligned} \quad (2.2.15)$$

where the index  $i$  is summed over all fermion flavours. Note that a mass cannot be explicitly written in the Lagrangian for the fermions, due to the fact that the SU(2)<sub>L</sub> and U(1)<sub>Y</sub> representations are chiral. Therefore, a fermion mass term of the form  $m\bar{\psi}\psi$  is not invariant. The fermion sector therefore consists only of gauge covariant kinetic terms.

### 2.2.3 The Higgs Sector

We saw in Sections 2.2.1 and 2.2.2 that simple bilinear mass terms of the form  $A_\mu^i A^{i,\mu}$  and  $m\bar{\psi}\psi$  are not gauge invariant and therefore are not allowed in the Standard Model. In order to describe

the experimentally observed masses of the particles, a mechanism is required to generate the masses in the theory by breaking the gauge symmetry. The Higgs mechanism [46–48] provides a procedure of spontaneous symmetry breaking to generate the masses of the massive vector bosons, the  $W^\pm$  and  $Z$ . The Lagrangian of the Higgs sector takes the form

$$\mathcal{L}_{\text{Higgs}} = \left(D_\mu \phi\right)^\dagger D^\mu \phi - V(\phi), \quad (2.2.16)$$

where the field  $\phi = (\phi^+, \phi^0)^\text{T}$ , is a complex scalar field, which transforms as  $(\mathbf{1}, \mathbf{2}, \frac{1}{2})$  under  $\text{SU}(3)_c \times \text{SU}(2)_L \times \text{U}(1)_Y$ , transforming trivially under  $\text{SU}(3)_c$ . Therefore the covariant derivative takes the form

$$D_\mu \phi = \left(\partial_\mu + ig' B_\mu Y + ig W_\mu^i T^i\right) \phi, \quad (2.2.17)$$

which, from Equation 2.2.16, leads to three and four point interactions with the vector bosons.

The potential term

$$V(\phi) = -\mu^2 \phi^\dagger \phi + \lambda (\phi^\dagger \phi)^2, \quad (2.2.18)$$

is symmetric under  $\text{U}(1)$  transformations, and has a steady ground state if  $\lambda > 0$ . For the Higgs mechanism to provide the masses of the particles, we require  $\mu^2 < 0$ , thus allowing for symmetry breaking<sup>¶</sup>.

It is convenient to write the scalar field in Hermitian form

$$\phi = \frac{1}{\sqrt{2}} \begin{pmatrix} \phi_1 + i\phi_2 \\ \phi_3 + i\phi_4 \end{pmatrix}, \quad (2.2.19)$$

---

<sup>¶</sup>If  $\mu^2 > 0$ , then the minimum of the potential would be at  $|\phi| = 0$ , thus  $\text{SU}(2)_L \times \text{U}(1)_Y$  is unbroken at the minimum

where  $\phi_i = \phi_i^\dagger$  are four real scalar fields. Due to the symmetry of the potential, we are free to choose a vacuum expectation value for  $\phi$  such that  $\langle 0 | \phi_i | 0 \rangle = 0$  for  $i = 1, 2, 4$  and  $\langle 0 | \phi_3 | 0 \rangle = v$ , where  $v \geq 0$  [26,27]. The vacuum expectation value of the scalar field is therefore  $\phi_0 = \frac{1}{2}(0, v)^T$ . Expanding the potential in  $\phi_0$ , we find

$$V(\phi_0) = -\frac{1}{2}\mu^2 v^2 + \frac{1}{4}\lambda v^4. \quad (2.2.20)$$

If we require that  $\mu^2 < 0$ , then the  $v = 0$  case is unstable and thus we require the vacuum expectation value to be non-zero,  $v \neq 0$ , therefore breaking the  $SU(2)_L \times U(1)_Y$  symmetry. We can find the value of the vacuum expectation value by minimising the potential,

$$\left. \frac{\partial V(\phi)}{\partial \phi} \right|_{\phi_0} = v(-\mu^2 + \lambda v^2) = 0, \quad (2.2.21)$$

which, at the minimum, has the solution

$$v = \sqrt{\frac{\mu^2}{\lambda}}. \quad (2.2.22)$$

As the vacuum is electrically neutral, the symmetry breaking pattern follows

$$SU(2)_L \times U(1)_Y \rightarrow U(1)_Q, \quad (2.2.23)$$

retrieving the QED gauge group,  $U(1)_Q$ , associated with the unbroken generator  $Q$ , defined in Equation 2.2.14. Consequently, we expect a massless boson associated with the unbroken QED, which is identified as the photon field,  $A_\mu$ . Furthermore, as  $SU(3)_c$  also remains unbroken, the eight gluons,  $G_\mu^a$ , remain massless. However, as a result of the symmetry breaking, we see that the remaining vector bosons gain a mass.

We can examine the generation of the masses by parameterising the scalar field in the unitary gauge [28]

$$\phi(x) = U(x) \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ v + H(x) \end{pmatrix}, \quad (2.2.24)$$

where  $v$  is the vacuum expectation value of  $\phi$  from Equation 2.2.22, and  $H(x)$  is an Hermitian scalar field with vacuum expectation value  $\langle 0 | H(x) | 0 \rangle = 0$ . In this parameterisation, the most general complex-valued two-component spinor is achieved by applying an  $SU(2)_L$  gauge transformation  $U(x)$ . We can reduce the field  $\phi$  to only one physical degree of freedom by applying a gauge transformation to eliminate  $U(x)$  from the Lagrangian [28].

We can now rewrite the Lagrangian from Equation 2.2.16 in the unitary gauge. First, consider the kinetic energy part,

$$(D_\mu \phi)^\dagger D^\mu \phi = \frac{1}{2} \begin{pmatrix} 0 & v + H(x) \end{pmatrix} \left[ \partial_\mu + ig' B_\mu Y + ig W_\mu^i I^i \right]^2 \begin{pmatrix} 0 \\ v + H(x) \end{pmatrix}, \quad (2.2.25)$$

which will yield kinetic and gauge interactions terms for the  $H(x)$  field, as well as terms that depend only on  $v$ . Expressing the covariant derivative in matrix form

$$\begin{aligned} D_\mu &= \partial_\mu + i \frac{g}{2} \begin{pmatrix} B_\mu & 0 \\ 0 & B_\mu \end{pmatrix} + i \frac{g}{2} \begin{pmatrix} W_\mu^3 & W_\mu^1 - iW_\mu^2 \\ W_\mu^1 + iW_\mu^2 & -W_\mu^3 \end{pmatrix}, \\ &= \partial_\mu + \frac{i}{2} \begin{pmatrix} g' B_\mu + g W_\mu^3 & g(W_\mu^1 - iW_\mu^2) \\ g(W_\mu^1 + iW_\mu^2) & g' B_\mu - g W_\mu^3 \end{pmatrix}, \end{aligned} \quad (2.2.26)$$

we can consider the terms from Equation 2.2.25 that depend only on  $v$ ,

$$(D_\mu\phi)^\dagger D^\mu\phi = \frac{v^2}{8} \left( g^2(W_\mu^1 - iW_\mu^2)(W^{1,\mu} + iW^{2,\mu}) + (g'B_\mu - gW_\mu^3)(g'B^\mu - gW^{3,\mu}) \right) + \mathcal{O}(H(x)). \quad (2.2.27)$$

Here we can identify the three massive bosons

$$W_\mu^\pm = \frac{1}{\sqrt{2}}(W_\mu^1 \mp iW_\mu^2), \quad \text{with mass} \quad m_W = \frac{gv}{2}, \quad (2.2.28)$$

$$Z_\mu = \frac{1}{\sqrt{g^2 + g'^2}}(gW_\mu^3 - g'B_\mu), \quad \text{with mass} \quad m_Z = \sqrt{g^2 + g'^2} \frac{v}{2}, \quad (2.2.29)$$

which correspond to the  $W^\pm$  and  $Z$  bosons. The final boson, orthogonal to the  $Z_\mu$ , remains massless and is identified as the photon,

$$A_\mu = \frac{1}{\sqrt{g^2 + g'^2}}(gW_\mu^3 + g'B_\mu), \quad \text{with mass} \quad m_A = 0. \quad (2.2.30)$$

These masses have been experimentally observed to be  $m_W = 80.38$  GeV,  $m_Z = 91.19$  GeV and  $m_A < 1 \times 10^{-18}$  eV [43]. The mixing between the  $Z$  and photon can be expressed by defining the *weak mixing angle*

$$\begin{pmatrix} Z_\mu \\ A_\mu \end{pmatrix} = \begin{pmatrix} \cos \theta_w & -\sin \theta_w \\ \sin \theta_w & \cos \theta_w \end{pmatrix} \begin{pmatrix} W_\mu^3 \\ B_\mu \end{pmatrix}, \quad (2.2.31)$$

where we can identify

$$\cos \theta_w = \frac{g}{\sqrt{g^2 + g'^2}}, \quad \text{and} \quad \sin \theta_w = \frac{g'}{\sqrt{g^2 + g'^2}}. \quad (2.2.32)$$

By rewriting the covariant derivative in terms of the mass eigenstates, one can identify electron charge,

$$e = \frac{gg'}{\sqrt{g^2 + g'^2}}, \quad (2.2.33)$$

and consequently show that the coupling  $g = e/\sin \theta_w$ . Therefore, we see that the masses of the  $W$  and  $Z$  bosons are not independent, and are related by

$$m_W = m_Z \cos \theta_w, \quad (2.2.34)$$

and as a result all processes involving an exchange of a  $W$  or  $Z$  boson can be written in terms of  $e$ ,  $\theta_w$  and  $m_W$ <sup>‡</sup>.

Returning to the kinetic terms in Equation 2.2.25, we can now consider all terms by expressing the fields in the mass eigenstate form,

$$(D_\mu \phi)^\dagger D^\mu \phi = \frac{1}{2} \partial_\mu H \partial^\mu H + \left[ m_W^2 W_\mu^+ W^{\mu-} + \frac{1}{2} m_Z^2 Z_\mu Z^\mu \right] \left( 1 + \frac{H}{v} \right)^2, \quad (2.2.35)$$

from which we can see that the gauge bosons couple to the  $H(x)$  field proportionally to their mass.

Finally, one can resolve the dynamics of the  $H(x)$  field by examining the potential of the Higgs Lagrangian, expanded in the unitary gauge,

---

<sup>‡</sup>This is certainly true for tree level processes. See References [27, 28] for a discussion of higher level processes.

$$V(\phi) = +\frac{\mu^4}{4\lambda} + \mu^2 H^2 + \lambda v H^3 + \frac{1}{4}\lambda H^4. \quad (2.2.36)$$

We see that the  $H(x)$  field is a scalar with mass

$$m_H = \sqrt{2}\mu = \sqrt{2}\lambda v, \quad (2.2.37)$$

which we identify as the Higgs boson<sup>\*\*</sup>. Furthermore, from the potential terms we see that the Higgs has three and four point self interaction terms. These dynamics are key to determining whether the Higgs-like particle discovered in 2012 is indeed the Standard Model Higgs boson.

In summary, we have generated the masses of the gauge bosons by breaking the symmetry from  $SU(2)_L \times U(1)_Y \rightarrow U(1)_Q$ , driven by the potential  $V(\phi)$  taking on a non-zero vacuum expectation value. The  $SU(2)_Y$  generators are broken, and are associated with massive gauge bosons, whilst a remaining unbroken generator is associated with a massless boson. As a byproduct of this symmetry breaking, we gain a massive scalar, the Higgs boson. The Higgs boson was observed in 2012 by the ATLAS [44] and CMS [45] experiments, which are situated at interaction points along the beamline of the Large Hadron Collider at CERN.

#### 2.2.4 The Yukawa Sector

In the Standard Model, ordinary bilinear mass terms are not allowed for the fermions as the left and right-chiral parts transform differently under  $SU(2)_L$ . Using the mechanism outlined in Section 2.2.3, it is possible to write gauge invariant terms which couple the left and right-chiral parts of the fermions, via the scalar field  $\phi$ . This is written in the Yukawa part of the Standard Model Lagrangian,

---

<sup>\*\*</sup>Note that this is a tree level mass, and the Higgs is sensitive to mass corrections from new physics at high energies. References [26, 28] discuss this further.

$$\mathcal{L}_{\text{Yukawa}} = -\lambda_l^{ij} \bar{l}_L^i \phi e_R^j - \lambda_u^{ij} \bar{q}_L^i \tilde{\phi} u_R^j - \lambda_d^{ij} \bar{q}_L^i \phi d_R^j + \text{h.c.}, \quad (2.2.38)$$

where the scalar field  $\tilde{\phi} = i\sigma^2 \phi^\dagger$  has been defined to obtain an invariant term for the up-type quarks. The Yukawa couplings,  $\lambda_f^{ij}$ , are complex-valued elements of the  $3 \times 3$  matrices,  $\Lambda_f$ , where  $i, j \in 1, 2, 3$ . The  $\Lambda_f$  matrices have 18 free parameters, however these can be reduced by diagonalising the  $\Lambda_f$  matrices,

$$\Lambda_f = V_{fL} \hat{\Lambda}_f V_{fR}^\dagger, \quad (2.2.39)$$

where  $V_{fL}$  and  $V_{fR}$  are unitary matrices and  $\hat{\Lambda}_f = \text{diag}(\lambda_f^1, \lambda_f^2, \lambda_f^3)$  is a real valued, diagonal matrix. The fermion fields are therefore redefined:

$$u_L^i \rightarrow V_{uL}^{ij} u_L^j, \quad u_R^i \rightarrow V_{uR}^{ij} u_R^i, \quad (2.2.40)$$

$$d_L^i \rightarrow V_{dL}^{ij} d_L^j, \quad d_R^i \rightarrow V_{dR}^{ij} d_R^i, \quad (2.2.41)$$

$$l_L^i \rightarrow V_{lL}^{ij} l_L^j, \quad e_R^i \rightarrow V_{eR}^{ij} e_R^i. \quad (2.2.42)$$

The matrices  $V_{fL}$  and  $V_{fR}$  do not affect the lepton kinetic terms as they are unitary matrices. This is not entirely true for the quarks. As left and right-chiral quark fields interact with identical couplings under  $\text{SU}(3)_c$ , then the matrices  $V_u$  and  $V_d$  will commute with the covariant derivative, therefore the kinetic terms corresponding to QCD remain unchanged. However, in the EW sector, quarks are mixed by weak interactions involving the charged  $W$  bosons, and flavour-changing terms, such as  $\bar{u}_L^i \gamma^\mu d_L^j$ , are present in the Lagrangian<sup>††</sup>. Applying the field

---

<sup>††</sup>Note that, in the Standard Model, there are no weak neutral flavour-changing currents. This is an area that is being searched for at modern collider experiments.



redefinition to this flavour-changing current term we see

$$\frac{1}{\sqrt{2}}\bar{u}_L^i\gamma^\mu d_L^i \rightarrow \bar{u}_L^i\gamma^\mu(V_u^\dagger V_d)d_L^i, \quad (2.2.43)$$

where the unitary matrices  $V_u$  and  $V_d$  do not cancel each other. Therefore, the charged weak flavour-changing current links the up-type quarks with the down-type quarks via a rotation described by the unitary matrix,

$$V_{\text{CKM}} = V_u^\dagger V_d. \quad (2.2.44)$$

This matrix is the so-called Cabibbo-Kobayashi-Maskawa (CKM) matrix [49, 50], which allows for the parameterisation of the quark flavour mixing. The CKM matrix has nine free parameters: three rotation angles, and six phases. By redefining the quark fields using a global phase transformation, one can remove five of the phases from the CKM matrix, leaving three angles and one phase. This remaining phase makes some couplings to the  $W^+$  matrix complex, and is the only CP violating parameter in the Standard Model [28].

Returning to Equations 2.2.40 to 2.2.42, these field redefinitions only have the effect of reducing the number of free parameters in  $\Lambda_f$  to three real valued parameters for each type of fermion,  $\lambda_f^i$ . The Yukawa part of the Standard Model Lagrangian can now be written as

$$\mathcal{L}_{\text{Yukawa}} = -\lambda_l^i \bar{l}_L^i \phi e_R^i - \lambda_u^i \bar{q}_L^i \tilde{\phi} u_R^i - \lambda_d^i \bar{q}_L^i \phi d_R^i + \text{h.c.} \quad (2.2.45)$$

The particle spectrum for the fermions of the Standard Model can be retrieved by expanding  $\mathcal{L}_{\text{Yukawa}}$  using the definition of the scalar field in the unitary gauge from Equation 2.2.24,

$$\mathcal{L}_{\text{Yukawa}} = -\frac{1}{\sqrt{2}} \left[ \lambda_l^i v \bar{l}_L^i e_R^i + \lambda_u^i v \bar{u}_L^i u_R^i + \lambda_d^i v \bar{d}_L^i d_R^i \right. \\ \left. + \lambda_l^i \bar{l}_L^i H e_R^i + \lambda_u^i \bar{u}_L^i H u_R^i + \lambda_d^i \bar{d}_L^i H d_R^i \right] + \text{h.c.}, \quad (2.2.46)$$

where we obtain mass terms of the form

$$-\frac{\lambda_f^i v}{\sqrt{2}} (\bar{f}_L^i f_R^i + \bar{f}_R^i f_L^i), \quad (2.2.47)$$

and thus, the masses of the up and down type quarks, and the leptons can be identified as

$$m_u^i = \frac{\lambda_u^i v}{\sqrt{2}}, \quad m_d^i = \frac{\lambda_d^i v}{\sqrt{2}}, \quad m_l^i = \frac{\lambda_l^i v}{\sqrt{2}}, \quad (2.2.48)$$

respectively. We see that the Higgs boson interacts with the fermions, with a coupling proportional to the mass of the fermion.

Note, that in the Standard Model, there is no right-chiral field for the neutrinos and therefore a mass is not generated in this way. If a right-chiral neutrino field,  $\nu_R$ , was introduced to the theory, then a Yukawa term for the neutrino could be written as

$$-\lambda_\nu^i \bar{l}_L^i \tilde{\phi} \nu_R^i + \text{h.c.}, \quad (2.2.49)$$

and the neutrino would acquire a mass, which, following the symmetry from the quark sector, should be comparable to the electron's mass. However, neutrinos have been observed to have extremely small masses. Following arguments of naturalness, it is therefore assumed that a right-chiral neutrino field does not exist, and that neutrinos must gain their mass from another

mechanism.

## 2.3 The Parton Shower

With the ability to accelerate protons to higher energies than leptons, modern collider experiments are preferentially hadron colliders. Heralded as discovery machines, hadron colliders have a significant energy advantage, however the collision environment in hadron colliders is much more complex than their lepton counterparts. Collision events can result in many particles being produced, and are accompanied by large Standard Model backgrounds due to the composite nature of the colliding hadrons. For this reason, the theoretical description of such processes can become highly challenging, and since the 1980s sophisticated computational methods have been developed to combat this challenge. The most successful approach to simulating high energy collision events in hadron colliders has been the development of Monte-Carlo event generators [29, 30]. These algorithms exploit the ability to factorise QCD cross sections, allowing for the separation of perturbative and non-perturbative parts [23–25]. In this section, we give a brief overview of one of the cornerstones of modern event generation, the perturbative process of the QCD parton shower. A detailed review of event generation for high energy hadron colliders can be found in References [29–33].

### 2.3.1 Collinear Factorisation

The QCD parton shower process evolves the system from a high energy, low multiplicity state (the outcome of the hard process) down to a low energy, high multiplicity state (where hadronisation occurs) by successively radiating additional partons. The shower process is dominated by soft and collinear emissions: soft emission occurs when a virtual particle moves on-shell by radiating a particle with vanishing four momentum,  $k \approx 0$ , and collinear emission occurs when an incoming massless particle,  $i$ , splits into two massless particles,  $j$  and  $k$ , which have parallel

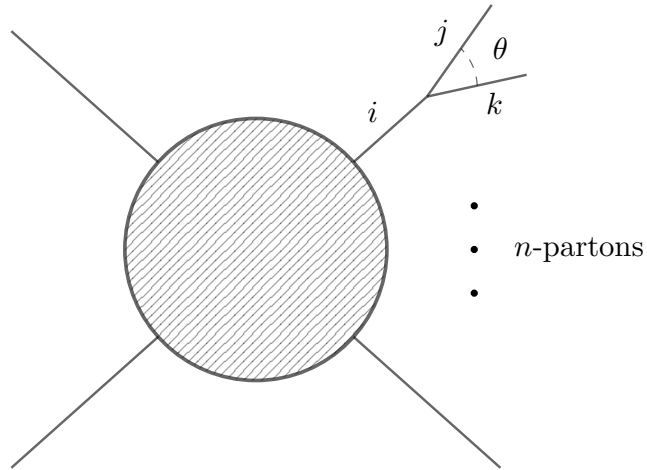


Figure 2.4: A process producing  $n$ -partons in the final state, where one parton,  $i$ , splits to partons  $j$  and  $k$ .

four momenta, such that the total momentum of the original particle,  $P$ , is distributed between the final state particles as

$$p_j = zP, \quad p_k = (1 - z)P. \quad (2.3.1)$$

To illustrate the enhancement of the shower process in the collinear and soft limits, consider a process where radiation occurs from a final state parton, such that the momentum distribution goes like Equation 2.3.1, as shown in Figure 2.4. In the limit that partons  $j$  and  $k$  are collinear, and utilising the factorisation theorem [23–25], the partonic differential cross section for the  $(n+1)$ -parton process is

$$d\hat{\sigma}_{n+1} = d\hat{\sigma}_n \sum_i \int_{k_{\min}}^{k_{\max}} \frac{d|k_{\perp}|^2}{|k_{\perp}|^2} \int_{z_{\min}}^{z_{\max}} \frac{dz}{z} \frac{\alpha_s}{2\pi} P_{ji}(z), \quad (2.3.2)$$

where  $d\hat{\sigma}_n$  is the cross section for the  $n$ -parton process, and thus the  $(n + 1)$ -parton process factorises in the collinear limit [29]. The integration measures  $k_{\perp}$  and  $z$  are the trans-

verse momentum of the parton  $k$  with respect to parton  $j$ , and the fraction of parton  $i$ 's momentum carried away by  $j$  from the splitting,  $i \rightarrow jk$ . The  $P_{ji}(z)$  functions are the *Dokshitzer–Gribov–Lipatov–Altarelli–Parisi* (DGLAP) splitting kernels [51–53], which have the form

$$P_{gq}(z) = C_F \left[ \frac{1 + (1-z)^2}{z} \right], \quad (2.3.3)$$

for the splitting of a quark into a gluon and quark, and

$$P_{gg}(z) = C_A \left[ 2 \frac{1-z}{z} + z(1-z) \right], \quad P_{qg}(z) = n_f T_R \left[ z^2 + (1-z)^2 \right], \quad (2.3.4)$$

for the processes of a gluon splitting to two gluons and a gluon splitting to quark-antiquark pair, respectively [31]. Here, the splitting kernels<sup>‡‡</sup> are multiplied by a diagonal colour-charge factor  $C_{ii}$ , such that  $C_F = (N_c^2 - 1)/(2N_c)$ ,  $C_A = N_c$  and  $T_R = 1/2$ , and for QCD, which has the gauge structure  $SU(3)_c$ , the number of colours is  $N_c = 3$ . Equation 2.3.2 exhibits a collinear divergence as  $k_\perp \rightarrow 0$ , and therefore is regularised by a lower limit on the transverse momentum. The upper limit of the transverse momentum integral depends on the physical energy scale of the given process. The limits on the momentum fraction,  $z$ , depend on the kinematics and resolution of the system [29]. We note that the splitting kernels from Equations 2.3.3 and 2.3.4 exhibit a singularity as the emitted parton becomes soft,  $z \rightarrow 0$ . The splitting kernels are derived in the collinear limit, and therefore describe the hard-collinear contribution for values of  $z > 0$ .

In Equation 2.3.2, we saw that the  $(n + 1)$ -parton process can be factorised in terms of the  $n$ -parton cross section, suggesting that the additional parton  $k$  is independent of the rest of the process. This argument can be iterated in the limit that the emissions are collinear, consequently all subsequent emissions will be mutually independent. Therefore, using collinear factorisation,

---

<sup>‡‡</sup>Note that these splitting kernels are not regularised. For information on the regularised splitting kernels, see References [29, 32, 33].

we can approximate the cross section for a final state with many particles, given a scattering process that gives well separated partons in the final state. In reality, there will be emission which has a non-zero angle separating partons  $j$  and  $k$  in Figure 2.4, however, as we have seen, emission is enhanced in the collinear limit, and therefore collinear factorisation provides a good approximation of the cross section [29, 33].

The cross section is proportional to the square of the amplitude of a specific process, thus the cross section is related to probability. We can therefore interpret the factor

$$d\mathcal{P} = \sum_i \frac{d|k_\perp|^2}{|k_\perp|^2} \int_{z_{\min}}^{z_{\max}} \frac{dz}{z} \frac{\alpha_s}{2\pi} P_{ji}(z), \quad (2.3.5)$$

as the probability for the emission of a parton with transverse momentum  $k_\perp$ , and momentum fraction  $z$ . Note that this is the probability of resolvable emission, and in reality non-resolvable emission is possible, where the final state radiation is not observable due to the resolution of the detector. Preserving unitarity, we can define the probability that no emission occurred in a defined interval as

$$\Delta_i(t_0, t + \delta t) = \Delta_i(t_0, t) \left[ 1 - \sum_j \int_t^{t+\delta t} \frac{dt}{t} \int dz \frac{\alpha_s}{2\pi} P_{ji}(z) \right], \quad (2.3.6)$$

where here we have written the transverse momentum integral in terms of the virtuality<sup>§§</sup> of parton  $i$ ,  $t = p_i^2$ , such that

$$\int \frac{d|k_\perp|^2}{|k_\perp|^2} = \int \frac{dt}{t}. \quad (2.3.7)$$

The quantity  $\Delta_i(t_0, t + \delta t)$  from Equation 2.3.6, therefore, is the probability that no resolvable

---

<sup>§§</sup>Note that the shower may be ordered with respect to other variables that are linear in virtuality. For example, angle ordering is a commonly used approach for including soft emissions, and such approaches are discussed in References [29, 33].

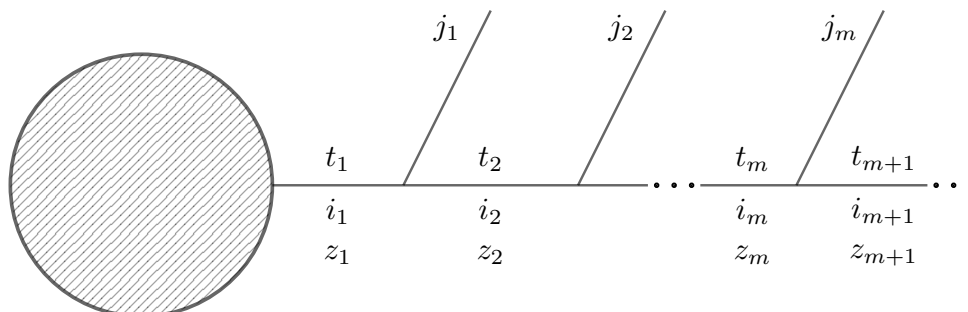


Figure 2.5: A parton of species  $i_1$ , with virtuality  $t_1$  and momentum fraction  $z_1 = 1$ , emerges from the hard process and showers, emitting  $n$  partons of species  $\{j_n\}$ . The shower is categorised by a set of virtualities and momentum fractions,  $(\{t_n\}, \{z_n\})$ , and the emitting parton's species at each step is denoted by  $i_n$ . Figure has been adapted from Reference [29].

emission occurred in the virtuality interval  $t_0$  to  $t + \delta t$ . By making  $\delta t$  small, it is possible to allow for only one emission per virtuality step. Therefore, one can Taylor expand Equation 2.3.6 to solve for  $\Delta_i(t_0, t)$  [29], such that

$$\Delta_i(t_0, t) = \exp \left[ - \sum_j \int_{t_0}^t \frac{dt'}{t'} \int_{z_{\min}}^{z_{\max}} dz \frac{\alpha_s}{2\pi} P_{ji}(z) \right]. \quad (2.3.8)$$

This quantity is the *Sudakov factor*, and describes the probability that parton  $i$  traverses the virtuality interval  $t_0$  to  $t$  without emitting any partons. Note that the Sudakov factor describes the probability that no emission of any kind occurs in the virtuality interval. Although we have only considered resolvable emissions, in fact, the Sudakov factor also sums virtual contributions to all orders. This is a result of preserving unitarity, as the sum of the emission and non-emission probabilities must be unity. Also, via unitarity, the resolvable emission probability gives information on the sum of the virtual and unresolvable real contributions, both of which are divergent on their own, but their sum is finite [33].

Having defined the Sudakov factor and resolvable emission probability, we are now in a

position to define the parton shower. Consider the shower from Figure 2.5, if the parton emerging from the hard process showers, radiating off partons of species  $j_i$ , then before each emission it will have a specific virtuality  $t_i$  and momentum fraction  $z_i$ . We can therefore characterise a parton shower with a set of virtualities and momentum fractions,  $(\{z_i\}, \{t_i\})$ , such that we can write down the probability for the parton shower that emits at each virtuality step  $\delta t$ ,

$$P_{i_1, i_2, \dots, i_n}^{j_1, j_2, \dots, j_n}(\{z_i\}, \{t_i\}) = \left[ \prod_{m=1}^n \Delta_i(t_m, t_{m+1}) \int d\tilde{z}_{m+1} \frac{\alpha_s}{2\pi} P_{i_{m+1} i_m}(\tilde{z}_{m+1}) \right], \quad (2.3.9)$$

where  $\tilde{z}_{m+1} = z_{m+1}/z_m$  is the momentum fraction carried away by the  $(m+1)^{\text{th}}$  parton. The Sudakov factor,  $\Delta_i(t_m, t_{m+1})$ , gives the probability that no resolvable partons have been emitted between each branching, and the integral includes the probability for each resolvable emission with momentum fraction  $\tilde{z}$ .

The Standard Model forbids the existence of free colour-charged particles, and colour-confinement in QCD restricts all long lived particles to exist in colour-neutral states, called hadrons. Consequently, final state particles produced by the parton shower in Equation 2.3.9 will never reach being completely on-shell, and at a specific energy scale,  $\Lambda_{\text{QCD}}$ , these partons hadronise. Monte Carlo approaches to simulating parton showers therefore include a cutoff to the parton cascade, running the algorithm from the hard process to the cutoff energy scale.

### 2.3.2 Monte Carlo Method

The main iterative step in a Monte Carlo parton shower algorithm is: given an initial virtuality and momentum fraction  $(t_i, z_i)$ , generate the values  $(t_{i+1}, z_{i+1})$ , for each step  $i$  in the evolution of the parton shower. Here we outline the Monte Carlo approach to simulating parton showers, summarising References [20, 29, 33].

The shower begins with a parton of virtuality  $t_1$  and momentum fraction  $z_1 = 1$  coming from the hard process. We then wish to generate  $t_2$ , such that the shower evolves through the



virtuality step  $t_1$  to  $t_2$  without emission. Utilising the Sudakov factor,  $t_2$  can be generated by solving

$$\Delta_i(t_2, t_1) = \mathcal{R}_1, \quad (2.3.10)$$

where  $\mathcal{R}_1$  is a random number that is distributed uniformly in  $[0, 1]$ . If  $t_2$  is less than the shower cutoff  $t_0$ , then no further branching occurs and the shower is terminated.

If the generated virtuality  $t_2$  passes the cutoff test, then, after the species of the emitted parton has been chosen, we must generate the momentum fraction  $\tilde{z}_{m+1} = z_{m+1}/z_m$  by solving

$$\int_{\epsilon}^{\tilde{z}_{m+1}} dz' \frac{\alpha_s}{2\pi} P_{i_{m+1}i_m}(z') = \mathcal{R}_2 \int_{\epsilon}^{1-\epsilon} dz' \frac{\alpha_s}{2\pi} P_{i_{m+1}i_m}(z'), \quad (2.3.11)$$

where  $\mathcal{R}_2$  is a second random number uniformly distributed in  $[0, 1]$ ,  $\epsilon$  is the infra-red cutoff for resolvable emission, and the subscripts  $i_m$  and  $i_{m+1}$  denote the emitting parton's species before and after the emission, as per the splitting functions in Equation 2.3.3 and 2.3.4. The integral on the right hand side of Equation 2.3.11 ensures that the splitting probabilities are normalised to unity.

To complete the shower step, the azimuthal angle  $\phi$  for the emitted parton ( $m+1$ ) must be generated by sampling from uniformly distributed angles in  $[0, 2\pi]$ . When a gluon is emitted, the generation of the azimuthal angle must take into account polarisation correlations, and these are discussed in detail in References [20, 33].

The above process is then repeated for all partons, including emitted partons, until no further emissions occur in the parton shower. Once the algorithm is complete, and momentum conservation at each branching has been applied, one obtains a list of four-momenta of the emitted partons and therefore a simulated scattering event [29, 33]. Here, we have considered evolving down in virtuality for a final state parton from a scattering event. In principle, this

algorithm can be used for incoming partons by evolving backwards from the hard process. Such initial-state parton showers are discussed in detail in References [29, 32, 33].

The parton shower algorithm outlined here is entirely in the collinear limit, and will be used in Chapters 4 and 5 to simulate a simple QCD collinear parton shower with  $n_f = 1$  on a quantum computer. This approach has been justified by utilising the fact that the cross section from Equation 2.3.2 is enhanced in the collinear limit. However, it can be shown that emission is also enhanced in the soft limit when the emitted parton is not collinear with any outgoing partons. Such emission is called *wide-angle soft* emission [29]. Due to this enhancement, it is not sufficient to neglect the terms from soft emission, and one must construct a method that allows for the inclusion of soft emission. One technique is the *dipole shower*.

### 2.3.3 The Dipole Shower

One method to augment the collinear parton shower to include the correct soft emission behaviour is to model the decay of a high energy state as a series of successive decays of colour-anticolour dipoles into lower energy dipoles [20, 54, 55]. Interpreting the showering process as a dipole shower addresses momentum conservation by considering only emissions which recoil against a pair of partons, rather than a single parton. Individual on-shell particles cannot decay into on-shell decay products that have non-vanishing angles or energies without violating momentum conservation. However, if we consider the final state  $q\bar{q}$ , where  $q$  and  $\bar{q}$  are both on-shell<sup>¶</sup>, then this dipole will readily decay to three on-shell particles without violating momentum conservation [20]. In the limit that  $N_c \rightarrow \infty$ , the  $q\bar{q}$  dipole decays to form two decay product dipoles. Therefore, the dipole shower model allows for the simulation of realistic parton showers with emissions that are not entirely collinear or soft. Furthermore, the dipole model allows for the straightforward treatment of interference patterns in soft gluon emissions, which

---

<sup>¶</sup>A quark,  $q$ , and antiquark,  $\bar{q}$ , emerging as a final state from a hard process carry opposite and compensating colours, forming a colour dipole, hence the term dipole shower.

is discussed in detail in References [20, 29, 30, 33].

To illustrate the dipole shower, we consider the emission of a (soft) gluon from a colour dipole,  $q\bar{q} \rightarrow qg\bar{q}$ , as described in Reference [3]. The inclusive decay probability for this process is given by the eikonal interference pattern,

$$\begin{aligned} d\mathcal{P}(q(p_I)\bar{p}(p_K) \rightarrow q(p_i)g(p_j)\bar{q}(p_k)) &\simeq \frac{ds_{ij}}{s_{IK}} \frac{ds_{jk}}{s_{IK}} C \frac{\alpha_s}{2\pi} \frac{2s_{ik}}{s_{ij}s_{jk}} \frac{d\phi}{2\pi}, \\ &\simeq \frac{ds_{ij}}{s_{IK}} \frac{ds_{jk}}{s_{IK}} C \frac{\alpha_s}{2\pi} \frac{2s_{IK}}{s_{ij}s_{jk}}, \end{aligned} \quad (2.3.12)$$

where the dipole invariant mass is given by  $s_{ab} = 2p_a p_b$ ,  $C$  is a colour-charge factor, and momentum conservation  $p_I + p_K = p_i + p_j + p_k$  implies the relation  $s_{IK} = s_{ik} + s_{ij} + s_{jk}$  between pre- and post-branching momenta. This decay can therefore be interpreted as the decay of a dipole with invariant mass  $s_{IK}$  into two dipoles of invariant masses  $s_{ij}$  and  $s_{jk}$ , which are linked by a soft gluon. Using the inclusive probability, the whole parton shower can therefore be written in terms of a generating functional  $\mathcal{F}_n$  [56], where we define  $\mathcal{F}_n$  in the recursive way,

$$\begin{aligned} \mathcal{F}_n(\Phi_n, t_0, t_n; O) &= \Delta(t_0, t_n) O(\Phi_n) \\ &+ \int_{t_0}^{t_n} dt d\xi \frac{d\phi}{2\pi} C \frac{\alpha_s}{2\pi} \frac{2s_{ik}(t, \xi)}{s_{ij}(t, \xi)s_{jk}(t, \xi)} \Delta(t, t_n) \mathcal{F}_n(\Phi_{n+1}, t_c, t; O), \end{aligned} \quad (2.3.13)$$

describing the contribution of exclusive, fixed particle multiplicity states  $\Phi$  to an observable  $O(\Phi)$ . Here we can rewrite the non-emission probability from Equation 2.3.8 in terms of the dipole invariant masses,

$$\Delta(t_0, t) = \exp \left( - \int_{t_0}^t dt d\xi \frac{d\phi}{2\pi} C \frac{\alpha_s}{2\pi} \frac{2s_{ik}(t, \xi)}{s_{ij}(t, \xi)s_{jk}(t, \xi)} \right). \quad (2.3.14)$$

The particular choice of  $t$  and  $\xi$  is the *phase space parameterisation* or *momentum mapping*, where  $t$  is the evolution variable of the parton shower, as before. In current, classical implementations of the parton shower,  $t$  and  $\xi$  are typically treated as continuous variables.

The first term in Equation 2.3.13 corresponds to the probability that the  $n$ -parton state does not emit any further partons above the virtuality cutoff,  $t_0 \approx \Lambda_{\text{QCD}}^2$ . The second term describes the probability that one or more emission occurred and is related to Equation 2.3.9. The approach to calculating the parton shower therefore depends on the choice of inclusive decay probabilities, the choice of evolution variable, and the momentum mapping between pre- and post-decay momenta,  $s_{ij}, s_{jk} \leftrightarrow t, \xi$ .

The dipole shower that is dominated by the emission of a soft gluon, described here, has been used in Chapter 6 for the simulation of the first realistic parton shower on a quantum device, producing synthetic data comparable to real data from experiments on the Large Electron-Positron collider. The approach uses a method of discretising the phase space available to subsequent gluon emissions, allowing for the simulation of the parton shower on a real quantum device on the `ibm_cloud`.

## Chapter 3

# Introduction to Quantum Computing

Alluding to the quote by Richard Feynman at the start of this thesis: Nature is quantum, therefore, if we want to simulate its intricacies and phenomena accurately, then we are going to have to do this using a quantum device, which contains all of the peculiarities of quantum mechanics inherently. Feynman made this observation in the early 1980s [4], and since then there has been great progress and success in Quantum Information Science (QIS); recently culminating in Peter W. Shor [57] and David Deutsch [58] being awarded the 2023 Breakthrough Prize in Fundamental Physics for their foundational work in the field of QIS, and the 2022 Nobel Prize in Physics being awarded to Alain Aspect, John F. Clauser and Anton Zeilinger for their pioneering work in QIS, conducting experiments establishing the violation of Bell inequalities in the Einstein-Podolsky-Rosen paradox [59–63].

Currently, there are several shortfalls in our ability to simulate high energy physics processes using conventional techniques. Despite impressive improvements in modern computational methods such as Lattice QCD, it is expected that classical devices, even highly advanced digital devices, will not be able to simulate the real time behaviour of particles in high energy collisions [64]. Quantum computers, with their inherent ability to retain the quantum nature of the system, offer an exciting window into the study of highly entangled hadronic matter.

Although a full simulation of a Quantum Field Theory (QFT) is currently over the horizon, even in its infancy quantum computing can provide a unique insight into quantum systems in a regime that has never been accessible before. In this Chapter a brief introduction to quantum computation is given, building up to the Quantum Walk framework in Section 3.2, which is used in Chapters 4 and 6 for the simulation of a QCD parton shower on a quantum device. In this thesis, we will concentrate on controllable superconducting qubit based devices, however promising alternative quantum computation methods are becoming available, such as photonic devices. An in depth review of quantum computation is available in References [65–68] and references therein. Appendix A provides a short guide on how to read quantum circuit diagrams, and includes a glossary of commonly used quantum gate operations. Supplementary material on quantum measurements and quantum errors is given in Appendices B and C.

## 3.1 Quantum Computation

### 3.1.1 The Qubit

Quantum computation is founded on the quantum bit, the *qubit*: the quantum analogue of the classical *bit*. The bit is a binary system which obeys Boolean logic and therefore can take on the value of 0 or 1. A collection of bits can then hold information, such as a number, in the binary basis. This information can be extracted at any point in time on a classical device by examining the exact state of each of the bits at that time. In contrast to the classical bit, the qubit is not restricted to a binary basis. Instead, the qubit is a two body quantum system, which in general can be represented in the computational basis such that the qubit state  $|\psi\rangle$  is a linear superposition of the two basis states  $|0\rangle$  and  $|1\rangle$ ,

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle, \quad (3.1.1)$$

where  $\alpha$  and  $\beta$  are complex amplitudes of the states  $|0\rangle$  and  $|1\rangle$  respectively. The qubit cannot be easily examined like a classical bit to retrieve  $\alpha$  and  $\beta$ . Instead, one must measure the qubit, projecting the state  $|\psi\rangle$  onto a classical bit, such that one obtains the 0 result with probability  $|\alpha|^2$  and the 1 result with probability  $|\beta|^2$ . The qubit state  $|\psi\rangle$  can be manipulated by single qubit gate operations to alter the outcome of the measurement. It will be shown that a series of gate operations can be applied to a set of qubits to create quantum algorithms.

To preserve unitarity, the complex amplitudes  $\alpha$  and  $\beta$  must obey  $|\alpha|^2 + |\beta|^2 = 1$ . Consequently, Equation 3.1.1 can be rewritten as

$$|\psi\rangle = e^{i\lambda} \left( \cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle \right), \quad (3.1.2)$$

where  $\theta$ ,  $\varphi$  and  $\lambda$  are real numbers\*. Only the relative phase between the two coefficients of the basis states has a physical meaning, and therefore the overall phase  $e^{i\lambda}$  has no observable effect and can thus be omitted to give

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle, \quad (3.1.3)$$

where  $0 \leq \theta \leq \pi$  and  $0 \leq \varphi < 2\pi$  now define a point on a three-dimensional unit sphere: the Bloch sphere. The Bloch sphere, shown in Figure 3.1, is a geometric representation of a single qubit only, and no generalisation to multiple qubits is known. Nevertheless, the Bloch sphere is a useful visualisation of qubit states and how they behave under single qubit gate operations [65].

Interpreting the qubit state  $|\psi\rangle$  in the Bloch sphere representation elucidates the potential information that can be held by the qubit. There are infinitely many points that exist on the Bloch sphere, and thus infinitely many states that the qubit can be in. However, in order to

---

\*See the definition of the  $U_3$  operation in Appendix A. The values of  $\theta$ ,  $\phi$  and  $\lambda$  correspond directly with Equation 3.1.2.

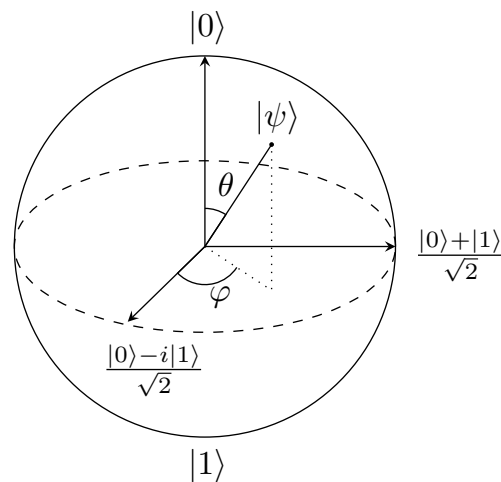


Figure 3.1: Bloch sphere representation of a qubit

retrieve information from the qubit, the qubit must be measured, collapsing the superposition to obtain either a 0 or 1 result and leaving the qubit in the  $|0\rangle$  or  $|1\rangle$  state respectively. If the qubit state  $|\psi\rangle$  was measured once, then one bit of information about the state of the qubit would be obtained. One can only fully determine  $\alpha$  and  $\beta$  from Equation 3.1.1 if infinitely many qubits, each in the identical state  $|\psi\rangle$ , were measured. Consequently, the qubit contains a lot of information which is practically hidden. How to utilise this hidden information is the key question in quantum algorithmic design. The true power of this hidden information is realised as the number of qubits increases.

### 3.1.2 Multiple Qubits

The state space of a quantum computer grows exponentially with the number of qubits, such that an  $n$ -qubit system has  $2^n$  basis states in the computational basis. These basis states take the form  $|q_1q_2, \dots, q_n\rangle$ , where  $q_i \in \{0, 1\}$ . For example, consider a two qubit system: there are four basis states, such that a general state of the system can be written as



$$|\psi\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle, \quad (3.1.4)$$

where  $\alpha_{ij}$  are complex amplitudes of the state  $|ij\rangle$ , for  $i, j \in \{0, 1\}$  [65]. As for the single qubit case, unitarity is preserved such that  $\sum_{i,j \in \{0,1\}} |\alpha_{ij}|^2 = 1$ . With multiple qubit systems, measurements can be carried out in several ways. One can measure the entire system or individual subsets of qubits. Different measurements have different effects on the state of the system after measurement, and can lead to interesting results. Quantum measurement is discussed in detail in Appendix B.

To illustrate the power of quantum states with multiple qubits, here we can construct the important two qubit state,

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \quad (3.1.5)$$

and consider the famous thought experiment outlined by Albert Einstein, Boris Podolsky and Nathan Rosen (EPR) [69]. Consider the scenario where the entangled state from Equation 3.1.5 is prepared, and then the first and second qubit are separated to positions A and B respectively, such that they are an arbitrary distance from each other. A measurement on the first qubit, at position A, would yield the 0 result with 50% probability, or the 1 result with 50% probability. If a measurement was subsequently performed on the second qubit, at position B, the outcome of the measurement will always be the same as the result from position A. Seemingly, information is being passed from one qubit to another instantaneously, violating relativity. It was therefore postulated that quantum mechanics was not a complete theory, and local hidden variables must be introduced to restore the causality and locality of the theory [69].

In response to the EPR thought experiment, John Bell proposed a test to determine whether there could be a local hidden variable description of quantum mechanics by applying a constraint

on the correlation between two measurements using the *Bell inequality* [70]. Recently, the 2022 Nobel Prize in Physics was awarded for the experimental proof that the EPR scenario violates the Bell inequality [59–63], confirming that there is no local hidden variable explanation of quantum mechanics. As a result, quantum computers have the ability to access highly entangled states which can be used for quantum algorithmic design. This result is a milestone in the formulation of quantum mechanics, and as a result the state from Equation 3.1.5 is often called a *Bell state*. In fact, the Bell state is at the heart of many quantum algorithms, including quantum teleportation and superdense coding.

### 3.1.3 Quantum Algorithms and Advantage

Quantum algorithms are constructed from unitary operations, *gates*, applied to a series of qubits. The qubits can be arranged into subsets called *registers*, and registers can be used to separate encoded information on the quantum device. As an example of a quantum algorithm, consider encoding the Bell state from Equation 3.1.5 onto a two qubit quantum device. The algorithm is constructed from two gates in series: a Hadamard gate applied to the first qubit, and a CNOT gate controlled from the first qubit and applied to the second qubit. Analytically, the effect of these gates on an initial state of  $|00\rangle$  is

$$\begin{aligned} |00\rangle &\xrightarrow{H} \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle), \\ &\xrightarrow{\text{CNOT}} \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \end{aligned} \quad (3.1.6)$$

This algorithm can be schematically illustrated using a quantum circuit, as shown in Figure 3.2, where the two qubits are represented by horizontal lines, and quantum gate operations are represented by boxes, controls and targets on the qubits. Appendix A gives an introduction to reading circuit diagrams. To retrieve information from the device, a measurement must be

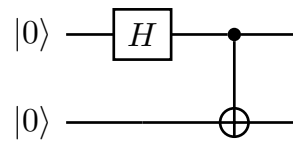


Figure 3.2: Schematic of the quantum circuit for encoding the Bell state on a quantum device.

performed, projecting onto a classical basis.

Although the encoding of the Bell state is a simple example, it demonstrates that a quantum device has the ability to access highly entangled states. Quantum computing consequently offers a potential paradigm shift in computational science, allowing for calculations to be performed in a regime that has never been accessible before. The potential of a quantum advantage over classical approaches is a driving force in algorithmic design, with examples of quantum advantage already being identified [57, 58]. One of the most famous examples is the Grover algorithm [71]. Based on Quantum Amplitude Amplification, the Grover algorithm is an optimal quantum algorithm [67] which can achieve a polynomial speedup for database search algorithms. In comparison to the classical case, which requires  $\mathcal{O}(N)$  queries of the database, the Grover algorithm requires  $\mathcal{O}(\sqrt{N})$  queries to successfully identify the marked state. A detailed overview of the Grover algorithm can be found in References [65, 67, 68, 71]. The Grover algorithm is a staple of quantum computing architecture, and can be formulated in many ways. Of particular interest for this thesis, the Grover algorithm can be formulated as an oracle-based quantum walk, and can be implemented on current quantum hardware and simulators.

## 3.2 The Quantum Walk

The quantum walk was first developed as the quantum analogue of the classical random walk, which describes the movement of a *walker* through some *position space*,  $\mathcal{H}_P$ . The classical random walk is based on *flipping* a coin such that the subsequent movement of the *walker* is determined by the outcome of the coin flip. The evolution of the quantum walk is fundamentally

different due to the unitary evolution of the isolated quantum system. In contrast to the random motion of the classical random walk, the evolution of the quantum walk is entirely deterministic<sup>†</sup> [72]. As a consequence, the term *quantum random walk* implies intermediate measurement. In this thesis, the quantum walks discussed do not involve intermediate measurements or decoherence effects during the running of the algorithm.

Quantum walks are a powerful technique which have been shown to achieve quantum advantage for certain cases such as search algorithms [73, 74] and simulated annealing [75, 76]. These algorithms are based on Markov Chain Monte Carlo (MCMC) algorithms. In classical computation, MCMC algorithms are the heart of modern computational methods<sup>‡</sup>. The speed of these algorithms is based on the so-called Markov chain *mixing time*, the time it takes for the algorithm to approach the equilibrium distribution. An interesting example of this quantum advantage is the quantisation of ergodic Markov chains<sup>§</sup> using Szegedy quantisation [73], which has been proven to achieve a quadratic speed up using the quantum walk model [77, 78].

With their potential quantum advantage and simple implementation, the quantum walk is a useful tool for quantum algorithmic design. Furthermore, their circuit architecture is compact, making the quantum walk model well suited to NISQ devices. In this section, we review the coined quantum walk in discrete time and quantum walks with memory. A detailed review of quantum walks is given in References [72, 79] and quantum walks with memory in Reference [80].

### 3.2.1 Discrete-Time Coined Quantum Walk

One can construct a classical random walk on a line such that the movement of the walker is dictated by a coin. If the coin lands on *heads* the walker moves in one direction, and if the coin lands on *tails* the walker moves in the opposite direction. The walk then proceeds by iteratively

---

<sup>†</sup>When a measurement is performed, the wavefunction collapses in a non-deterministic way and the outcome is probabilistic.

<sup>‡</sup>Though extensively used, it should be noted that classical MCMC algorithms are not optimal in general.

<sup>§</sup>A Markov chain is said to be ergodic if it is aperiodic and irreducible, consequently the Markov chain has a unique equilibrium distribution

flipping the coin and then moving the walker accordingly until the number of desired steps is reached. This type of random walk is called a coined walk. A detailed discussion on classical random walks is given in Reference [72]. Here, we consider the quantum analogue, the so-called *discrete-time coined quantum walk* [81–83].

The evolution of the coined quantum walk in discrete-time describes the movement of a walker in a Hilbert space,  $\mathcal{H}$ , comprising a *position* space,  $\mathcal{H}_P$ , augmented by a *coin* space,  $\mathcal{H}_C$ , such that

$$\mathcal{H} = \mathcal{H}_C \otimes \mathcal{H}_P. \quad (3.2.1)$$

It will become clear that, in analogy to the coined classical random walk, it is this additional coin space which defines the diffusion characteristics of the discrete-time quantum walk. The quantum walk proceeds in individual time steps constructed from two unitary operations. The first is the coin operation,  $C$ , which encodes the coin state in  $\mathcal{H}_C$  and is analogous to *flipping* a coin in the classical case. The subsequent shift operation,  $S$ , then moves the walker in  $\mathcal{H}_P$  depending on the state of the coin register, the outcome of the coin flip in the classical example. Combining the two operations, one can define a unitary operation which describes a single step of the quantum walk,

$$U = S \cdot (C \otimes \mathbb{I}), \quad (3.2.2)$$

where  $\mathbb{I}$  is the identity in  $\mathcal{H}_P$ . Applying this unitary operation iteratively without intermediate measurement describes the evolutionary dynamics of the quantum walk, such that an  $N$ -step quantum walk is described by  $U^N$ . It should be noted that, in the quantum case, from now on we describe the evolution of a set of states on a finite Hilbert space under unitary transformations, therefore the walker is only an intuitive representation of the diffusion characteristics within the

quantum walk [79].

To illustrate the diffusive characteristics of the quantum walk, we consider the simple example of a discrete-time coined quantum walk on a line<sup>¶</sup>. The position space  $\mathcal{H}_P$  is spanned by integer values on a one-dimensional lattice, such that

$$\mathcal{H}_P = \{|i\rangle : i \in \mathbb{Z}\}, \quad (3.2.3)$$

where the basis states  $|i\rangle$  correspond to the walker being located at position  $i$ . The walker can move along the number line in both the left and right directions, therefore the coin space  $\mathcal{H}_C$  takes the form

$$\mathcal{H}_C = \{|0\rangle, |1\rangle\}. \quad (3.2.4)$$

The coin register in this case can therefore be constructed from a single qubit. We require the shift operation  $S$  to be a unitary operation on the computational basis, controlling from the state of the coin. Therefore, we can define  $S$  as

$$S = |0\rangle\langle 0| \otimes \sum_{i=-\infty}^{\infty} |i-1\rangle\langle i| + |1\rangle\langle 1| \otimes \sum_{i=-\infty}^{\infty} |i+1\rangle\langle i|, \quad (3.2.5)$$

such that, if the coin register is in the  $|0\rangle$  state, the walker will move to the left, and if the coin register is in the  $|1\rangle$  state, the walker will move to the right. By construction, the walker moves position every step and cannot remain stationary<sup>‡</sup>. A schematic of a single step of the

---

<sup>¶</sup>It is easy to extend the quantum walk to higher dimensions by increasing the size of the Hilbert space available to the walker. See Chapters 5 and 6 for examples of higher dimensional quantum walks, used for the simulation of parton showers in high energy collisions.

<sup>‡</sup>This is not a general requirement for a quantum walk. An example of a so-called *lackadaisical* quantum walk, where the walker can remain in the same position, is provided in Reference [84].

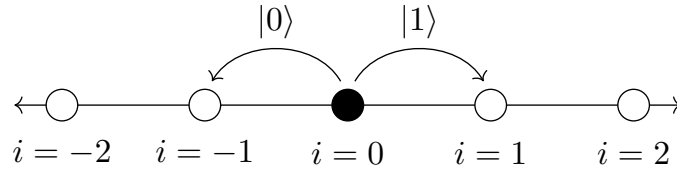


Figure 3.3: Schematic of discrete-time coined quantum walk on a line. The walker at position  $i = 0$  can move left or right depending on the coin state  $|0\rangle$  or  $|1\rangle$  respectively. Figure modified from Reference [2].

discrete-time coined quantum walk on a line is shown in Figure 3.3.

In order to elucidate the difference between the coined classical random walk and the discrete-time coined quantum walk, it is useful to consider the dynamics of the classical case first. As described above, the evolution of the classical case is defined by the outcome of a classical coin, which gives either *heads* or *tails* with certainty at each coin flip. Following the framework of the quantum walk on a line, here we equate *heads* with the  $|0\rangle$  state and *tails* with the  $|1\rangle$  state, such that the walker moves left and right for *heads* and *tails* respectively. Here we assume that an equal coin is used, and therefore the walker will move to the left 50% of the time, and to the right 50% of the time. Consequently, after  $N$  steps, one would expect to find the walker close to where it began the walk\*\*.

In contrast, the discrete-time quantum walk is based on a *quantum coin*. A commonly used quantum coin is the Hadamard coin, which puts a qubit in an equal superposition of the computational basis, such that, when applied to the coin qubit for the one-dimensional quantum walk, the coin state becomes

$$C|c\rangle = H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad (3.2.6)$$

with the coin qubit initialised in the ground state,  $|c\rangle = |0\rangle$ . The shift operation  $S$  from Equation 3.2.5 is then applied and the walker moves into a superposition of the position states,

---

\*\*This is a well known result, and the coined classical random walk can be modelled well with a Galton board.

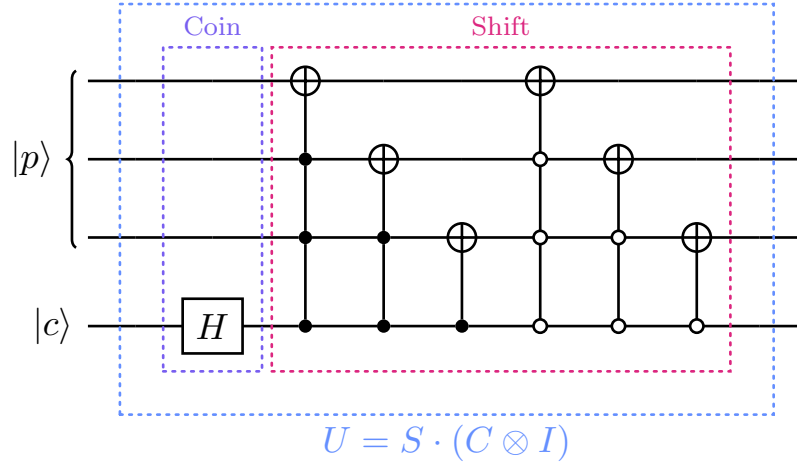


Figure 3.4: Quantum circuit diagram for the discrete-time quantum walk on a line with a Hadamard coin.

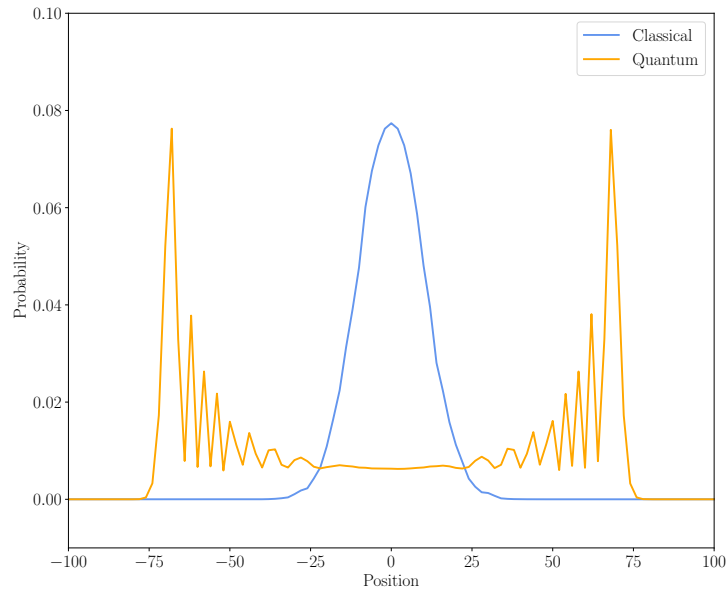
such that the state of the system after the coin and shift operations is

$$U |0\rangle \otimes |0\rangle = \frac{1}{\sqrt{2}} \left( |0\rangle \otimes |i = -1\rangle + |1\rangle \otimes |i = +1\rangle \right). \quad (3.2.7)$$

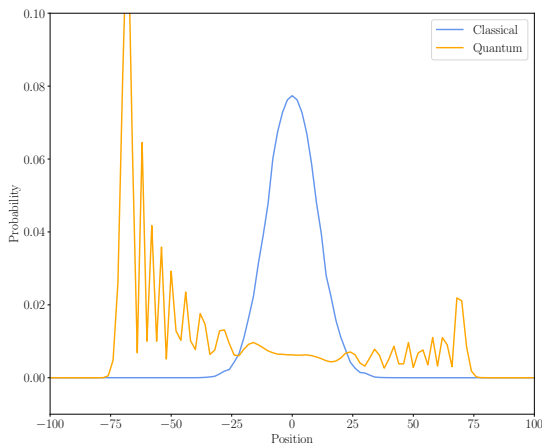
After the operation, the state is in a superposition of positions  $i = -1$  and  $i = 1$ . One can now see that the coin introduces the dynamics of the walker by constructing a superposition on the coin qubit, causing a superposition of positions in the position space [72]. If a measurement is carried out after the step, the walker would be identified in either the  $|i = -1\rangle$  or  $|i = +1\rangle$  state with equal probability, regaining the classical distribution. This process can be repeated, performing a measurement after every step. In this case, the classical distribution is retrieved from the quantum walk. However, if a measurement is not performed after each step, the correlations between different position states will propagate and cause interference. Figure 3.4 shows a schematic circuit diagram for the quantum walk algorithm.

Figure 3.5 shows a simulation of an  $N = 100$  step discrete-time coined quantum walk, compared to an  $N = 100$  step discrete coined classical random walk. For  $N = 100$  steps, every odd position will have a zero probability, by construction. For clarity, the zero-valued

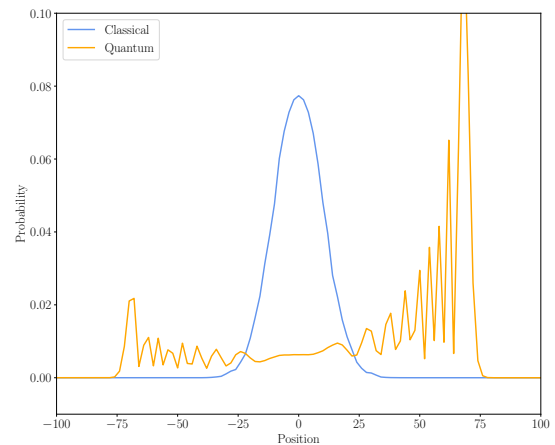




(a)



(b)



(c)

Figure 3.5: Simulation of a 100-step discrete-time quantum walk for a Hadamard coin with the coin state initialised in (a) a symmetric initial state  $|c\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$ , (b) the ground state  $|c\rangle = |0\rangle$ , and (c) the excited state  $|c\rangle = -|1\rangle$ , compared to a 100-step discrete classical random walk. The simulations have been run for  $10^6$  shots on the 32-qubit `ibmq_qasm_simulator`, and the classical distribution has been achieved by measuring the system at each step. Note that only non-zero values are shown. Figure (a) has been adapted from Reference [2].

probabilities have been removed from the distributions. As expected from before, the classical distribution approaches a Gaussian centred around the initial position of the walker (the origin in this case), with variance  $\sigma^2 = N$ . In stark contrast, the quantum walk distributions demonstrate visible interference patterns introduced by the correlations between the walker's positions. It is clear that the coin initialisation is a key aspect of the quantum walk algorithm, and directly affects the diffusive characteristics of the walk.

The quantum walk from Figure 3.5b initialises the coin in the  $|0\rangle$  state, therefore the coin operation is defined by Equation 3.2.6. The distribution is asymmetric and is shifted to the left, such that the walker is more likely to be measured in the negative  $|i\rangle$  region. Initially, the cause of this asymmetry is not obvious, as Equation 3.2.7 shows that the state after one step of the quantum walk is equivalent to the classical walk, after a measurement is performed. However, in the subsequent steps, the coin operation will also be applied to the  $|c\rangle = |1\rangle$  state. Applying the Hadamard coin to the  $|1\rangle$  state, one obtains

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \quad (3.2.8)$$

From this we can now see that a negative phase has been introduced on the  $|1\rangle$  state after the coin operation. Consequently, there will be more cancellations when the coin state is in the  $|1\rangle$  position, and thus the walker is moving to the right, than in the  $|0\rangle$  position, when the walker is moving to the left. As a result, we see constructive interference on the left side of the distribution, and destructive on the right side. This becomes clear in the analytical calculation of the first three steps of the quantum walk,

$$\begin{aligned}
|c\rangle \otimes |i\rangle = |0\rangle \otimes |0\rangle &\xrightarrow{U} \frac{1}{\sqrt{2}} (|0\rangle \otimes |-1\rangle + |1\rangle \otimes |1\rangle), \\
&\xrightarrow{U} \frac{1}{2} (|0\rangle \otimes |-2\rangle + (|0\rangle + |1\rangle) \otimes |0\rangle - |1\rangle \otimes |2\rangle), \\
&\xrightarrow{U} \frac{1}{2\sqrt{2}} (|0\rangle \otimes |-3\rangle + (2|0\rangle + |1\rangle) \otimes |-1\rangle \\
&\quad - |0\rangle \otimes |1\rangle + |1\rangle \otimes |3\rangle). \tag{3.2.9}
\end{aligned}$$

Figure 3.5c shows the distribution for the opposite coin initialisation,  $|c\rangle = -|1\rangle$ . Here there will be more negative phase states than positive phase states. Consequently, the right moving walkers interfere constructively and the left moving walkers interfere destructively, leading to the asymmetric distribution shifted to the right, the mirror image of Figure 3.5b. One can confirm this by calculating the first three steps of the quantum walk,

$$\begin{aligned}
|c\rangle \otimes |i\rangle = -|1\rangle \otimes |0\rangle &\xrightarrow{U} \frac{1}{\sqrt{2}} (-|0\rangle \otimes |-1\rangle + |1\rangle \otimes |1\rangle), \\
&\xrightarrow{U} \frac{1}{2} (-|0\rangle \otimes |-2\rangle + (|0\rangle - |1\rangle) \otimes |0\rangle - |1\rangle \otimes |2\rangle), \\
&\xrightarrow{U} \frac{1}{2\sqrt{2}} (-|0\rangle \otimes |-3\rangle - |1\rangle \otimes |-1\rangle \\
&\quad + (-|0\rangle + 2|1\rangle) \otimes |1\rangle + |1\rangle \otimes |3\rangle). \tag{3.2.10}
\end{aligned}$$

For both the quantum walks in Figures 3.5b and 3.5c, the walker appears to move away from the initial position, such that the distribution has a large peak at  $|N/\sqrt{2}|$  and a small probability at the origin. Such a distribution suggests that the walker's motion is *ballistic* [72], *id est* uniform motion away from the origin. Whether the walker's motion is ballistic or not can be determined in the symmetric case.

It is possible, and often desired, to remove the asymmetry of the Hadamard coin quantum

walk to obtain a symmetric distribution of the quantum walk. To do this, one must combine the two asymmetric walks in Figures 3.5b and 3.5c. This can be done by initialising the coin in a state such that the final state of the walker is a superposition of the two asymmetric quantum walks. We can define

$$|c\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle), \quad (3.2.11)$$

where the complex phase  $-i$  has been applied to the  $|1\rangle$  state [72, 79, 85]. As the Hadamard gate does not introduce any complex amplitudes, applying the coin to this state will result in the quantum walk from Figure 3.5b propagating in the real amplitudes, and the walk from Figure 3.5c propagating in the imaginary amplitudes. Therefore, the two asymmetric quantum walks will not interfere with each other. At the end of the algorithm, the position of the walker is measured, which adds the two asymmetric distributions. The result is shown in Figure 3.5a. The distribution is spread in the range  $[-N/\sqrt{2}, N/\sqrt{2}]$ , with high probability peaks at  $|N/\sqrt{2}|$ , and a uniform distribution around the origin, with small probability. It can be shown that the standard deviation of the discrete-time coined quantum walk on a line is approximately  $\sigma^2 \approx N^2$ , up to a numerical factor<sup>††</sup> [82, 85, 86]. We see that the quantum walk propagates quadratically faster through the position space than the classical Gaussian distribution, which has a standard deviation  $\sigma^2 = N$ .

Numerical calculations from Reference [72] show that the standard deviation of the discrete-time quantum walk on a line, with a Hadamard coin, is approximately  $\sigma = 0.54N$ . Returning to the discussion of ballistic motion, by comparing this result to the ballistic case,  $\sigma = N$ , we see that the Hadamard coined quantum walk's motion is ballistic, but runs at almost half the speed of a free particle [72]. However, in the symmetric case, the quantum walker has the ability to be

---

<sup>††</sup>A detailed analytical derivation of the standard deviation for a Hadamard coined quantum walk on an infinite line is provided in Chapter 5 of Reference [72].

measured in either the negative or positive directions with a large probability. This is strictly not possible in the classical case.

### 3.2.2 Quantum Walks with Memory

An interesting extension of the quantum walk is the quantum walk with memory. Analogous to classical random walks with memory, the quantum walk with memory allows for the simulation of arbitrary dynamics by augmenting the Hilbert space of the coined, discrete-time case with an additional *memory* space, such that the walk now exists in

$$\mathcal{H} = \mathcal{H}_C \otimes \mathcal{H}_P \otimes \mathcal{H}_M. \quad (3.2.12)$$

This additional memory space allows for the diffusive characteristics of the quantum walker to be modified depending on “memory” of the previous walker positions [87] and coin operations [80, 88]. It should be noted that the term memory here relates to information available to the quantum walk within the unitary evolution of the algorithm. This information cannot be accessed at a later time outside of the algorithm, or preloaded onto the device. Therefore, when referring to memory, we are not referring to the concept of qRAM [89].

The amount of memory the walker has affects the diffusive characteristics of the walk. If the walker has no memory, the quantum distribution from Figure 3.5 is retained, *id est* the ideal quantum limit. If the walker has full memory, *id est* the walker has information about all the previous positions or coin operations, then the classical distribution is obtained. One could make the connection between obtaining the classical distribution and performing a measurement at each step, however this is not the case. Importantly, in the quantum walk with memory, the evolution is entirely unitary, and no decoherence effects, including measurements, are present. Moreover, one can prove this by reversing time to regain the initial state, something that is not possible for a genuinely classical system. This has been shown, and is discussed in more detail,

in References [80, 88].

### 3.2.3 Quantum Walks for Simulation

Discrete-time quantum walks provide an interesting framework for quantum circuit design. The model is compact, with the circuit depth of the algorithm growing linearly with the number of steps in the walk. The quantum walk architecture is therefore well suited to designing algorithms for NISQ devices, which require shallow circuit depths to limit noise on the device. Furthermore, as illustrated in Section 3.2, quantum walks provide a potential quadratic speedup over classical random walks. Consequently, quantum walks offer an exciting opportunity for high energy physics, where computation time is becoming a limiting factor for simulations.

In this thesis, we present three novel quantum algorithms for the simulation of the parton shower process as a first step towards a fully quantum simulation of high energy collision events. It will be shown in Chapters 5 and 6 that the compact architecture of the quantum walk framework can be used to efficiently encode the QCD parton shower on a quantum device. Furthermore, these algorithms exploit the ability of the quantum walk with memory to simulate arbitrary dynamics through the population of a memory Hilbert space,  $\mathcal{H}_M$ , to successfully generate realistic parton showers on a quantum device [2, 3].

## Chapter 4

# Quantum Monte Carlo Parton Shower

The collision of hadrons at high energy colliders can result in the production of high multiplicity final states, the theoretical description of which can become highly complex. As outlined in Section 2.3, the calculation of multi-particle processes in QCD can be factorised such that short and long distant effects can be separated, allowing for the individual treatment of the perturbative and non-perturbative parts of the calculation [23–25]. A collision event of two protons can therefore be split into four main parts: the determination of the parton content of the proton using the parton density functions (PDFs), the calculation of the hard process, the parton shower, and hadronisation. Of the four main parts, the parton shower is entirely perturbative and, along with the hard process and the phase space integral, presents a bottleneck in the production of synthetic data for collider experiments.

As we move to an era of higher energy and higher luminosity collider experiments, new and innovative techniques must be developed to continue to produce accurate and efficient simulated collider data, which is crucial for the discovery of new physics. Quantum computing offers a powerful tool to combat this computational challenge, and has already demonstrated

an advantage over classical computational methods under certain conditions [57, 58, 71]. In this chapter, we present the first implementation of a QCD parton shower algorithm on a quantum device, summarising the findings of Reference [1]. The algorithm constructs a collinear QCD model with one gluon and one quark. The theoretical framework of the simple QCD model is outlined in Section 4.1. The quantum implementation of a general and extendable parton shower algorithm is outlined in Section 4.2, and the results from the simulation of two shower steps on the `ibmq_qasm_simulator` device are presented in Section 4.3.

## 4.1 Collinear Parton Shower on a Quantum Device

The simulation of parton showers with conventional classical methods can quickly become highly complex when describing collision events at high energy hadron colliders, such as the Large Hadron Collider at CERN. Event generators are crucial to the study of physics processes in high energy physics experiments, thus new methods are required to improve the accuracy and efficiency of parton shower algorithms, and to cope with the growing computational challenge presented by higher energy and higher luminosity colliders. In this section, we present the theoretical framework for a simple, collinear QCD model based on the Monte Carlo approach from Section 2.3.2. This simple framework allows for the implementation of a proof-of-principle shower algorithm on a quantum device, outlined in Section 4.2.

The algorithm has been designed to run on the `ibmq_qasm_simulator`, available on the `ibm_cloud`, which has the ability to simulate 32 qubits. To meet the qubit limitation, we consider a toy model of QCD consisting of one quark flavour and one gluon. We restrict the model to massless particles and collinear splittings only, where the DGLAP splitting kernels from Equations 2.3.3 and 2.3.4 have been used. Taking  $n_f = 1$  and  $N_c = 3$ , the splitting kernels become



$$P_{gq}(z) = \frac{4}{3} \left[ \frac{1 + (1-z)^2}{z} \right], \quad (4.1.1)$$

$$P_{gg}(z) = 3 \left[ 2 \frac{1-z}{z} + z(1-z) \right], \quad (4.1.2)$$

$$P_{qq}(z) = \frac{1}{2} \left[ z^2 + (1-z)^2 \right], \quad (4.1.3)$$

where  $z$  is the fraction of parton  $i$ 's momentum carried away from the splitting  $i \rightarrow jk$  by parton  $j$ . As a further simplification, we discretise the shower into individual shower steps and only evolve the variable  $z$ , removing the need to keep track of individual particle kinematics. We can therefore write the Sudakov factor from Equation 2.3.8 in the simplified form

$$\Delta_j(z_1, z_2) = \exp \left[ -\alpha_s^2 \int_{z_1}^{z_2} dz P_{ji}(z) \right], \quad (4.1.4)$$

such that the total non-emission probability at step  $N$  is defined as

$$\Delta_{\text{tot}}(z_1, z_2) = \Delta_g^{n_g}(z_1, z_2) \Delta_q^{n_q}(z_1, z_2) \Delta_{\bar{q}}^{n_{\bar{q}}}(z_1, z_2), \quad (4.1.5)$$

where  $n_g$ ,  $n_q$  and  $n_{\bar{q}}$  are the number of gluons, quarks, and antiquarks present in the shower at step  $N$  respectively. The parton shower allows for only one emission per shower step, thus  $(n_g + n_q + n_{\bar{q}}) \leq N$ . To simplify the shower further we neglect the running of the strong coupling and set  $\alpha_s = 1$ .

Current state of the art event generators aim to solve the showering process numerically, treating the evolution variable as a continuous degree of freedom. This approach is unsuitable for (current) quantum devices, which are intrinsically discretised into qubits. Therefore, to

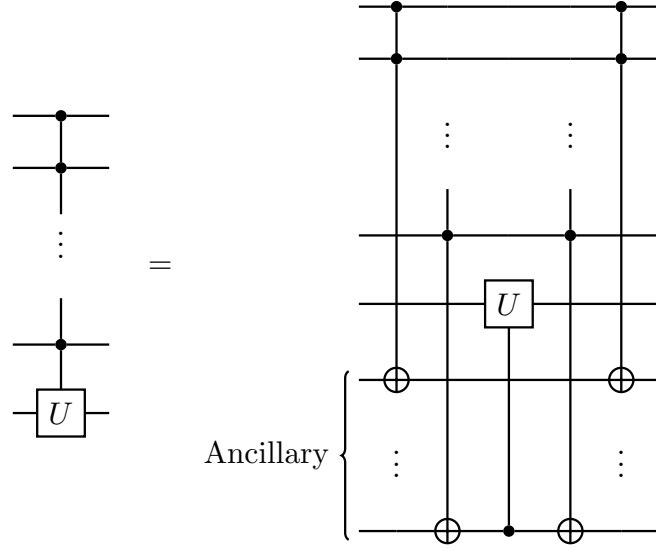


Figure 4.1: Naive decomposition of an  $n$ -controlled gate operation using  $(n-1)$  ancillary qubits.

implement a quantum parton shower algorithm, we discretise the evolution variable, evolving the system down in discrete steps of the momentum fraction  $z$ .

## 4.2 Quantum Circuit Implementation

Having outlined the theoretical framework of the parton shower, we can now build a quantum circuit implementation of a quantum parton shower algorithm, inspired by the classical Monte Carlo approach [1]. The quantum circuit is constructed from four registers: the *parton* register, *counting* register, *emission* register, and *history* register. The algorithm is discretised into individual shower steps, with each step operation consisting of four operations: the *counting* operation, *emission* operation, *history* operation, and *update* operations. Each of the operations is derived from the naive decomposition of a multi-controlled rotation in Figure 4.1, which is supported by  $(n - 1)$  ancillary qubits for an  $n$ -control rotation.

The step operation is applied iteratively to achieve the parton shower, reflecting the conventional Monte Carlo algorithm method from Section 2.3.2. A measurement is performed at the end of the algorithm only, projecting the final state of the system onto a register of classical bits,

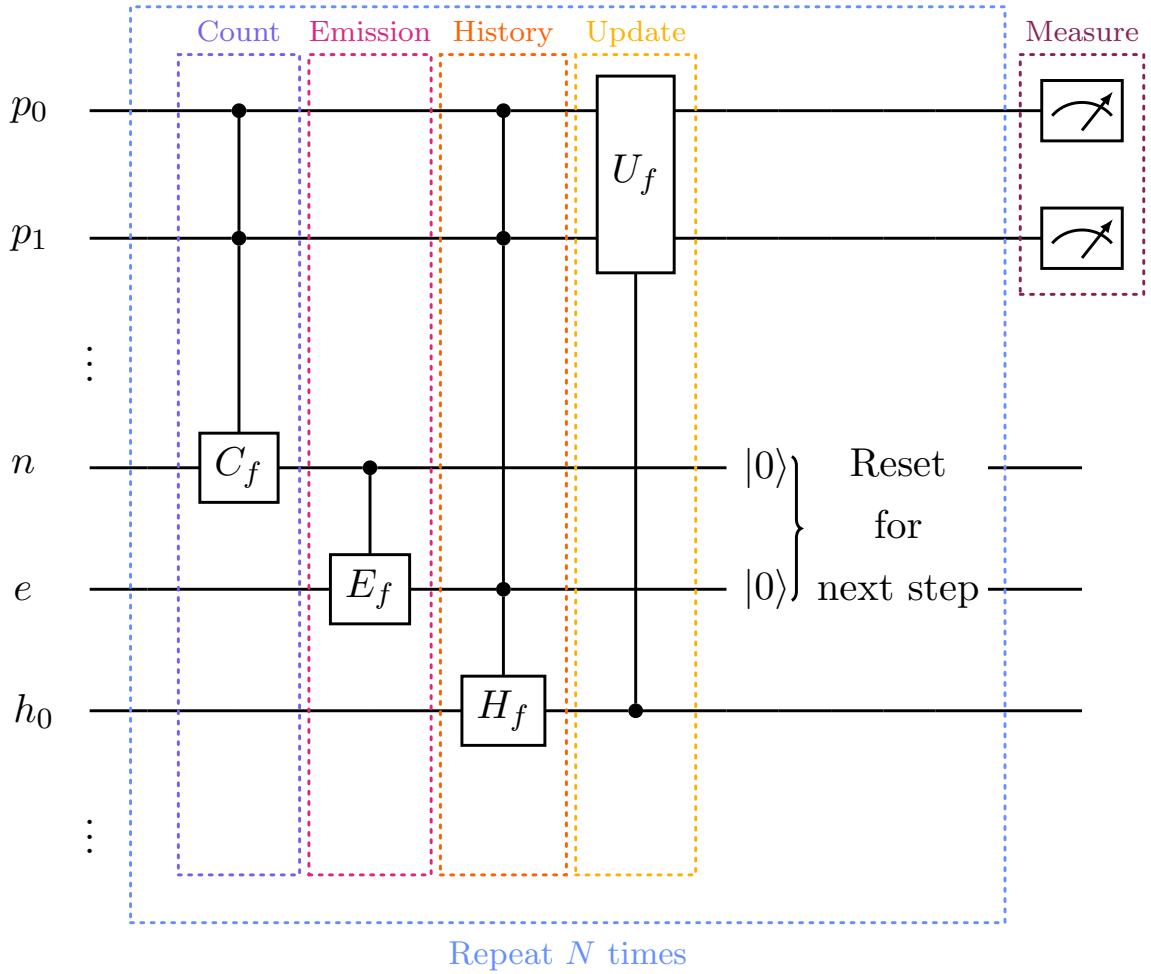


Figure 4.2: Schematic circuit diagram for the quantum parton shower algorithm, with the ability to simulate a collinear, toy QCD model. The circuit is constructed from four registers: the parton register  $p_i$ , the counting register  $n$ , the emission register  $e$ , and the history register  $h_i$ . The algorithm proceeds in individual steps comprised of four unitary operations: the counting operations  $C_f$ , the emission operations  $E_f$ , the history operations  $H_f$ , and the update operations  $U_f$ . The subscript  $f$  is used here to denote that these operations are applied for each parton in the shower, and differ at each step. After  $N$  steps, the parton register is measured and the particle content of the shower is obtained.

retrieving information about a single final state configuration. The algorithm is then repeated to achieve a probability distribution of shower histories. A schematic of the quantum parton shower circuit is shown in Figure 4.2. Below, we define each of the operations in the algorithm and explain how the evolution of the parton shower is handled.

The parton shower starts with a single initial parton, which can be either a gluon or a (anti)quark, and showers the parton according to the theoretical outline above. The species of the parton is encoded on the parton register using a three qubit basis, such that the states  $|100\rangle$ ,  $|001\rangle$  and  $|011\rangle$  describe a gluon, quark and antiquark respectively. The absence of a parton is represented by the null state,  $|000\rangle$ . The initial parton is encoded on the  $p_0$  sub-register, with subsequent emissions populating the  $p_i$  sub-register, for  $i > 0$ . With a three qubit basis, it is easy to extend the model to include more flavours of quarks; one has the ability to encode seven parton species and one null state using three qubits.

### 4.2.1 Counting Operation

The first operation in the algorithm is the counting operation,  $C_f$ , which seeks to determine the parton content of the shower at the start of each step. The operation is constructed from three individual counting mechanisms, one for each species of parton, and is individually applied to each parton sub-register. The counting operation controls from the state of the parton sub-registers  $p_i$  and populates the counting register  $n$  to record the number of partons in the shower. Figure 4.3 shows a schematic of the count operation applied to the parton sub-register  $p_i$ . Note that here we divide the counting register into three sub-registers  $n_g$ ,  $n_q$ , and  $n_{\bar{q}}$ , each of which contains a set of qubits sufficient to encode the number of each parton species in the computational basis. The number of qubits required for the counting register therefore scales as

$$n_{\text{qubits}}^{\text{count}} = 3 \left\lceil \log_2(N + 1) \right\rceil, \quad (4.2.1)$$

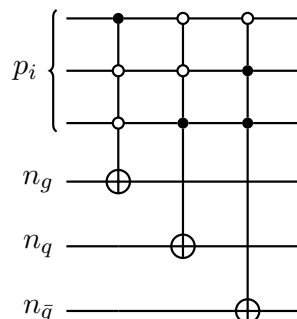


Figure 4.3: The count operation controls from the parton sub-register  $p_i$  and updates the counting registers  $n_i$  according to the particle content of the shower.

*id est* each sub-register grows logarithmically. It should be noted that this is the most general scaling. In fact, due to the nature of the parton splittings, one can streamline this scaling and remove redundant qubits. For a two-step parton shower, the counting register requires only four qubits: two qubits for the gluon counting sub-register, and one for the quark and antiquark sub-registers. This is because there cannot be more than one (anti)quark after the first step, but there can be two gluons. The streamlining of the counting register reduces the overall number of qubits needed to simulate a two-step shower, allowing for the algorithm to be run on the `ibmq_qasm_simulator`.

The count operation is constructed from a series of multi-controlled gate operations: CCNOT and CNOT gates. For each particle sub-register, 15 gates are required at each step to implement the count operation. However, as we only allow for one emission per step, we know the maximum amount of partons allowed at any point in the parton shower. Thus, we require  $15N$  operations at step  $N$  of the algorithm, as only  $N$  partons can be present in the shower at this time. The number of gate operations required to implement the count operation therefore scales as

$$n_{\text{gates}}^{\text{count}} = 15 \frac{N(N+1)}{2}. \quad (4.2.2)$$

### 4.2.2 Emission Operation

Following the Monte Carlo shower process from Section 2.3.2, the next step for the quantum shower algorithm is to deal with the Sudakov factors and determine whether an emission occurs in a momentum interval  $z_1$  to  $z_2^*$ . The non-emission probability is encoded onto the device using the rotation

$$E_f = \begin{pmatrix} \sqrt{\Delta_{\text{tot}}(z_1, z_2)} & -\sqrt{1 - \Delta_{\text{tot}}(z_1, z_2)} \\ \sqrt{1 - \Delta_{\text{tot}}(z_1, z_2)} & \sqrt{\Delta_{\text{tot}}(z_1, z_2)} \end{pmatrix}, \quad (4.2.3)$$

where  $\Delta_{\text{tot}}$  is defined in Equation 4.1.5. The emission operation controls from the counting registers and acts on the emission register  $e$  with the  $E_f$  rotation.

Consider the first step in the shower, the emission register is constructed from a single qubit. The state of the qubit after the emission operation gives information about whether the parton emits a further parton or not. The emission operation for the first step in the shower is shown schematically in Figure 4.4. For each species of parton, the emission operation is applied such that the controlled rotation  $E_f$  constructs a superposition of states  $|0\rangle$  and  $|1\rangle$  on the emission qubit,  $e$ , which correspond to the non-emission and emission outcomes with probability  $\Delta_{\text{tot}}$  and  $(1 - \Delta_{\text{tot}})$  respectively. If the parton register  $p_i$  is initialised in a superposition of the three states corresponding to a gluon, quark and antiquark, then the state of the emission register after the count and emission operations will contain information about each of the species' emission probabilities, in superposition.

As the shower proceeds, the emission operation becomes more complicated. After the first step, the parton content of the shower can increase. Therefore, the emission operation must control from all possible combinations of the partons in the shower, controlling from multiple

---

\*Note, for the quantum shower we do not include an automatic cutoff as in the Monte Carlo approach in Section 2.3.2. Instead, we “hardcode” the shower termination, which is constrained by the quantum resources available.

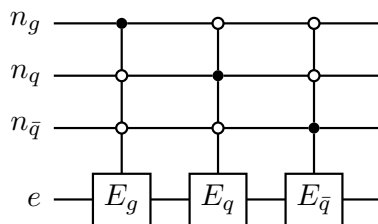


Figure 4.4: Controlling from the number of partons present in the shower, the emission gate encodes the Sudakov factor onto the device by rotating the emission qubit  $e$  by the  $E_f$  rotation.

counting sub-registers  $n_f$  at once in order to implement Equation 4.1.5 correctly. We have defined the shower model such that only one emission occurs per step, hence the quantum shower algorithm needs to select which parton is allowed to emit. This choice will be dealt with by the update operation. However, to allow for the choice to be made, the emission probabilities for each parton must be recorded. For a shower with  $N$  steps, there are a maximum of  $N$  possible partons before the final emission, therefore the emission register scales as

$$n_{\text{qubits}}^{\text{emission}} = N, \quad (4.2.4)$$

such that the emission probabilities of the parton in sub-register  $p_i$  are encoded onto the emission register qubit  $e_i$ .

The emission operation is constructed from a series of CCNOT gates and a controlled rotation, following the form shown in Figure 4.1. The number of CCNOT gates required to decompose a multi-controlled rotation is  $2(n - 1)$ , where  $n$  is the number of controls. Therefore, the number of gates required to implement the emission operation is

$$n_{\text{gate}}^{\text{emission}} = 12 \left( 3 \left\lceil \log_2(N + 1) \right\rceil - 1 \right) + 6N. \quad (4.2.5)$$

Once again, this can be streamlined when tailoring the algorithm to simulate only two steps. Since we know that only one parton is present in the first step of the algorithm, then only

three controlled rotations are needed to correctly implement the emission operation for the first step, thus removing the need for computationally costly CCNOT gates. Furthermore, due to the streamlining of the number of qubits needed in the counting register, the emission operation for the second step has less controls, four qubit controls instead of six. Therefore, fewer CCNOT gates are needed to implement the multi-controlled unitary rotation. Although we use the `ibmq_qasm_simulator` for this example, which can simulate a fully fault tolerant device, shallow circuit depths are crucial in order to obtain practical results from NISQ devices. Any reduction in the number of gate operations will have a direct affect on the fidelity of the results. This will be discussed in further detail in Chapter 6 and Appendix C.

### 4.2.3 History Operation

Having established whether an emission has occurred or not, and encoded this information onto the emission register, the next step of the algorithm is to determine which species the emitting parton will split to. This is handled by the history operation, which encodes the splitting probabilities onto the device through the history rotation,

$$H_f = \begin{pmatrix} \sqrt{1 - \hat{P}(z)} & -\sqrt{\hat{P}(z)} \\ \sqrt{\hat{P}(z)} & \sqrt{1 - \hat{P}(z)} \end{pmatrix}, \quad (4.2.6)$$

where  $\hat{P}(z) = (P_{ji}(z)/P_{\text{tot}}(z))$  and we have defined

$$P_{\text{tot}} = n_g (P_{qg} + P_{gg}) + n_q P_{gq} + n_{\bar{q}} P_{g\bar{q}}. \quad (4.2.7)$$

From this equation, we see that the two possible gluon splittings,  $g \rightarrow gg$  and  $g \rightarrow q\bar{q}$ , lead to an added complication in the implementation of the history operation. To accommodate for splittings, the history register is partitioned into three sub-registers, the gluon histories, quark



histories, and antiquark histories, where the gluon history sub-register requires twice the amount of qubits as the (anti)quark sub-register. The history register is directly needed as a control in the update operation, therefore we will be required to extend the history register at each step. The scaling of the history register with respect to the number of steps  $N$  is given by

$$n_{\text{qubits}}^{\text{history}} = 4 \frac{N(N+1)}{2}. \quad (4.2.8)$$

The general form of the history operation is schematically shown in Figure 4.5. The operation controls from the parton sub-registers,  $p_i$ , and the emission register,  $e$ , and applies the  $H_f$  rotation from Equation 4.2.6 to the history sub-registers,  $h_i^f$ , encoding the splitting information onto the device. For example, consider a gluon in the first parton sub-register,  $p_0$ . If the emission operation determines that an emission has occurred, then the history operation will be applied to the  $h^g$  sub-register for this parton. The probability for the gluon to split to two gluons is encoded onto the  $h_0^g$  sub-register by the  $H_{gg}$  operation. A superposition of the states  $|0\rangle$  and  $|1\rangle$  with probability  $(1 - \hat{P}(z))$  and  $\hat{P}(z)$  respectively is constructed on the device, corresponding to the splitting probabilities for the gluon in the  $p_0$  sub-register. This will then be repeated for the second gluon splitting kernel, and for all other partons in the shower with the corresponding splitting kernels and  $H_f$  operation.

Once again, we can streamline this operation by tailoring the circuit for the two-step shower. Exploiting the symmetry in the splitting function for a quark and an antiquark, we require only one history sub-register for the quark and antiquark species. We will see how this is practical in the update operation. Furthermore, Figure 4.5 shows the two gluon history rotations controlled and applied separately. To reduce the number of CCNOT gates, in practice the  $H_{gg}$  and  $H_{qg}$  rotations can be applied to  $h_0^g$  and  $h_1^g$  at the same time, without the need to control twice from the same state on the parton register  $p_i$ .

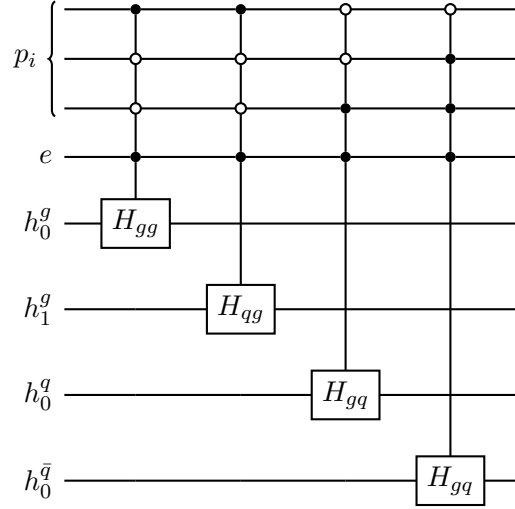


Figure 4.5: The history operation controls from the parton sub-registers,  $p_i$ , and the emission register,  $e$ , and acts on the history sub-registers,  $h_i^f$ , with the  $H_f$  rotation. The  $H_f$  rotation encodes the information about which species the emitting parton will split to onto the history register. Note that the two gluon history rotations  $H_{gg}$  and  $H_{qg}$  can be applied at the same time to reduce the number of CCNOT gates, but here they have been displayed separately for clarity.

Similarly to the emission operation, the history operation is constructed from a series of CCNOT gates and a controlled-rotation<sup>†</sup>, following the decomposition of the multi-controlled unitary rotation in Figure 4.1. The number of gates required to implement the history gate at step  $N$  is given by

$$n_{\text{gates}}^{\text{history}} = 16 \frac{N(N+1)}{2}. \quad (4.2.9)$$

Due to the quadratic scaling in both the number of gates and the number of qubits required in the history register, the history operation is the most costly operation in the quantum parton shower algorithm. Although the scaling in qubits is important, and is the focus of the implementation in Reference [1], the scaling of the number of gates is arguably more dangerous for NISQ devices, which have low coherence times and large environmental errors. For the proof-of-principle

<sup>†</sup>The streamlined version of the  $H_{gg}$  and  $H_{qg}$  operations follows the routine in Figure 4.1, but instead of one controlled rotation, there are two.

quantum parton shower outlined here, coherence times and CNOT errors are not a prevalent error, as the algorithm is designed to be run on the `ibmq_qasm_simulator` device, which simulates a fully fault tolerant device.

#### 4.2.4 Resetting the Emission and Counting Registers

Now that the full splitting probability has been encoded onto the device, the history register holds all the information needed to propagate the shower to the next step, inheriting the Sudakov factor information from the emission operation. Before updating the particle content of the shower, one can now return the counting and emission registers back to the ground state in the computation basis. This can be achieved by applying the emission and then counting operations in reverse. It should be noted that, although referred to as *resetting* here, returning the counting and emission registers to the ground state is an entirely unitary operation. No decoherence has been introduced to the circuit, nor has a measurement been performed at this intermediate stage of the algorithm.

Resetting the emission and counting registers to the ground state allows for the reuse of the registers without having to introduce new qubits, and reduces the scaling of the number of required qubits to those shown in Equations 4.2.1 and 4.2.4.

#### 4.2.5 Update Operation

The final operation in a shower step is to update the particle content of the shower, depending on the splitting probabilities encoded on the history registers. The update operation is an iterative application of the circuit routine shown in Figure 4.6 to all populated parton sub-registers,  $p_i$ . The first gate group populates the  $p_j$  sub-register with a gluon, simulating the emission of a gluon from a (anti)quark<sup>‡</sup>.

---

<sup>‡</sup>We can group the splittings  $q \rightarrow gq$  and  $\bar{q} \rightarrow g\bar{q}$  together for the update operation due to the symmetry in the splitting kernels from Equation 2.3.3 for the quark and antiquark.

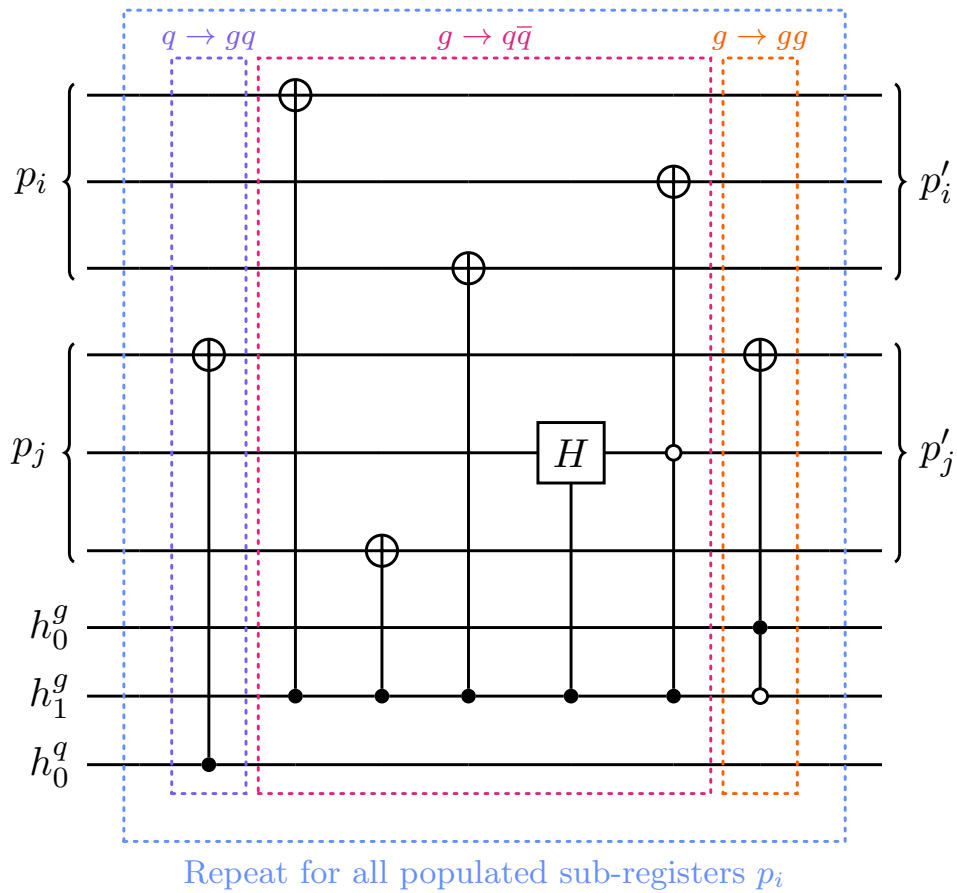


Figure 4.6: Update operation sub-routine applied to all populated parton sub-registers  $p_i$ . The first gate group shows the creation of a gluon from the  $q \rightarrow gq$  splitting. Note that this can also be for a  $\bar{q} \rightarrow g\bar{q}$  splitting. The centre gate group shows the splitting  $g \rightarrow q\bar{q}$  and the final gate group shows the splitting  $g \rightarrow gg$ . The  $|0\rangle$ -control on the  $h_1^g$  history qubit gives preference to  $g \rightarrow q\bar{q}$  splittings. The parton registers are then passed to the next step.

The second gate group deals with the splitting of a gluon to a quark-antiquark pair. We see from Figure 4.6 that the first step is to return the  $p_i$  sub-register to the null state, before populating  $p_i$  and  $p_j$  with a quark state. In this unconstrained evolution, we do not have an ordering to the quark-antiquark pair, therefore either of the parton sub-registers,  $p_i$  or  $p_j$ , can take on the antiquark state. To simulate this on the device, we construct a superposition of the quark and antiquark states on both the  $p_i$  and  $p_j$  sub-registers using a controlled-Hadamard gate and subsequent CCNOT gate. If the controlled-Hadamard gate yields a  $|0\rangle$  state on the middle qubit of  $p_j$ , *id est*  $p_j$  stays in the quark state, then the CCNOT gate creates an antiquark state in  $p_i$ . Otherwise, if the controlled-Hadamard gate yields a  $|1\rangle$  state on the middle qubit of  $p_j$ , *id est* an antiquark state on  $p_j$ , then the CCNOT gate has no effect, and  $p_i$  remains in the quark state. Since the Hadamard gate constructs an equal superposition, the quark-antiquark and antiquark-quark states on the  $p_i$  and  $p_j$  sub-registers are constructed in equal superposition.

The final gate group in the update operation shown in Figure 4.6 corresponds to the splitting of a gluon into two gluons. Similarly to the emission of a gluon from a (anti)quark, this is a simple population of the  $p_j$  sub-register with the gluon state. However, here we introduce a distinct feature of the shower model: the preferential treatment of the splitting  $g \rightarrow q\bar{q}$  over  $g \rightarrow gg$ . This is encoded onto the device by using a  $|0\rangle$ -control on the  $h_1^g$  qubit when populating  $p_j$ . Without tracking the kinematics of the parton shower, this preference is a practical method of implementing the splittings. However, this may be detrimental to the fidelity of the results obtained from the quantum parton shower algorithm. Nevertheless, we will see in Section 4.3 that this is a reasonable approximation for the shower model, as collinear radiation from gluons is dominated by the emission of further gluons. Therefore, there is only a small crossover when  $h_0^g$  and  $h_1^g$  are both in the  $|1\rangle$  state after the history operation, having minimal affect on the two-step shower. We will see that this preference is removed/replaced in more realistic algorithms shown in Chapters 5 and 6.

The update sub-routine shown in Figure 4.6 is then applied iteratively to all parton sub-registers,  $p_i$ . The shower model has been defined to allow for only one emission per step, therefore, if more than one parton has been calculated to emit in the emission and history operations, then the algorithm must “choose” which parton is allowed to emit. Here, we accommodate this choice by allowing for the “oldest” partons in the shower to emit first, *id est* the update operation will allow for parton  $i$  to emit over parton  $j$  if  $i > j$ . The number of operations to implement the update operation is

$$n_{\text{gates}}^{\text{update}} = 7 \frac{N(N+1)}{2}. \quad (4.2.10)$$

Once again, we see quadratic scaling in the number of required gates. However, it should be noted that the update gate is constructed mainly from single controlled gates, which are not as computationally expensive as multi-controlled gates, such as the CCNOT. Therefore, the update operation is not as costly as the counting and history operations, which grow quadratically in the number of required CCNOT gates.

After the update operation has been completed, the step is then repeated for the desired number of steps, constructing a highly entangled final state that holds the information of all possible shower histories. Measurement of the parton register then projects out a single final state configuration. The shower is then repeated to build a probability distribution for the shower process.

### 4.3 Results

The collinear QCD quantum parton shower has been run for two shower steps on the `ibmq_qasm_simulator` device for 10,000 shots of the algorithm. For ease of implementation, each shower step has been run with a fixed interval  $0.3 \leq z \leq 0.5$ , where  $z$  is the fraction of parton  $i$ 's momentum carried

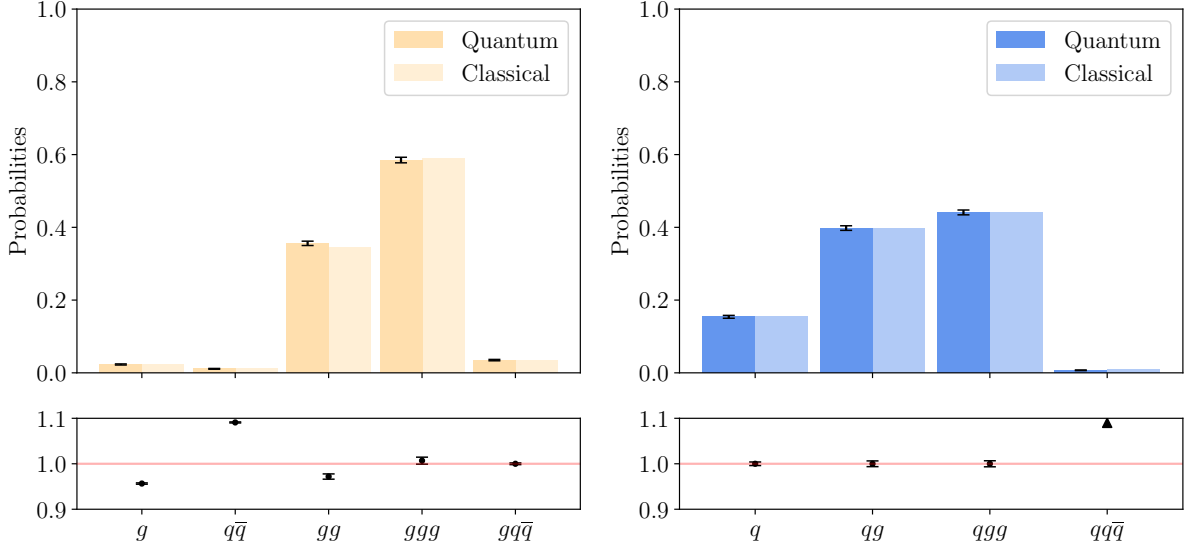


Figure 4.7: Comparison between the `ibmq_qasm_simulator` and analytical calculations for a two-step quantum parton shower with momentum interval  $0.3 \leq z \leq 0.5$ . The algorithm has been run for an initial particle of a gluon (left) and a quark (right). Note that the distribution of the antiquark initial state will be identical to the quark initial state. The algorithm has been run for  $10^4$  shots on the quantum simulator, without a noise profile. Figure adapted from Reference [1].

away from the splitting  $i \rightarrow jk$  by parton  $j$ , and is the evolution variable of the shower. The `ibmq_qasm_simulator` device has been run without a noise profile, therefore representing a fully fault tolerant device. The results are displayed in Figure 4.7 and show good agreement with the expected final state configuration rates calculated analytically using the collinear splitting functions and Sudakov factors. The antiquark distribution is equivalent to the quark distribution due to the symmetry of the (anti)quark splitting kernel in Equation 2.3.3, and therefore has been omitted.

For both distributions in Figure 4.7, there is an excess in the shower histories that involve a  $g \rightarrow q\bar{q}$  splitting, which is likely due to the preference given to this splitting over  $g \rightarrow gg$  in the update operation. This preference allows for a practical implementation of the splittings onto the device, ensuring that the splitting of  $g \rightarrow q\bar{q}$  occur in the shower for the momentum interval chosen here. This therefore allows for the validity of the algorithm to be tested, demonstrating

that the  $g \rightarrow q\bar{q}$  splittings are implemented correctly as a proof of principle. However, in improved algorithms this preference has been removed/replaced to increase the realism of the shower. For the two-step shower, the splittings involving  $g \rightarrow q\bar{q}$  have a small contribution to the final probability density and therefore the approximation is acceptable for the comparison in Figure 4.7.

Here we have presented a general and extendable algorithm for the simulation of a collinear QCD parton shower on a quantum device, inspired by the classical Monte Carlo method. From now on, we will refer to this shower algorithm as the quantum Monte Carlo parton shower. Designed to run on the 32-qubit `ibmq_qasm_simulator`, the shower algorithm has the ability to simulate two shower steps and is constructed from 31 qubits and 444 gate operations (169 single qubit gates, 217 CCNOT gates and 58 CNOT gates<sup>§</sup>). The scaling of the algorithm has been shown to be quadratic in the number of qubits and the number of gates required. This quadratic scaling makes an implementation on a NISQ device unfeasible due to the large circuit depth and the many multi-controlled qubit gates. In order to obtain high fidelity results from a NISQ device, the circuit depth of the parton shower simulation must be dramatically reduced. Chapter 5 shows that exploiting the quantum walk framework, the collinear parton shower can be implemented efficiently on the `ibmq_qasm_simulator`, achieving shower depths comparable to a realistic parton shower at a high energy hadron collider.

---

<sup>§</sup>Note these qubit and gate numbers are for the streamlined version of the algorithm



## Chapter 5

# Quantum Walk Approach to Shower Algorithms

In Chapter 4, we have seen the implementation of a quantum parton shower algorithm inspired by the classical Monte Carlo approach to parton showers, outlined in Section 2.3.2. This quantum Monte Carlo parton shower is the first quantum implementation of a parton shower with the ability to simulate a toy QCD model and is a proof-of-principle use case for quantum computation for high energy physics purposes. However, it was shown that the algorithm scales quadratically in both the number of required qubits and circuit depth. This scaling makes a practical implementation on a NISQ device unfeasible due to the low qubit limit and decoherence times. Even when running on the `ibmq_qasm_simulator`, which simulates a full fault tolerant device\*, the quadratic scaling in the number of qubits required is a limiting factor, restricting the simulation to a maximum of two shower steps. This is relatively shallow when compared to shower depths that result from a high energy hadron collision, such as those which occur at the Large Hadron Collider. Therefore, to simulate realistic showers, we must find a way to reduce

---

\*Note that the `ibmq_qasm_simulator` device simulates a fully fault tolerant quantum computer by calculating the state vector of the system analytically. Therefore, the simulation of large circuits can quickly become highly computationally expensive.

the amount of quantum resources needed to implement each step of the shower.

In this chapter, we present a novel quantum parton shower algorithm which utilises the compact circuit architecture of the quantum walk described in Section 3.2, summarising the findings of Reference [2]. Section 5.1 extends the theoretical framework from Section 4.1, abstracting the parton shower to the quantum walk framework. In Section 5.2 we give a simple example of a scalar shower to illustrate how the movement of a walker on a one-dimensional number line can model a shower algorithm, leading to the simulation of a collinear QCD parton shower in Section 5.3. In Section 5.4, we will see that the paradigm shift from a classical Monte Carlo inspired algorithm to a quantum architectural circuit design will provide a significant improvement in the realism of the parton shower, achieving realistic shower depths and reducing the quantum resources required.

## 5.1 The Parton Shower as a Quantum Walk

The quantum walk parton shower follows the same theoretical framework as the quantum Monte Carlo parton shower, presented in Section 4.1. The shower is discretised into individual steps and we allow for only one emission per step. In the quantum walk regime it is useful to combine the Sudakov factor,  $\Delta$ , defined in Equation 4.1.5, and the splitting kernel,  $P$ , defined in Equation 4.2.7, to express the overall probability for emission at a step as

$$\hat{P}'_{ji} = (1 - \Delta_{\text{tot}}) \times P_{\text{tot}}. \quad (5.1.1)$$

We will see that this probability will be directly encoded onto the device through the coin operation in the quantum walk.

The quantum walk framework allows for a greater number of steps to be simulated, thus the evolution of the shower is important. We handle this by evolving the evolution variable

$z$  to lower and lower momenta, exponentially with the number of steps. Once again, we do not include an automatic cutoff mechanism in the quantum walk parton shower, and instead “hardcode” the shower termination at a desired number of steps. Furthermore, it will be shown that the individual treatment of the quark and antiquark is removed to streamline the algorithm, without reducing its generality.

## 5.2 Scalar Quantum Walk Shower

To implement a shower algorithm as a quantum walk (with memory), we use the framework outlined in Section 3.2. The quantum walk is constructed from the tensor product of two Hilbert spaces: the position space  $\mathcal{H}_P$  and the coin space  $\mathcal{H}_C$ . The architecture can be augmented by a further Hilbert space,  $\mathcal{H}_M$ , allowing for the walker to have memory of previous occupied positions and/or previous coin operations. The quantum walk framework is discretised into individual steps constructed from the *coin* operation, which acts on  $\mathcal{H}_C$ , and the *shift* operation, which controls from the  $\mathcal{H}_C$  and acts on  $\mathcal{H}_P$ . To introduce memory to the walk, the memory operation is applied as desired in the step<sup>†</sup>. For the quantum walk parton shower, each step operation is entirely unitary and no decoherence effects are introduced during the algorithm.

Analogously to the quantum walk, the shower model presented in Section 4.1 is discretised into individual steps. Therefore, the quantum walk framework appears to be a natural framework for the simulation of showers. In this section we will outline the simulation process using a simple scalar shower example. Consider a toy model with one particle,  $\phi$ , that can split via the mechanism  $\phi \rightarrow \phi\phi$  according to the probability

$$\hat{P}'_{\phi\phi} = (1 - \Delta_\phi) \times P_{\phi\phi}. \quad (5.2.1)$$

---

<sup>†</sup>The memory operation can be tailored, however it must be unitary to retain quantum diffusive effects.

We can therefore define a rotation which can encode this probability onto the device, such that

$$C_{\phi\phi} = \begin{pmatrix} \sqrt{1 - \hat{P}'_{\phi\phi}} & -\sqrt{\hat{P}'_{\phi\phi}} \\ \sqrt{\hat{P}'_{\phi\phi}} & \sqrt{1 - \hat{P}'_{\phi\phi}} \end{pmatrix}. \quad (5.2.2)$$

We can make a direct connection between this rotation and the coin operation from the quantum walk, such that the characteristics of the coin are encoded using  $C_{\phi\phi}$ . Acting on a single qubit,  $C_{\phi\phi}$  yields the states  $|0\rangle$  and  $|1\rangle$  corresponding to the non-emission,  $(1 - \hat{P}'_{\phi\phi})$ , and emission,  $\hat{P}'_{\phi\phi}$ , probabilities respectively. Therefore, we can use this rotation as the coin operation to encode the splitting probabilities on the coin space,  $\mathcal{H}_C$ , which, because we only have one splitting, spans  $\{|0\rangle, |1\rangle\}$ .

Having defined a tailored coin operation to reflect the splitting probabilities, we can now utilise the position space  $\mathcal{H}_P$  to encode the number of particles in the shower. For the case of the scalar shower, the number of particles in the shower can only increase through the  $\phi \rightarrow \phi\phi$  splitting, and there is no mechanism to diminish the number of scalars. Therefore, we can restrict  $\mathcal{H}_P$  to span only zero and positive integers,  $\{|i\rangle : i \in \mathbb{N}_0\}$ , where  $|i\rangle$  represents  $i$  particles in the computational basis. Using this position space, and the coin operation from Equation 5.2.2, a circuit for a simple one-dimensional quantum walk can be constructed, schematically shown in Figure 5.1. The circuit is constructed from three registers: the position register,  $x$ , which encodes the number of particles in the shower using the computational basis, the coin register,  $c$ , and memory register,  $m$ , which can be used to store previous coin operations or walker positions. The circuit is supported by an ancillary qubit,  $a$ . The number of particles that can be simulated by the quantum walk shower scales exponentially with the number of qubits in the position register,  $n_x$ , such that

$$\text{Maximum number of particles} = 2^{n_x} - 1. \quad (5.2.3)$$

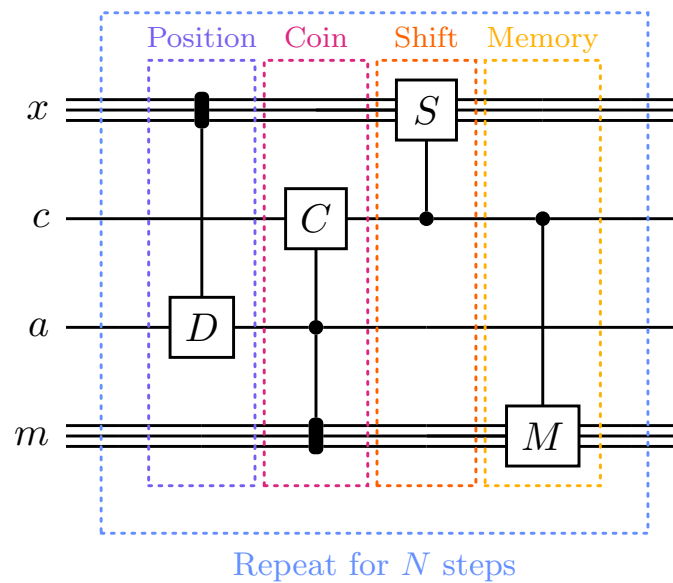


Figure 5.1: Schematic circuit diagram for a quantum walk with the ability to simulate a scalar particle cascade. Each step of the algorithm is constructed from three main operations: the *coin* operation, which encodes the splitting probabilities onto the device, the *shift* operation, which updates the particle content of the shower, and the *memory* operation, which can be used to store previous coin outcomes. The coin operation is supported by a *position check* operation, which ensures the correct implementation of the splitting probabilities. The step operation is then repeated for  $N$  steps, before measuring the number of particles in the final state, encoded in the position register.

The algorithm is discretised into individual steps, corresponding to steps in a parton shower. Each step is split into three main operations: the *coin* operation, the *shift* operation, and the *memory* operation, which are described below. We will see that to implement the splittings properly, the coin operation needs to be supported by the *position check* operations, resulting in a position dependent coin operation.

### 5.2.1 Coin Operation

Following the framework of the one-dimensional, discrete-time quantum walk from Section 3.2, the first operation in each step is the coin operation. To simulate the shower algorithm, we require this coin operation to encode the splitting probabilities on the coin space,  $\mathcal{H}_C$ , such that, after applying the coin rotation from Equation 5.2.2, the coin register,  $c$ , is in a superposition of the states  $|0\rangle$  and  $|1\rangle$  with probabilities corresponding to non-emission and emission respectively.

From Equation 5.2.1, we see that the coin operation includes the Sudakov factor and the splitting kernels, which depend on the shower content and shower evolution variable,  $z$ , at each step. Consequently, to correctly implement the splitting probabilities in the coin operation, the particle content of the shower must be known. The coin operation is therefore supported by the position check operation, shown in the first gate group of Figure 5.1. The position check operation is constructed from a series of CCNOT operations which control from the position of the walker, and by construction the number of particles in the shower, and update an ancillary qubit accordingly. The coin operation, shown in the second gate group of Figure 5.1, can then control from the ancillary qubit and act on the coin register,  $c$ , with the correct splitting probability,  $\hat{P}'$ . After applying the coin operation, the ancillary qubit can be returned to the ground state by applying the position check operation in reverse. This therefore allows for the same ancillary qubit to be used for each position check. Together, the position check and the coin operation, followed by the position check in reverse, form a multi-controlled rotation of the form shown in Figure 4.1. Introducing a memory operation to the quantum walk complicates this equivalence,

as will become clear when outlining the memory operation below.

The coin and the position check operations are then applied for each combination of intermediate shower states. The number of qubits required for the coin register,  $c$ , is equal to the number of splitting kernels, which in this case is one. Similarly, the number of qubits required for the ancillary register,  $a$ , is equal to the number of species in the shower, which again is one for this case. The particles are encoded in the position register,  $x$ , for which the number of required qubits grows logarithmically with the maximum number of shower steps (*id est* the number of particles which can be simulated). Therefore, the number of gates needed for the position check and coin operations after  $N$  steps is

$$n_{\text{gates}}^{\text{coin}} = N \left( 2 \left\lceil \log_2(N_{\text{max}} + 1) \right\rceil - 1 \right), \quad (5.2.4)$$

where  $N_{\text{max}}$  is the maximum number of steps in the shower, corresponding to the maximum number of particles which could be produced in the process.

### 5.2.2 Shift Operation

From Section 3.2, we see that the outcome of the coin dictates the dynamics of the walker. For the scalar shower, the splitting probabilities have been encoded as a coin operation onto the coin space,  $\mathcal{H}_C$ . The next task is to control from the coin register,  $c$ , and apply the shift operation accordingly. We have defined the simple model to include only the splitting  $\phi \rightarrow \phi\phi$ , and constructed the shower model such that only one emission can occur per shower step. Thus, there is no mechanism that can diminish the number of scalars in the shower. Consequently, the shift operation will only require a routine that either increments the position of the walker at each step, corresponding to an emission, or leaves the position of the walker unchanged, when there is no emission. A schematic of the shift operation for the scalar shower is shown in Figure 5.2. We construct the circuit to increment the walker's position if the coin operation

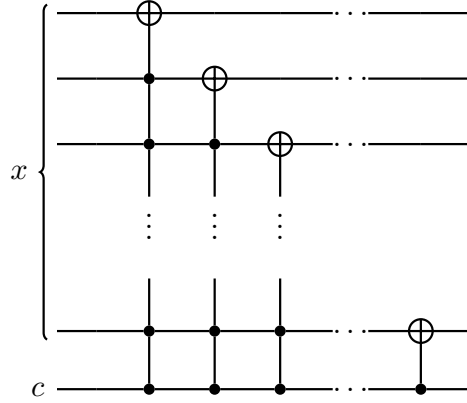


Figure 5.2: Schematic of the circuit implementation for the shift operation in the scalar shower model. If, after the coin operation, the coin is in the  $|1\rangle$  state and a splitting has occurred, then the position of the walker is incremented using a series of CCNOT gates. Otherwise, if a splitting has not occurred and the coin is in the  $|0\rangle$  state, then the walker remains stationary.

from Equation 5.2.2 yields a  $|1\rangle$  state on the coin register. Note that, although not needed here, a decrement operation can be constructed from either controlling from  $|0\rangle$  states and not  $|1\rangle$  states, or applying the increment gate in reverse. This will be useful in Section 5.3 when discussing the QCD shower, which has splittings that can diminish the number of partons.

The number of gates required to implement the shift operation is the same for each step, therefore the circuit depth of the shift operation for an  $N$ -step shower grows linearly with  $N$ , such that

$$n_{\text{gates}}^{\text{shift}} = N \left( \left\lceil \log_2(N_{\text{max}}) \right\rceil \right)^2. \quad (5.2.5)$$

where  $N_{\text{max}}$  is the maximum number of steps in the shower, corresponding to the maximum number of particles that can be simulated. The shift operation for each step requires  $\left( \left\lceil \log_2(N_{\text{max}}) \right\rceil \right)^2$  gates to increment the position of the walker, which is equivalent to the number of qubits in the position register squared.



### 5.2.3 Memory Operation

Having defined a coin and shift operation, a conventional quantum walk can be constructed for our example. However, due to the definition of the coin operation in Equation 5.2.2, we require that  $C_{\phi\phi}$  is applied to a  $|0\rangle$  state on the coin register to correctly encode the splitting probabilities onto the device. After a shower step has been implemented, it is possible that the coin register is in the  $|1\rangle$  state. If we apply  $C_{\phi\phi}$  to a qubit in the  $|1\rangle$  state, then we encode the emission probability onto the  $|0\rangle$  state and the non-emission probability onto the  $|1\rangle$  state; the opposite of what is required to encode the splitting probabilities correctly. This mislabelling can be dealt with by allowing the quantum walk to have memory of previous coin operations through the use of the memory operation.

The memory operation is constructed from a series of CCNOT gates, which populates the memory register with the state of the coin after the coin operation has been applied. This can then be used in subsequent steps to ensure that the correct splitting functions are encoded onto the device. There is some freedom in the circuit design of the memory operation, however the operation is required to be unitary. The simplest memory operation can be thought of as a series of *additional coins*, and using this model one can introduce arbitrary dynamics into the algorithm [88]. If we required full memory, as described in Section 3.2.2, this additional coins memory model would require a coin qubit for each step, with the memory and coin registers being synonymous. In the full memory regime, the classical limit is obtained, however, and very importantly, no decoherence effects are introduced during the algorithm. The additional coin memory operation scales linearly with the number of steps in the shower, such that

$$n_{\text{gates}}^{\text{memory}} = 2N, \quad (5.2.6)$$

where the number of required qubits is equal to the number of steps in the algorithm. For

simplicity, we describe the quantum walk parton shower with an additional coin memory model<sup>‡</sup>.

The memory operation is not needed in the shower algorithm until after the second step, as the memory operation can be replicated using the position check operation. Therefore, the walker retains quantum diffusive effects. With this streamlining, the gate requirement of the memory operation reduces to  $n_{\text{gates}}^{\text{memory}} = 2(N - 2)$ .

With the definition of the memory operation, the scalar shower can now be modelled as a quantum walk with memory, as outlined in Section 3.2.2. The algorithm is fully extendable to an  $N$  step shower. The qubit requirement grows at most linearly with the number of steps<sup>§</sup>, and the circuit depth grows linearly with the number of steps. Furthermore, by examining the quantum walk's Hilbert space, the model can be extended to include more complicated branchings and model a QCD parton shower.

### 5.3 Quantum Walk Parton Shower

Following on from the scalar shower outlined in Section 5.2, it is possible to build a quantum walk with memory that can simulate the discrete, collinear QCD model from Section 4.1. This can be achieved by examining the position space,  $\mathcal{H}_P$ , and coin space,  $\mathcal{H}_C$ , of the one-dimensional quantum walk. Previously, we have seen that the number of particles of a single species can be encoded onto a one-dimensional position register, for which  $\mathcal{H}_P$  spans  $\{|i\rangle : i \in \mathbb{N}_0\}$ . The single-splitting kernel for the scalar has been encoded onto the coin register through the rotation in Equation 5.2.2, such that the coin space spans  $\{0, 1\}$ .

To extend the model to include more particle species and their corresponding splitting kernels, we must increase the dimension of  $\mathcal{H}_P$  to the number of particle species, and  $\mathcal{H}_C$  to the number of splitting kernels. For the toy model of QCD presented in Section 4.1, there are

---

<sup>‡</sup>One should note that it is possible to reset the coin qubit at each step, essentially measuring the coin register at each step and enforcing that the coin is in the  $|0\rangle$  state at the start of every step.

<sup>§</sup>With a more sophisticated memory operation, this linear scaling can be reduced to logarithmic scaling. However, to reduce the circuit depth, the linear scaling has been retained

two parton species: one gluon and one quark flavour. We can therefore extend the dimension of  $\mathcal{H}_P$  to two dimensions, allowing for the number of gluons to be encoded in one dimension, which we denote as the  $x$  position register, and the number of quarks to be encoded in the second dimension, which we will call the  $y$  position register. The position space,  $\mathcal{H}_P$ , now spans  $\{|i\rangle, |j\rangle; i, j \in \mathbb{N}_0\}$ . Note that the branchings that are possible for the gluon and quark, shown in Equations 4.1.1 to 4.1.3, can only produce quarks in quark-antiquark pairs. Therefore, we can use the  $y$  position register to encode the number of quark-antiquark pairs, removing the need for an extra register for the antiquark.

We must now extend the coin space,  $\mathcal{H}_C$ , to incorporate the three splitting kernels for the gluon and quarks. This is achieved by increasing the number of coins to three, therefore increasing the Hilbert space available to the walker, such that  $\mathcal{H}_C$  spans  $\{0, 1\}^3$ . A schematic of the quantum walk with memory for the simulation of the QCD model is shown in Figure 5.3.

With the redefinition of  $\mathcal{H}_P$  and  $\mathcal{H}_C$  we can now extend the quantum walk with memory framework from Section 5.2 to simulate a discrete, QCD parton shower. The shower follows the same procedure as the scalar shower example and has been designed and run on the `ibmq_qasm_simulator` device, allowing for the simulation of 31 shower steps. Below, we outline how the coin and shift operations are adapted to incorporate the additional parton species and splittings. Section 5.4 discusses the results of the algorithm compared to a like-for-like classical implementation. The possible advantage given by the quantum walk for the parton shower is then discussed in Chapter 7.

### 5.3.1 Coin Operation

For the QCD parton shower, there are three splitting kernels which are encoded onto three coin qubits using controlled rotations of the form shown in Equation 5.2.2. Similarly to the quantum walk shower from Section 5.2, the coin operation is supported by the position check operation. Having extended the position space to two-dimensions, the position check needs to control from

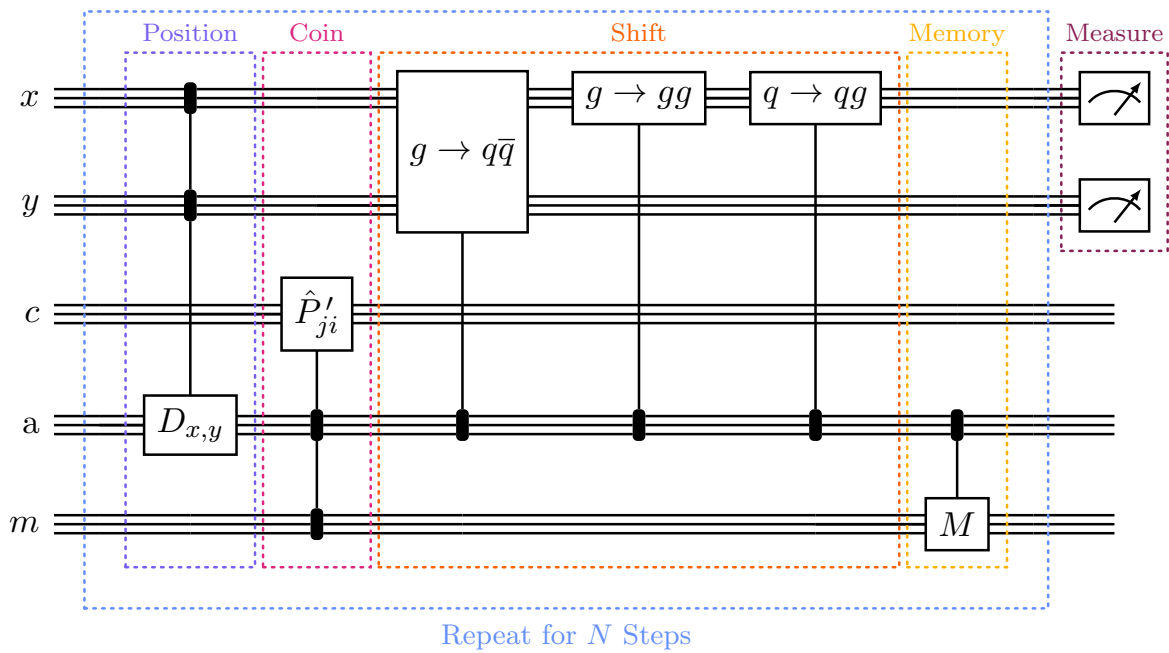


Figure 5.3: Schematic circuit diagram for the quantum walk parton shower with the ability to simulate a discrete, collinear QCD parton shower. The algorithm is built from a quantum walk with memory framework in two-dimensions and is constructed from three main operations: the *coin* operation, which encodes the splitting functions on the device, the *shift* operation, which moves the walker along a two-dimensional lattice to update the shower content, and the *memory* operation, which allows for previous coin operations to be retained for later use in the algorithm. The coin operation is supported by a *position check* operation to ensure the splittings are implemented correctly.

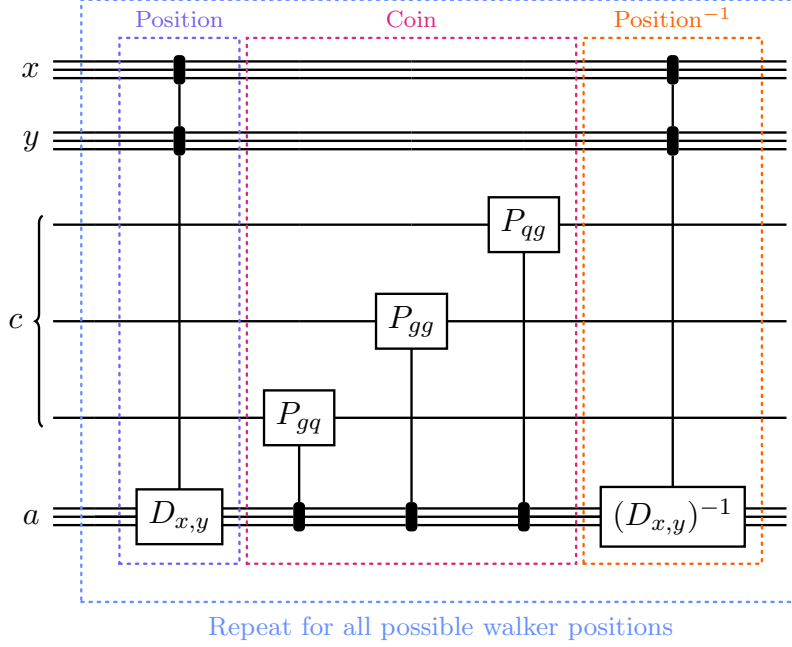


Figure 5.4: A schematic circuit diagram of the *position check* and *coin* operation for the quantum walk parton shower algorithm. The position check operation allows for the correct application of the splitting probabilities by controlling from the position of the walker and populating an ancillary register. The coin operation then controls from this ancillary register and applies the correct splitting probabilities which corresponds to the shower content. The ancillary register can then be returned to the ground state by applying the position check in reverse.

both the  $x$  and  $y$  registers to populate the ancillary register. The number of gates needed to successfully apply the position check and coin operations for the two-dimensional quantum walk parton shower is therefore

$$n_{\text{gates}}^{\text{coin}} = N \left( 4 \left\lceil \log_2(N_{\text{max}} + 1) \right\rceil + 1 \right), \quad (5.3.1)$$

where  $N_{\text{max}}$  is the maximum number of steps that the algorithm can simulate. The ancillary register can then be returned to the ground state by applying the position check in reverse. Figure 5.4 shows a schematic of the position check and coin operations for the QCD parton shower algorithm. Each of the splitting probabilities corresponding to the processes  $g \rightarrow q\bar{q}$ ,  $g \rightarrow gg$  and  $q \rightarrow qg$  are therefore encoded onto one of the coin qubits.

### 5.3.2 Shift Operation

Having encoded the correct splitting probabilities onto the device using the coin operation, the shift operation can now update the shower content accordingly. There are three processes that need to be implemented for the quantum walk parton shower:  $g \rightarrow q\bar{q}$ ,  $g \rightarrow gg$  and  $q \rightarrow qg$ . The splitting of a gluon to a quark-antiquark pair is the most complicated process to implement due to the reduction in the number of gluons, a process which was not encountered in the previous example. Here we will describe how the shower content is updated for the  $g \rightarrow q\bar{q}$  splitting, with the remaining splittings following on simply from the shift operation described in Section 5.2, incrementing the number of gluons by applying the circuit from Figure 5.2 to the  $x$  register.

To correctly implement the  $g \rightarrow q\bar{q}$  splitting, we need to decrease the number of gluons by one and then increase the number of quark-antiquark pairs by one. Figure 5.5 shows how this is done in the quantum circuit architecture. Controlling from the coin register, a decrement operation is applied to the  $x$  register, which contains information on the number of gluons in the shower. A controlled increment operation is then applied to the  $y$  position register, which contains information on the number of quark-antiquark pairs. Consequently, we can visualise the simulation of the splitting of a gluon into a quark-antiquark pair as the movement of the walker in the two-dimensional lattice. After  $N$  steps have been applied, the  $x$  and  $y$  registers are measured, and the position of the walker corresponds to the shower content after  $N$  steps. The movement of the walker in the 2D lattice is shown visually in Figure 5.6, where Figure 5.6a shows the results from a two-step parton shower with the splitting kernels from Equations 4.1.1 to 4.1.3. To demonstrate how the walker moves through the entire lattice, Figure 5.6b shows the result of a two-step parton shower where the splitting functions have been enhanced to increase the probability of emission.

The number of gates required to implement the  $g \rightarrow q\bar{q}$  operation at each step is  $2 \lceil \log_2(N_{\max}) \rceil^2$ , and therefore the number of gates to implement the shift operation for the quantum walk parton

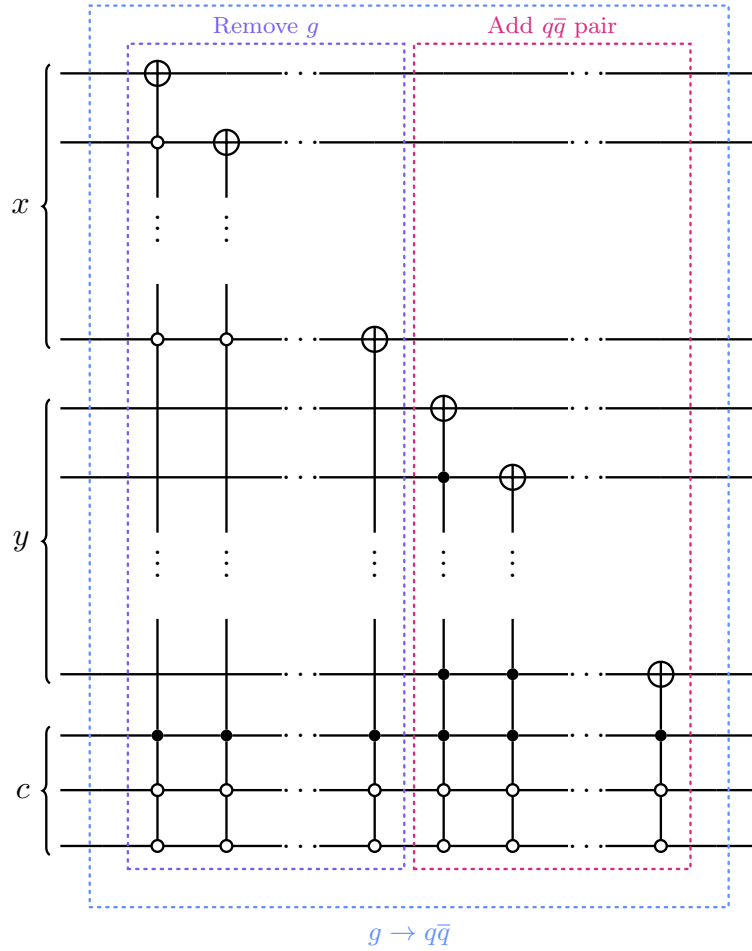


Figure 5.5: Quantum circuit implementation for the  $g \rightarrow q\bar{q}$  shift operation. The first gate group, “Remove  $g$ ”, decrements the number of gluons encoded on the  $x$  position register, reflecting the gluon splitting. The second gate group, “Add  $q\bar{q}$  pair”, increments the position of the walker in the  $y$  position register, corresponding to adding a quark-antiquark pair to the shower content. The  $g \rightarrow q\bar{q}$  controls from a  $|1\rangle$  state on the first coin qubit, which encodes the splitting probability  $\hat{P}'_{gg}$ , and a  $|0\rangle$  state on the second and third coin qubits, which correspond to the  $\hat{P}'_{gq}$  and  $\hat{P}'_{qg}$  splitting probabilities respectively.

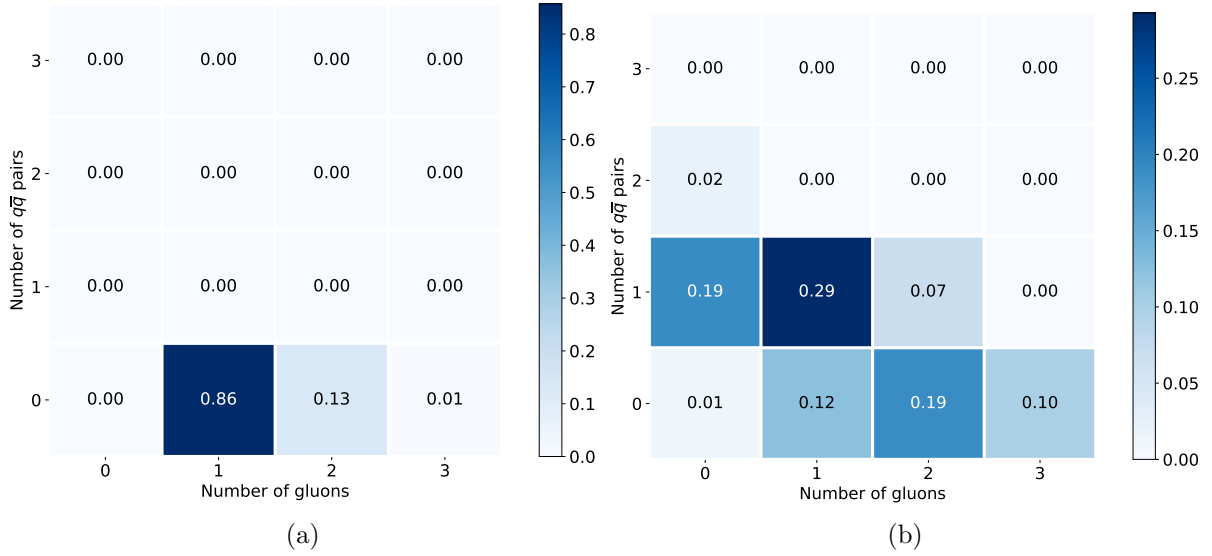


Figure 5.6: Visualisation of the quantum walk parton shower algorithm. The walker’s position on the two-dimensional lattice constructed from the  $x$  and  $y$  positions corresponds to the particle content of the parton shower. Figure (a) shows the result from a two-dimensional parton shower using the DGLAP splitting kernels, (b) shows modified splitting kernels to show how the walker spreads through the lattice. Figure from Reference [2].

shower scales with the number of steps  $N$  as

$$n_{\text{gates}}^{\text{shift}} = 4N \left( \left\lceil \log_2(N_{\text{max}}) \right\rceil \right)^2, \quad (5.3.2)$$

where  $N_{\text{max}}$  is the maximum number of steps in the shower.

The shift operation for the quantum walk parton shower implements the splitting probabilities differently from the quantum Monte Carlo parton shower outlined in Chapter 4, which allows for a preference to be given to the splitting  $g \rightarrow q\bar{q}$  over  $g \rightarrow gg$ . Furthermore, the quantum Monte Carlo parton shower gives preference to the order in which the partons in the shower were produced, allowing for the first parton to split at each step before considering subsequent partons. In the quantum walk parton shower, no preference is given to any branching process. Figure 5.5 shows that the shift operation controls from the  $|1\rangle$  state on only one of the coin qubits, and  $|0\rangle$  otherwise. This means that the only coin operation that yielded a  $|1\rangle$  state on



the coin register is the coin operation corresponding to the splitting implemented in the shift. If two coin operations yield a  $|1\rangle$  state on the coin register, then there is an ambiguity as to which splitting should be applied, and therefore the walker remains stationary.

The quantum walk parton shower also calculates the splittings without having any knowledge of which parton was produced at which step. Therefore, no preference can be given to any of the partons in the shower, and only the splitting which yields a  $|1\rangle$  state on one of the coin qubits is applied. This reflects the process used in the Monte Carlo approach outlined in Section 2.3.2 more closely. The quantum walk framework consequently provides a natural approach to simulating a parton shower.

### 5.3.3 Memory Operation

There is an amount of freedom in how the memory operation for the collinear, QCD quantum walk parton shower can be implemented. It is logical to consider the simple method of additional coins, used in the memory operation from Section 5.2. Similarly to the scalar shower, the memory operation is only required after the second step for the quantum walk parton shower, retaining quantum diffusive effects. With this approach, the scaling of the gates and qubits is equivalent to the scalar shower.

More sophisticated memory operations can be designed to reduce the quantum resources needed to implement the routine, or to gain some circuit architecture design advantage. For example, the state before splitting can be encoded onto a memory register and then used in subsequent operations as a control for the coin operation. It should be noted that a method of resetting the qubits without the need for a measurement or additional computational qubits may be achievable in the future. For now, however, one must use computational qubits to implement operations of this type.

## 5.4 Results

A 31-step quantum walk parton shower has been run on the `ibmq_qasm_simulator` device for 500,000 shots of the algorithm, with a gluon as the initial shower particle. The results obtained from the quantum device are shown in Figure 5.7 and have been compared to a like-for-like classical algorithm. Figures 5.7a and 5.7b show the number of gluons measured in the final state in association with zero or one quark-antiquark pairs. We see that the quantum walk parton shower shows good agreement with the like-for-like classical algorithm. A final state containing one or more quark-antiquark pairs is unlikely, producing low statistics. A further validation of the algorithm has therefore been conducted with splitting functions that enhance the  $g \rightarrow q\bar{q}$  and the  $q \rightarrow qg$  splittings to achieve more final states with one or more quark-antiquark pairs. The results from this test are shown in Figure 5.8, and once again the results show a good agreement with the classical algorithm, confirming that the quantum walk parton shower can replicate a discrete, collinear parton shower successfully. Even with the enhanced splitting functions, the probability of producing a final state with two or more quark-antiquark pairs is less than  $10^{-5}$ .

The quantum walk framework provides a distinct increase in the number of shower steps that can be simulated on the quantum device, reducing the number of qubits and the number of gate operations required to implement each shower step. The quantum walk parton shower has the ability to simulate over 15 times the number of shower steps than the quantum Monte Carlo parton shower from Chapter 4, allowing for a more realistic parton shower depth to be simulated.

In order to construct a fair comparison between the quantum walk parton shower and the quantum Monte Carlo parton shower, we can consider the quantum resources required to simulate a two-step shower in the discrete, collinear QCD toy model comprising one gluon and one quark flavour. The quantum walk parton shower can recreate the results shown in Figure 4.7 with a circuit that requires 9 qubits and 203 gate operations (59 single qubit gates, 98 CNOT

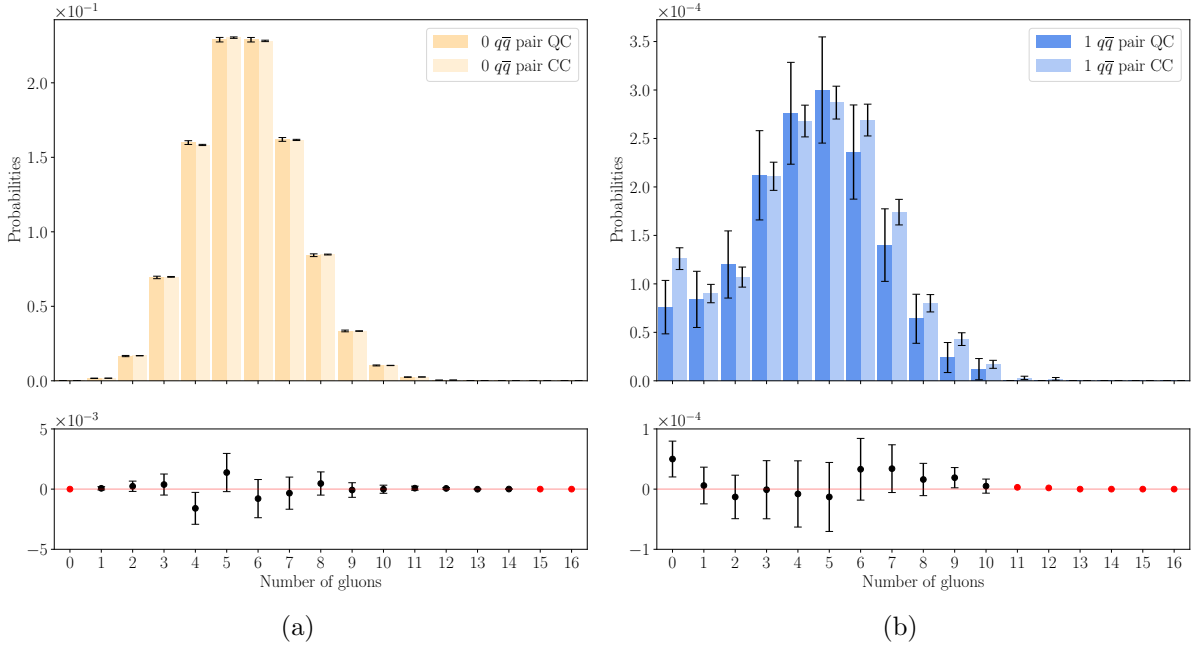


Figure 5.7: Comparison between the probability distributions for the number of gluons in the final state in association with zero quark-antiquark pairs (left) and exactly one quark-antiquark pair (right) for classical and quantum algorithms. The quantum algorithm has been run on the `ibmq_qasm_simulator` for  $5 \times 10^5$  shots on the device, and the classical algorithm has been run for  $10^6$  shots. Figure adapted from Reference [2].

gates and 46 CNOT gates) [2], compared to the 31 qubits and 444 gate operations (169 single qubit gates, 217 CCNOT gates and 58 CNOT gates) [1] required to implement the Monte Carlo quantum shower<sup>¶</sup>. This increase in qubit efficiency is a result of the logarithmic scaling of the position registers, and by extension the number of final state partons that can be simulated, in the quantum walk parton shower, compared to the linear scaling of the quantum Monte Carlo parton shower. Furthermore, the circuit depth of the quantum walk parton shower grows linearly with the number of steps in the shower, compared to the quadratic scaling of the quantum Monte Carlo parton shower.

<sup>¶</sup>Any implementation of a controlled unitary rotation has been counted as a CNOT gate.

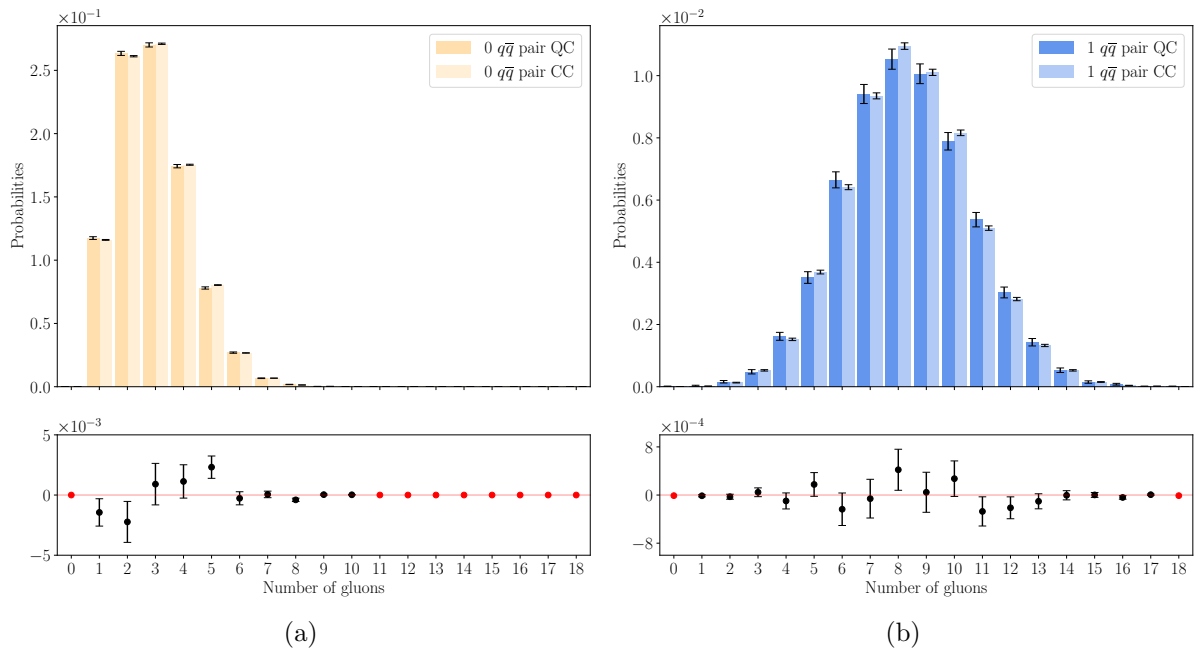


Figure 5.8: Comparison between the probability distributions for the number of gluons in the final state in association with zero quark-antiquark pairs (left) and exactly one quark-antiquark pair(right) for classical and quantum algorithms, using modified splitting kernels. The quantum algorithm has been run on the `ibmq_qasm_simulator` for  $10^5$  shots on the device, and the classical algorithm has been run for  $10^6$  shots. Figure adapted from Reference [2].

## Chapter 6

# Discrete QCD Parton Shower

The shower algorithms presented in Chapters 4 and 5 have calculated the shower evolution in the collinear limit only. In this limit, successive decay steps factorise into independent quasi-classical steps, which can be implemented on the quantum device using the Sudakov factor and the collinear DGLAP splitting kernels to simulate parton cascades. From Equation 2.3.2 we see that the cross section is enhanced as emission becomes collinear, therefore one can argue that the collinear limit is a good approximation of the cross section. However, as discussed in Section 2.3, the cross section is also enhanced by soft radiation,  $k^\mu \rightarrow 0$ , which is not necessarily collinear. In the soft limit, interference contributions only allow for partial factorisation, therefore one must construct a new scheme to deal with the interference patterns. Interestingly, the leading order contributions to the decay rate in the collinear limit are included in the soft decay rate. Therefore, we can utilise the soft limit as a starting point for the simulation of a parton cascade from a high energy collision. In this limit, the decay of a high energy state proceeds as a colour-anticolour dipole cascade, as outlined in Section 2.3.3.

To improve the realism of the quantum shower algorithms presented so far, one must devise a method which allows for the inclusion of soft emission in the quantum shower algorithm. In this chapter, we outline a quantum parton shower algorithm with the ability to simulate

a colour dipole cascade and produce synthetic data which is comparable to real data from an experiment. In Section 6.1 we describe a method that discretises QCD and allows for the dipole shower to be implemented on a NISQ device. The implementation of a quantum circuit for this shower is outlined in Section 6.2, and the first comparison between synthetic data generated by the `ibm_algiers` device and real data from a high energy collider experiment is presented in Sections 6.3 and 6.4.

## 6.1 Discretising QCD

In this section we perform several abstractions of the dipole shower presented in Section 2.3.3, leading to the discrete QCD model presented in Reference [90]. Through this discretisation, we will find that the shower can be described by creating and sampling from a set of primitive fractal structures, followed by a translation of the chosen primitive into scattering event momenta. It will be shown that the generation of the fractal structures has an elegant quantum circuit implementation, suitable for NISQ devices [3].

Following the method outlined in References [3, 90], we start by removing the independent treatment of the decay probability and the momentum space integration. This is achieved by absorbing the non-uniform probability density from Equation 2.3.12 into the integration measure; parameterising the phase space in terms of the gluon's transverse momentum,

$$k_{\perp}^2 = \frac{s_{ij}s_{jk}}{s_{IK}}, \quad (6.1.1)$$

and rapidity,

$$y = \frac{1}{2} \ln \left( \frac{s_{ij}}{s_{jk}} \right). \quad (6.1.2)$$

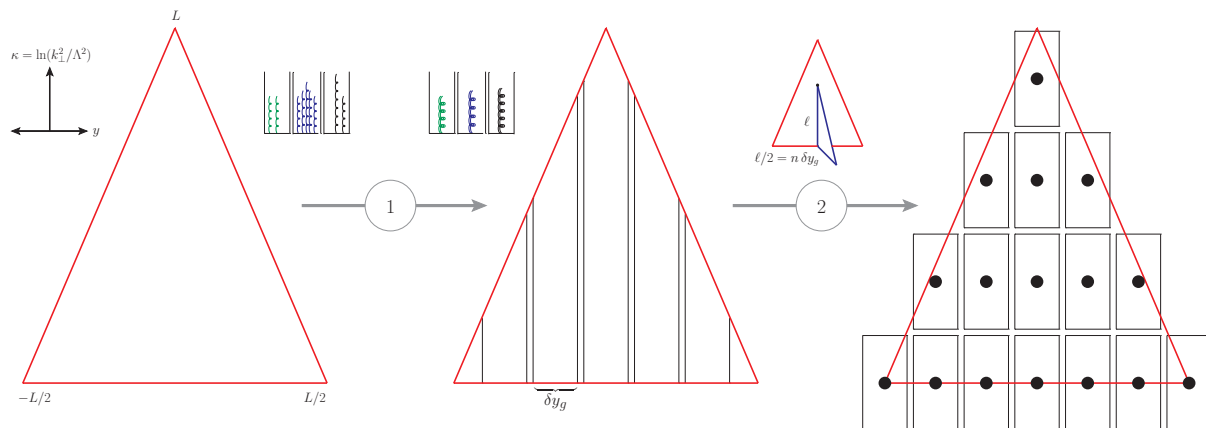


Figure 6.1: Parameterising the phase space in terms of the gluon's transverse momentum,  $k_{\perp}$ , and rapidity,  $y$ , allowed dipole decays are constrained to a triangular region of phase space. The phase space of effective gluon emission is discretised since ① the running-coupling effects dictate that gluons emitted within a rapidity gap of  $\delta y_g$  are reabsorbed, acting coherently as single effective gluon. The  $\kappa$  (proportional to  $k_{\perp}^2$ ) dimension is consequently quantised due to ② the additional phase space folds opening as a result of gluon emission also being discretised in units of  $\delta y_g$ . Figure from Reference [3]. See the main text and Reference [3] for more details.

This choice of phase space parameterisation allows for the simplification of the inclusive probability, such that Equation 2.3.12 becomes

$$d\mathcal{P}(q(p_I)\bar{q}(p_K) \rightarrow q(p_i)g(p_j)\bar{q}(p_k)) \simeq \frac{C\alpha_s}{\pi} d\kappa dy, \quad (6.1.3)$$

where we have defined  $\kappa = \ln(k_{\perp}^2/\Lambda^2)$ , introducing the arbitrary mass scale  $\Lambda^2$ . As a result, allowed dipole decays are constrained to a triangular region of height and baseline  $L = \ln(s_{IK}/\Lambda^2)$  in the  $(y, \kappa)$  phase space. The left hand section of Figure 6.1 illustrates this phase space region centred around  $y = 0$ .

An emission of a gluon from the dipole increases the phase space available to subsequent emissions. This increase in rapidity span is a direct consequence of the colour charge of the emitted gluon, and is described in detail in Reference [90]. The extra phase space can be conveniently *added* through a method called *folding out*, where the initial phase space triangle is extended in the positive  $y$  direction by an additional (folded) triangular phase space, called

a *fold*. Since the folds are themselves additional phase space regions which may contribute to subsequent decays, the process of folding out may repeat on the folds, leading to the construction of a fractal structure. An example of such a fractal structure is shown in the top left panel of Figure 6.2. Each fold from the parent triangle occurs at a height  $\kappa$  which is lower than the parent fold's highest  $\kappa$  value. Therefore, eventually the folds will be small enough to terminate the shower. This fractal structure is the basis of successful classical dipole shower algorithms such as ARIADNE [91, 92].

Through the phase space parameterisation and the folding out method, we already see the simplification of the shower process. Complicated splitting kernels are subsumed into the phase space parameterisation, and importantly, for fixed coupling  $\alpha_s$ , the inclusive decay rate is uniformly distributed in the  $(y, \kappa)$ -plane. This approach therefore allows for straightforward sampling algorithms. However,  $\kappa$  and  $y$  are still treated as continuous variables, thus the dipole shower is not yet suited for quantum circuit implementations on current quantum devices.

The shower model can be simplified even further by examining the effect of the transverse momentum dependent running coupling. Changing from a constant coupling to a running coupling allows for higher order corrections, which stem from *vertex* and *self-energy* corrections, to be accounted for in the shower calculation. In other words, one is able to take into account the *polarisability properties of the vacuum*, *id est* newly emitted gluons are surrounded by a set of virtual partons, which can be emitted and/or absorbed [90]. Therefore, the vacuum can influence the dynamics of emitted gluons.

We can write the transverse momentum dependent running coupling as

$$\alpha_s(k_{\perp}^2) = \frac{12\pi}{33 - 2n_f} \frac{1}{\ln(k_{\perp}^2/\Lambda_{\text{QCD}}^2)}, \quad (6.1.4)$$

where the  $n_f$ -dependent term arises from the  $g \rightarrow q\bar{q}$  splitting. Neglecting the splitting of a gluon



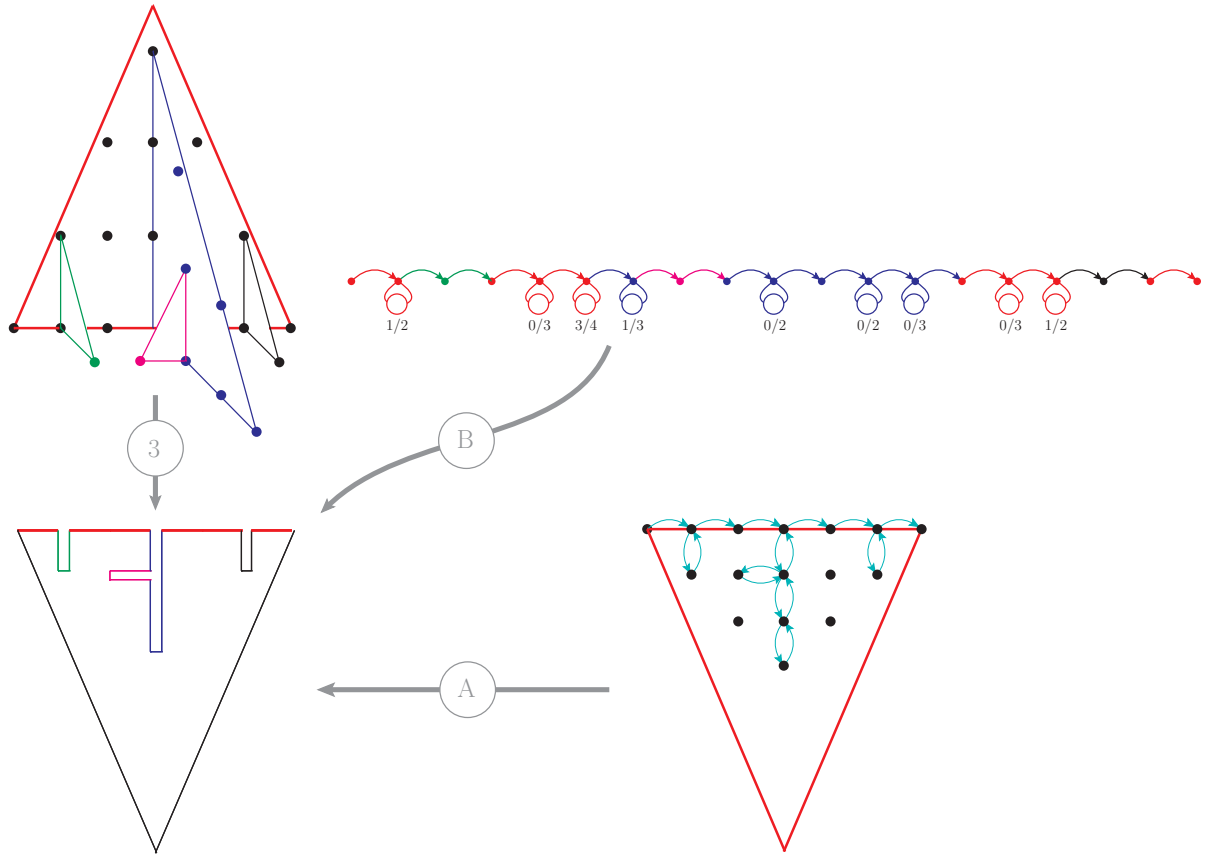


Figure 6.2: The discrete QCD parton shower produces a fractal structure of folds extending from an original triangular phase space region. The algorithm can be reinterpreted as a one-dimensional random walk since ③ the baseline of the fractal encodes all necessary information. The *grove* structure can be ① generated by a heavily constrained two-dimensional random walk that moves through a lattice with spacing  $\delta y_g$ . Due to the low fractal dimension of the baseline, ② the grove structure can be constructed from a one-dimensional random walk. For the one-dimensional walk ②, the notation  $n/n_{\text{tiles}}^{\text{slice}}$  indicates that tile  $n$  at height  $h = n\delta y_g$  has been selected. The dipole invariants ( $\ln(s_{ij})$ ) can be read directly from the grove by following the baseline from particle  $i$  to particle  $j$ , and skipping segments whose colour was created and reabsorbed along the way (*exempli gratia* skipping the pink segments when calculating the invariant mass of the green and blue gluon tips.). Figure from Reference [3].

to quarks, and identifying the arbitrary mass scale in  $\kappa$  as  $\Lambda_{\text{QCD}}^2$ , we can write Equation 6.1.3 as

$$d\mathcal{P}(q(p_I)\bar{q}(p_K) \rightarrow q(p_i)g(p_j)\bar{q}(p_k)) \simeq \frac{d\kappa}{\kappa} \frac{dy}{\delta y_g}, \quad (6.1.5)$$

where  $\delta y_g = 11/6$ , and in the leading-colour limit  $C \rightarrow C_A/2 = 3/2$  for any dipole decay.

As argued in detail in Reference [90], one can interpret the running coupling renormalisation group equation as a gain-loss equation. Through this interpretation, one finds that gluons that are within a rapidity range  $\delta y_g$  reabsorb and act coherently as one single *effective gluon*, as illustrated by ① in Figure 6.1. Therefore, the baseline of each rapidity region is discretised into *slices* of width  $\delta y_g$ , corresponding to a “distance” between two effective gluons which will not be reabsorbed again. The rapidity span has now been discretised, however  $\kappa$  is still a continuous variable. Returning to the folding out mechanism, we see that the additional fold extends into positive  $y$  by  $l/2$ , where  $l$  is the “height” in phase space that the new fold is produced from, *id est* the  $\kappa$  value of the emitted effective gluon. The rapidity span of this additional fold is also discretised into slices of width  $\delta y_g$ . Consequently, through the relation  $l/2 = n\delta y_g$ , the  $\kappa$  range is discretised into steps of  $2\delta y_g$ . Thus, the dipole shower can be modelled by generating effective gluons at the centre of *tiles* of dimension  $\delta y_g \times 2\delta y_g$ , which cover the phase space triangle, and subsequent folds. The right panel of Figure 6.1 shows the quantised triangular phase space available to dipole decays. The abstractions that brought us to this discretised phase space are the basis of the discrete QCD model presented in Reference [90].

Each rapidity slice is now made up of a number of tiles,  $n_{\text{tiles}}^{\text{slice}}$ , and each slice can be treated separately. The exclusive decay rate can now be calculated by substituting Equation 6.1.5 into the second term of Equation 2.3.13, achieving the simple form

$$dt d\xi \frac{d\phi}{2\pi} C \frac{\alpha_s}{2\pi} \frac{2s_{ik}(t, \xi)}{s_{ij}(t, \xi)s_{jk}(t, \xi)} \Delta(t_n, t) = \frac{d\kappa}{\kappa} \exp\left(-\int_{\kappa}^{\kappa_{\max}} \frac{d\bar{\kappa}}{\bar{\kappa}}\right) = \frac{d\kappa}{\kappa_{\max}}. \quad (6.1.6)$$

Therefore, the  $\kappa$ -value of the emitted effective gluon from each rapidity slice is simply (number of tiles)<sup>-1</sup>, corresponding to the probability of selecting a specific tile at height  $\kappa$ . This is remarkably straightforward in comparison to conventional parton shower approaches, which sample the non-emission probability,  $\Delta$ , using the so called *veto* algorithm [32, 33, 93, 94].

The discrete QCD method can then be applied for each slice in the initial triangular phase space region. Once the effective gluons have been selected, the new triangles can be folded out from the initial triangle. These will also be discretised into tiles of dimension  $\delta y_g \times 2\delta y_g$ , and the algorithm can be applied recursively for all subsequent folds. A fractal structure will then be constructed, as shown in the top left of Figure 6.2. However, the algorithm can be simplified even further by recognising that the height and  $y$ -range of each fold is redundant. All the information that is needed to calculate the momentum invariants,  $s_{ij}$ , can be read directly from the baseline of the fractal structure, represented by ③ in Figure 6.2. The shortest distance along the baseline between two “turning points”  $i$  and  $j$  can be shown to equal  $\ln(s_{ij}/\Lambda^2)$  [3]. From now on we will call this baseline a *grove*. Together with knowledge of the overall centre-of-mass energy, and uniformly sampled azimuthal decay angles  $\phi$ , the grove is sufficient to reconstruct post-decay kinematics of the shower.

Examining the algorithm further, we see that the grove structures are confined to a triangular region which is smaller than the original triangle, and that points on the baseline are separated from their nearest neighbours by  $\delta y_g$ , as shown in the bottom right of Figure 6.2. Therefore, one can construct a lattice of points within the area of the original triangle, through which a two-dimensional random walk can move to generate a grove structure. The horizontal movement of

the walker through the lattice shown in the bottom right of Figure 6.2 represents the algorithm moving to the next rapidity slice in the original fold. Any vertical movement of the walker then represents folding out an additional phase space triangle from the original triangle, and by extension the emission of an effective gluon. However, the dynamics of this two-dimensional walk are heavily constrained since it cannot extend beyond arbitrarily small or large  $\kappa$  values, and the  $y$ -value of each step cannot decrease. Therefore, the simplest algorithm to generate the grove structures is a one-dimensional random walk with variable step length, represented by the schematic shown in the top right of Figure 6.2.

The one-dimensional random walk proceeds by starting at the left most part of the grove. This initial position corresponds to the quark of the quark-antiquark dipole, and is marked as such. A single recursive step of the algorithm then starts; moving along the baseline by  $\delta y_g$  with unit probability, representing moving to the next rapidity slice in the original phase space triangle. The number of tiles in the slice,  $n_{\text{tiles}}^{\text{slice}}$ , is calculated, and a tile is selected such that the effective gluon is emitted at height  $h = n\delta y_g$ , with  $n \in \{0, 1, \dots, n_{\text{tiles}}^{\text{slice}} - 1\}$ . The probability of selecting a specific tile in the slice is  $P = 1/n_{\text{tiles}}^{\text{slice}}$ . If the tile at  $h = 0$  has been selected, then the walker remains stationary. If the tile at height  $h = n\delta y_g$ , for  $n \neq 0$ , then the walker “steps into the fold”, representing the folding out of an additional phase space, shown by a change of colour in Figure 6.2. This is simulated by the walker taking  $n$  steps to represent the new fold. The position is marked as an effective gluon. The walker then “steps out of the fold”, moving back to the baseline of the original triangle by taking a further  $n$  steps. This recursive step is then repeated for all rapidity slices in the original fold, until the walker reaches the right most part of the grove structure, when this position is marked as the antiquark from the dipole.

The algorithm described above simulates all folds from the original phase space triangle, however it currently neglects emission from the secondary folds, extended from the initial phase space triangle. To remedy this, when stepping into and out of the fold, the algorithm is applied in

the same way, allowing for the emission of effective gluons from subsequent folds. As a result all possible grove structures allowed for the dipole shower can be generated using this random walk approach. This very compact algorithm is inspired by considering the constraints for a quantum implementation, which will be outlined in Section 6.2. The moderate resources available from NISQ devices favour compact algorithms, which require shallow circuit depths and only have a small number of computational qubits. The simplification of the classical algorithm as a result of these considerations is an excellent example of the synergy between classical and quantum algorithmic design, and is a major result of Reference [3].

### 6.1.1 Generating Scattering Events from Groves

Once the grove has been constructed, the event data for the scattering event can be generated from the information encoded in the structure of the grove. The post-decay momenta are created iteratively from the grove, starting with the highest- $\kappa$  effective gluons first\*. For every effective gluon  $j$  emitted from a dipole  $IK$ , we can read off the invariant dipole masses  $s_{ij}$ ,  $s_{jk}$ , and  $s_{IK}$  from the grove. The post-branching momenta are then produced by generating a uniformly distributed azimuthal angle,  $\phi$ , and employing the momentum mapping from Reference [95], which provides a scheme that does not distinguish between individual partons, thus matching the dipole shower well.

To alleviate some of the most prominent effects of the phase space discretisation, we follow the strategy of Reference [90] and distribute the  $y$  and  $\kappa$ -values of the effective gluons in the highest and second highest tiles of each rapidity slice. This ensures that we are sampling from a good model for the highest  $\kappa$  gluons. The method relies on the fact that the  $(y, \kappa)$ -plane is uniformly covered as a result of Equation 6.1.3, thus the selection probability for each of the tiles is proportional to their area. If a tile protrudes outside of the allowed triangular phase space region, then the probability of selecting that tile would be less than a tile which is fully inside

---

\*Going from top to bottom in the triangular phase space region.

the allowed phase space region. In the random walk algorithm, this caveat is neglected, and all tiles have the same probability of being selected. This difference is corrected in post-processing by weighting each event with a factor

$$w = \prod_{i_s} \frac{(\text{area of tile picked in slice } i_s)/(\text{area of slice } i_s)}{(\text{number of tiles in slice } i_s)^{-1}}. \quad (6.1.7)$$

To truly appreciate the improvement provided by the quantum-inspired algorithm, we note that all possible grove structures can be enumerated for a fixed centre-of-mass energy as a result of the discretisation of the  $(y, \kappa)$  plane. Therefore, one can use the random walk algorithm outlined above to “pre-compute” all grove structures and their rates before simulating event data. Event generation then becomes the simple and efficient process of selecting a grove structure according to its probability, and then setting up the particle momentum vectors defined by the grove. Furthermore, the event generation step does not need to know “how” the groves and their rates have been generated. Consequently, we can use an elegant quantum algorithm to generate the groves, outlined in Section 6.2.

## 6.2 Quantum Circuit Implementation

The discrete QCD parton shower presented in Section 6.1 has been shown to have a compact one-dimensional random walk implementation that improves the efficiency of the approach outlined in Reference [90]. Utilising the quantum walk with memory framework from Section 3.2.2, the discrete QCD model has an elegant implementation on a quantum device. In this section we outline a general and extendable circuit implementation of the discrete QCD model on a quantum device, laying the groundwork for a quantum algorithm with the ability to generate synthetic data comparable to real data from the Large Electron Positron collider, presented in Section 6.3. This is the first quantum assisted event generator that can produce realistic collider

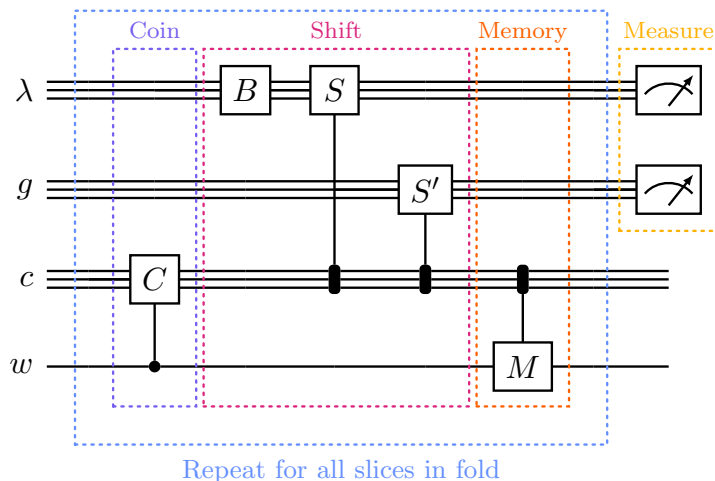


Figure 6.3: Schematic circuit of the quantum discrete QCD algorithm constructed from four registers: the grove-baseline register,  $\lambda$ , the gluon register,  $g$ , the coin register,  $c$ , and the memory register,  $m$ . The algorithm is based on a quantum walk with memory framework and is discretised into individual steps comprising three main operations: the coin operation, which encodes the probability of selecting a specific tile in the rapidity slice, the shift operation, which moves the walker according to the outcome of the coin operation, and the memory operation, which can be used to record previous coin outcomes. The step is then applied iteratively for all rapidity slices in the phase space before a measurement of the  $\lambda$  and  $g$  registers is performed. See the main text for more details on each operation. Figure has been adapted from Reference [3].

pseudo-data, and examines the possibility of a new paradigm in event generation.

The circuit is constructed from a quantum walk with one-dimensional memory (*id est* the memory register consists of one qubit.), where the Hilbert space of the walker is of the form shown in Equation 3.2.12. Here, we separate the position space,  $\mathcal{H}_P$ , into two separate subspaces: the grove-baseline space,  $\mathcal{H}_\lambda$ , in which information of the grove structure is encoded, and the effective gluon space,  $\mathcal{H}_g$ , which records the emission of an effective gluon. The position space is augmented by the coin space,  $\mathcal{H}_C$ , and the memory space,  $\mathcal{H}_M$ . The quantum circuit is constructed from four registers which correspond to each of the Hilbert spaces: the grove-baseline register,  $\lambda$ , the gluon register,  $g$ , the coin register,  $c$ , and the memory register,  $m$ . A schematic of the quantum circuit is shown in Figure 6.3.

The algorithm follows the framework of the quantum walk with memory, and is discretised

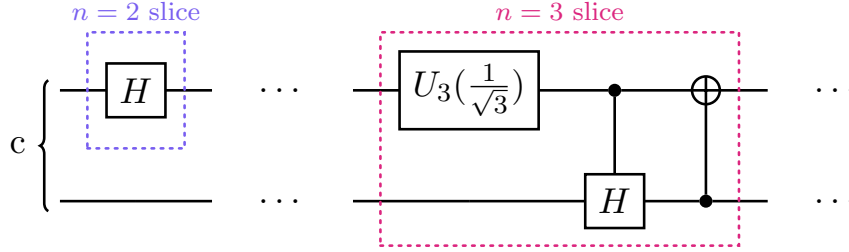


Figure 6.4: Coin operations for (left)  $n = 2$  and (right)  $n = 3$  tiles in a rapidity slice. Figure adapted from Reference [3].

into individual steps comprising three main operations: the coin operation, the shift operation and the memory operation. Following the structure of the one-dimensional random walk from Section 6.1, the algorithm is initialised in the left-most position of the phase space region, shown in Figure 6.2, and will propagate through each rapidity slice to simulate the shower process.

### 6.2.1 Coin operation

A step begins with the coin operation,  $C$ , which constructs an equal superposition of  $n$  states on the coin register of  $m$  qubits, such that

$$C |0\rangle^{\otimes m} = \frac{1}{\sqrt{n}} \left( |0\rangle + |1\rangle + \dots + |n-1\rangle \right), \quad (6.2.1)$$

where  $n = n_{\text{tiles}}^{\text{slice}}$  for each rapidity slice, and the states  $|i\rangle$  are represented in the computational basis,  $|q_1 q_2 \dots q_m\rangle$  with  $q_i \in \{0, 1\}$ . At each step, the coin register now holds the probabilities for selecting a tile in the slice.

Note that, for each slice, only  $\log_2 \left( n_{\text{tiles}}^{\text{slice}} \right)$  qubits are required to build an equal superposition of  $n_{\text{tiles}}^{\text{slice}}$  states. Consider a slice which has two tiles, *exempli gratia* the second slice from the left in Figure 6.1. Only one qubit is required to construct the state



$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad (6.2.2)$$

on the coin register, such that  $|0\rangle$  corresponds to picking the slice at  $h = 0$ , and  $|1\rangle$  corresponds to picking the tile at  $h = \delta y_g$ . This is constructed using a single Hadamard gate operation on the first qubit of the coin register, as shown in the left-hand gate group of Figure 6.4.

As shown in Figure 6.1, as the walker moves along the baseline to rapidity slices with more than two tiles, a more complicated coin operation is going to be needed. Consider a slice with three tiles, the required coin operation yields the state

$$\frac{1}{\sqrt{3}}(|00\rangle + |10\rangle + |01\rangle), \quad (6.2.3)$$

which is constructed on a two qubit basis. This can be achieved using the right-hand gate group of Figure 6.4. A similar approach can be used to construct the superposition with  $n$  states from Equation 6.2.1 on  $m$  qubits.

For a centre-of-mass energy,  $E_{\text{CM}} = 91.2$  GeV, typical of collisions at the Large Electron Positron collider, the initial triangular phase region has only five slices, with a maximum of three tiles in the centre slice. Therefore, to construct the algorithm to produce synthetic data for the Large Electron Positron collider, the coin operation can be constructed from the circuits shown in Figure 6.4. To implement the shower correctly, the coin can be controlled by a memory register. For  $E_{\text{CM}} = 91.2$  GeV, the memory operation is needed only once, and this will be described in detail below.

## 6.2.2 Shift Operation

Following the framework of the quantum walk with memory, having implemented the coin operation, the next task is to implement the shift operation depending on the outcome of the

coin. To simulate the discrete QCD parton shower, we divide the shift operation into three sub-operations, two of which act on the sub-space  $\mathcal{H}_\lambda$ , and the final operation acts on the sub-space  $\mathcal{H}_g$ .

The first of the sub-operations is the baseline operation,  $B$ , which increments the walker's position in  $\mathcal{H}_\lambda$ , corresponding to moving the walker along the baseline of the initial phase space triangle, to the next rapidity slice. This happens at each step of the algorithm, as the walker must always move to the next slice in the initial triangle. The implementation of this in the quantum algorithm is evident from the circuit diagram shown in Figure 6.3, where the  $B$  operation is applied irrespective of the outcome of the coin.

The second sub-operation is the  $\lambda$ -shift,  $S$ , which propagates the walker in  $\mathcal{H}_\lambda$  depending on the outcome of the coin operation. The  $\lambda$ -shift corresponds to folding out from the initial phase space triangle, shifting the walker by  $2n\delta y_g$  when the coin operation yields the  $|n\rangle$  state. The third sub-operation, the gluon shift,  $S'$ , is then applied to encode the emission of an effective gluon in  $\mathcal{H}_g$ .

It is possible that subsequent emissions can occur from the additional folded out phase spaces. To calculate these correctly, the quantum discrete QCD parton shower algorithm is applied recursively to the subsequent folds, but without the baseline sub-operation to avoid double counting in the  $\lambda$ -measure. The shift operation therefore encodes information about the grove structures in full superposition on the quantum device. This information can be retrieved by measuring the  $\lambda$  and  $g$  registers to reconstruct the scattering events.

All three sub-operations can be implemented using a circuitry scheme to increment the position of the walker in the computational basis, such as the one shown in Figure 5.2. The circuit depth of the shift operation at each step depends on the number of tiles in the slice being examined. Therefore, the circuit depth of each shift operation differs at each step: first increasing, until the walker reaches the centre rapidity slice, and then decreasing, after. Therefore, the total circuit

depth of the algorithm does not scale with each step, like the algorithms shown in Chapters 4 and 5, but instead scales as a function of the centre-of-mass energy<sup>†</sup>. This will be discussed in Section 6.3, when implementing the shower algorithm for a centre-of-mass scale  $E_{\text{CM}} = 91.2 \text{ GeV}$  to generate synthetic data comparable to experimental data from the Large Electron Positron collider.

### 6.2.3 Memory Operation

The final operation of the step is then the memory operation,  $M$ , which can be optionally applied at each step to record the previous outcome of the coin operation. Memory of the previous coin outcomes allows for conditional coin operations to be applied, ensuring that the state in Equation 6.2.1 can be correctly implemented onto the coin register.

Once again, there is an amount of freedom available in how the memory operation can be implemented. For  $E_{\text{CM}} = 91.2 \text{ GeV}$ , the memory operation need only be applied once. Therefore, the simplest implementation is to use the additional coins method described in Chapter 5.

Combing the coin, shift, and memory operations, the step operation can then be applied iteratively for all rapidity slices in the original phase space region, until the walker reaches the right-most position on the grove. At the end of the algorithm, information about the generated grove structure can be retrieved from the device by measuring the grove-baseline register,  $\lambda$ , and the gluon register,  $g$ . This can then be passed to the event generation process described in Section 6.1 to produce synthetic data from the information stored within the grove structure.

## 6.3 Generating Collider Events on a Quantum Computer

The general quantum algorithm shown in Section 6.2 is the first of its kind to be comparable to real, archival collider data. In this section we will demonstrate this by designing the algorithm

---

<sup>†</sup>The rapidity span available to gluon emission is proportional to the dipole centre-of-mass, as shown in Equation 6.1.2

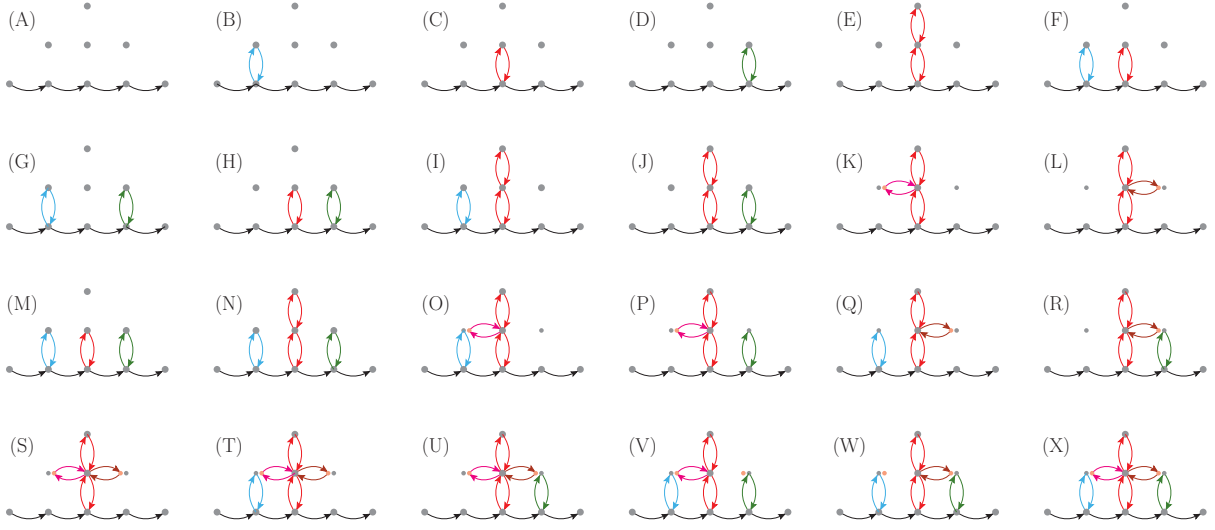


Figure 6.5: Set of all possible primitive grove structures for  $E_{\text{CM}} = 91.2$  GeV. The groves progress from (A) minimal emission, to (X) maximal emission. Figure from Reference [3].

to simulate the evolution of a quark-antiquark final state produced at  $E_{\text{CM}} = 91.2$  GeV, typical of events observed at experiments on the Large Electron Positron collider. As described in Section 6.1, a side-effect of the discrete QCD method is that all possible grove structures can be enumerated and sampled from. For  $E_{\text{CM}} = 91.2$  GeV, there are a total of 24 primitive grove structures which can be generated<sup>‡</sup>, with a tertiary fold being the maximum fractal depth. The 24 unique grove structures are shown in Figure 6.5 and labelled A-X, with A having no effective gluon emission, and X having maximal effective gluon emission.

To simulate collision data from the Large Electron Positron collider, the circuit requires 15 qubits (Five qubits for the  $\lambda$  register, six qubits for the  $g$  register, three qubits for the  $c$  register, and one qubit for the  $m$  register), and 116 gate operations (102 multi-qubit and 14 single qubit gate operations). For  $E_{\text{CM}} = 91.2$  GeV, the algorithm requires memory of dimension one, *id est* the memory register consists of only one qubit. In fact, the memory operation is only required once during the algorithm, after the first step, to return the coin register back to the ground state. Therefore, quantum diffusive effects are retained.

<sup>‡</sup>The number of unique grove structures that can be generated depends on the cutoff. Here, we have chosen  $\Lambda_{\text{QCD}} \in [0.1, 1]$  GeV.

The circuit is compact, requiring only a small number of qubits and it has a shallow circuit depth. The circuit is well within the capabilities of the `ibmq_qasm_simulator` device, which simulates a fully fault tolerant quantum computer. However, despite the efficient implementation of the quantum discrete QCD algorithm, the circuit presented in Figure 6.3 is still beyond the reach of NISQ devices, such as the `ibm_algiers` 27-qubit quantum computer, introducing large noise effects and producing low fidelity data. Therefore, the circuit must be streamlined in order to retrieve practical results from the quantum computer.

### 6.3.1 Tailoring the Quantum Circuit

For the specific example of  $E_{\text{CM}} = 91.2$  GeV, the circuit implementation from Figure 6.3 can be streamlined at the expense of generality and extendability, but at the benefit of a dramatic reduction in the quantum resources required. Knowing the pre-determined primitive grove structures from Figure 6.5, it is possible to remove the calculation of the baseline of the grove structure from the quantum circuit. The specific grove structure generated can then be interpolated from the gluon emission information encoded in  $\mathcal{H}_g$ , using the primitive grove structures, A-X. Therefore, reducing the position space,  $\mathcal{H}_P$ , by removing the  $\mathcal{H}_\lambda$  space, restricts the quantum walker to

$$\mathcal{H}' = \mathcal{H}_g \otimes \mathcal{H}_C \otimes \mathcal{H}_M, \quad (6.3.1)$$

and allows for the equivalent simulation of a parton shower from a quark-antiquark dipole with  $E_{\text{CM}} = 91.2$  GeV, whilst dramatically reducing the required circuit resources to 10 qubits and 21 gate operations (12 multi-qubit, and nine single qubit gate operations) only.

Crucially, the number of multi-qubit gate operations, which are decomposed into a series of CNOT gates, is decreased by a factor of five. The CNOT gate is associated with higher error rates than single qubit operations, therefore they are one of the major sources of quantum

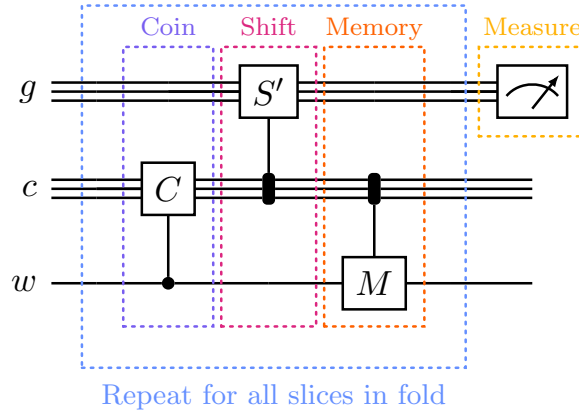


Figure 6.6: Schematic of the tailored circuit for generating grove structures for  $E_{CM} = 91.2$  GeV. Each step of the quantum walk with memory is constructed from three operations: the coin operation, which encodes the probabilities to selecting tile  $n$  from the rapidity slice, the shift operation, which adjusts the position of the walker if a gluon has been emitted, and the memory operation, which is used once to record the state of the coin after the first coin operation. Figure adapted from Reference [3].

error<sup>§</sup>. Reducing these operations, and the depth of the circuit, will improve coherence within the device, and allow for practical results to be retrieved from the quantum computer. The streamlined circuit is shown schematically in Figure 6.6.

### 6.3.2 Generating Grove Structures on a Quantum Device

Having streamlined the quantum circuit to dramatically reduce the circuit depth of the algorithm, the quantum discrete QCD model is now viable for implementation on a NISQ device. Figure 6.7 shows the results of the grove generation step, obtained from the 32-qubit `ibmq_qasm_simulator`<sup>¶</sup> and the 27-qubit `ibm_algiers` quantum computer. The quantum results have been directly compared to an analytically calculated grove generation rates.

The `ibmq_qasm_simulator` has been run for 100,000 shots of the algorithm and shows good agreement with the analytically calculated grove rates, as expected from a fully fault tolerant

<sup>§</sup>Appendix C discusses the main sources of error on quantum devices.

<sup>¶</sup>Note that the `ibmq_qasm_simulator` has been run using the full algorithm presented in Section 6.2 as a proof of principle.

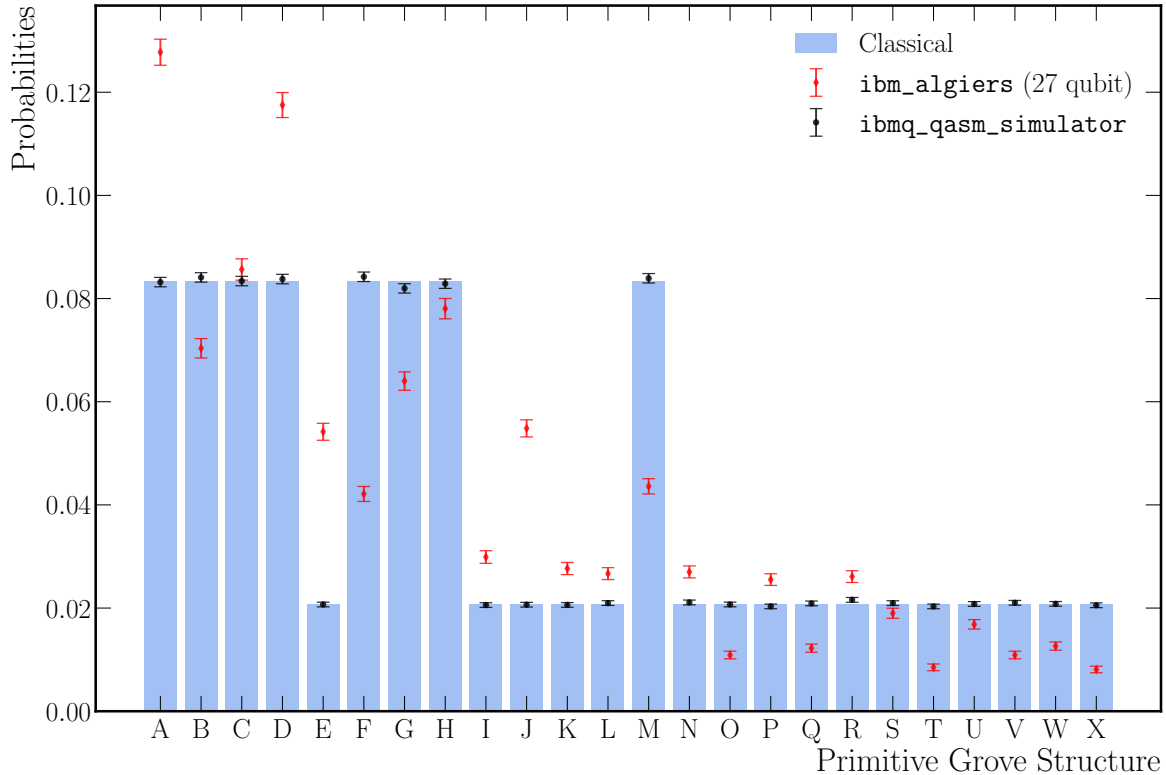


Figure 6.7: Generation of the primitive grove structures for a centre-of-mass energy of  $E_{\text{CM}} = 91.2$  GeV using the `ibmq_qasm_simulator` and `ibmq_algiers` devices. The results from the simulator and quantum computer have been compared to analytically calculated classical distribution. Figure from Reference [3].

simulation of a quantum device. As a proof of principle test, the grove structures generated by the `ibmq_qasm_simulator` have been fed into the event generation step from Section 6.1 and compared to the classical discrete QCD algorithm. The results are shown in Figure 6.8. We see an exact agreement between the classical algorithm and quantum simulation, demonstrating the ability of the quantum algorithm to successfully simulate the discrete QCD model.

The 27-qubit `ibmq_algiers` device has been run for 20,000 shots using the `ibm_cloud` platform to generate grove structures for a centre-of-mass energy  $E_{\text{CM}} = 91.2$  GeV. From the comparison in Figure 6.7, it is clear that the `ibmq_algiers` device returns noisy results, differing greatly from the simulator and analytical rates for the grove structures. The main source of error is likely due to the number of CNOT gate operations in the algorithm. The CNOT gate is associated with a

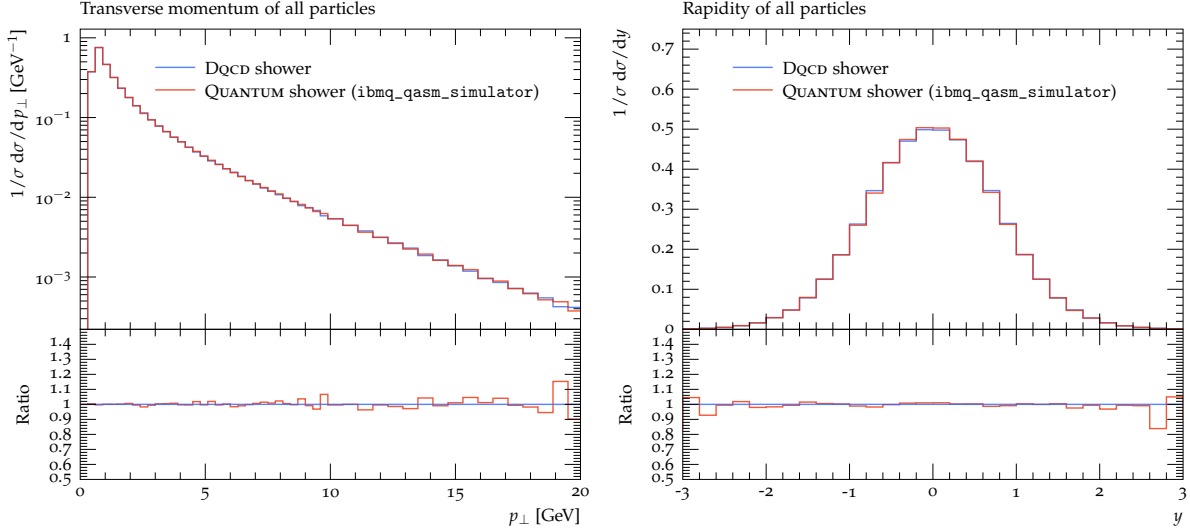


Figure 6.8: Comparison between the results obtained from the classical discrete QCD algorithm, and the quantum discrete QCD algorithm. The quantum algorithm has been run on the `ibmq_qasm_simulator` device, which simulates a fully fault tolerant device. Thus, as expected, we see exact agreement between the two algorithms. Figure from Reference [3].

higher implementation error than single qubit gate operations, and can introduce environmental noise into the system, resulting in decoherence. Consequently, the number of CNOT gates needed in the circuit directly impacts the fidelity of the results.

Furthermore, an interesting artefact of the `ibm.qalgiers` results in Figure 6.7 is the tendency to emit too few gluons. This propensity towards groves with fewer gluons is likely due to the relaxation time of the qubits in the gluon register,  $g$ . The relaxation time, called the *T1 time*, is the time it takes for an excited qubit to decay back down to the ground state,  $|0\rangle$ . For the tailored circuit in Figure 6.6, the  $|0\rangle$  state in the gluon register,  $g$ , is associated with an absence of emission. It is therefore likely that this preference towards groves with few emissions is due to the circuit architecture, and that the T1 time of some of the gluon register qubits is less than the algorithm run time.

A detailed discussion of quantum errors relating to this algorithm is given in Reference [3] and Appendix C. It should be noted that is possible to employ error mitigation schemes to suppress common errors such as readout and CNOT errors, and there are many examples in the



literature [96–98]. However, it will be shown below that, remarkably, error mitigation is not needed for the quantum discrete QCD algorithm.

### 6.3.3 Event Generation using the Quantum Device

Having generated the grove structures using the `ibm_algiers` device, we can now use the results to generate scattering data and compare measurements of selected kinematic distributions performed on these simulated data with archival collider data. We will highlight and examine different aspects of the simulation by comparing both the event shape and jet observables. To demonstrate the capability of the quantum discrete QCD shower algorithm, we will consider four main measurements: *thrust* [99], *energy-energy correlation* (EEC) [100], *Durham 2-jet rate*,  $y_{23}$  [101], and *jet mass difference*,  $M_D$  [99].

The event shape variable thrust,  $T$ , characterises the “shape” of an event, for example giving a measure of whether the distributions of hadrons is “pencil-like” ( $T \approx 1$ ) or “planar-circular” ( $T \approx 0.5$ ) [33]. Therefore, studying thrust values as  $T \rightarrow 1$  is an excellent probe of the modelling of soft and collinear emission in the parton shower, and more moderate  $T$  values give a measure of the high-energy gluon emissions. Another event shape variable that can be used to probe the QCD evolution is the EEC, which measures the correlation of hadronic energy flow in an event [33]. If primary partons mostly retain their direction, as would be the case for the asymptotically free limit of QCD, then one would expect a particle at angle  $\chi$  to be accompanied by another particle at angle  $\chi + \pi$ . The parton shower then proceeds through collinear emission, one then would expect a cloud of collimated particles around the primary particle  $\chi$ . As well as collinear radiation, we expect soft radiation and strong-coupling QCD effects, which contribute to intermediate EEC values,  $-0.5 \lesssim \cos \chi \lesssim 0.5$  [3, 33].

The Durham 2-jet rate,  $y_{23}$ , is a jet observable that defines the “closeness” of the second and third-hardest jets in a collision event. Jets with high energy and large angular separation are said to be well separated, and have high  $y_{23}$  values. At such values of  $y_{23}$ , the spectrum is

dominated by higher order, real emission corrections. At intermediate values of  $y_{23}$ , the jets will start to coalesce, and the spectrum will become susceptible to parton shower evolution. For low values of the 2-jet rate, the jets will no longer be separated, and the distribution is dominated by non-perturbative effects [3]. The jet mass difference,  $M_D$ , is another jet observable that can be used to study the dynamics of the parton shower algorithm. The variable  $M_D$  measures the difference between the invariant mass of particles in the left and right hemispheres of the detector. If  $M_D \approx 0$ , the two hemispheres of the detector are equally populated, and thus  $M_D = 0$  describes perfectly balanced jets. Values of  $M_D \lesssim \frac{1}{3}$  probe the modelling of hard and multiple soft gluon emission, while large  $M_D$  probes multiple hard gluon emission [3].

We can now use these event shape and jet observables to determine the performance of the quantum discrete QCD parton shower algorithm run on the `ibm_almogiers` quantum computer. To facilitate the generation of the scattering data, we embed the quantum algorithm into a classical toolchain. First, the momentum distribution of the quark-antiquark pair in the process  $e^+e^- \rightarrow Z/\gamma^* \rightarrow q\bar{q}$  is determined classically for a centre-of-mass energy  $E_{\text{CM}} = 91.2$  GeV. The 27-qubit `ibm_almogiers` device then generates the parton shower evolution of the quark-antiquark dipole, generating the groves with the rates shown in Figure 6.7. These groves are then used to generate the kinematics of the system, and are stored in a Les Houches Event (LHE) file. This can then be passed to PYTHIA, which performs the parton-to-hadron conversion using the Lund string model [102]. Finally, RIVET [103] is then used to compare the synthetic data produced by the `ibm_almogiers` quantum device to data taken by experiments at the Large Electron Positron collider.

Figure 6.9 shows sample comparisons for the thrust, EEC, Durham 2-jet rate, and jet mass difference. To produce these comparisons, the parton shower mass scale,  $\Lambda$ , has been chosen as  $\Lambda = 0.4$  GeV, the parameters of the Lund symmetric fragmentation function [102] have been set to  $a = 2$  and  $b = 3/5$ , and the width of the non-perturbative string- $p_{\perp}$  has been set to

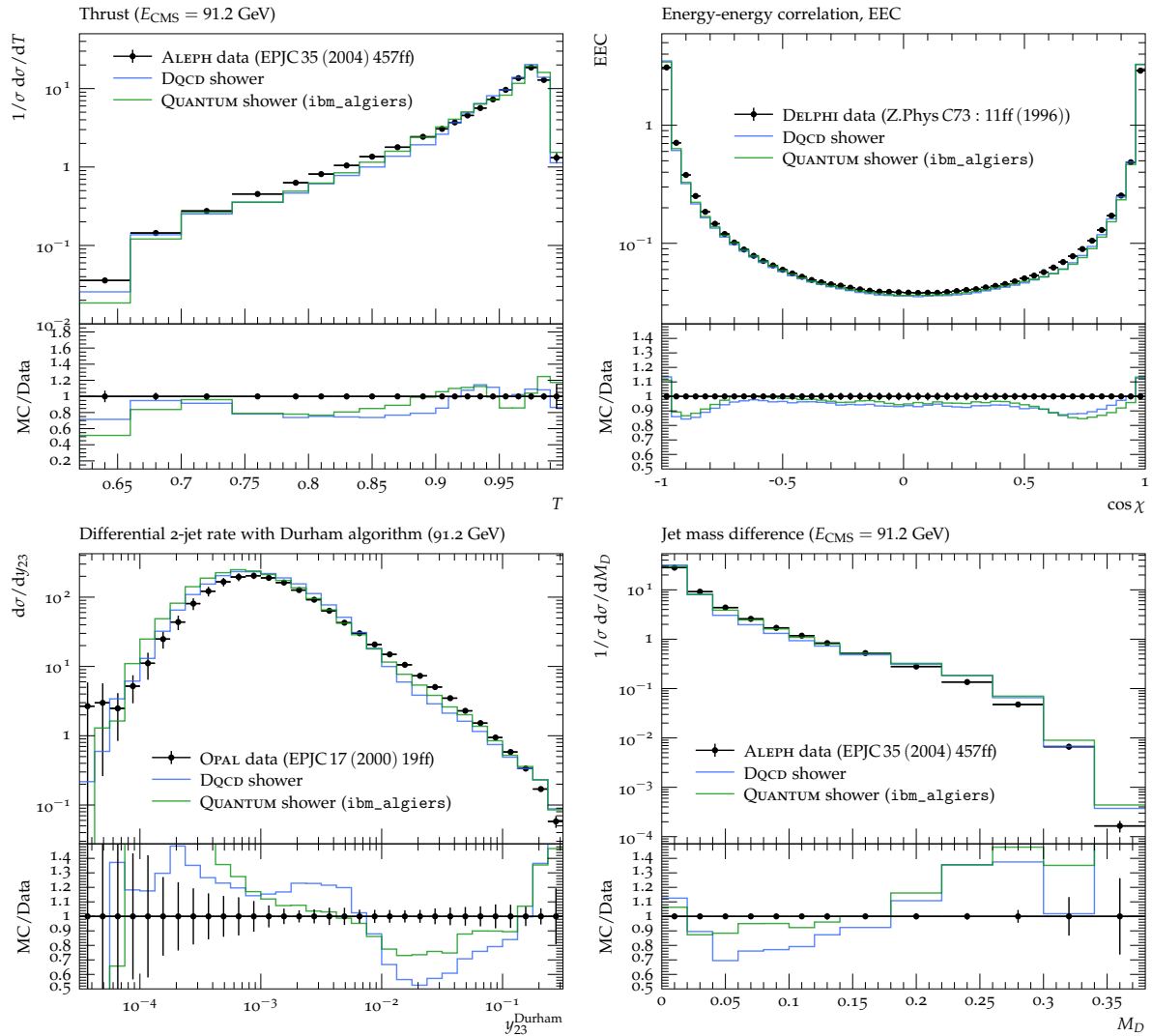


Figure 6.9: Sample comparisons of synthetic data produced by the discrete QCD model (DQCD) and the quantum algorithm run on the `ibm_algiers` device to data taken at the Large Electron Positron collider. The quantum parton shower results from the `ibm_algiers` device have not been corrected for quantum errors. Figure from Reference [3].

$\sigma = 0.1$  GeV [3].

First, we compare the classical discrete QCD parton shower, DQCD, presented in Section 6.1 to data from the Large Electron Positron collider<sup>‡</sup>. We see that the DQCD approach gives very satisfactory agreement with the archival data, especially given then simple implementation of algorithm. From the jet mass difference and Durham 2-jet rate plots, we see that the DQCD algorithm obtains an excess for states with multiple high-energy gluons. This is expected, as the rate of soft-gluon emission in the hard region is greater than the true matrix element calculation. Improving the description of the shower outside of the soft limit will improve the agreement in these areas. Furthermore, a better description outside the soft limit will also improve the thrust range  $0.75 \lesssim T \lesssim 0.9$ . We see from large  $T$  values, and moderate EEC values, that the description of soft gluon emission is well modelled.

Having examined the agreement between the DQCD model and the collider data, we can now examine the agreement between the DQCD model, the collider data and the synthetic data produced using the `ibm_algiers` device. We see that, once again, the agreement between the `ibm_algiers` device and the data is encouraging, even without error correction on the device. An interesting factor of the quantum results is that the tendency to generate grove structures with “too few” emissions actually slightly improves the overall data description. Although this would be removed with error mitigation, the agreement of uncorrected quantum data may hint at possible improvements of the classical algorithm, once again highlighting the synergy between quantum and classical algorithmic design.

### 6.3.4 Tuning Simulation Parameters

To avoid giving an artificially positive impression of the capabilities of the DQCD and quantum showers, so far no dedicated program to *tune* the parameters of the simulation has been de-

---

<sup>‡</sup>Note that the data description of the DQCD model is equivalent to the data description of the quantum simulation, as shown by the agreement in Figure 6.8.

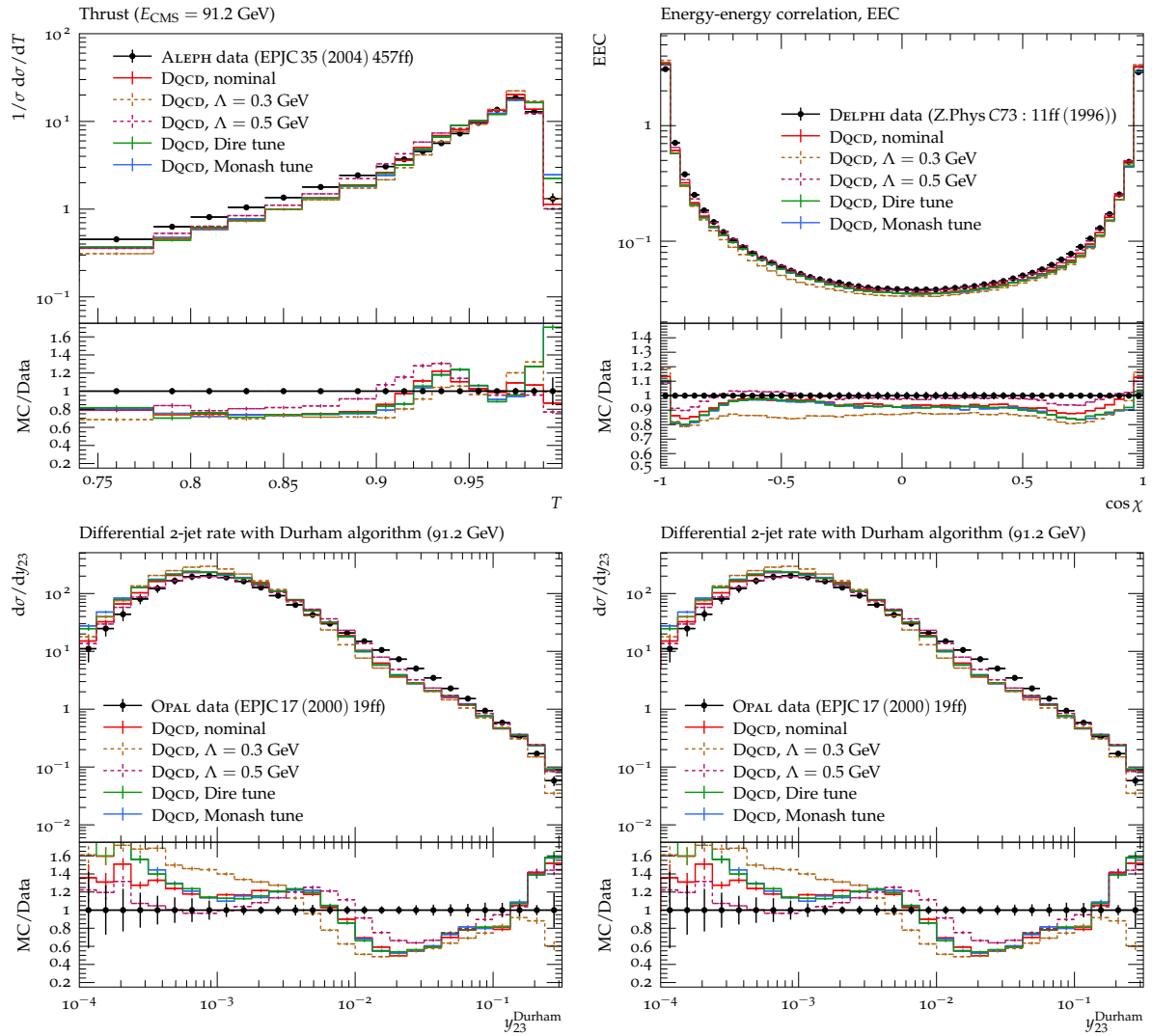


Figure 6.10: Sample comparisons of synthetic data produced by the discrete QCD (DQCD) model for various values of the mass scale,  $\Lambda$ . Variations of the mass scale have non-negligible uncertainties for observables that are dominated by perturbative QCD effects. Figure from Reference [3].

signed to improve the overall data description shown in Figure 6.9. This is the first quantum algorithm to be compared to “real-life” particle physics data [3], and even without tuning, the data comparison is very encouraging. However, it is interesting to examine the effects of tuning on the distributions from Figure 6.9.

The discrete QCD method models the entirely perturbative aspects of the showering process, with all non-perturbative parts of the evolution dealt with by the Lund string model in PYTHIA. As a result, the only tuneable parameter in the perturbative part is the mass scale  $\Lambda$ . Figure 6.10 shows that variations of this mass scale have non-negligible uncertainties. This is entirely expected from a leading-logarithm model.

We can examine the tuning of the non-perturbative model by applying changes to the PYTHIA tune. We see that these lead to small variations in the model, best seen in observables which are dominated by non-perturbative dynamics. Examples of such observables are shown in Figure 6.11, and present mild dependence on the perturbative mass scale  $\Lambda$ , but are sensitive to changes in the PYTHIA tune. We see that both the Monash [104] and DIRE [105] parton shower tunes yield a better data agreement than the nominal parameters outlined above. Therefore, and as expected, it is confirmed that a dedicated tuning of the DQCD shower can improve the description of the data [3].

## 6.4 Results and Overview

Here, we have presented a general and extendable approach to simulating realistic collider event data using a quantum device. To achieve this, we have re-interpreted the conventional parton shower framework, which relies on an intricate method of rejection sampling, as a one-dimensional random walk through the multi-gluon phase space. The novel algorithm is a reformulation of the discrete QCD model presented in Reference [90], and it has led to the development of an elegant quantum shower implementation, utilising the quantum walk frame-

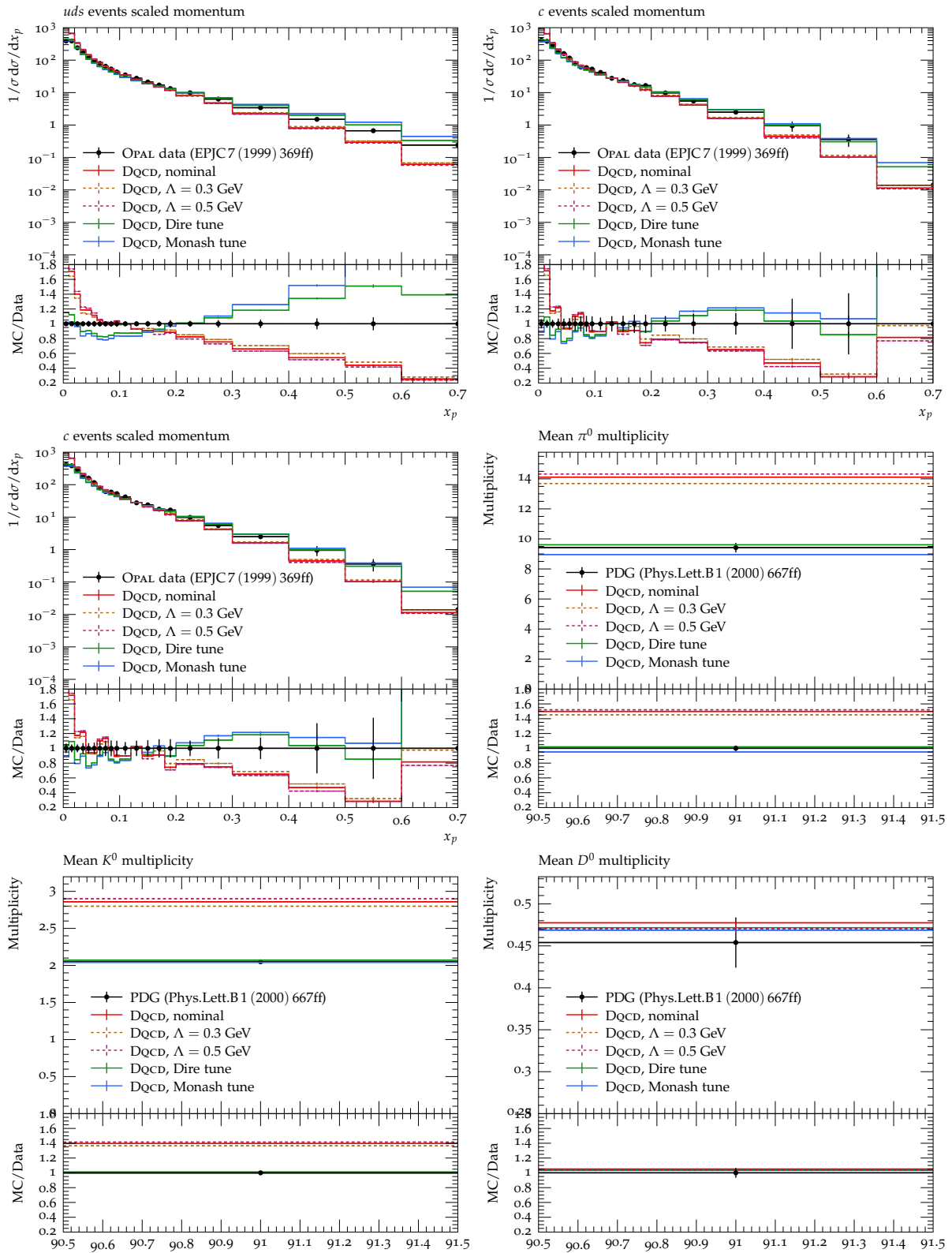


Figure 6.11: A comparative study on the affect of varying the tune for observables dominated by non-perturbative QCD effects. It is clear that applying the default PYTHIA Monash tune, or the DIRE tune leads to better data description. Figure from Reference [3].

work.

The algorithm provides a remarkable improvement on the realism of the parton shower in comparison to previously discussed shower algorithms in Chapters 4 and 5. It is the first quantum algorithm to produce a result comparable to “real life” particle physics data, and Figure 6.9 shows the encouraging data comparison achieved. Furthermore, it is the first time a realistic parton shower algorithm has been simulated using a real, quantum computer. The `ibm.algiers` device is a 27-qubit platform which uses an IBM Falcon r5.11 processor and is categorised as a NISQ device. Therefore, in order to obtain practical results from the device, the quantum shower algorithm has been streamlined; tailoring the implementation to data from the Large Electron Positron collider. Disagreement with the classical discrete QCD model, DQCD, has been identified as errors within the device, primarily from CNOT errors and the T1 time of the qubits. This is explained in detail in Reference [3] and Appendix C.



## Chapter 7

# Quantum Advantage and the Future of Quantum Parton Showers

When developing a quantum algorithm, it is interesting to consider what benefits one can gain from using the quantum device. With the ability to form highly entangled systems, quantum computers provide an opportunity to perform calculations in a regime never before accessible. Utilising this unique feature is the goal of any quantum algorithmic architecture, and can be used to achieve a speed up of classical approaches [57,58,71], or, and perhaps more fundamentally, to exploit the quantum nature of the system and improve the understanding of quantum mechanical systems [4,64].

In this chapter, we consider the potential quantum advantage of the parton shower algorithms outlined in Chapters 4, 5 and 6. Section 7.1 discusses the potential speedup that quantum walks can provide under certain conditions, and how this applies to the quantum walk parton shower from Chapter 5. Section 7.2 discusses the reduction in the need for error correction to achieve practical results from a quantum device, highlighted by the success of the quantum discrete QCD algorithm from Chapter 6. Finally, Section 7.3 looks to the future of event generation on a quantum computer.

## 7.1 Speedup via Quantum Walks

As has been shown in Section 2.3, conventional parton shower algorithms proceed via a Markov Chain Monte Carlo (MCMC) process to sample from the probability distribution of shower final states. Beyond high energy particle physics applications, MCMC algorithms are also the heart of modern computation and have a wide variety of applications in many fields\*. As a result, a speedup of MCMC approaches using a quantum device would be invaluable, and has been the subject of intense research [106–108]. Of particular interest to this thesis, quantum walks have been shown to provide a quadratic speedup of Monte Carlo methods, under certain conditions [73–78].

The mechanism underlying the speedup of MCMC approaches using quantum walks is the so-called Markov chain *mixing time*, defined as the time it takes for the algorithm to approach the equilibrium distribution [109, 110]. To understand the speedup from quantum walks, we will first consider the mixing time of a classical random walk described by the stochastic matrix  $M$ . The mixing time is directly related to the difference,  $\delta$ , between the largest and second largest eigenvalues of  $M$ , the so-called *spectral-gap* of the matrix [78, 79, 83]. For minimal  $\delta$ , the mixing time of the a classical random walk is proportional to  $\delta^{-1}$ . If we now examine a quantum walk described by the matrix  $M'$ , it can be shown that the spectral-gap,  $\delta'$ , of the quantum walk is quadratically larger than its classical counterpart [73, 78]. As a result, the quantum walk therefore has the potential to provide a quadratic speedup up in the mixing time, and by extension a quadratic speedup over classical MCMC algorithms [75, 76].

It should be noted that the speedup of any quantum algorithm is heavily dependent on the physical implementation of the circuit. Therefore, only in combination with an efficient circuit implementation can the quantum walk provide a quadratic speed up over classical MCMC algorithms. This speedup has not yet been proven in general for all quantum walk frameworks,

---

\*Despite their success, it is worth noting that MCMC algorithms are not optimal and often contribute to large computational overheads, such as the generation of synthetic data for high energy colliders.

and over all MCMC algorithms. However it is widely expected that this speedup will be the case in general, with recent results providing upper bounds on the mixing time of quantum walks have been estimated [111, 112].

### 7.1.1 Quantum Walk Parton Shower

There are many parallels between the quantum walk parton shower from Chapter 5, and the quantum walks which utilise the Szegedy quantisation of Markov chains [73, 113–115], a method which has been proven to provide the quadratic speed up in mixing times for reversible, ergodic Markov chains<sup>†</sup>. To illustrate this relation, consider the observable corresponding to the number of gluons in the final state,  $\langle n_g \rangle$ , which is encoded in the quantum walk parton shower as the position of the walker in the  $x$ -dimension of the position space,  $\mathcal{H}_P$ .

During the evolution of the shower, the number of gluons can be enhanced through the splittings  $g \rightarrow gg$  and  $q \rightarrow qg$ , or diminished through the splitting  $g \rightarrow q\bar{q}$ . As a result, the Markov chain representing  $\langle n_g \rangle$  can be modelled as a binary tree walk, with the ability to move left or right in the  $x$ -dimension at each step. Thus, the walker can be found in any state in the  $\langle n_g \rangle$  state after  $N$  steps with finite probability. Therefore, as there are no transient states, the Markov chain representing  $\langle n_g \rangle$  is irreducible. Furthermore, the parton shower process is aperiodic, and, by construction, the quantum algorithm is reversible. The Markov chain representing  $\langle n_g \rangle$  is thus reversible and ergodic, and there is a direction connection between the quantum walk parton shower and Szegedy quantisation of Markov Chains. As a consequence, we can postulate that the quantum walk parton shower has the potential to benefit from a quadratic speed up.

Here, we do not consider an analytical proof of such a speedup, and highlight some key differences between the quantum walk parton shower and Szegedy quantisation. Firstly, the approach taken by Szegedy [73] is not a coined quantum walk, and is an abstraction of the

---

<sup>†</sup>A Markov chain is said to be ergodic if it is aperiodic and irreducible

Grover algorithm [71], which is implemented using an oracle. It has been shown that Szegedy quantisation can be replicated by coined quantum walks [113, 114], and quantum walks with memory [115], however there are currently strict conditions on the coin and memory operations, with general architectures remaining an active field of research.

Without an analytical proof, one can only conjecture on the potential speedup on offer for the quantum walk parton shower from Section 3.2. Instead, the main finding of Reference [2] is the dramatic improvement in the quantum resources required to simulate the collinear, QCD shower over the quantum Monte Carlo shower.

## 7.2 Redundancy of Quantum Error Correction

A limiting factor for current quantum architectures is the amount of quantum resources available. In their infancy, gate quantum devices have a small number of qubits, with little fault tolerance and short coherence times. Furthermore, NISQ devices become untenable to quantum errors from many decoherence effects within the device, ultimately delivering low fidelity results. Therefore, a major challenge facing quantum circuit architects is how to design algorithms which will return practical results. One solution is to employ sophisticated error mitigation techniques to minimise the decoherence affects within the device.

The major result of the quantum discrete QCD parton shower from Chapter 6 is that no error correction schemes are required to obtain a good description of archival collider data [3], allowing for the algorithm to be implemented and run on the `ibm_algiers` NISQ device. Despite achieving a device output which is far from ideal (see Figure 6.7), due to the fact that the physical observables are supported by a mixture of several parton shower structures, the results are remarkably robust to quantum errors, as seen in Figure 6.9. Therefore, in conjunction with the improvements in physical shower aspects (inclusion of soft-gluon coherence, running coupling effects, and realistic scattering data generation) and the reduction in the required quantum

resources required compared to all other quantum shower algorithms, the quantum discrete QCD parton shower drastically alleviates the requirement for fully fault tolerant devices.

### 7.3 Towards a Fully Quantum Event Generator

Given the rapid improvement in the design of quantum circuit architectures for the simulation of parton showers, demonstrated by the progression from the two-step, collinear shower presented in Section 4.1, to the realistic dipole shower from Chapter 6, it is interesting to examine the future of event generation on a quantum device.

One of the obvious extensions of the discrete QCD algorithm is the addition of  $g \rightarrow q\bar{q}$  splittings, which are currently neglected in the shower. Reference [116] may provide inspiration on how to develop the discrete QCD model to achieve this, along with the addition of hard-collinear corrections, and sub-leading colour terms. Furthermore, a thorough programme of dedicated event generator parameter tuning is needed for the quantum shower algorithm to improve the data description. Although the addition of these aspects will undoubtedly ameliorate the current implementation of the discrete QCD model, to achieve a competitive model it will be necessary to incorporate precision QCD calculations into the discrete QCD framework through matching [117–119] and merging [120–124] methods [3].

To develop a fully quantum event generator, quantum algorithms will have to be devised for all steps of the event generation process. Already, there have been attempts to design quantum circuits with the ability to simulate each of the steps in the process. A quantum machine learning approach to determine the parton content of the proton is presented in Reference [125], estimating “quantum” parton density functions. Reference [1] proposes a circuit for sampling from the highest-momentum transfer interactions by exploiting the equivalence between the mathematical form of a helicity spinor and a qubit. The inclusion of colour calculations is an important feature of modern event generators, and an area which could benefit greatly

from quantum devices. Reference [126] presents the first steps towards the quantum simulation of the colour part of perturbative QCD. Finally, the event generator needs to sample from non-perturbative hadronisation models. Reference [127] shows results from a NISQ device, modelling string breaking in the simplified Schwinger model. Together with the quantum discrete QCD algorithm from Chapter 6, these advances give an encouraging outlook for quantum event generators.

## Chapter 8

# Conclusions and Outlook

The production of high quality synthetic data is crucial for the analysis and interpretation of measurements from high energy collider experiments. The theoretical description of proton-proton collisions at the Large Hadron Collider is highly complex, relying on event generator algorithms to construct realistic pseudo scattering event data. The cornerstone of the event generation process is the QCD parton shower, which evolves the system from the high momentum transfer of the hard process down to the hadronisation scale, where non-perturbative effects dominate. Together with the calculation of the hard process and phase space integral, the parton shower presents one of the bottlenecks in the production of synthetic data for collider experiments, becoming a limiting factor for the event generation process as we move to the era of higher energies and higher luminosities.

Quantum computing offers a potential solution, promising to provide a quantum advantage to a plethora of computational and theoretical problems in physics. Furthermore, all processes in high energy physics are inherently quantum mechanical, and thus particle physics is a natural testbed for quantum algorithmic design. In this thesis, we present three novel quantum computing algorithms for the simulation of the QCD parton showering process, following the development of the quantum simulation of parton showers in References [1, 2], summarised in

Chapters 4 and 5, leading to the first realistic quantum simulation of a QCD parton shower in Reference [3], summarised in Chapter 6. These algorithms provide the first step towards a fully quantum event generator that can benefit from the advantages that computing in the quantum regime can provide.

In Chapter 4 we present a proof-of-principle algorithm for the quantum simulation of a QCD parton shower, the quantum Monte Carlo parton shower. Inspired by the classical Monte Carlo approach, the quantum algorithm has the ability to simulate a simplified QCD model comprising one gluon and one quark flavour, which can split according to the collinear DGLAP splitting functions only. The quantum circuit architecture has been designed for the `ibmq_qasm_simulator` device, a 32-qubit quantum simulator with the ability to simulate a fully fault tolerant digital quantum computer. Constructed from 31 qubits and 444 gate operations (275 multi-qubit and 169 single-qubit gate operations), the quantum Monte Carlo parton shower can simulate two shower steps. Figure 4.7 shows the comparison of results from the quantum parton shower algorithm and analytically calculated rates for each final state. We see that the quantum Monte Carlo parton shower shows good agreement with the expected rates, with the exception of channels which contain a  $g \rightarrow q\bar{q}$  process in the shower history. These channels exhibit an excess in comparison to the analytical rates due to the preference given to the  $g \rightarrow q\bar{q}$  splitting over the  $g \rightarrow gg$  splitting in this implementation. The preference for a gluon to split to quarks has been included to demonstrate the correct implementation of the  $g \rightarrow q\bar{q}$  process in the circuit architecture. Examining the scaling of the quantum resources required, we find that the quantum Monte Carlo parton shower scales quadratically in both the number of gate operations and qubits, specifically requiring a large number of controlled gate operations. Controlled gate operations are associated with a high error rate on the quantum device, consequently the implementation of the quantum Monte Carlo parton shower on a NISQ device is currently out of reach. Furthermore, the quadratic scaling in resources limits simulation to very shallow shower



depths. To simulate a shower depth typical of collision events at modern collider experiments, one must utilise compact quantum circuit architectures, such as the quantum walk.

Chapter 5 outlines a quantum walk approach to simulating the collinear QCD parton shower on a quantum device. Reframing the parton shower as a quantum walk across a two-dimensional lattice, the quantum resources required to implement a single shower step can be drastically reduced. The number of qubits scales logarithmically with the number of shower steps, allowing for the simulation of a 31-step parton shower on the `ibmq_qasm_simulator` device. The probability distributions for the number of gluons measured in the final state associated with zero and one quark-antiquark pair are shown in Figure 5.7. The quantum walk parton shower exhibits good agreement with a like-for-like classical parton shower, simulated using the Monte Carlo approach from Section 2.3.2. Due to the low probability of obtaining a final state with one or more quark-antiquark pairs, a further validation of the quantum walk parton shower has been performed in Figure 5.8. Once again, we see good agreement between the quantum simulator and the classical algorithm.

Further to the efficient qubit consumption of the quantum walk parton shower, the circuit is also compact, growing linearly with the number of shower steps. The quantum walk parton shower is therefore much more efficient than the quantum Monte Carlo parton shower, requiring only 9 qubits and 203 gate operations (144 multi-qubit and 59 single-qubit gate operations) to replicate the two-step shower from Chapter 4. However, despite the drastic improvement in the efficiency of the algorithm, the quantum walk parton shower can still only simulate a collinear QCD model, restricted to one flavour of quark. To include kinematics and simulate a realistic parton shower, the approach must again be reimaged for the quantum device.

The conventional approach to parton showering, which uses variations of the “veto” algorithm to perform rejection sampling, can be reframed as a one-dimensional random walk over the baseline of a multi-gluon phase space structure, called a grove. Chapter 6 outlines how this

approach is achieved by reformulating the discrete QCD method of Reference [90], allowing for the result of the parton shower evolution to be pre-selected before multi-particle momenta are generated, all the while ensuring the on-shell conditions and the conservation of momentum. By abstracting the random walk to the quantum walk framework, the discrete QCD algorithm has the ability to synthesise realistic particle collision events by sampling parton shower configurations on a quantum device, forming the basis of the quantum discrete QCD algorithm. As an application, a circuit has been composed to simulate collision events with  $E_{\text{CM}} = 91.2$  GeV, typical of collisions at the Large Electron Positron collider. Due to the compact architecture of the one-dimensional quantum walk, the grove structures have been generated and sampled on the 27-qubit `ibm_algiers` quantum computer using a streamlined circuit comprising 10 qubits and 21 gate operations (12 multi-qubit and nine single-qubit gate operations). The uncorrected results from the `ibm_algiers` device have been compared with results from the `ibmq_qasm_simulator` device and analytically calculated decay rates in Figure 6.7. It is clear that the `ibm_algiers` NISQ device suffers from substantial decoherence effects during the calculation. These noise effects have been identified as errors introduced by CNOT operations, and short T1 times of the qubits.

Despite a far from ideal distribution obtained from the `ibm_algiers` device, the generation of scattering events from the grove structures is remarkably robust to noise. Figure 6.9 shows the comparison between data from experiments at the Large Electron Positron collider and synthetic data produced by the classical and quantum discrete QCD algorithms. The data description is encouraging and represents the first successful comparison between synthetic data produced by a quantum computer and “real-life” experimental data from a collider. The quantum discrete QCD algorithm therefore improves on all previous quantum shower algorithms in several physics aspects (incorporating soft-gluon coherence, running coupling effects, and realistic scattering event generation), as well as the quantum resources required. Furthermore, the quantum discrete

QCD algorithm reduces the necessity for fully fault tolerant quantum devices, demonstrating that NISQ devices can already be used to obtain practical results by constructing efficient quantum circuit architectures.

The parton shower algorithms presented in this thesis stand as a proof-of-principle study, providing the first step towards the design of a fully quantum event generator. To achieve the goal of a event generation process which can exploit the benefits of quantum computation, future work must focus on the development of quantum circuits for all steps in the process. As outlined in Section 7.3, designs and proof-of-principle attempts have already begun to be made. Along with the parton showers presented here, these developments move the prospect of a quantum event generator ever closer, and perhaps even in not-so-distant reach.

Finally, it is interesting to consider the future of quantum computing in high energy physics. Quantum computing offers an exciting opportunity for the particle physics community, providing a platform with the ability to simulate physics in a regime never before accessible. However, it is not a trivial task to identify which areas of particle physics will benefit from quantum computing, nor is it trivial to design quantum circuits which can be realised on current and future quantum hardware. Nevertheless, it is exciting to see how the rapid advancements in algorithmic design, exploiting efficient quantum architectures, and reimagining approaches to problems in fundamental physics are already redefining “the horizon” for practical quantum computing. With such swift advancements, the practical use of quantum devices is starting to move into sight, rather than being over the horizon.

# Bibliography

- [1] K. Beperi, S. Malik, M. Spannowsky, and S. Williams, “Towards a quantum computing algorithm for helicity amplitudes and parton showers,” *Physical Review D*, vol. 103, no. 7, p. 076020, 2021.
- [2] K. Beperi, S. Malik, M. Spannowsky, and S. Williams, “Quantum walk approach to simulating parton showers,” *Physical Review D*, vol. 106, no. 5, p. 056002, 2022.
- [3] G. Gustafson, S. Prestel, M. Spannowsky, and S. Williams, “Collider events on a quantum computer,” *Journal of High Energy Physics*, vol. 11, p. 035, 2022.
- [4] R. P. Feynman, “Simulating physics with computers,” *International Journal of Theoretical Physics*, vol. 21, pp. 467–488, 1982.
- [5] S. P. Jordan, K. S. M. Lee, and J. Preskill, “Quantum algorithms for fermionic quantum field theories,” 2014. E-print: arXiv:1404.7115.
- [6] L. García-Álvarez, J. Casanova, A. Mezzacapo, I. L. Egusquiza, L. Lamata, G. Romero, and E. Solano, “Fermion-Fermion Scattering in Quantum Field Theory with Superconducting Circuits,” *Physical Review Letters*, vol. 114, no. 7, p. 070502, 2015.
- [7] Z. Davoudi, A. F. Shaw, and J. R. Stryker, “General quantum algorithms for hamiltonian simulation with applications to a non-abelian lattice gauge theory,” 2023. E-print: arXiv:2212.14030.

- 
- [8] A. Kan and Y. Nam, “Lattice quantum chromodynamics and electrodynamics on a universal quantum computer,” 2022. E-print: arXiv:2107.12769.
- [9] A. Ciavarella, N. Klco, and M. J. Savage, “Trailhead for quantum simulation of  $su(3)$  yang-mills lattice gauge theory in the local multiplet basis,” *Physical Review D*, vol. 103, p. 094501, 2021.
- [10] B. Nachman, D. Provasoli, W. A. de Jong, and C. W. Bauer, “Quantum algorithm for high energy physics simulations,” *Physical Review Letters*, vol. 126, p. 062001, 2021.
- [11] T. Li, X. Guo, W. K. Lai, X. Liu, E. Wang, H. Xing, D.-B. Zhang, and S.-L. Zhu, “Partonic collinear structure by quantum computing,” *Physical Review D*, vol. 105, p. L111502, 2022.
- [12] A. Blance and M. Spannowsky, “Quantum machine learning for particle physics using a variational quantum classifier,” *Journal of High Energy Physics*, vol. 2021, no. 2, pp. 1–20, 2021.
- [13] A. Blance and M. Spannowsky, “Unsupervised event classification with graphs on classical and photonic quantum computers,” *Journal of High Energy Physics*, vol. 2021, no. 8, pp. 1–26, 2021.
- [14] K. Terashi, M. Kaneda, T. Kishimoto, M. Saito, R. Sawada, and J. Tanaka, “Event classification with quantum machine learning in high-energy physics,” *Computing and Software for Big Science*, vol. 5, pp. 1–11, 2021.
- [15] S. L. Wu, J. Chan, W. Guan, S. Sun, A. Wang, C. Zhou, M. Livny, F. Carminati, A. Di Meglio, A. C. Li, *et al.*, “Application of quantum machine learning using the quantum variational classifier method to high energy physics analysis at the lhc on ibm quantum computer simulator and hardware with 10 qubits,” *Journal of Physics G: Nuclear and Particle Physics*, vol. 48, no. 12, p. 125003, 2021.

- [16] A. Mott, J. Job, J.-R. Vlimant, D. Lidar, and M. Spiropulu, “Solving a higgs optimization problem with quantum annealing for machine learning,” *Nature*, vol. 550, no. 7676, pp. 375–379, 2017.
- [17] C. Csáki, S. Lombardo, and O. Telem, “Tasi lectures on non-supersymmetric bsm models,” 2018. E-print: arXiv:1811.04279.
- [18] J. D. Lykken, “Beyond the standard model,” 2010. E-print: arXiv:1005.1676.
- [19] S. P. Martin, “A Supersymmetry Primer,” in *Perspectives on Supersymmetry*, pp. 1–98, World Scientific, 1998.
- [20] C. Bierlich, S. Chakraborty, N. Desai, L. Gellersen, I. Helenius, P. Ilten, L. Lönnblad, S. Mrenna, S. Prestel, C. T. Preuss, T. Sjöstrand, P. Skands, M. Uthmeim, and R. Verheyen, “A comprehensive guide to the physics and usage of PYTHIA 8.3,” *SciPost Physics Codebases*, p. 8, 2022.
- [21] M. Bahr *et al.*, “Herwig++ Physics and Manual,” *The European Physical Journal C*, vol. 58, pp. 639–707, 2008.
- [22] T. Gleisberg, S. Hoeche, F. Krauss, A. Schaliche, S. Schumann, and J.-C. Winter, “SHERPA 1. alpha: A Proof of concept version,” *Journal of High Energy Physics*, vol. 02, p. 056, 2004.
- [23] G. T. Bodwin, “Factorization of the Drell-Yan Cross-Section in Perturbation Theory,” *Physical Review D*, vol. 31, p. 2616, 1985. [Erratum: *Physical Review D* 34, 3932 (1986)].
- [24] J. C. Collins, D. E. Soper, and G. F. Sterman, “Factorization for Short Distance Hadron-Hadron Scattering,” *Nuclear Physics B*, vol. 261, pp. 104–142, 1985.
- [25] J. C. Collins, D. E. Soper, and G. Sterman, “Soft gluons and factorization,” *Nuclear Physics B*, vol. 308, no. 4, pp. 833–856, 1988.

- [26] P. Langacker, *The Standard Model and Beyond*. Series in High Energy Physics, Cosmology and Gravitation, Taylor & Francis, 2010.
- [27] M. Schwartz, *Quantum Field Theory and the Standard Model*. Quantum Field Theory and the Standard Model, Cambridge University Press, 2014.
- [28] M. Peskin and D. Schroeder, *An Introduction to Quantum Field Theory*. Advanced book classics, Avalon Publishing, 1995.
- [29] A. Buckley, C. White, and M. White, *Practical Collider Physics*. Institute of Physics, 2021.
- [30] A. Buckley, J. Butterworth, S. Gieseke, D. Grellscheid, S. Höche, H. Hoeth, F. Krauss, L. Lönnblad, E. Nurse, P. Richardson, S. Schumann, M. H. Seymour, T. Sjöstrand, P. Skands, and B. Webber, “General-purpose event generators for LHC physics,” *Physics Reports*, vol. 504, no. 5, pp. 145–233, 2011.
- [31] S. Marzani, G. Soyez, and M. Spannowsky, *Looking Inside Jets*. Springer International Publishing, 2019.
- [32] S. Höche, “Introduction to parton-shower event generators,” in *Theoretical Advanced Study Institute in Elementary Particle Physics: Journeys Through the Precision Frontier: Amplitudes for Colliders*, pp. 235–295, 2015.
- [33] R. K. Ellis, W. J. Stirling, and B. R. Webber, *QCD and collider physics*, vol. 8. Cambridge University Press, 2011.
- [34] C. N. Yang and R. L. Mills, “Conservation of isotopic spin and isotopic gauge invariance,” *Physical Review*, vol. 96, pp. 191–195, 1954.
- [35] C. Gattringer and C. B. Lang, *Quantum chromodynamics on the lattice*, vol. 788. Berlin: Springer, 2010.

- [36] D. J. Gross and F. Wilczek, “Asymptotically free gauge theories. i,” *Physical Review D*, vol. 8, pp. 3633–3652, 1973.
- [37] S. Weinberg, “Non-abelian gauge theories of the strong interactions,” *Physical Review Letters*, vol. 31, pp. 494–497, 1973.
- [38] H. Fritzsch, M. Gell-Mann, and H. Leutwyler, “Advantages of the Color Octet Gluon Picture,” *Physics Letters B*, vol. 47, pp. 365–368, 1973.
- [39] S. L. Glashow, “Partial Symmetries of Weak Interactions,” *Nuclear Physics*, vol. 22, pp. 579–588, 1961.
- [40] S. Weinberg, “A model of leptons,” *Physical Review Letters*, vol. 19, pp. 1264–1266, 1967.
- [41] A. Salam, “Weak and Electromagnetic Interactions,” *Conference Proceedings C*, vol. 680519, pp. 367–377, 1968.
- [42] “Tikz example: Standard model of particle physics.” <https://texample.net>. Accessed: 16-02-2023.
- [43] R. L. Workman and Others, “Review of Particle Physics,” *Progress of Theoretical and Experimental Physics*, vol. 2022, p. 083C01, 2022.
- [44] The ATLAS Collaboration, “Observation of a new particle in the search for the standard model higgs boson with the ATLAS detector at the LHC,” *Physics Letters B*, vol. 716, no. 1, pp. 1–29, 2012.
- [45] The CMS Collaboration, “Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC,” *Physics Letters B*, vol. 716, no. 1, pp. 30–61, 2012.
- [46] P. W. Higgs, “Broken symmetries and the masses of gauge bosons,” *Physical Review Letters*, vol. 13, pp. 508–509, 1964.



- [47] F. Englert and R. Brout, “Broken symmetry and the mass of gauge vector mesons,” *Physical Review Letters*, vol. 13, pp. 321–323, 1964.
- [48] G. S. Guralnik, C. R. Hagen, and T. W. B. Kibble, “Global conservation laws and massless particles,” *Physical Review Letters*, vol. 13, pp. 585–587, 1964.
- [49] N. Cabibbo, “Unitary symmetry and leptonic decays,” *Physical Review Letters*, vol. 10, pp. 531–533, 1963.
- [50] M. Kobayashi and T. Maskawa, “CP Violation in the Renormalizable Theory of Weak Interaction,” *Progress of Theoretical Physics*, vol. 49, pp. 652–657, 1973.
- [51] Y. L. Dokshitzer, “Calculation of the Structure Functions for Deep Inelastic Scattering and  $e^+ e^-$  Annihilation by Perturbation Theory in Quantum Chromodynamics,” *Soviet Physics Journal of Experimental and Theoretical Physics*, vol. 46, pp. 641–653, 1977.
- [52] V. N. Gribov and L. N. Lipatov, “Deep inelastic  $e p$  scattering in perturbation theory,” *Soviet Journal of Nuclear Physics*, vol. 15, pp. 438–450, 1972.
- [53] G. Altarelli and G. Parisi, “Asymptotic Freedom in Parton Language,” *Nuclear Physics B*, vol. 126, pp. 298–318, 1977.
- [54] S. Gieseke, P. Stephens, and B. Webber, “New formalism for QCD parton showers,” *Journal of High Energy Physics*, vol. 12, p. 045, 2003.
- [55] S. Schumann and F. Krauss, “A Parton shower algorithm based on Catani-Seymour dipole factorisation,” *Journal of High Energy Physics*, vol. 03, p. 038, 2008.
- [56] S. Höche and S. Prestel, “The midpoint between dipole and parton showers,” *European Physical Journal C*, vol. 75, no. 9, p. 461, 2015.

- 
- [57] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 1997.
- [58] D. Deutsch and R. Jozsa, “Rapid solution of problems by quantum computation,” *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, vol. 439, no. 1907, pp. 553–558, 1992.
- [59] A. Aspect, P. Grangier, and G. Roger, “Experimental tests of realistic local theories via bell’s theorem,” *Physical Review Letters*, vol. 47, pp. 460–463, 1981.
- [60] A. Aspect, P. Grangier, and G. Roger, “Experimental realization of einstein-podolsky-rosen-bohm gedankenexperiment: A new violation of bell’s inequalities,” *Physical Review Letters*, vol. 49, pp. 91–94, 1982.
- [61] A. Aspect, J. Dalibard, and G. Roger, “Experimental test of bell’s inequalities using time-varying analyzers,” *Physical Review Letters*, vol. 49, pp. 1804–1807, 1982.
- [62] S. J. Freedman and J. F. Clauser, “Experimental test of local hidden-variable theories,” *Physical Review Letters*, vol. 28, pp. 938–941, 1972.
- [63] G. Weihs, T. Jennewein, C. Simon, H. Weinfurter, and A. Zeilinger, “Violation of bell’s inequality under strict einstein locality conditions,” *Physical Review Letters*, vol. 81, pp. 5039–5043, 1998.
- [64] J. Preskill, “Simulating quantum field theory with a quantum computer,” *Proceeding of Science*, vol. LATTICE2018, p. 024, 2019.
- [65] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
- [66] R. de Wolf, “Quantum computing: Lecture notes,” 2019. E-print: arXiv:1907.09415.

- [67] E. Rieffel and W. Polak, *Quantum Computing: A Gentle Introduction*. The MIT Press, 2011.
- [68] S. Barnett, *Quantum Information*. USA: Oxford University Press, Inc., 2009.
- [69] A. Einstein, B. Podolsky, and N. Rosen, “Can quantum-mechanical description of physical reality be considered complete?,” *Physical Review*, vol. 47, pp. 777–780, 1935.
- [70] J. S. Bell, “On the einstein podolsky rosen paradox,” *Physics Physique Fizika*, vol. 1, pp. 195–200, 1964.
- [71] L. K. Grover, “A fast quantum mechanical algorithm for database search,” in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pp. 212–219, 1996.
- [72] R. Portugal, *Quantum Walks and Search Algorithms*. Springer Publishing Company, Incorporated, 2013.
- [73] M. Szegedy, “Quantum speed-up of markov chain based algorithms,” in *45th Annual IEEE Symposium on Foundations of Computer Science*, pp. 32–41, 2004.
- [74] N. Shenvi, J. Kempe, and K. B. Whaley, “Quantum random-walk search algorithm,” *Physical Review A*, vol. 67, p. 052307, 2003.
- [75] P. Wocjan and A. Abeyesinghe, “Speedup via quantum sampling,” *Physical Review A*, vol. 78, p. 042336, 2008.
- [76] R. D. Somma, S. Boixo, H. Barnum, and E. Knill, “Quantum simulations of classical annealing processes,” *Physical Review Letters*, vol. 101, p. 130504, 2008.
- [77] F. Magniez, A. Nayak, J. Roland, and M. Santha, “Search via quantum walk,” *SIAM Journal on Computing*, vol. 40, no. 1, pp. 142–164, 2011.

- 
- [78] J. Lemieux, B. Heim, D. Poulin, K. Svore, and M. Troyer, “Efficient Quantum Walk Circuits for Metropolis-Hastings Algorithm,” *Quantum*, vol. 4, p. 287, 2020.
- [79] J. Kempe, “Quantum random walks: An introductory overview,” *Contemporary Physics*, vol. 44, no. 4, pp. 307–327, 2003.
- [80] P. P. Rohde, G. K. Brennen, and A. Gilchrist, “Quantum walks with memory provided by recycled coins and a memory of the coin-flip history,” *Physical Review A*, vol. 87, p. 052302, 2013.
- [81] Y. Aharonov, L. Davidovich, and N. Zagury, “Quantum random walks,” *Physical Review A*, vol. 48, pp. 1687–1690, 1993.
- [82] A. Ambainis, E. Bach, A. Nayak, A. Vishwanath, and J. Watrous, “One-dimensional quantum walks,” in *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, STOC '01, (New York, NY, USA), p. 37–49, Association for Computing Machinery, 2001.
- [83] D. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani, “Quantum walks on graphs,” in *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, STOC '01, (New York, NY, USA), p. 50–59, Association for Computing Machinery, 2001.
- [84] T. G. Wong, “Grover search with lackadaisical quantum walks,” *Journal of Physics A: Mathematical and Theoretical*, vol. 48, no. 43, p. 435304, 2015.
- [85] A. Nayak and A. Vishwanath, “Quantum walk on the line,” 2000. E-print: arXiv:quant-ph/0010117.
- [86] D. A. Meyer, “From quantum cellular automata to quantum lattice gases,” *Journal of Statistical Physics*, vol. 85, no. 5-6, pp. 551–574, 1996.

- [87] E. Camilleri, P. Rohde, and J. Twamley, “Quantum walks with tuneable self-avoidance in one dimension,” *Scientific Reports*, vol. 4, p. 4791, 2014.
- [88] T. A. Brun, H. A. Carteret, and A. Ambainis, “Quantum walks driven by many coins,” *Physical Review A*, vol. 67, p. 052317, 2003.
- [89] V. Giovannetti, S. Lloyd, and L. Maccone, “Quantum random access memory,” *Physical Review Letters*, vol. 100, no. 16, 2008.
- [90] B. Andersson, G. Gustafson, and J. Samuelsson, “Discrete qcd, a new approximation for qcd cascades,” *Nuclear Physics B*, vol. 463, no. 2, pp. 217–237, 1996.
- [91] L. Lönnblad, “Ariadne version 4 — a program for simulation of qdc cascades implementing the colour dipole model,” *Computer Physics Communications*, vol. 71, no. 1, pp. 15–31, 1992.
- [92] B. Andersson, G. Gustafson, and J. Samuelsson, “Discrete QCD: A New approximation for QCD cascades,” *Nuclear Physics B*, vol. 463, pp. 217–237, 1996.
- [93] M. H. Seymour, “Matrix-element corrections to parton shower algorithms,” *Computer Physics Communications*, vol. 90, no. 1, pp. 95–101, 1995.
- [94] S. Platzer and M. Sjö Dahl, “The Sudakov Veto Algorithm Reloaded,” *European Physical Journal Plus*, vol. 127, p. 26, 2012.
- [95] A. Gehrmann-De Ridder, M. Ritzmann, and P. Skands, “Timelike dipole-antenna showers with massive fermions,” *Physical Review D*, vol. 85, p. 014013, 2012.
- [96] A. A. Clerk, M. H. Devoret, S. M. Girvin, F. Marquardt, and R. J. Schoelkopf, “Introduction to quantum noise, measurement, and amplification,” *Reviews of Modern Physics*, vol. 82, pp. 1155–1208, 2010.

- 
- [97] D. A. Lidar and T. A. Brun, *Quantum error correction*. Cambridge university press, 2013.
- [98] S. J. Devitt, W. J. Munro, and K. Nemoto, “Quantum error correction for beginners,” *Reports on Progress in Physics*, vol. 76, no. 7, p. 076001, 2013.
- [99] A. Heister, G. Dissertori, A. Collaboration, *et al.*, “Studies of qcd at e (+) e (-) centre-of-mass energies between 91 and 209 gev,” *The European Physical Journal C*, vol. 35, no. 4, pp. 457–486, 2004.
- [100] D. Collaboration, “Tuning and test of fragmentation models based on identified particles and precision event shape data delphi collaboration,” *Zeitschrift für Physik C Particles and Fields*, vol. 73, pp. 11–59, 1997.
- [101] J. collaboration, P. Pfeifenschneider, O. Collaboration, and G. Abbiendi, “Qcd analyses and determinations of in annihilation at energies between 35 and 189 gev,” *The European Physical Journal C-Particles and Fields*, vol. 17, no. 1, pp. 19–51, 2000.
- [102] B. Andersson, G. Gustafson, G. Ingelman, and T. Sjostrand, “Parton Fragmentation and String Dynamics,” *Physics Reports*, vol. 97, pp. 31–145, 1983.
- [103] A. Buckley, J. Butterworth, D. Grellscheid, H. Hoeth, L. Lonnblad, J. Monk, H. Schulz, and F. Siegert, “Rivet user manual,” *Computer Physics Communications*, vol. 184, pp. 2803–2819, 2013.
- [104] P. Skands, S. Carrazza, and J. Rojo, “Tuning PYTHIA 8.1: the Monash 2013 Tune,” *European Physical Journal C*, vol. 74, no. 8, p. 3024, 2014.
- [105] J. Isaacson and S. Prestel, “Stochastically sampling color configurations,” *Physical Review D*, vol. 99, no. 1, p. 014021, 2019.

- [106] A. Montanaro, “Quantum speedup of monte carlo methods,” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 471, no. 2181, p. 20150301, 2015.
- [107] R. Orús, S. Mugel, and E. Lizaso, “Quantum computing for finance: Overview and prospects,” *Reviews in Physics*, vol. 4, p. 100028, 2019.
- [108] S. Woerner and D. J. Egger, “Quantum risk analysis,” *npj Quantum Information*, vol. 5, no. 1, p. 15, 2019.
- [109] D. A. Levin and Y. Peres, *Markov chains and mixing times*, vol. 107. American Mathematical Soc., 2017.
- [110] A. Sinclair, *Algorithms for random generation and counting: a Markov chain approach*. Springer Science & Business Media, 2012.
- [111] Y. Atia and S. Chakraborty, “Improved upper bounds for the hitting times of quantum walks,” *Physical Review A*, vol. 104, p. 032215, 2021.
- [112] S. Chakraborty, K. Luh, and J. Roland, “How fast do quantum walks mix?,” *Physical Review Letters*, vol. 124, p. 050501, 2020.
- [113] T. G. Wong, “Equivalence of Szegedy’s and coined quantum walks,” *Quantum Information Processing*, vol. 16, pp. 1–15, 2017.
- [114] R. Portugal, “Establishing the equivalence between szegedy’s and coined quantum walks using the staggered model,” *Quantum Information Processing*, vol. 15, pp. 1387–1409, 2016.
- [115] D. Li, Y. Liu, Y.-G. Yang, J. Xu, and J.-B. Yuan, “Szegedy quantum walks with memory on regular graphs,” *Quantum Information Processing*, vol. 19, pp. 1–12, 2020.

- 
- [116] G. Gustafson, “Multiplicity distributions in qcd cascades,” *Nuclear Physics B*, vol. 392, no. 1, pp. 251–278, 1993.
- [117] S. Frixione and B. R. Webber, “Matching NLO QCD computations and parton shower simulations,” *Journal of High Energy Physics*, vol. 06, p. 029, 2002.
- [118] P. Nason, “A New method for combining NLO QCD with shower Monte Carlo algorithms,” *Journal of High Energy Physics*, vol. 11, p. 040, 2004.
- [119] S. Frixione, P. Nason, and C. Oleari, “Matching NLO QCD computations with Parton Shower simulations: the POWHEG method,” *Journal of High Energy Physics*, vol. 11, p. 070, 2007.
- [120] S. Catani, F. Krauss, R. Kuhn, and B. R. Webber, “QCD matrix elements + parton showers,” *Journal of High Energy Physics*, vol. 11, p. 063, 2001.
- [121] L. Lonnblad, “Correcting the color dipole cascade model with fixed order matrix elements,” *Journal of High Energy Physics*, vol. 05, p. 046, 2002.
- [122] M. L. Mangano, M. Moretti, and R. Pittau, “Multijet matrix elements and shower evolution in hadronic collisions:  $Wb\bar{b} + n$  jets as a case study,” *Nuclear Physics B*, vol. 632, pp. 343–362, 2002.
- [123] S. Mrenna and P. Richardson, “Matching matrix elements and parton showers with HERWIG and PYTHIA,” *Journal of High Energy Physics*, vol. 05, p. 040, 2004.
- [124] J. Alwall *et al.*, “Comparative study of various algorithms for the merging of parton showers and matrix elements in hadronic collisions,” *European Physical Journal C*, vol. 53, pp. 473–500, 2008.
- [125] A. Pérez-Salinas, J. Cruz-Martinez, A. A. Alhajri, and S. Carrazza, “Determining the proton content with a quantum computer,” *Physical Review D*, vol. 103, p. 034027, 2021.



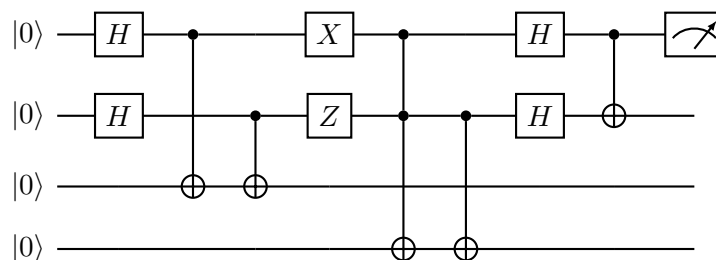
- 
- [126] H. A. Chawdhry and M. Pellen, “Quantum simulation of colour in perturbative quantum chromodynamics,” 2023. E-print: arXiv:2303.04818.
- [127] N. Klco, E. F. Dumitrescu, A. J. McCaskey, T. D. Morris, R. C. Pooser, M. Sanz, E. Solano, P. Lougovski, and M. J. Savage, “Quantum-classical computation of Schwinger model dynamics using quantum computers,” *Physical Review A*, vol. 98, no. 3, p. 032331, 2018.
- [128] A. A. Clerk, M. H. Devoret, S. M. Girvin, F. Marquardt, and R. J. Schoelkopf, “Introduction to quantum noise, measurement, and amplification,” *Reviews of Modern Physics*, vol. 82, no. 2, pp. 1155–1208, 2010.
- [129] N. A. of Engineering, *Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2018 Symposium*. National Academies Press (US), 2019.

# Appendix A

## Quantum Circuit Diagrams

Quantum circuit diagrams are a compact and efficient way of symbolically representing quantum algorithms, by decomposing the unitary process into a series of qubit initialisations, gate operations and measurements applied to individual qubits. This appendix is a brief guide to reading quantum circuit diagrams and contains a short glossary of commonly used quantum gate operations, useful for the reading of this thesis. An in depth review of quantum circuitry, and quantum computing as a whole, can be found in Reference [65].

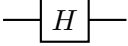

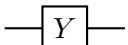

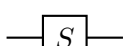
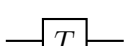
An example of a quantum circuit comprising single and multi-qubit operations, with a measurement on the first qubit is shown below.

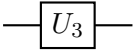
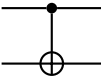
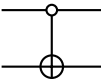
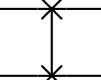
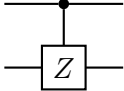


The quantum circuit diagram is read from left to right, with time on the horizontal axis. Consequently, the two Hadamard gates (see Section A.1) are implemented first in the algorithm,

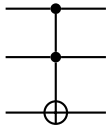
and the measurement on the first qubit is performed at the end of the algorithm. Horizontal lines represent qubits and vertical lines represent operations applied to multiple qubits. Gate operations are generally represented by boxes on qubit lines. Note that sometimes quantum algorithms can have custom gates. These must be unitary in order to be implemented on a quantum device, and can be decomposed into the universal gate set outlined in Section [A.1](#).

## A.1 Glossary of Quantum Gate Operations

Hadamard		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Pauli-X		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Phase		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
T-gate		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$

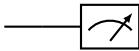
$U_3(\theta, \varphi, \lambda)$		$\begin{bmatrix} \cos(\frac{\theta}{2}) & -e^{i\lambda} \sin(\frac{\theta}{2}) \\ e^{i\varphi} \sin(\frac{\theta}{2}) & e^{i(\varphi+\lambda)} \cos(\frac{\theta}{2}) \end{bmatrix}$
controlled-NOT (CNOT)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
$ 0\rangle$ -controlled-NOT		$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
SWAP		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
controlled-Z		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$

Toffoli (CCNOT)



$$\begin{bmatrix}
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
 \end{bmatrix}$$

Measurement



Projection onto classical bit\*

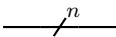
Qubit



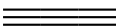
Classical bit



$n$  qubits



Qubit bundle




---

\*See Appendix B for a detail discussion of quantum measurements.

## Appendix B

# Quantum Measurement

Quantum algorithms proceed under unitary evolution, and are constructed from a series of unitary operations applied to a set of qubits. In order to retrieve information from the quantum system, a measurement must be performed. After measurement, the system is no longer closed, and thus the evolution of the system is not necessarily unitary. This appendix provides a short overview of quantum measurement for the purposes of understanding the algorithms in the main text, and will concentrate on measurement in the computational basis. A detailed overview of quantum measurement, including projective measurement and positive operator-valued measurement, is given in References [65, 66] and references therein.

Following the convention outlined in Reference [65], if a measurement is performed on a general state  $|\psi\rangle$ , the probability that result  $m$  occurs is given by

$$p(m) = \langle \psi | M_m^\dagger M_m | \psi \rangle, \tag{B.0.1}$$

where the  $M_m$  is the measurement operator for result  $m$ . The state after measurement is therefore

$$|\psi\rangle \xrightarrow{M_m} |\psi'\rangle = \frac{M_m |\psi\rangle}{\sqrt{\langle\psi| M_m^\dagger M_m |\psi\rangle}}, \quad (\text{B.0.2})$$

where the factor  $\sqrt{\langle\psi| M_m^\dagger M_m |\psi\rangle}$  normalises the post-measurement state  $|\psi'\rangle$ . The set of measurement operators  $\{M_m\}$  satisfy the completeness relation  $\sum_m M_m^\dagger M_m = \mathbb{I}$ , such that, equivalently, the probabilities sum to one for all  $|\psi\rangle$ ,

$$\sum_m p(m) = \sum_m \langle\psi| M_m^\dagger M_m |\psi\rangle = 1. \quad (\text{B.0.3})$$

## B.1 Computational Basis

For most of this thesis, the term measurement relates to measuring in the computational basis, id est the basis  $|0\rangle$  and  $|1\rangle$ . Using the notation from above, one can define the measurement of a single qubit, with arbitrary state

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle, \quad (\text{B.1.1})$$

where  $\alpha$  and  $\beta$  are complex amplitudes. There are two possible results from a measurement of this state: the 0 and the 1 result. These are defined by the measurement operators  $M_0$  and  $M_1$  respectively, such that

$$M_0 = |0\rangle\langle 0|, \quad \text{and} \quad M_1 = |1\rangle\langle 1|. \quad (\text{B.1.2})$$

It is easy to show that these operators are Hermitian and satisfy the completeness relation, as required. Applying the measurement operator for the 0 result to the state in Equation B.1.1, one obtains

$$\begin{aligned}
p(0) &= \langle \psi | M_0^\dagger M_0 | \psi \rangle \\
&= \langle \psi | M_0 | \psi \rangle \\
&= (\langle 0 | \alpha^* + \langle 1 | \beta^* ) | 0 \rangle \langle 0 | (\alpha | 0 \rangle + \beta | 1 \rangle ) \\
&= |\alpha|^2 \langle 0 | 0 \rangle \langle 0 | 0 \rangle + |\beta|^2 \langle 1 | 0 \rangle \langle 0 | 1 \rangle \\
&= |\alpha|^2,
\end{aligned} \tag{B.1.3}$$

where the orthonormality of the computational basis has been applied. Similarly, applying the measurement operator for the 1 result,

$$\begin{aligned}
p(1) &= \langle \psi | M_1^\dagger M_1 | \psi \rangle \\
&= (\langle 0 | \alpha^* + \langle 1 | \beta^* ) | 1 \rangle \langle 1 | (\alpha | 0 \rangle + \beta | 1 \rangle ) \\
&= |\beta|^2,
\end{aligned} \tag{B.1.4}$$

The post-measurement states are therefore

$$|\psi'\rangle = \frac{M_0 |\psi\rangle}{|\alpha|} \quad \text{and} \quad |\psi'\rangle = \frac{M_1 |\psi\rangle}{|\beta|}. \tag{B.1.5}$$

Note that for the one qubit the factors  $M_0 |\psi\rangle = \alpha$  and  $M_1 |\psi\rangle = \beta$ , and so the post-measurement states effectively reduce to  $|0\rangle$  and  $|1\rangle$  respectively.

In the multi-qubit picture, subsets of the qubits can be measured as well as the full state of the system. For example, consider the state

$$|\psi\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle, \tag{B.1.6}$$



where  $\alpha_{ij}$  is the complex amplitude for the state  $|ij\rangle$ , where  $i, j \in \{0, 1\}$ . If we measure the first qubit only, then we gain the 0 result with probability  $(|\alpha_{00}|^2 + |\alpha_{01}|^2)$ , such that the post-measurement state is

$$|\psi'\rangle = \frac{\alpha_{00} |00\rangle + \alpha_{01} |01\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}}, \quad (\text{B.1.7})$$

where the state is now normalised by the factor  $\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}$ .

## Appendix C

# Quantum Errors

Without understanding possible error contributions, the use of NISQ devices can become impractical, and lead to low fidelity results. Unlike classical devices, quantum computers have a small amount of qubits and therefore cannot rely on bit redundancy for error correction. Until this is the case, the fidelity of quantum algorithms depends heavily on quantum error correction and mitigation schemes. Currently, there is no general error correction method, and no quantum device is fully fault tolerant. In this Appendix, we will introduce the two main sources of errors which affect the quantum parton shower algorithms, specifically the quantum discrete QCD algorithm. Namely, we will consider errors from multi-qubit operations and the T1 time of the qubits. This Appendix has been designed as a very brief introduction and identification of the quantum errors that are present for the quantum parton showers. An in depth review of quantum errors is available in References [97, 98, 128, 129].

### Multi-qubit Gate Operation

Multi-qubit gate operations are typically decomposed into a series of CNOT and rotation gates during transpilation\*. The CNOT operation has a larger associated error rate than single-qubit

---

\*The process of preparing a circuit for implementation on a real quantum device.

gate operations. Therefore, as the number of CNOT operations increase in the algorithm, the fidelity of the results is likely to reduce as more errors are introduced into the calculation. Furthermore, if multi-qubit operations require SWAP operations (id est swapping the state of the qubits) to correctly implement the routine, then this will also increase the number of CNOT operations required, as one SWAP operation requires three CNOT gates. Therefore, even if a small number of CCNOT operation are used in the algorithm, together with the required SWAP operations, the number of CNOT operations required can be an order of magnitude higher. This consequently introduces many errors into the calculation, and can lead to unfeasible circuit implementations for NISQ devices.

### **Qubit Relaxation Times**

Another primary source of error in results obtained from a NISQ quantum device comes from the relaxation time of the qubits, the so-called *T1 time*. The T1 time of a qubit is the time it takes for the qubit to decay from the excited state, the  $|1\rangle$  state in the computational basis, to the ground state, the  $|0\rangle$  state. The T1 time is therefore a measure of the time in which practical and meaningful algorithms can be run. As an example of the T1 time being a limiting factor, the results from Figure 6.7 show a preference for grove structures with fewer emissions. This has been attributed to qubits in the gluon register relaxing to their ground state, therefore representing a state with fewer emissions.

# List of Figures

2.1 Schematic of a  $pp \rightarrow t\bar{t}$  event from Reference [20]. The schematic has been simplified from a real PYTHIA event, such that 1) shower branchings and final-state hadrons are less numerous, 2) the effect of recoil is not correctly illustrated, 3) weak decays of light hadrons are not included, and 4) incoming momenta are depicted as crossed ( $p \rightarrow -p$ ). . . . . 23

2.2 Pictorial representation of the construction of the field tensor from four consecutive compactors forming an infinitesimal square loop in the  $(\hat{n}_1, \hat{n}_2)$  plane. Figure adapted from Reference [28]. . . . . 28

2.3 The Standard Model of particle physics consists of three generations of fermions and the gauge bosons. Displayed in the white coloured boxes are the six quarks, which interact via the strong and electroweak forces. In the light blue boxes are the leptons, the first row includes the charged leptons which interact via the electroweak force, and the second row shows the neutrinos, which interact via the weak force only. The gauge bosons are the gluons (red), which mediate the strong force, the photon (green), which mediates the electromagnetic force, and the weak bosons (orange), which mediate the weak force. Finally, the Higgs boson, which provides a mechanism to generate the masses of the particles, is shown in grey. The Figure has been adapted from Reference [42] to include the latest observed

	values from the Particle Data Group [43]. . . . .	37
2.4	A process producing $n$ -partons in the final state, where one parton, $i$ , splits to partons $j$ and $k$ . . . . .	52
2.5	A parton of species $i_1$ , with virtuality $t_1$ and momentum fraction $z_1 = 1$ , emerges from the hard process and showers, emitting $n$ partons of species $\{j_n\}$ . The shower is categorised by a set of virtualities and momentum fractions, $(\{t_n\}, \{z_n\})$ , and the emitting parton's species at each step is denoted by $i_n$ . Figure has been adapted from Reference [29]. . . . .	55
3.1	Bloch sphere representation of a qubit . . . . .	64
3.2	Schematic of the quantum circuit for encoding the Bell state on a quantum device. . . . .	67
3.3	Schematic of discrete-time coined quantum walk on a line. The walker at position $i = 0$ can move left or right depending on the coin state $ 0\rangle$ or $ 1\rangle$ respectively. Figure modified from Reference [2]. . . . .	71
3.4	Quantum circuit diagram for the discrete-time quantum walk on a line with a Hadamard coin. . . . .	72
3.5	Simulation of a 100-step discrete-time quantum walk for a Hadamard coin with the coin state initialised in (a) a symmetric initial state $ c\rangle = \frac{1}{\sqrt{2}}( 0\rangle + i 1\rangle)$ , (b) the ground state $ c\rangle =  0\rangle$ , and (c) the excited state $ c\rangle = - 1\rangle$ , compared to a 100-step discrete classical random walk. The simulations have been run for $10^6$ shots on the 32-qubit <code>ibmq_qasm_simulator</code> , and the classical distribution has been achieved by measuring the system at each step. Note that only non-zero values are shown. Figure (a) has been adapted from Reference [2]. . . . .	73
4.1	Naive decomposition of an $n$ -controlled gate operation using $(n-1)$ ancillary qubits. . . . .	82
4.2	Schematic circuit diagram for the quantum parton shower algorithm, with the ability to simulate a collinear, toy QCD model. The circuit is constructed from . . . . .	

	four registers: the parton register $p_i$ , the counting register $n$ , the emission register $e$ , and the history register $h_i$ . The algorithm proceeds in individual steps comprised of four unitary operations: the counting operations $C_f$ , the emission operations $E_f$ , the history operations $H_f$ , and the update operations $U_f$ . The subscript $f$ is used here to denote that these operations are applied for each parton in the shower, and differ at each step. After $N$ steps, the parton register is measured and the particle content of the shower is obtained. . . . .	83
4.3	The count operation controls from the parton sub-register $p_i$ and updates the counting registers $n_i$ according to the particle content of the shower. . . . .	85
4.4	Controlling from the number of partons present in the shower, the emission gate encodes the Sudakov factor onto the device by rotating the emission qubit $e$ by the $E_f$ rotation. . . . .	87
4.5	The history operation controls from the parton sub-registers, $p_i$ , and the emission register, $e$ , and acts on the history sub-registers, $h_i^f$ , with the $H_f$ rotation. The $H_f$ rotation encodes the information about which species the emitting parton will split to onto the history register. Note that the two gluon history rotations $H_{gg}$ and $H_{qg}$ can be applied at the same time to reduce the number of CCNOT gates, but here they have been displayed separately for clarity. . . . .	90
4.6	Update operation sub-routine applied to all populated parton sub-registers $p_i$ . The first gate group shows the creation of a gluon from the $q \rightarrow gq$ splitting. Note that this can also be for a $\bar{q} \rightarrow g\bar{q}$ splitting. The centre gate group shows the splitting $g \rightarrow q\bar{q}$ and the final gate group shows the splitting $g \rightarrow gg$ . The $ 0\rangle$ -control on the $h_1^g$ history qubit gives preference to $g \rightarrow q\bar{q}$ splittings. The parton registers are then passed to the next step. . . . .	92
4.7	Comparison between the <code>ibmq_qasm_simulator</code> and analytical calculations for a	

two-step quantum parton shower with momentum interval  $0.3 \leq z \leq 0.5$ . The algorithm has been run for an initial particle of a gluon (left) and a quark (right). Note that the distribution of the antiquark initial state will be identical to the quark initial state. The algorithm has been run for  $10^4$  shots on the quantum simulator, without a noise profile. Figure adapted from Reference [1]. . . . . 95

5.1 Schematic circuit diagram for a quantum walk with the ability to simulate a scalar particle cascade. Each step of the algorithm is constructed from three main operations: the *coin* operation, which encodes the splitting probabilities onto the device, the *shift* operation, which updates the particle content of the shower, and the *memory* operation, which can be used to store previous coin outcomes. The coin operation is supported by a *position check* operation, which ensures the correct implementation of the splitting probabilities. The step operation is then repeated for  $N$  steps, before measuring the number of particles in the final state, encoded in the position register. . . . . 101

5.2 Schematic of the circuit implementation for the shift operation in the scalar shower model. If, after the coin operation, the coin is in the  $|1\rangle$  state and a splitting has occurred, then the position of the walker is incremented using a series of CCNOT gates. Otherwise, if a splitting has not occurred and the coin is in the  $|0\rangle$  state, then the walker remains stationary. . . . . 104

5.3 Schematic circuit diagram for the quantum walk parton shower with the ability to simulate a discrete, collinear QCD parton shower. The algorithm is built from a quantum walk with memory framework in two-dimensions and is constructed from three main operations: the *coin* operation, which encodes the splitting functions on the device, the *shift* operation, which moves the walker along a two-dimensional lattice to update the shower content, and the *memory* operation, which allows for

- previous coin operations to be retained for later use in the algorithm. The coin operation is supported by a *position check* operation to ensure the splittings are implemented correctly. . . . . 108
- 5.4 A schematic circuit diagram of the *position check* and *coin* operation for the quantum walk parton shower algorithm. The position check operation allows for the correct application of the splitting probabilities by controlling from the position of the walker and populating an ancillary register. The coin operation then controls from this ancillary register and applies the correct splitting probabilities which corresponds to the shower content. The ancillary register can then be returned to the ground state by applying the position check in reverse. . . . . 109
- 5.5 Quantum circuit implementation for the  $g \rightarrow q\bar{q}$  shift operation. The first gate group, “Remove  $g$ ”, decrements the number of gluons encoded on the  $x$  position register, reflecting the gluon splitting. The second gate group, “Add  $q\bar{q}$  pair”, increments the position of the walker in the  $y$  position register, corresponding to adding a quark-antiquark pair to the shower content. The  $g \rightarrow q\bar{q}$  controls from a  $|1\rangle$  state on the first coin qubit, which encodes the splitting probability  $\hat{P}'_{qg}$ , and a  $|0\rangle$  state on the second and third coin qubits, which correspond to the  $\hat{P}'_{gg}$  and  $\hat{P}'_{gq}$  splitting probabilities respectively. . . . . 111
- 5.6 Visualisation of the quantum walk parton shower algorithm. The walker’s position on the two-dimensional lattice constructed from the  $x$  and  $y$  positions corresponds to the particle content of the parton shower. Figure (a) shows the result from a two-dimensional parton shower using the DGLAP splitting kernels, (b) shows modified splitting kernels to show how the walker spreads through the lattice. Figure from Reference [2]. . . . . 112
- 5.7 Comparison between the probability distributions for the number of gluons in the



- final state in association with zero quark-antiquark pairs (left) and exactly one quark-antiquark pair(right) for classical and quantum algorithms. The quantum algorithm has been run on the `ibmq_qasm_simulator` for  $5 \times 10^5$  shots on the device, and the classical algorithm has been run for  $10^6$  shots. Figure adapted from Reference [2]. . . . . 115
- 5.8 Comparison between the probability distributions for the number of gluons in the final state in association with zero quark-antiquark pairs (left) and exactly one quark-antiquark pair(right) for classical and quantum algorithms, using modified splitting kernels. The quantum algorithm has been run on the `ibmq_qasm_simulator` for  $10^5$  shots on the device, and the classical algorithm has been run for  $10^6$  shots. Figure adapted from Reference [2]. . . . . 116
- 6.1 Parameterising the phase space in terms of the gluon’s transverse momentum,  $k_{\perp}$ , and rapidity,  $y$ , allowed dipole decays are constrained to a triangular region of phase space. The phase space of effective gluon emission is discretised since ① the running-coupling effects dictate that gluons emitted within a rapidity gap of  $\delta y_g$  are reabsorbed, acting coherently as single effective gluon. The  $\kappa$  (proportional to  $k_{\perp}^2$ ) dimension is consequently quantised due to ② the additional phase space folds opening as a result of gluon emission also being discretised in units of  $\delta y_g$ . Figure from Reference [3]. See the main text and Reference [3] for more details. . 119
- 6.2 The discrete QCD parton shower produces a fractal structure of folds extending from an original triangular phase space region. The algorithm can be reinterpreted as a one-dimensional random walk since ③ the baseline of the fractal encodes all necessary information. The *grove* structure can be ④ generated by a heavily constrained two-dimensional random walk that moves through a lattice with spacing  $\delta y_g$ . Due to the low fractal dimension of the baseline, ⑤ the

- grove structure can be constructed from a one-dimensional random walk. For the one-dimensional walk  $\textcircled{B}$ , the notation  $n/n_{\text{tiles}}^{\text{slice}}$  indicates that tile  $n$  at height  $h = n\delta y_g$  has been selected. The dipole invariants ( $\ln(s_{ij})$ ) can be read directly from the grove by following the baseline from particle  $i$  to particle  $j$ , and skipping segments whose colour was created and reabsorbed along the way (*exempli gratia* skipping the pink segments when calculating the invariant mass of the green and blue gluon tips.). Figure from Reference [3]. . . . . 121
- 6.3 Schematic circuit of the quantum discrete QCD algorithm constructed from four registers: the grove-baseline register,  $\lambda$ , the gluon register,  $g$ , the coin register,  $c$ , and the memory register,  $m$ . The algorithm is based on a quantum walk with memory framework and is discretised into individual steps comprising three main operations: the coin operation, which encodes the probability of selecting a specific tile in the rapidity slice, the shift operation, which moves the walker according to the outcome of the coin operation, and the memory operation, which can be used to record previous coin outcomes. The step is then applied iteratively for all rapidity slices in the phase space before a measurement of the  $\lambda$  and  $g$  registers is performed. See the main text for more details on each operation. Figure has been adapted from Reference [3]. . . . . 127
- 6.4 Coin operations for (left)  $n = 2$  and (right)  $n = 3$  tiles in a rapidity slice. Figure adapted from Reference [3]. . . . . 128
- 6.5 Set of all possible primitive grove structures for  $E_{\text{CM}} = 91.2$  GeV. The groves progress from (A) minimal emission, to (X) maximal emission. Figure from Reference [3]. . . . . 132
- 6.6 Schematic of the tailored circuit for generating grove structures for  $E_{\text{CM}} = 91.2$  GeV. Each step of the quantum walk with memory is constructed from

	three operations: the coin operation, which encodes the probabilities to selecting tile $n$ from the rapidity slice, the shift operation, which adjusts the position of the walker if a gluon has been emitted, and the memory operation, which is used once to record the state of the coin after the first coin operation. Figure adapted from Reference [3]. . . . .	134
6.7	Generation of the primitive grove structures for a centre-of-mass energy of $E_{\text{CM}} = 91.2$ GeV using the <code>ibmq_qasm_simulator</code> and <code>ibm_algiers</code> devices. The results from the simulator and quantum computer have been compared to analytically calculated classical distribution. Figure from Reference [3]. . . . .	135
6.8	Comparison between the results obtained from the classical discrete QCD algorithm, and the quantum discrete QCD algorithm. The quantum algorithm has been run on the <code>ibmq_qasm_simulator</code> device, which simulates a fully fault tolerant device. Thus, as expected, we see exact agreement between the two algorithms. Figure from Reference [3]. . . . .	136
6.9	Sample comparisons of synthetic data produced by the discrete QCD model (DQCD) and the quantum algorithm run on the <code>ibm_algiers</code> device to data taken at the Large Electron Positron collider. The quantum parton shower results from the <code>ibm_algiers</code> device have not been corrected for quantum errors. Figure from Reference [3]. . . . .	139
6.10	Sample comparisons of synthetic data produced by the discrete QCD (DQCD) model for various values of the mass scale, $\Lambda$ . Variations of the mass scale have non-negligible uncertainties for observables that are dominated by perturbative QCD effects. Figure from Reference [3]. . . . .	141
6.11	A comparative study on the affect of varying the tune for observables dominated by non-perturbative QCD effects. It is clear that applying the default <code>PYTHIA</code>	

Monash tune, or the DIRE tune leads to better data description. Figure from Reference [3]. . . . .	143
---	-----





*“A man is a success if he gets up in the morning and gets to bed at night, and in between he does what he wants to do”*

- Bob Dylan, 1967

