

Towards a quantum computing algorithm for helicity amplitudes and parton showers

Khadeejah Bepari^{1,*}, Sarah Malik^{2,†}, Michael Spannowsky^{1,‡} and Simon Williams^{2,§}

¹*Institute for Particle Physics Phenomenology, Department of Physics, Durham University, Durham DH1 3LE, United Kingdom*

²*High Energy Physics Group, Blackett Laboratory, Imperial College, Prince Consort Road, London SW7 2AZ, United Kingdom*



(Received 21 December 2020; accepted 18 March 2021; published 26 April 2021)

The interpretation of measurements of high-energy particle collisions relies heavily on the performance of full event generators, which include the calculation of the hard process and the subsequent parton shower step. With the continuous improvement of quantum devices, dedicated algorithms are needed to exploit the potential quantum that computers can provide. We propose general and extendable algorithms for quantum gate computers to facilitate calculations of helicity amplitudes and the parton shower process. The helicity amplitude calculation exploits the equivalence between spinors and qubits and the unique features of a quantum computer to compute the helicities of each particle involved simultaneously, thus fully utilizing the quantum nature of the computation. This advantage over classical computers is further exploited by the simultaneous computation of s- and t-channel amplitudes for a $2 \rightarrow 2$ process. The parton shower algorithm simulates collinear emission for a two-step, discrete parton shower. In contrast to classical implementations, the quantum algorithm constructs a wave function with a superposition of all shower histories for the whole parton shower process, thus removing the need to explicitly keep track of individual shower histories. Both algorithms utilize the quantum computers ability to remain in a quantum state throughout the computation and represent a first step towards a quantum computing algorithm describing the full collision event at the LHC.

DOI: [10.1103/PhysRevD.103.076020](https://doi.org/10.1103/PhysRevD.103.076020)

I. INTRODUCTION

Modern collider experiments such as the Large Hadron Collider (LHC) at CERN depend heavily on the modeling of particle collisions and simulations of detector response to examine physics processes within the experiments. This modeling is used to construct different possible outcomes from particle collisions, used both for the identification of certain physical processes and for the construction of event backgrounds. Consequently, such simulations play a crucial role in modern high energy physics, and are usually carried out by Monte Carlo event generators such as PYTHIA [1], HERWIG [2], and SHERPA [3].

The theoretical description of LHC events can be highly complex. In a typical event, hundreds of particles are

produced as a result of the evolution of an event from the collision of two protons to the formation of long-lived hadrons, leptons, and photons. The collision process can be separated into several stages. The protons consist of many partons, each carrying a fraction of the total proton energy. When protons collide, two of their partons can interact with each other via a large momentum transfer, thereby giving rise to the so-called hard interaction. In this part of the collision, large interaction scales are probed, possibly accessing new physics. However, if color-charged particles are produced during the hard interaction process, they are likely to emit further partons. This results in a parton shower, providing a mechanism that evolves the process from the hard interaction scale down to the hadronization scale $\mathcal{O}(\Lambda_{\text{QCD}})$, where nonperturbative processes rearrange the partons into color-neutral hadrons.

The hard interaction and the parton shower are the two parts of the event evolution that can be described perturbatively and largely independently of nonperturbative processes, as a result of the factorization theorem [4]. In addition, together with the phase space integration, they are by far the most time-consuming parts of the event generation and pose, therefore, the bottleneck in the generation of pseudo-data for ongoing measurements at the LHC.

*khadeejah.bepari@durham.ac.uk

†sarah.malik@imperial.ac.uk

‡michael.spannowsky@durham.ac.uk

§s.williams19@imperial.ac.uk

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Funded by SCOAP³.

For a mathematical description of the hard interaction, scattering matrix elements are calculated, which nowadays rely on helicity amplitude methods to cope with the ever-increasing complexity of the partonic scattering process [5,6]. Instead, the parton shower is technically implemented through a Markov chain Monte Carlo algorithm ordered in some measure of showering time t , where splitting functions define the probability for a parton to branch into two partons and Sudakov factors [7] determine the probability for the system not to change between two shower times,¹ t_{in} and t_{end} . Recent developments in combining helicity amplitudes with the parton shower have been shown to improve the theoretical description of scattering events, including multiple jets [9–14], in hypothesis testing [15,16] and in particular in the construction of spin-dependent parton showers [17].

With practical quantum computers becoming available, there has been growing interest in harnessing the power and advantages that these machines may provide. This interest extends to applying the abilities of quantum computers to describe processes in field theories, with the hope of exploiting the intrinsic “quantumness” of these novel machines to calculate quantum phenomena efficiently. Current quantum computers are divided into two classes: quantum annealers and universal gate quantum computers (GQCs). The former is based on the adiabatic theorem of quantum mechanics to find the ground state of a complex system. Quantum annealers perform continuous-time quantum computations and are therefore well-suited to study the dynamics of quantum systems, even quantum field theories [18,19], and in solving optimization problems, e.g., applied to Higgs phenomenology [20]. However, they are not universal. Despite their severe limitations due to the relatively small number of qubits of current machines, GQCs are a popular choice for the implementation of algorithms to calculate multiparticle processes [21–31], often with field theories mapped onto a discrete quantum walk [32–35] or a combined hybrid classical/quantum approach [36–39].

This paper presents a first step towards a generic implementation of quantum algorithms, applicable to QGC devices, for two key components of the event generation in high-energy collisions: the calculation of the hard process in terms of helicity amplitudes and the simulation of the parton shower.² These algorithms utilize the unique features of a quantum device and demonstrate distinct advantages over the classical implementations.

As depicted in Fig. 1, quantum computing (QC) calculations proceed in general in three stages: (i) encoding of the initial state, i.e., an initial wave function, using a

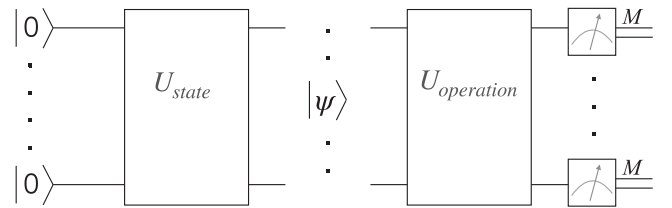


FIG. 1. Schematic setup of generic quantum computing calculations with the following steps: (i) the encoding of the initial state, (ii) the application of (unitary) operations, and (iii) the measurement of the transformed state.

specific representation of the problem, (ii) applying unitary operations on this state, which on a GQC is realized through circuits, and (iii) measuring a specific property of interest, i.e., a projection, onto the final state vector.

Following this structure, we will elucidate how the calculation of a hard process in terms of helicity amplitudes or the parton shower can be performed using a GQC. Specifically, we use the IBM Q Experience [40], which provides access to a range of public access quantum computers and a 32-qubit quantum simulator [41]. We have designed the circuits with a focus on limiting the number of qubits needed to perform the calculations. While our code can be run on a real quantum device, the current quantum machines cannot outperform classical computers. The quantum circuits presented here, therefore, serve as a template and nucleus for future developments.

This paper is organized as follows: In Sec. II, we motivate and detail the implementation of our QC algorithm for helicity amplitudes for a $1 \rightarrow 2$ process and a $2 \rightarrow 2$ scattering process, Sec. III contains the description of our two-step parton shower algorithm, and Sec. IV offers a summary and conclusions.

II. HELICITY AMPLITUDE ALGORITHM

Scattering processes are calculated using conventional techniques by squaring the scattering amplitude and then performing a sum of all possible helicity processes using trace techniques. For a process with N possible Feynman diagrams, this results in N^2 terms in the squared amplitude. Therefore, for processes with a large number of Feynman diagrams, such calculations become extremely complicated. In contrast, helicity amplitude calculations provide a more efficient way of calculating such processes, as one calculates the amplitude for a specific helicity setup. The different helicity combinations do not interfere, and therefore the full amplitude can be obtained by summing the squares of all possible helicity amplitudes.

Helicity amplitude calculations are based on the manipulation of helicity spinors. As the Lorentz group Lie algebra can be written as the direct sum of two $SU(2)$ subalgebras, i.e., $so(3,1) = su(2) \oplus su(2)$, there are two specific complex representations, each specified by two degrees

¹For more details see [8] and references therein.

²A first implementation of a parton shower algorithm was provided in [26], where interference effects in the parton shower evolution were studied.

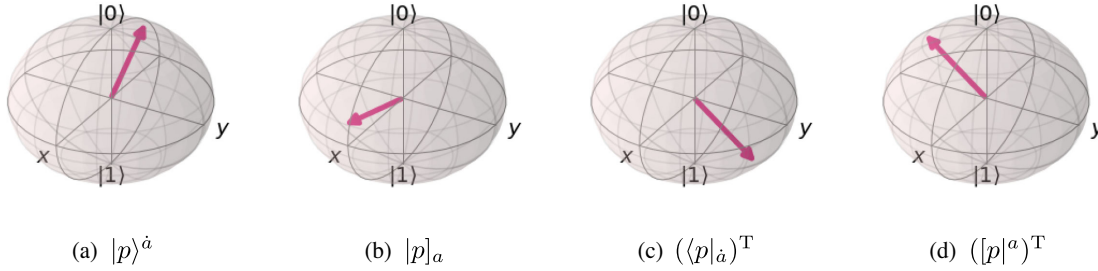


FIG. 2. A visualization of the helicity spinors $|p\rangle^{\dot{a}}$, $\langle p|_{\dot{a}}$, $(|p]_a)^T$, and $([p|^a)^T$ for $\theta = \pi/4$, $\phi = \pi/2$, $E = 1/2$ on the Bloch sphere, following the choice of representation of Eq. (2.1).

of freedom which solve the massless Weyl equation: a right-handed Weyl spinor, associated with the representation $(\frac{1}{2}, 0)$, and a left-handed Weyl spinor, associated with the representation $(0, \frac{1}{2})$. Consequently and for concreteness, the helicity spinor $|p\rangle^{\dot{a}}$ for a massless state can be chosen to be expressed as

$$|p\rangle^{\dot{a}} = \sqrt{2E} \begin{pmatrix} \cos \frac{\theta}{2} \\ \sin \frac{\theta}{2} e^{i\phi} \end{pmatrix}, \quad (2.1)$$

associated with momentum p^μ and energy E , such that $p^\mu p_\mu = -m^2$ using the $\eta_{\mu\nu} = \text{diag}(-1, +1, +1, +1)$ metric convention. This spinor is parametrized by the angles θ and ϕ , where the other spinors $\langle p|_{\dot{a}}$, $|p]_a$, and $[p|^a$ are related by $p_{\dot{a}b} = -|p]_a \langle p|_{\dot{b}}$ and $p^{\dot{a}b} = -[p|^a |p]_b$. The correspondence between the two-dimensional helicity spinors and four-component Dirac spinors associated with Feynman rules is demonstrated in the Appendix B.

To facilitate and implement such calculations on a GQC, we use *qubits*, the quantum analogue of the *bit* for classical computation. The state of the qubit is defined on a two-dimensional complex vector space with states $|0\rangle$ and $|1\rangle$ forming the orthonormal basis for this space. A qubit can thus be formed by a linear superposition of these orthonormal basis states. By considering a general qubit parametrized by two angles,

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle = \begin{pmatrix} \cos \frac{\theta}{2} \\ \sin \frac{\theta}{2} e^{i\phi} \end{pmatrix}, \quad (2.2)$$

we can represent the qubit on a three-dimensional unit sphere called the Bloch sphere. Performing unitary operations on qubit states corresponds to rotating states in the Bloch sphere.

Remarkably, comparing Eqs. (2.1) and (2.2), helicity spinors can be represented through a qubit, modulo an overall normalization factor $\sqrt{2E}$, and the calculation of helicity amplitudes follows an identical structure to the one shown in Fig. 1, i.e., quantum operators act on an initial state to eventually perform the projection onto a final state. In contrast to classical computers, where all numerical

quantities are converted into a binary system representation and on which an algorithm is applied and then transformed back into quantities that can be understood in terms of a numerical result, in a quantum computing algorithm, the helicity spinor is a faithful representation of the object the circuit directly operates on. The spinors can be directly represented as vectors on the Bloch sphere, which provides the most efficient encoding of the state on which the algorithm operates. This indicates that GQCs provide an ideal framework for the calculation of helicity amplitudes.

Consequently, we will exploit that the spinors used to calculate helicity amplitudes naturally live in the same representation space as qubits. This motivates the manipulation of the direct correspondence of the θ and ϕ variables of the qubit states and helicity spinors to represent the spinors on a quantum circuit. We further encode operators acting on spinors as quantum circuits of unitary operations. These can be applied to qubits (rotating vectors on the Bloch sphere) to calculate helicity amplitudes. The helicity spinors $|p\rangle^{\dot{a}}$, $\langle p|_{\dot{a}}$, $|p]_a$, and $[p|^a$ are visualized for $\theta = \pi/4$, $\phi = \pi/2$, $E = 1/2$, as vectors on the Bloch sphere in Fig. 2, in direct analogy to their respective qubit representation.

This study aims to create the basic building blocks to encode spinor helicity calculations on a quantum circuit. These basic building blocks are then used to construct quantum algorithms for two simple examples of helicity calculations: (i) the contraction of an external polarization vector corresponding to a $g \rightarrow q\bar{q}$ vertex, and (ii) the construction of s- and t-channel amplitudes for a $q\bar{q} \rightarrow q\bar{q}$ process with identical initial and final quark flavors. ‘‘Helicity registers’’ are crucially introduced into these circuits to control the helicity of each particle involved. In addition, we introduce a superposition state between the helicity qubits of $|+\rangle = |1\rangle$ and $|-\rangle = |0\rangle$ by applying Hadamard gates to the helicity registers. In doing so, we can calculate both helicities of each particle involved simultaneously, thus fully utilizing the quantum nature of the computation. This advantage is further exploited by the simultaneous computation of s- and t-channel amplitudes for the $q\bar{q} \rightarrow q\bar{q}$ process.

This section is organized as follows: A description of the quantum circuit for the $1 \rightarrow 2$ process of $q \rightarrow gq$ is given in

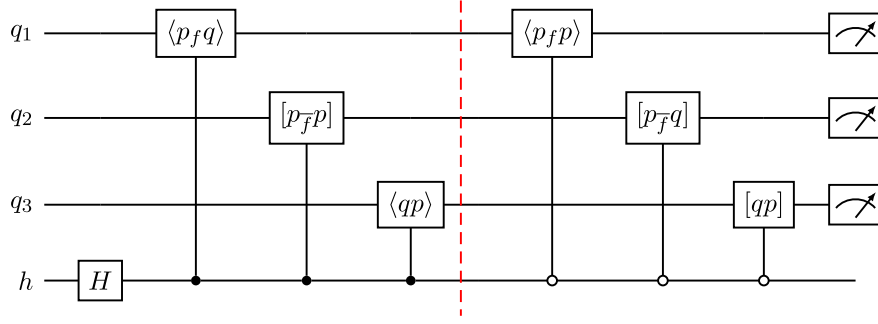


FIG. 3. $gq\bar{q}$ vertex circuit. The amplitude for the process is calculated on the q_i qubits, which are controlled from the helicity register. The q_i qubits are then measured by the quantum computer.

Sec. II B, together with a comparison of the results of the algorithm as run on a real machine and a simulator, the quantum circuit and the results for the $2 \rightarrow 2$ process of $q\bar{q} \rightarrow q\bar{q}$ are given in Sec. II C, and a brief discussion of the generalization of the algorithm to $2 \rightarrow n$ processes follows in Sec. II D.

A. Constructing helicity spinors and scalar products on the Bloch sphere

The helicity spinors have been implemented on the quantum circuit by constructing Bloch sphere representations, like the ones shown in Fig. 2. The helicity spinor decompositions are outlined in detail in Appendix C. They utilize the Qiskit $U_3(\theta, \phi, \lambda)$ gate, which applies a rotation to a single qubit. The rotation is defined by

$$U_3(\theta, \phi, \lambda) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -e^{i\lambda} \sin(\frac{\theta}{2}) \\ e^{i\phi} \sin(\frac{\theta}{2}) & e^{i(\phi+\lambda)} \cos(\frac{\theta}{2}) \end{pmatrix}. \quad (2.3)$$

A simple U_3 gate acting on a $|0\rangle$ state has been used to create the $|q\rangle^{\dot{a}}$ spinor, where θ and ϕ variables of the U_3 gate correspond to the θ and ϕ variables of the helicity spinor. The $|q\rangle_a$ spinor has been created by sequentially applying a U_3^\dagger rotation and a NOT gate, where here the θ and λ variables of the U_3 gate correspond to the θ and ϕ variables of the $|q\rangle_a$ spinor.

To construct the scalar products $\langle pq \rangle$ or $[pq]$ on a quantum computer, 2×2 unitary gates $U_{\langle p}$ and $U_{[p}$ were created such that when they act on the $|q\rangle^{\dot{a}}$ and $|q\rangle_a$ spinors, respectively, the scalar product values correspond to the first component of the final qubit state, i.e., the complex coefficient associated with the $|0\rangle$ state. It should be noted that the factors of $\sqrt{2E}$ in the definition of the helicity spinors have not been accounted for such that the spinor-qubit states are normalized to one on the quantum register. As a consequence, these factors must be added after the results have been obtained from the quantum computer.

B. $1 \rightarrow 2$ amplitude calculation

A simple application of the helicity amplitude approach is the calculation of a $1 \rightarrow 2$ process. Here we will consider the process of $q \rightarrow gq$ by calculating the $gq\bar{q}$ vertex,

$$\mathcal{M}_{gq\bar{q}} = \langle p_f | \bar{\sigma}_\mu | p_{\bar{f}} \rangle \epsilon_{\pm}^\mu, \quad (2.4)$$

where p_f and $p_{\bar{f}}$ are the momenta associated with the fermion and antifermion, respectively. The gluon polarization vectors are defined as [42]

$$\epsilon_+^\mu = -\frac{\langle q | \bar{\sigma}^\mu | p \rangle}{\sqrt{2} \langle qp \rangle}, \quad \epsilon_-^\mu = -\frac{\langle p | \bar{\sigma}^\mu | q \rangle}{\sqrt{2} [qp]}. \quad (2.5)$$

From this, it is possible to create a circuit where each four-vector present in the amplitude, i.e., the fermion antifermion vertex and polarization vector, is calculated individually on a series of four qubits. This is done by using the corresponding Pauli gates for each four-vector component on each qubit. However, this will lead to a large circuit depth due to the number of gates required to do such a calculation. Therefore it is useful to simplify the expression for the amplitude using the Fierz identity,

$$\langle p | \bar{\sigma}^\mu | q \rangle \langle k | \bar{\sigma}_\mu | l \rangle = 2 \langle pk \rangle [ql]. \quad (2.6)$$

With this, the amplitude for the $gq\bar{q}$ vertex becomes

$$\mathcal{M}_+ = -\sqrt{2} \frac{\langle p_f q \rangle [p_{\bar{f}} p]}{\langle qp \rangle}, \quad \mathcal{M}_- = -\sqrt{2} \frac{\langle p_f p \rangle [p_{\bar{f}} q]}{[qp]}. \quad (2.7)$$

As a consequence of this simplification, the number of qubits needed to calculate the amplitude on the quantum computer can be reduced from 10 to 4. The circuit for calculating this amplitude is shown in Fig. 3. The three q_i qubits calculate the three scalar products from Eq. (2.7) using the gate decompositions outlined in Appendix C. These rotation gates are controlled from the helicity register, h . If h is in the $|1\rangle$ state, then the helicity is

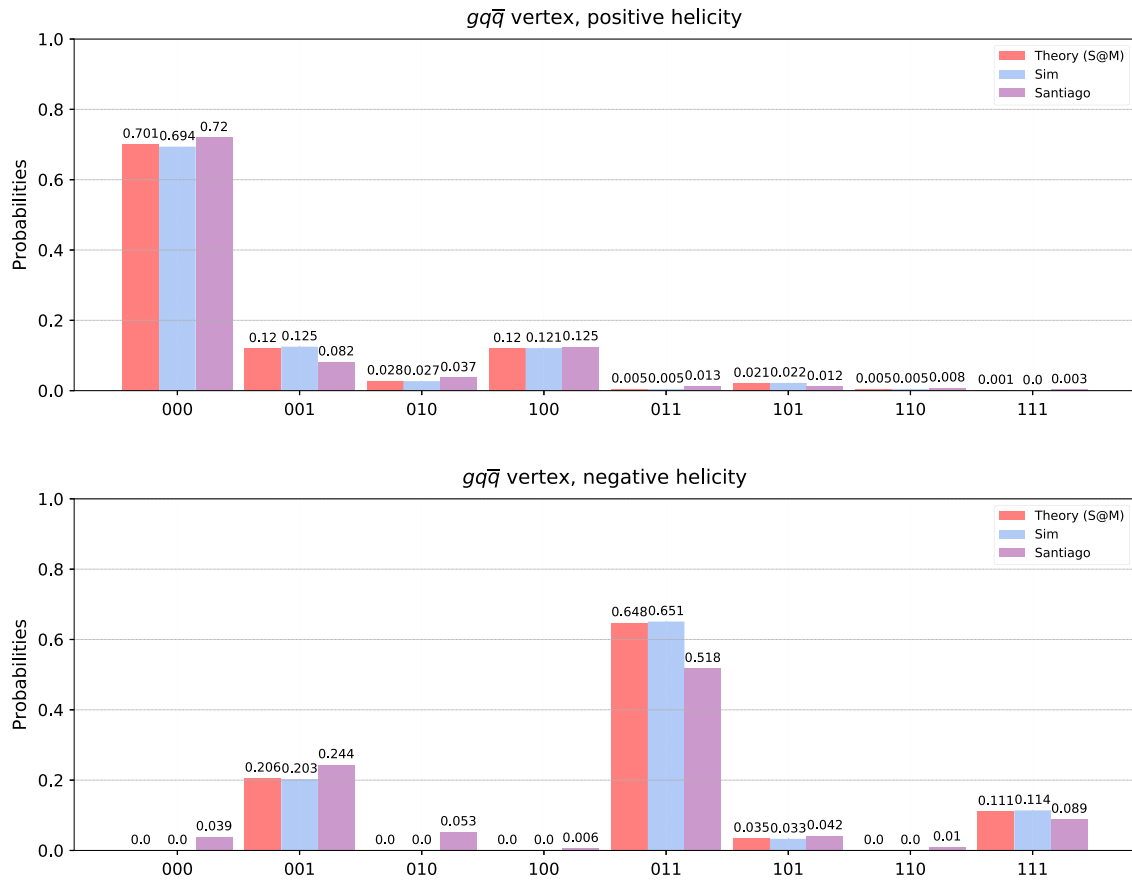


FIG. 4. Results for the $q \rightarrow g\bar{q}$ helicity amplitude calculation. The comparison is between the theoretically calculated probability distribution, the quantum simulator, and the real quantum computer.

positive and the \mathcal{M}_+ amplitude is calculated; if h is in the $|0\rangle$ state, then the helicity is negative and the \mathcal{M}_- amplitude is calculated. This introduces direct entanglement between the helicity qubit, h , and the calculation qubits, q_i . As the same calculation qubits are used to calculate both the positive and negative helicity cases, a superposition of both helicity calculations is introduced to the q_i qubits. The three calculation qubits, q_i , are then measured by the quantum machine. How this algorithm scales for processes involving a large number of final state particles is discussed in detail in Sec. II D.

Figure 4 shows the results of the algorithm for a random selection of small scattering angles, with runs on the IBM Q 32-qubit Quantum Simulator [41] and the IBM Q 5-qubit Santiago Quantum Computer [43]; both of which have been compared to theoretical predictions of the probability distributions extrapolated directly from analytic calculations of the helicity amplitude, calculated using the S@M software [44]. The simulator has been run without a noise profile for 10,000 shots. The results agree well with theoretically predicted values, to within 1σ . From these distributions, one can determine the helicity setup of the process and consequently reconstruct the helicity amplitudes.

The Santiago machine has been run on the maximum shot setting of 8192 for 100 runs, leading to a total of 819,200 shots of the algorithm. Figure 4 shows that the quantum computer's performance does not match that of a perfect machine. This is expected for current quantum devices, as the fidelity of such quantum computers remains low, especially for algorithms with many multiqubit operations. However, effective error mitigation schemes can be applied to the results and the steps taken for this algorithm have been outlined below. As a further check of the algorithm's performance, the quantum simulator has also been rerun with the noise profile of the Santiago device and a comparison between this and the quantum computer is shown and discussed in Appendix D.

The results from the quantum computer, shown in Fig. 4, have been achieved by isolating the individual helicity processes on the quantum circuit, and removing the superposition between the positive and negative processes. The full amplitude is achieved through the implementation of a Hadamard gate on the helicity qubit, which puts the system into a superposition state of the positive and negative processes. The helicity of the process is then determined by measuring the helicity register. The qubit setup chosen here has been used in order to best reduce the controlled-NOT

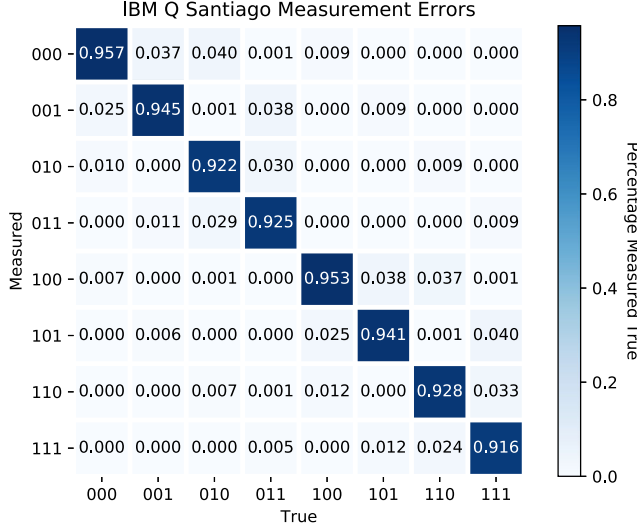


FIG. 5. IBM Q Santiago 5-qubit quantum computer response matrix for the measurement of the error correction on the 4-qubit helicity amplitude calculation algorithm.

(CNOT) qubit errors and limits the number of SWAP operations needed in the algorithm. The Santiago machine is a 5-qubit quantum computer, with all qubits connected inline to their adjacent qubit. The helicity qubit, h , from Fig. 3 has been assigned to the fourth qubit of the Santiago machine, with the q_i qubits on the second, third, and fifth qubits of the Santiago machine. The optimum qubit setup would have the h qubit fully connected to the q_i qubits, thus fully minimizing the SWAP operation errors. However, the available machines with such qubit mapping on the public IBM Q experience have a lower quantum volume than the Santiago machine, which reports a quantum volume of 32. Consequently, the trade of ideal qubit mapping for a better quantum volume has been made.

One of the key sources of error in the quantum computer is readout noise. Error mitigation methods have been used to optimize the output from the quantum computer and reduce readout noise effects. This has been done using the Qiskit Ignis software [40], which provides tools for noise characterization and error correction based on noise models of the quantum machines. The method involves testing simple qubit states on a series of calibration circuits, which are run using the quantum simulator with the noise profile of the Santiago machine. The response matrix created from this is shown in Fig. 5. This response matrix is calculated immediately before running the algorithm and then applied to the machine results to obtain the error corrected results, as shown in Fig. 4.

In addition to the readout noise error mitigation, one could improve the accuracy of the output through post-selection of the zero states in the results from Fig. 4, which are artifacts of multiplying out the qubit state probabilities. Postselection would allow for the user to condition the output of the quantum device in a way similar to the

response matrix method detailed above. Currently, post-processing methods are not supported on the IBM Q machines, however in the future this could lead to an increase in the fidelity of such calculations using the quantum devices.

C. 2 → 2 amplitude calculation

Extending from the 1 → 2 case in Sec. II B, the implementation of a full helicity amplitude calculation for the s- and t-channels of a 2 → 2 scattering process is presented here.³ As an example, we consider a $q\bar{q} \rightarrow q\bar{q}$ process. The initial state quark and antiquark are labeled as particles 1 and 2 respectively and the final state quark and antiquark as 3 and 4. In total, there are only four nonzero helicity configurations possible for each s- and t-channel process. The relevant amplitudes are

$$\begin{aligned}\mathcal{M}_{s(+---)} &= -\langle 2|\bar{\sigma}^\mu|1\rangle \frac{1}{s_{12}} [3|\sigma_\mu|4], \\ \mathcal{M}_{s(+--+)} &= -\langle 2|\bar{\sigma}^\mu|1\rangle \frac{1}{s_{12}} \langle 3|\bar{\sigma}_\mu|4\rangle,\end{aligned}\quad (2.8)$$

and

$$\begin{aligned}\mathcal{M}_{t(++--)} &= -\langle 3|\bar{\sigma}^\mu|1\rangle \frac{1}{s_{13}} [2|\sigma_\mu|4], \\ \mathcal{M}_{t(+++-)} &= -\langle 3|\bar{\sigma}^\mu|1\rangle \frac{1}{s_{13}} \langle 2|\bar{\sigma}_\mu|4\rangle,\end{aligned}\quad (2.9)$$

where the $+/-$ signs denote the helicity of the outgoing-particles 1, 2, 3, and 4, and

$$s_{ij} = -(p_i + p_j)^2 = \langle ij\rangle [ji]. \quad (2.10)$$

The other nonzero amplitudes are obtained by complex conjugation.

The calculation is performed in the center-of-mass (CM) frame and the momenta of individual particles are defined such that the only dependent input variable is the angle θ through which the quark (and antiquark) is scattered. In the CM frame, the overall magnitude of energy E associated with the momenta of each particle also drops out of the final helicity amplitude and is therefore not considered in this example.

In the “all-outgoing” convention of spinor-helicity formalism [42], the momenta of incoming particles are flipped so that the incoming quark (1) [antiquark (2)] is mapped to an outgoing antiquark (quark) with opposite helicity. In the quantum algorithm, each quark-antiquark vertex is calculated on a 4-qubit quantum register, q_i . The outgoing antifermion spinor, \bar{q}/q , is implemented on the vertex

³Note, for the calculation of the 1 → 3 case only minor modifications are needed.

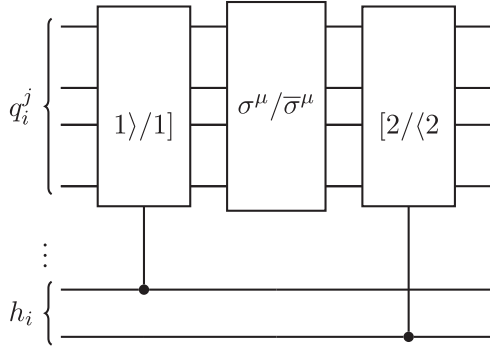


FIG. 6. Circuit for the $q\bar{q} \rightarrow q\bar{q}$ process helicity amplitude calculation. The q_i^j registers are used to calculate the $q\bar{q}$ vertices, and these are controlled from the helicity registers, h_i , which dictate the helicity configuration of the process.

quantum register, q_i^j , followed by the two-dimensional representation of the gamma matrices, $\sigma^\mu/\bar{\sigma}^\mu$, and then finally the vertex is closed with the opposite helicity outgoing fermion spinor, $[q/\langle q]$. A single qubit, s , is used to calculate the denominator of the gluon propagator. The calculation is controlled both from the helicity registers, h_i , which determine what helicity configuration the particles are in, and the amplitude qubit, p , which controls whether the s- or t-channel process is calculated. A schematic of the quantum circuit is shown in Fig. 6. Through this implementation, each component of the helicity amplitude can be calculated and extracted from the machine.

This method is powerful as it allows for each component of the calculation to be extracted, however it leads to a complicated circuit, especially if one implements a method of dealing with incorrect helicity setups. As in Sec. II B, the circuit can be simplified by directly calculating the scalar products required for the final amplitudes. The amplitudes given in Eqs. (2.8) and (2.9) can be simplified using

Eq. (2.6) (and that $[p|\sigma^\mu|q\rangle = \langle q|\bar{\sigma}^\mu|p]$) to give the final forms,

$$\mathcal{M}_{s_{(++++)}} = 2 \frac{\langle 24 \rangle [31]}{\langle 12 \rangle [21]}, \quad \mathcal{M}_{s_{(----)}} = 2 \frac{\langle 23 \rangle [41]}{\langle 12 \rangle [21]}, \quad (2.11)$$

and

$$\mathcal{M}_{t_{(++++)}} = 2 \frac{\langle 34 \rangle [21]}{\langle 13 \rangle [31]}, \quad \mathcal{M}_{t_{(----)}} = 2 \frac{\langle 32 \rangle [41]}{\langle 13 \rangle [31]}. \quad (2.12)$$

Using these expressions, the number of qubits needed for the circuit is reduced from 17 to 12 qubits. Another advantage is that the machine now only has to read out three qubits, where previously eight qubits were read out per run. On these three qubits, each of the scalar products is calculated. The quark-antiquark vertex scalar products from the numerator are calculated on the first two qubits, and the denominator of the gluon propagator is calculated on the third qubit. Only one scalar product needs to be calculated for the denominator since [42]

$$\langle ij \rangle = [ji]^*, \quad (2.13)$$

therefore the second scalar product can be determined from the same qubit.

This simplified circuit is run on the IBM Q 32-qubit Quantum Simulator [41] for 10,000 runs and compared to theoretically calculated probability distributions, extrapolated directly from analytic calculations of the helicity amplitude, calculated using the s@m software [44]. Using the equivalence between helicity spinors and orthogonal pure state qubits, these theoretical predictions have been obtained from the probabilities of each of the qubits to be in the $|0\rangle$ or $|1\rangle$ state, which correspond to the magnitude squared of the upper and lower components of the helicity

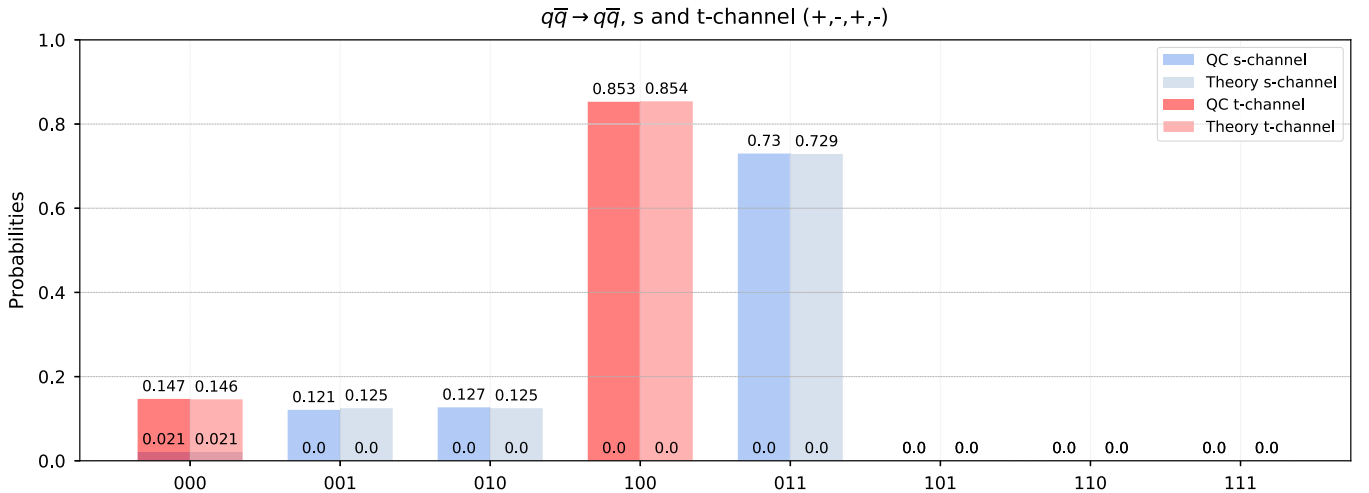


FIG. 7. Comparison between theoretically predicted qubit final state probabilities and 32-qubit quantum simulator output for the s- and t-channel $q\bar{q} \rightarrow q\bar{q}$ process in the $(+, -, +, -)$ helicity configuration. The quark (antiquark) scattering angle has been chosen as $\theta_3 = \frac{\pi}{4}$.

spinor, respectively. The results from the quantum simulator show that the output of the quantum circuit lies within 1σ of the theoretically predicted probability distribution and are shown in Fig. 7 for both the s- and t-channel in a specific helicity configuration.

D. Generalization to $2 \rightarrow n$ amplitude calculations

It can be shown, using the BCFW recursion formula [45,46] and the relations in Eq. (2.5), that scattering amplitudes for massless partons can be reduced to a combination of scalar products between helicity spinors.⁴ Consequently, the algorithm presented in Secs. II B and II C can be generalized to multiparticle amplitudes straightforwardly as the tools are already created, namely the circuit decompositions of the helicity spinors from Appendix C. Since E is an input to the calculation of the squared matrix element even for very large n , including the multiplicative factors of $\sqrt{2E}$ for each particle remains straightforward. The number of calculation qubits, q_i , and the number of helicity qubits, h_i , needed in the algorithm both scale linearly with the number of final state particles, n . As the number of helicity qubits, h_i , scales linearly, then so does the number of work qubits needed in the algorithm. Each scalar product calculation requires two spinor operations, and so the algorithm can be easily extended without adding disproportionate complexity. The circuit depth scales linearly with an increase in the number of scalar products, calculated on the q_i qubits, and the number of helicity qubits, h_i , added to the circuit.

It is interesting and practical to consider the extension of the simple helicity amplitude algorithms presented here to more complicated processes that are likely to be present in high-energy collisions, such as those studied at the LHC. As we have seen in Sec. II B, modern public access quantum computers do not perform to a standard where one could extrapolate accurate calculations of helicity amplitudes, even for a single vertex. However, the performance of public access computers is well below that of state of the art machines, such as the IBM 53-qubit machine and the Honeywell machine. The latter, in unpublished work, claims to have the world's best Quantum Volume of 64 [47]. Such computers do not have the same restrictions as the smaller, less capable public access machines. The more powerful machines offer more choice for qubit setup and mapping, and the ability to perform more operations

⁴A well-known example is the Parke-Taylor formula for a $2 \rightarrow n$ gluon scattering process, where the gluons i and j have helicity $(-)$ and all other gluons have helicity $(+)$. Then the formula provides the following expression for the amplitude \mathcal{A}_n :

$$\mathcal{A}_n[1^+ \dots i^- \dots j^- \dots n^+] = (-g_s)^{n-2} \frac{\langle ij \rangle^4}{\langle 12 \rangle \langle 23 \rangle \dots \langle n1 \rangle}. \quad (2.14)$$

before decoherence in the machine starts to affect the circuit output. We can speculate that the algorithms presented here would be very accurate on these machines, especially the vertex calculation, which comprises a maximum of only 33 operations across 4 qubits. Currently, the algorithm is restricted to massless partons by exploiting the BCFW recursion formula. The advancement in quantum technologies will make the implementation of the full algorithm, capable of simulating massive partons, computationally feasible, thus extending the ability of the proposed helicity amplitude algorithms.

The main difficulty of extending such algorithms for helicity amplitude calculations on quantum computers comes not only from limitations due to the number of qubits, but also the machine's fault tolerance. The more complicated the helicity amplitude calculation, the more operations are needed to calculate it. Therefore, a machine needs not only sufficient qubits but also the ability to implement many operations without excess noise. The algorithms proposed here also rely on fault tolerant implementation of multiqubit gates. As a result the immediate challenge is not the number of qubits available, but the number of operations that can be reliably implemented on the circuit. With advancements in the Quantum Volume of quantum computers [48], this limitation will likely be overcome on current hardware. It is possible that near-future computers will have the ability to perform accurate and precise calculations and also have a large number of qubits. IBM recently announced their road map for the future and the goal of having machines with the number of qubits exceeding 1,000 by 2023 [49]. Therefore, it is highly likely that these near-future devices will be able to perform precise helicity amplitude calculations for processes with a large number of particles.

III. PARTON SHOWER ALGORITHM

After the hard process is calculated, the next step in simulating a scattering event at a high-energy collider experiment is the parton shower stage. The parton shower evolves the scattering process from the hard interaction scale down to the hadronization scale. We propose an algorithm for simulating a QCD parton shower using IBM Quantum Experience [40] software and hardware. The quantum circuit has been implemented to simulate a two-step QCD parton shower with collinear splittings only. Section III A provides the theoretical outline for the shower algorithm and discusses the splitting functions and probability calculations implemented in the quantum circuit. A brief overview of the quantum circuit is given in Sec. III B, and a comparison between the results of the algorithm and theoretically calculated probability distributions are discussed in Sec. III C. A glossary of quantum logic gates is given in Appendix A and a detailed overview of the quantum circuit for the algorithm in Appendix E.

A. Theoretical outline of shower algorithm

We present a parton shower algorithm with the ability to simulate a general, discrete QCD parton shower, harnessing the quantum computers ability to remain in a quantum state throughout the algorithm. In contrast to classical methods, the algorithm does not need to explicitly keep track of individual shower histories. Instead, our algorithm constructs and maintains a wave function that consists of a superposition of all possible shower histories, with the final measurement projecting out a specific quantity of the final state. Consequently, the algorithm presented inherently simulates the quantum interference between all possible final states, without the need for extensive computational logic present in current classical algorithms. In a classical algorithm, a physically meaningful quantity can only be extracted from a parton shower calculation after summing over all possible shower histories, requiring them to be stored on a physical memory device. Our quantum algorithm avoids the need for such an intermediate step, as the measurement is performed on the superposition of all shower histories directly.

The goal is to create the foundation for constructing a general quantum algorithm that can simulate a full QCD parton shower. To comply with the current capabilities of public access quantum computers and simulators provided by IBM Quantum Experience [40], the algorithm presented here uses a simplified model consisting of one flavor of quark and a gluon. This reduces the number of qubits needed, and the algorithm can be run on the IBM Q 32-qubit Quantum Simulator [41]. To further reduce the number of required qubits, only collinear splittings are considered within the model. By neglecting the soft-limit, there is no need to keep track of the detailed kinematics of the particles in the shower history.

Collinear emission occurs when a parton splits into two massless particles which have parallel four-momenta, such that the total momentum, P , is distributed between the particles as

$$p_i = zP, \quad p_j = (1-z)P, \quad (3.1)$$

and thus, $(p_i + p_j)^2 = P^2 = 0$ [50].

The emission probabilities in the algorithm are calculated using the collinear splitting functions outlined in [51–54]. A consequence of the collinear limit being a semiclassical interpretation with 1-to-2 splittings leads to the presence of a diagonal color charge in the splitting functions, C_{ii} . The splitting for a quark to a gluon and a quark, with momentum fractions z and $1-z$, respectively, is described at leading order by

$$P_{q \rightarrow qg}(z) = C_F \frac{1 + (1-z)^2}{z}, \quad (3.2)$$

with $C_F = 4/3$. The gluon splitting can be divided into two parts, with the first describing the splitting of a gluon to a

quark-antiquark pair and the second describing the splitting of a gluon to two gluons,

$$\begin{aligned} P_{g \rightarrow q\bar{q}}(z) &= n_f T_R (z^2 + (1-z)^2), \\ P_{g \rightarrow gg}(z) &= C_A \left[2 \frac{1-z}{z} + z(1-z) \right], \end{aligned} \quad (3.3)$$

where $C_A = 3$ and $T_R = 1/2$. Here, n_f is the number of massless quark flavors, and T_R is the color factor. It should be noted that both splitting functions have a soft singularity at $z = 0$; the hard-collinear limit only takes into account finite z .

In addition to calculating the splitting functions, the Sudakov factors have been used to determine whether an emission occurred in the step. The Sudakov factors for a QCD process are given by [7]

$$\Delta_{i,k}(z_1, z_2) = \exp \left[-\alpha_s^2 \int_{z_1}^{z_2} P_k(z') dz' \right], \quad (3.4)$$

and are used to calculate the nonemission probability. The running of the strong coupling, α_s , is not simulated in this algorithm and for ease has been set to one. For any given step N , there are N possible particles present, and so the probability that none of the particles split is given by

$$\Delta_{\text{tot}}(z_1, z_2) = \Delta_g^{n_g}(z_1, z_2) \Delta_q^{n_q}(z_1, z_2) \Delta_{\bar{q}}^{n_{\bar{q}}}(z_1, z_2). \quad (3.5)$$

Finally, the probability of a certain splitting is therefore obtained from

$$\text{Prob}_{k \rightarrow ij} = (1 - \Delta_k) \times P_{k \rightarrow ij}(z). \quad (3.6)$$

To implement the algorithm efficiently, preference has been given to gluons splitting to a quark-antiquark pair. This splitting preference implementation is explained in depth in Appendix E, but, for definiteness, the probability of a gluon splitting to two gluons is calculated as

$$\text{Prob}_{g \rightarrow gg} = (1 - \Delta_g) \times (1 - P_{g \rightarrow q\bar{q}}(z)) \times P_{g \rightarrow gg}(z). \quad (3.7)$$

For the energy scale considered here, this should have a small affect on the results as $P_{g \rightarrow q\bar{q}}(z) \ll P_{g \rightarrow gg}(z)$.⁵

The theoretical outline above describes a simplified parton shower as a toy model that is possible to simulate using the public access IBM Q 32-qubit Quantum

⁵The quantum algorithm proposed in Sec. III aims to calculate the perturbative evolution from the hard interaction scale of the collision event down to the hadronization scale. Other quantum algorithms have been proposed to calculate parton correlation and structure functions in terms of gauge-invariant light-front operators [55,56]. As both algorithms are aiming to calculate different yet highly important quantities for the prediction of collision events, these different methods complement each other well.

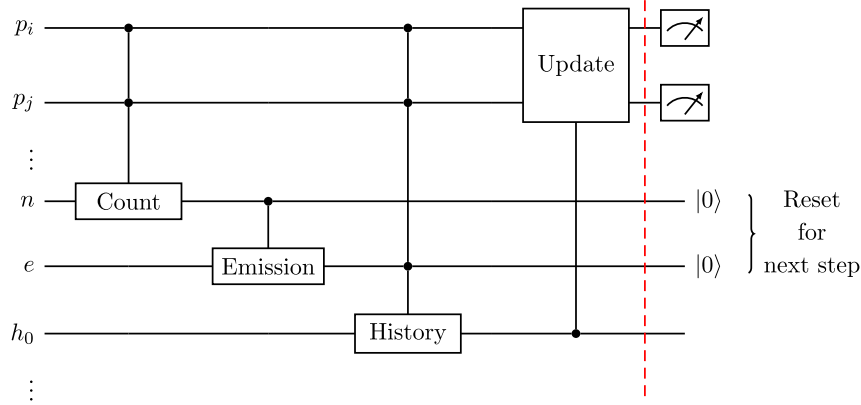


FIG. 8. Circuit diagram for one step of the algorithm. The circuit comprises particle registers, emission registers, and history registers.

Simulator [41]. Here, the restrictions come from the number of qubits available on the simulator. The toy model is, of course, a simplification of the current state-of-the-art classical parton shower algorithms, which have the functionality to include the running of the strong coupling constant, incorporate soft emissions, and do not need to discretize the energies at which the splittings can occur. However, future fault-tolerant quantum devices with enough qubits will be able to support the full simulation of a parton shower. The simplified model presented here serves as a proof-of-concept for a general QCD parton shower quantum algorithm.

B. Implementation on quantum circuit

A quantum circuit has been constructed to simulate a parton shower with collinear splittings. The circuit is comprised of particle registers, emission registers, and history registers and uses a total of 31 qubits. The algorithm is discretized into individual steps. An emission can occur in each step, and the probabilities are calculated from the splitting functions and the Sudakov factors. To meet the 32-qubit limit of the IBM Q Quantum Simulator [41], the algorithm has been limited to two steps, but it is generally extendable. Figure 8 shows the circuit diagram for a single step.

The algorithm follows a similar method to that described in [26], first counting the particles present in the simulation, determining whether an emission has occurred and if so, assessing which splitting did occur, then finally updating the particle content of the simulation. In contrast to the method shown by [26], the algorithm presented here has the ability to simulate a QCD process with splittings for both gluons and quarks implemented using the splitting functions outlined in Eqs. (3.2) and (3.3). The addition of such splitting functions leads to significant changes to the algorithm compared to that presented in [26], specifically in the history and update gates of the algorithm, shown in Fig. 8. The implementation of these gates is outlined in detail in Appendix E. Unlike the algorithm presented

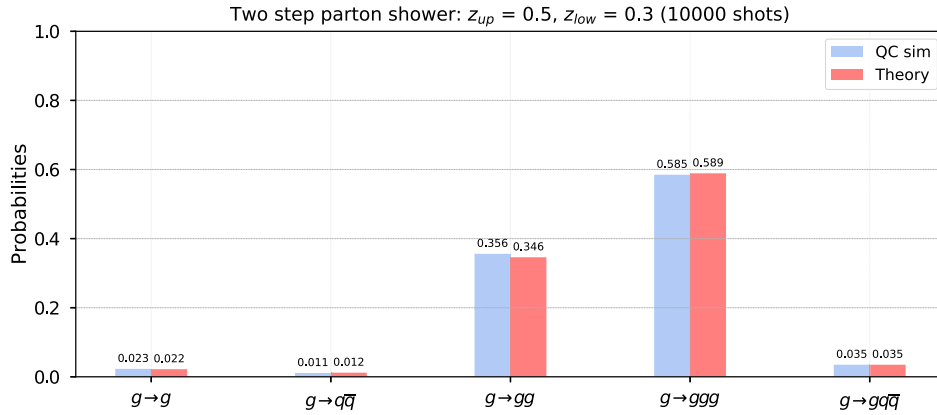
in [26], we have chosen not to introduce flavor mixing at the start of the algorithm. Instead, the superposition and interference between the possible output states are introduced in the tailored history and update gates. These gate processes are the basis of the quantum advantage provided by this algorithm. The quantum computer constructs a wave function of all possible shower histories in full superposition, thus the difficulty of keeping track of all individual histories in classical algorithms is removed. At the end of the algorithm, a physically meaningful quantity is then projected out by measurement. With the ability to simulate gluon and quark splittings, the algorithm is thus well-suited to hadronic parton shower simulation and provides the foundations for a general parton shower algorithm for use on a QQC.

The parton shower algorithm is designed to operate on the public access IBM Q 32-qubit Quantum Simulator [41], which allows for a total of two steps to be simulated on the machine. As the machine is a simulator, it does not suffer from noise or a limit on the number of operations due to qubit decoherence effects, therefore giving a simulation of a perfect machine. As a consequence, error checking is easily done with direct comparison to theoretically predicted probability distributions, and this is discussed in Sec. III C.

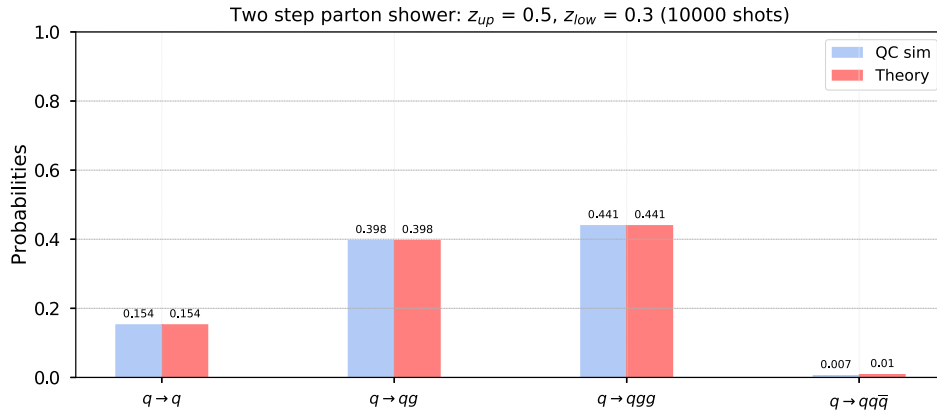
One of the main benefits of using a QC algorithm for the simulation of QCD parton showers over classical methods is the computational simplicity of the algorithm. When dealing with interference of different splittings in the shower process, the algorithm presented here offers a much less computationally complex approach than that provided by modern Monte Carlo event generators. This is achieved by utilizing the unique ability to maintain the quantum computer in a fully quantum state throughout the algorithm, and only collapse to a classical circuit by measurement at the end of the process. This allows for the system to account for all possible parton shower histories simultaneously. In contrast, modern Monte Carlo methods must manually keep track of the particle splitting histories to consider all possible contributions to a specific final state. For a two-step, discrete parton shower, this is a relatively easy task for

a modern Monte Carlo generator. However, the quantum computing field is still in its infancy; the true potential of quantum computing for simulating QCD parton showers will become apparent with the advancement of quantum technologies. With more available qubits and machines

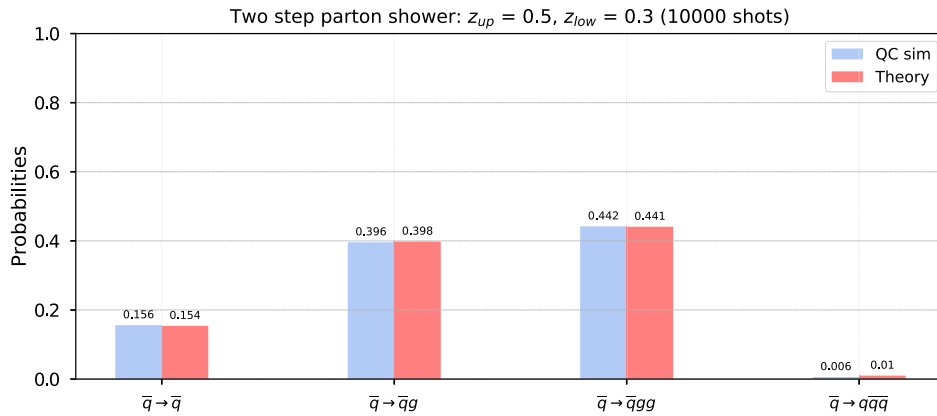
with improved hardware, the algorithm presented here will have the ability to simulate quantum effects, without the extensive and complex computational logic that a classical computer would need. Therefore, quantum computers offer an avenue to explore processes that contain a large number



(a) Initial particle a gluon.



(b) Initial particle a quark.



(c) Initial particle an antiquark.

FIG. 9. Results from the quantum circuit compared to theoretical predictions for two steps of the parton shower with momentum interval of $z_{\text{lower}} = 0.3$ to $z_{\text{upper}} = 0.5$ and the initial state particle of (a) gluon, (b) quark, and (c) antiquark.

of shower particles, thus requiring complicated parton histories and computing power, not currently achievable with modern classical techniques. Beyond QCD parton showers, this feature of a quantum computing algorithm can be of particular interest for cosmic-ray air showers, where millions of long-lived particles are simulated [57,58].

C. Results of parton shower

A comparison of the output from the parton shower algorithm and theoretical predictions of the splitting probabilities is made, and the results are shown in Fig. 9. The algorithm was run for 10,000 shots using the IBM Q 32-qubit Quantum Simulator [41], with a momentum interval of $z_{\text{lower}} = 0.3$ to $z_{\text{upper}} = 0.5$, and no noise simulation. Here, the theoretical predictions have been calculated using the collinear splitting functions from Eqs. (3.2) and (3.3), using the method outlined in Sec. III A. The z value used for the particle splitting probabilities from Eq. (3.7) is the midpoint of the momentum interval used in the algorithm. The results are in agreement with the theoretically calculated probabilities to within 1σ .

A consequence of running the algorithm on a quantum simulator is that there will be no noise in the results, unlike a real quantum computer. Therefore, problems with the algorithm can be identified through direct comparison with the theoretical calculations. In the future, if the algorithm can be run on a real quantum computer with enough qubits, then IBM Q offers a range of noise reducing schemes for its devices through the Qiskit software [40]. Another advantage of using the quantum simulator is that it automatically chooses an optimum qubit setup. In a real quantum computer, the user has to select a qubit mapping in order to optimize the operation of the computer. For future use of the algorithm, this can be done using the calibration data provided by IBM Q.

IV. SUMMARY AND CONCLUSIONS

Accurate modeling of complex collisions at experiments like the LHC relies on theoretical calculations of multi-particle processes. Such calculations can be factorized into the hard interaction, which models the large momentum transfer, and the parton shower, which models the evolution from the hard interaction scale down to the hadronization scale.

We present general and extendable quantum computing algorithms to calculate the hard interaction and parton shower processes, as a first step towards a quantum computing algorithm to describe the full collision event at the LHC.

These algorithms demonstrate distinct advantages over their corresponding classical implementations, summarized in the following:

- (i) The hard interaction calculation proceeds via helicity amplitudes and exploits the equivalence of spinors and qubits. It uses qubits to represent spinors and encodes operators as a series of gates in the quantum circuit. In a classical algorithm, the spinors have to be encoded as binary numbers, an algorithm acting on binary numbers has to be designed, and eventually the result must be transformed back into real/complex numbers for postprocessing. In contrast, the one-to-one correspondence between spinors and qubits in our quantum algorithm allows the entire calculation to be performed directly on the spinors.
- (ii) Quantum algorithms are shown for two simple examples of helicity calculations: the $q \rightarrow qg$ process and the $q\bar{q} \rightarrow q\bar{q}$ process. The algorithm constructs a full wave function for the complete process with a superposition between the helicity qubits. In doing so, it calculates the positive and negative helicities of each particle involved, simultaneously. This unique advantage offered by the quantum machine is further exploited in the simultaneous computation of s- and t-channel amplitudes for the $q\bar{q} \rightarrow q\bar{q}$ process. The algorithm is designed to be general and extendable and expected to scale linearly with the number of particles, hence, on par with its classical implementation.
- (iii) The parton shower algorithm simulates collinear emission in a two-step, discrete parton shower. In contrast to classical implementations of parton showers, where individual shower histories have to be stored on a physical memory device and analyzed in their entirety to provide information on a physical quantity, our quantum computing algorithm constructs a wave function for the whole parton shower process, containing a superposition of all possible shower histories. Therefore, we do not need to keep track of individual shower histories explicitly, and a desired physical quantity of the shower process can be directly obtained through a measurement of the wave function. This also leaves open the possibility of performing further postprocessing on the full parton shower wave function.

Advancements in quantum technologies will allow both algorithms to be extended to more complex processes without adding disproportionate computational complexity. With IBM recently setting their goal of exceeding 1,000 qubits by 2023 [49] and advancements in the development of devices with better Quantum Volume [48], we are on the brink of a quantum revolution. These developments will allow the algorithms presented in this paper to be extended to reflect the processes seen in experiments such as the LHC.

ACKNOWLEDGMENTS

We would like to acknowledge the use of the IBM Q for this work. We are grateful to the authors of [26]

for answering questions on the circuit presented in their work and for sharing their preliminary codes. M. S. would like to thank Steve Abel and Daniel Maitre for helpful discussions. K. B. and M. S. are supported by the STFC under Grant No. ST/P001246/1. S. M. and S. W. are supported by a Grant from the Royal Society.

APPENDIX A: QUANTUM LOGIC GATE DEFINITIONS

- (i) *NOT gate*—a *NOT* gate is a single qubit operation which flips the state of the qubit:

$$\text{NOT}|0\rangle = |1\rangle, \quad \text{NOT}|1\rangle = |0\rangle.$$

The circuit representation of a *NOT* gate is:



- (ii) *CNOT gate*—a *controlled-NOT* (*CNOT*) gate is a 2-qubit operation which flips the state of a target qubit dependent on the state of a control qubit:

$$\begin{aligned} \text{CNOT}|00\rangle &= |00\rangle, & \text{CNOT}|01\rangle &= |01\rangle, \\ \text{CNOT}|10\rangle &= |11\rangle, & \text{CNOT}|11\rangle &= |10\rangle. \end{aligned}$$

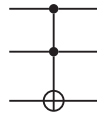
Here, the first qubit is the control. The circuit representation of a *CNOT* gate is:



- (iii) *Toffoli gate* (*CCNOT*)—A Toffoli gate is a 3-qubit operation, which is just a further extension of the *NOT* gate with two control qubits:

$$\begin{aligned} \text{CCNOT}|000\rangle &= |000\rangle, & \text{CCNOT}|001\rangle &= |001\rangle, \\ \text{CCNOT}|100\rangle &= |100\rangle, & \text{CCNOT}|010\rangle &= |010\rangle, \\ \text{CCNOT}|110\rangle &= |111\rangle, & \text{CCNOT}|111\rangle &= |110\rangle. \end{aligned}$$

The circuit representation of a Toffoli gate is:



- (iv) *Hadamard gate*—a Hadamard gate is a purely quantum logic gate and does not have a classical logic gate equivalent. A Hadamard gate is a single qubit operation which puts a qubit into a superposition:

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

The Hadamard gate can be controlled, and so is only applied depending on the state of the control qubit. The circuit representation of a Hadamard gate is:



APPENDIX B: DIRAC AND HELICITY SPINOR CORRESPONDENCE

The following demonstration of the correspondence between Dirac spinors and Helicity spinors can be seen in Chapter 2 of [42].

Fermion and antifermion spinors satisfy the Dirac equations such that,

$$(\not{p} + m)u(p) = 0, \quad (-\not{p} + m)v(p) = 0, \quad (\text{B1})$$

where both equations have independent solutions which can be labeled by subscripts $s = \pm$. One can move to a basis where the \pm denotes spin up/down along the z -axis, by ensuring that spinors u_{\pm} and v_{\pm} are eigenstates of the z -component of the spin-matrix in the rest frame. For massless fermions, \pm denotes the helicity, the projection of the spin along the momentum of the particle. These spinors are also associated with the conventional Feynman rules for external fermions, e.g., $v_{\pm}(p)$ for an outgoing antifermion and $\bar{u}_{\pm}(p)$ for an outgoing fermion.

For the massless case, the Dirac equations reduce to

$$\not{p}v_{\pm}(p) = 0, \quad \bar{u}_{\pm}(p)\not{p} = 0, \quad (\text{B2})$$

where $v_{\pm}(p)$ and $u_{\pm}(p)$ are the wave functions associated with outgoing antifermions and fermions, respectively. For this case the wave functions are related as $u_{\pm} = v_{\mp}$ and $\bar{v}_{\pm} = \bar{u}_{\mp}$. The two independent solutions of the Dirac equations can be written as

$$v_{+}(p) = \begin{pmatrix} |p\rangle_a \\ 0 \end{pmatrix}, \quad v_{-}(p) = \begin{pmatrix} 0 \\ |p\rangle_{\dot{a}} \end{pmatrix}, \quad (\text{B3})$$

and

$$\bar{u}_{-}(p) = (0 \quad \langle p|_{\dot{a}}), \quad \bar{u}_{+}(p) = (\langle p|^a \quad 0), \quad (\text{B4})$$

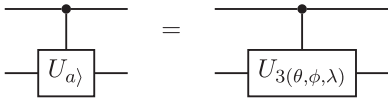
where the angle and square spinors are two-component spinors that satisfy the massless Weyl equation.

APPENDIX C: HELICITY AMPLITUDE GATE DECOMPOSITIONS

- (i) *U_a gate*—The U_a takes the form of a conventional U_3 rotation gate,

$$U_a = U_3(\theta, \phi, \lambda) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -e^{i\lambda} \sin(\frac{\theta}{2}) \\ e^{i\phi} \sin(\frac{\theta}{2}) & e^{i(\phi+\lambda)} \cos(\frac{\theta}{2}) \end{pmatrix}. \quad (\text{C1})$$

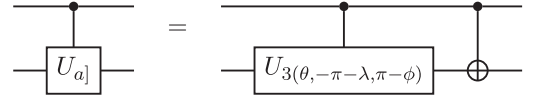
Therefore, the circuit representation is just a qiskit U_3 rotation,



(ii) $U_{|a\rangle}$ gate—The $U_{|a\rangle}$ has the matrix form

$$U_{|a\rangle}(\theta, \phi, \lambda) = \begin{pmatrix} -e^{-i\lambda} \sin(\frac{\theta}{2}) & e^{-i(\phi+\lambda)} \cos(\frac{\theta}{2}) \\ \cos(\frac{\theta}{2}) & e^{-i\phi} \sin(\frac{\theta}{2}) \end{pmatrix}. \quad (\text{C2})$$

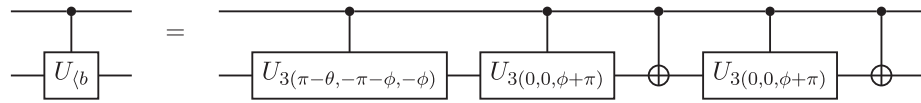
Therefore, this gate has the circuit representation:



(iii) $U_{\langle b}$ gate—The $U_{\langle b}$ has the matrix form

$$U_{\langle b}(\theta, \phi) = \begin{pmatrix} -e^{i\phi} \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \\ \cos(\frac{\theta}{2}) & e^{-i\phi} \sin(\frac{\theta}{2}) \end{pmatrix}. \quad (\text{C3})$$

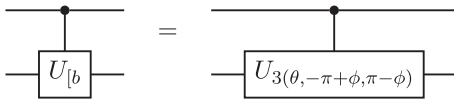
Therefore, this gate has the circuit representation:



(iv) $U_{|b}$ gate—The $U_{|b}$ has the matrix form

$$U_{|b}(\theta, \phi) = \begin{pmatrix} \cos(\frac{\theta}{2}) & e^{-i\phi} \sin(\frac{\theta}{2}) \\ -e^{i\phi} \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix}. \quad (\text{C4})$$

Therefore, this gate has the circuit representation:



APPENDIX D: HELICITY AMPLITUDE CALCULATION CIRCUIT DIAGRAMS AND FURTHER RESULTS

1. 1 → 2 amplitude calculation

Here we present the detailed circuit diagram for the $q \rightarrow gq$ process, shown in Fig. 10, which is implemented using the helicity amplitude gate decompositions outlined in Appendix C. This demonstrates the simplification achieved by using fully contracted helicity amplitudes in the calculation, with a scalar product calculated on each

qubit. The first slice in the circuit diagram (to the left of the vertical dashed line) calculates the positive helicity controlling from the h register in the $|1\rangle$ state, and the second slice (to the left of the vertical dashed line) controls from the h register in the $|0\rangle$ state and calculates the negative helicity process. A superposition of both the positive and negative processes, and thus the full amplitude, is achieved by implementing a Hadamard gate on the helicity qubit, h .

In Fig. 11, a comparison between the output of the IBM Q Santiago 5-qubit Quantum Computer [43] and the IBM Q 32-qubit Quantum Simulator [41] run with the Santiago device's noise profile is presented. The quantum computer has been run for 100 runs of 8192 shots, giving a total of 819,200 shots on the circuit, and the simulator has been run for 10,000 shots. Here we see a more reasonable agreement between the noisy simulator and the output from the quantum computer than the comparison to the perfect machine simulator from Sec. II B. However, it should be noted that the noise profile used in the noisy simulation is only an approximation of the real quantum computer errors. Noise profiles are built from a limited number of parameters and are based on average measurements of qubit errors [40]. As a result, some discrepancies are present

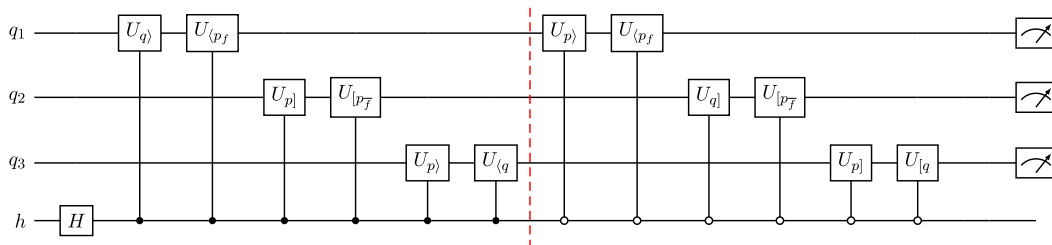


FIG. 10. Detailed circuit diagram for the $q \rightarrow gq$ helicity amplitude calculation, using gate decompositions outlined in Appendix C. The amplitude for the process is calculated on the q_i qubits, which are controlled from the helicity register. The q_i qubits are then measured by the quantum computer.

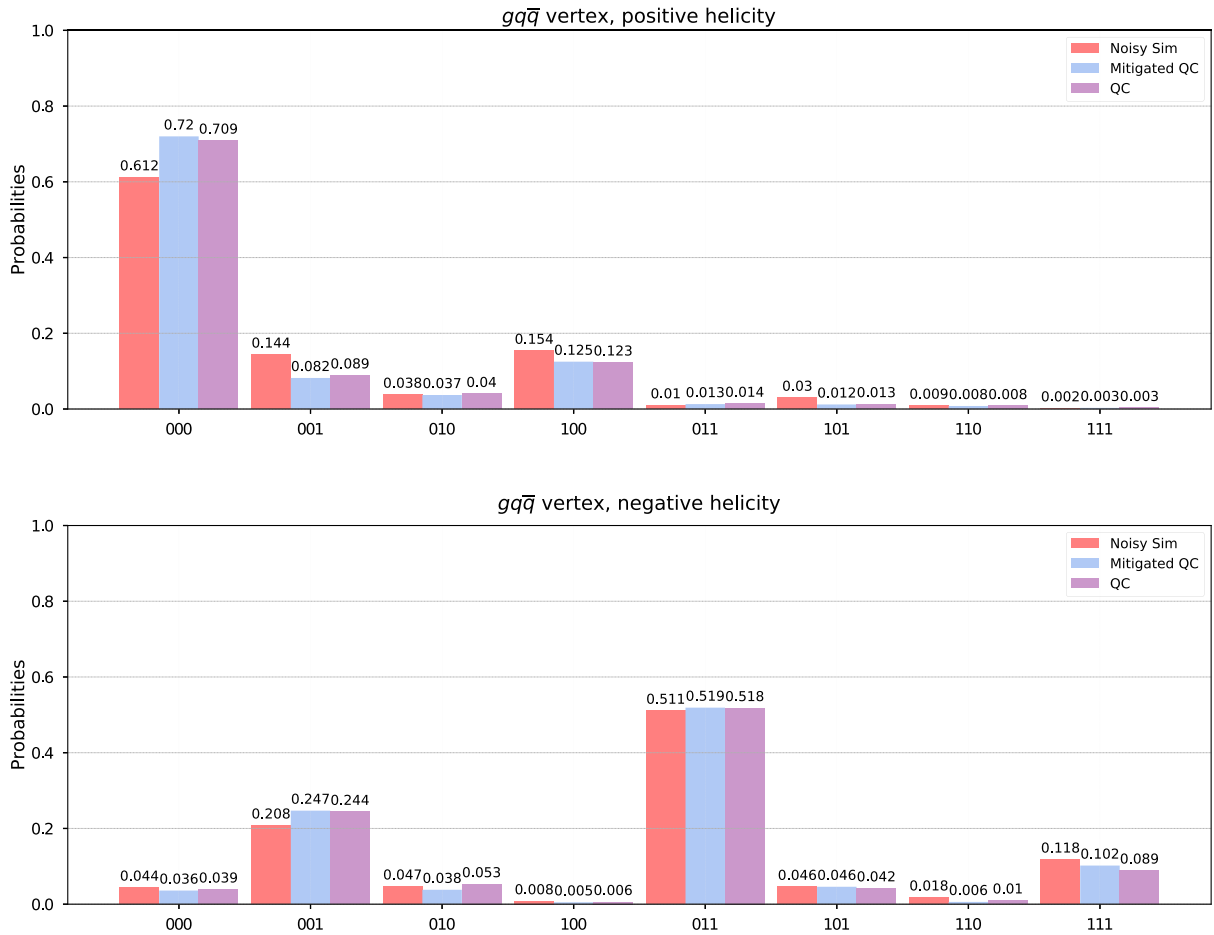


FIG. 11. Results for the $q \rightarrow g\bar{q}$ helicity amplitude calculation. Comparison between the results from a quantum simulator with the relevant machine noise profile, the results from the Santiago quantum computer and the error mitigated results from the quantum computer.

between the quantum simulator with the Santiago device's noise profile and the real device. These can be attributed to noise not accounted for in the quantum computer.

While obtaining the results, we noticed a discrepancy between separate runs on the negative helicity case. By changing the qubit mapping in the measurement process, this was identified as a tuning error on the entangling gate between qubits two and three of the Santiago machine. This error was later fixed, and the results shown were obtained from runs with a fully functioning machine. To further validate the results, a series of runs were performed on the IBM Q Valencia 5-qubit machine [59], which has a Quantum Volume of 16. The results confirmed that the Santiago machine was working correctly.

APPENDIX E: DETAILED QUANTUM CIRCUIT FOR COLLINEAR PARTON SHOWER ALGORITHM

The algorithm presented here follows a similar method to that outlined in [26]. In contrast, the algorithm does not

introduce flavor mixing, but does simulate a vector boson with the possibility of boson splittings. As a result, the algorithm presented here includes tailored history and update gates to deal with the increased splitting channels. Shown in Fig. 8, the circuit is comprised of four tailored gate operations: the count, emission, history, and update gates. The particle identity is encoded using a 3-qubit base, and the following qubit combinations have been chosen for each type of particle:

$$p \begin{cases} \text{gluon} & \begin{array}{|c|cc} \hline & \text{quark} & \text{antiquark} \\ \hline p_0 & 1 & 0 & 0 \\ p_1 & 0 & 0 & 1 \\ p_2 & 0 & 1 & 1 \\ \hline \end{array} \end{cases} \quad (\text{E1})$$

Using a 3-qubit base, it is possible to simulate seven different types of particle and one null state. Therefore, the algorithm could be easily extended to accommodate more quark flavors if more qubits were available.

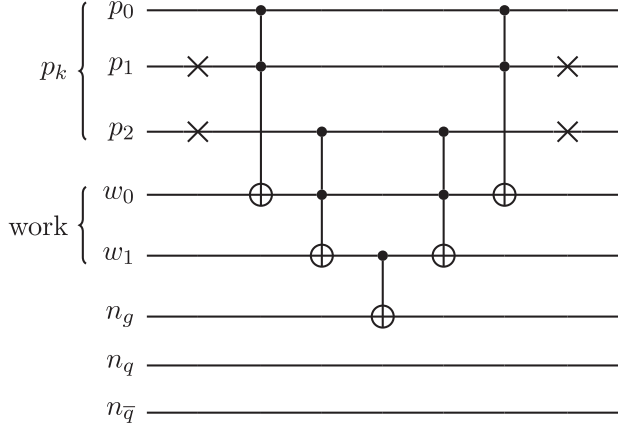


FIG. 12. Count gate circuit decomposition for counting a gluon in the particle register. To complete the count gate, this is repeated for all other possible particle types by applying different combinations of NOT gates.

1. Count gate

The count gate is comprised of three individual counting mechanisms for each type of particle, and is applied to each particle register individually. The algorithm utilizes a series of NOT, *controlled*-NOT (CNOT), and Toffoli (CCNOT) gates to update the count registers, n_i , depending on the type of particle represented in the particle register. Figure 12 shows the counting mechanism for a gluon, controlling only from the gluon state outlined in Appendix (E1).

The total number of count registers, n_i , used in the algorithm is four. As the particle count registers are updated at the beginning of a step, the maximum number of gluons that can be present is two, and the maximum number of quarks/antiquarks is one. Therefore, for this algorithm, only two gluon count registers and one quark/antiquark count register are required. Ideally, one would have the same

number of count registers for each particle type, which would be equal to the step number. However, due to the limited number of available qubits, this has not been possible here.

2. Emission gate

The emission gate implements the Sudakov factors from Eq. (3.5) by defining a U_3 rotation that can be applied to the emission register, e . The structure of this rotation takes the same form as that presented in [26],

$$U_e = \begin{pmatrix} \sqrt{\Delta_{\text{tot}}(z_1, z_2)} & -\sqrt{1 - \Delta_{\text{tot}}(z_1, z_2)} \\ \sqrt{1 - \Delta_{\text{tot}}(z_1, z_2)} & \sqrt{\Delta_{\text{tot}}(z_1, z_2)} \end{pmatrix}, \quad (\text{E2})$$

where Δ_{tot} is given in Eq. (3.5). This rotation changes the state of the emission gate, e , to $|1\rangle$ if there is an emission, and keeps it in state $|0\rangle$ if there is no emission. Nonemission probabilities (Sudakov factors) are used due to the Qiskit [40] definition of a qubit state,

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (\text{E3})$$

Similarly to the Count gate, the Emission gate is constructed from a series of NOT gates which determine the target state, and a series of CCNOT gates which implement the operation if the target state is present. The circuit diagram for the Emission gate is shown in Fig. 13. Here, the emission is determined by controlling from the particle count gates. If the desired particles are present, then the emission rotation from Eq. (E2) is applied to the emission register. As only one emission can occur in a single step, then only one emission qubit is needed per step.

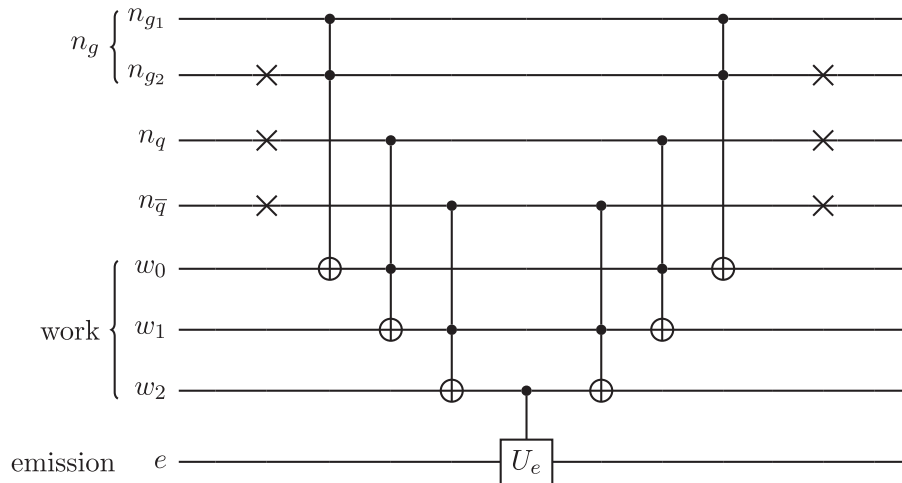


FIG. 13. Emission gate for a single gluon in the first particle register. Here, the U_e is a U_3 rotation is used to implement the Sudakov factors.

3. History gate

The history gate is the most complicated implementation in the algorithm. This is largely due to the fact that a gluon can split to either a gluon pair, or a quark-antiquark pair. As a consequence this requires two calculations of splitting probabilities for a gluon, as outlined in Eq. (3.7). These probabilities are implemented by controlling from present particles and applying a rotation to the relevant history register; again taking a form similar to the one presented in [26],

$$U_h = \begin{pmatrix} \sqrt{1 - \frac{P_{k \rightarrow ij}(z)}{P_{\text{tot}}(z)}} & -\sqrt{\frac{P_{k \rightarrow ij}(z)}{P_{\text{tot}}(z)}} \\ \sqrt{\frac{P_{k \rightarrow ij}(z)}{P_{\text{tot}}(z)}} & \sqrt{1 - \frac{P_{k \rightarrow ij}(z)}{P_{\text{tot}}(z)}} \end{pmatrix}, \quad (\text{E4})$$

where P_{tot} is defined as

$$P_{\text{tot}}(z) = n_g(P_{g \rightarrow q\bar{q}} + P_{g \rightarrow gg}) + n_q P_{q \rightarrow qg} + n_{\bar{q}} P_{\bar{q} \rightarrow \bar{q}g}. \quad (\text{E5})$$

Here the nonsplitting probabilities are used in the diagonal elements due to the definition of the qubit states outlined in Eq. (E3).

The history gate used in this algorithm differs from [26], such that it controls from the particle registers and not the count registers. This is to reduce the number of count registers needed in the algorithm. For this algorithm, the history rotation needs to know which particle is being considered and which particle register it is in so that the correct rotation can be applied to the correct history qubit. This could be done by increasing the count registers by one qubit per particle type every step, to have a count register

for each possible particle in a specific particle register. However, this need for more counting qubits can be reduced by simply controlling from the particle registers themselves; this is shown for a gluon in Fig. 14. This can be done without impacting the rest of the algorithm, as long as there are enough count qubits to count the number of present particles correctly. This is because the emission gate does not need to know what specific particle is present in which particle register, just how many particles are present.

Once the particle content of the simulation has been assessed, the history rotations, U_h , from Eq. (E4) are applied to the relevant history registers. The first, labeled g_1 , is for the $g \rightarrow q\bar{q}$ splitting, and the second, labeled g_2 , is for the $g \rightarrow gg$ splitting. Note that both of these rotations could result in a splitting, and thus rotate the history qubit to the $|1\rangle$ state. Therefore, they are applied to different history registers. In general, one could have a condition on the second rotation, such that it is not applied if the first rotation yields a $|1\rangle$, but in this algorithm, this condition was carried forward to the update gate (see Sec. E4). As a result of these different splittings, the required number of qubits needed for the history register in each step is $3N$, where N is the step number. As a result, the qubits needed for the history register for the whole algorithm scales as $(3N(N+1)/2)$. Figure 11 shows the history gate for the first step, thus three qubits are needed for the history register: two for the gluon splittings, and one for the quark/antiquark splittings.

4. Update gate

The final gate in the algorithm is the update gate, which, if an emission has occurred, changes the particle content of

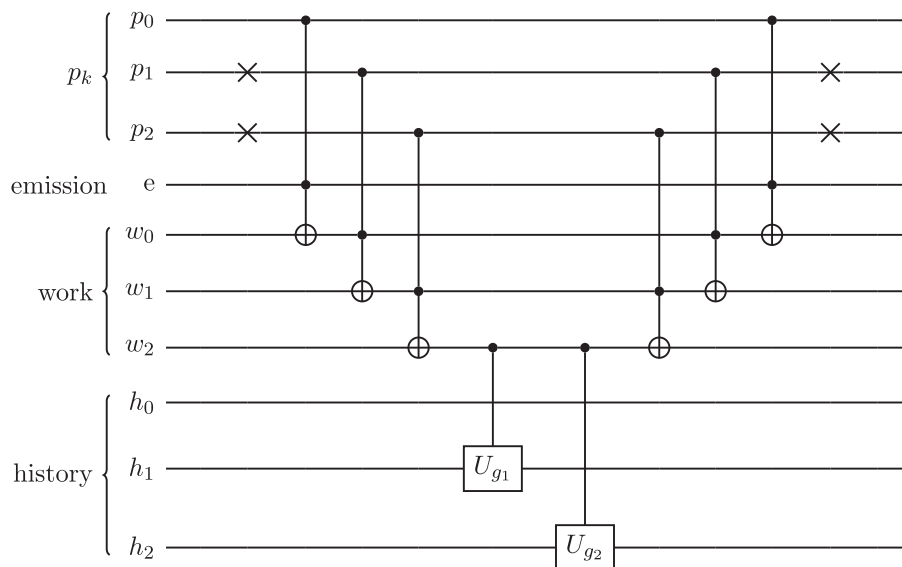


FIG. 14. History gate for a single gluon in the first step. Here the U_h gate is a U_3 rotation used to implement the splitting probabilities.

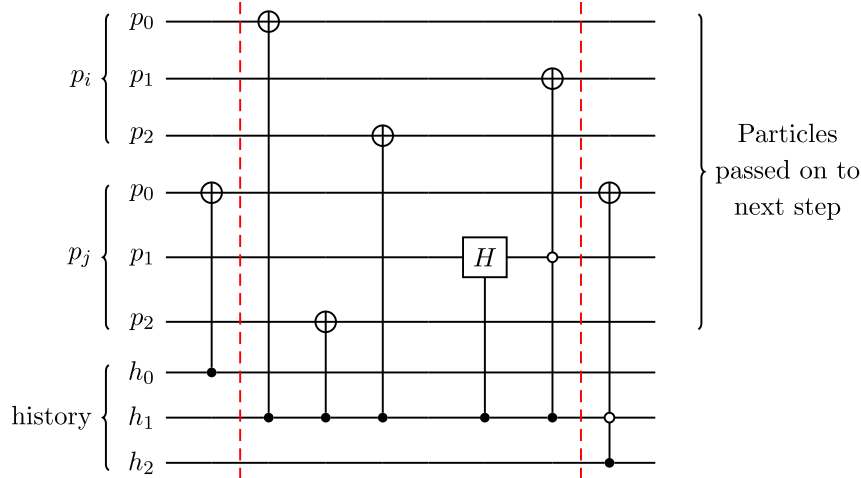


FIG. 15. Update gate for the first step of the algorithm. Each slice is a different update mechanism: the far left slice updates $q \rightarrow q\bar{q}$ splittings, the center slice updates $g \rightarrow q\bar{q}$ splittings, and the far right slice updates $g \rightarrow gg$ splittings.

the particle registers accordingly. Figure 15 shows the update gate from the first step, which is the simplest update gate in the algorithm. The circuit shown is sliced into individual updates. The first slice on the left shows the addition of a new gluon to the particle register. This is controlled from the quark/antiquark history gate, and so corresponds to the $\binom{-}{q} \rightarrow \binom{-}{q}g$ process.

The middle slice in Fig. 15 shows the update of a gluon splitting to a quark-antiquark pair controlled from the $g \rightarrow q\bar{q}$ history register. The first three CNOT gates of this slice put the particle registers into a state of two quarks. The update gate then utilizes a *controlled-Hadamard* gate, putting the p_{j_i} qubit in a superposition of $|0\rangle$ and $|1\rangle$ states. The final gate of the slice controls from the history register, but also controls on a $|0\rangle$ state on the p_{j_i} . At the point of measurement, if p_{j_i} is measured as a $|0\rangle$ state, then the p_k register represents an antiquark and p_j represents a quark. If p_{j_i} is measured in the $|1\rangle$ state, then the p_k register represents a quark and the p_j register represents an antiquark. In a sense, this *controlled-Hadamard* gate and subsequent CCNOT gate

put the particle registers p_j and p_k into a superposition of quark-antiquark and antiquark-quark states.

The final slice on the right of the circuit diagram in Fig. 15 shows the update gate corresponding to the $g \rightarrow gg$ process. This simple update changes the p_{j_0} qubit to a $|1\rangle$ state controlled from the history register, like the quark/antiquark update gate. However, this is where the algorithm adds a preference to $g \rightarrow q\bar{q}$ process over the $g \rightarrow gg$ process. The CCNOT gate for the final slice in Fig. 15 also controls from a $|0\rangle$ state on the $g \rightarrow q\bar{q}$ history qubit. Therefore a gluon can only split to a gluon pair if the history gate for a gluon splitting to a quark-antiquark pair is in the $|0\rangle$ state. This is an acceptable approximation because the splitting probabilities for $g \rightarrow q\bar{q}$ are a lot less than for $g \rightarrow gg$. Consequently, there is only a small probability that they are both in the $|1\rangle$ state at any one time. However, it is possible that this may be a limitation in comparison to current classical parton shower algorithms provided by packages such as PYTHIA [1], HERWIG [2], and SHERPA [3], as these give more complex weightings to the different splitting channels.

- [1] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, Christine O. Rasmussen, and P. Z. Skands, An introduction to PYTHIA 8.2, *Comput. Phys. Commun.* **191**, 159 (2015).
- [2] M. Bahr *et al.*, Herwig++ physics and manual, *Eur. Phys. J. C* **58**, 639 (2008).
- [3] T. Gleisberg, S. Hoeche, F. Krauss, A. Schalicke, S. Schumann, and J.-C. Winter, SHERPA 1. alpha: A Proof of concept version, *J. High Energy Phys.* **02** (2004) 056.

- [4] J. C. Collins, D. E. Soper, and G. F. Sterman, Factorization of Hard Processes in QCD, *Adv. Ser. Dir. High Energy Phys.* **5**, 1–91 (1989).
- [5] S. J. Parke and T. Taylor, An Amplitude for n Gluon Scattering, *Phys. Rev. Lett.* **56**, 2459 (1986).
- [6] F. A. Berends and W. Giele, Recursive calculations for processes with n gluons, *Nucl. Phys.* **B306**, 759 (1988).
- [7] V. V. Sudakov, Vertex parts at very high-energies in quantum electrodynamics, *Sov. Phys. JETP* **3**, 65 (1956), http://www.jetp.ac.ru/cgi-bin/dn/e_003_01_0065.pdf.

- [8] A. Buckley *et al.*, General-purpose event generators for LHC physics, *Phys. Rep.* **504**, 145 (2011).
- [9] S. Catani, F. Krauss, R. Kuhn, and B. Webber, QCD matrix elements + parton showers, *J. High Energy Phys.* **11** (2001) 063.
- [10] 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, *Nucl. Phys.* **B632**, 343 (2002).
- [11] L. Lonnblad, Correcting the color dipole cascade model with fixed order matrix elements, *J. High Energy Phys.* **05** (2002) 046.
- [12] S. Hche and S. Prestel, The midpoint between dipole and parton showers, *Eur. Phys. J. C* **75**, 461 (2015).
- [13] N. Fischer and S. Prestel, Combining states without scale hierarchies with ordered parton showers, *Eur. Phys. J. C* **77**, 601 (2017).
- [14] N. Fischer, A. Lifson, and P. Skands, Helicity antenna showers for hadron colliders, *Eur. Phys. J. C* **77**, 719 (2017).
- [15] S. Prestel and M. Spannowsky, HYTREES: Combining matrix elements and parton shower for hypothesis testing, *Eur. Phys. J. C* **79**, 546 (2019).
- [16] D. E. Soper and M. Spannowsky, Finding physics signals with event deconstruction, *Phys. Rev. D* **89**, 094005 (2014).
- [17] P. Richardson and S. Webster, Spin correlations in parton shower simulations, *Eur. Phys. J. C* **80**, 83 (2020).
- [18] S. Abel, N. Chancellor, and M. Spannowsky, Quantum computing for quantum tunnelling, *Phys. Rev. D* **103**, 016008 (2021).
- [19] S. Abel and M. Spannowsky, Observing the fate of the false vacuum with a quantum laboratory, [arXiv:2006.06003](https://arxiv.org/abs/2006.06003).
- [20] A. Mott, J. Job, J.-R. Vlimant, D. Lidar, and M. Spiropulu, Solving a higgs optimization problem with quantum annealing for machine learning, *Nature (London)* **550**, 375 (2017).
- [21] S. P. Jordan, K. S. Lee, and J. Preskill, Quantum computation of scattering in scalar quantum field theories, *Quantum Inf. Comput.* **14**, 1014 (2014).
- [22] L. García-Álvarez, J. Casanova, A. Mezzacapo, I. Egusquiza, L. Lamata, G. Romero, and E. Solano, Fermion-Fermion Scattering in Quantum Field Theory with Superconducting Circuits, *Phys. Rev. Lett.* **114**, 070502 (2015).
- [23] S. P. Jordan, K. S. M. Lee, and J. Preskill, Quantum algorithms for fermionic quantum field theories, [arXiv:1404.7115](https://arxiv.org/abs/1404.7115).
- [24] S. P. Jordan, H. Krovi, K. S. M. Lee, and J. Preskill, BQP-completeness of scattering in scalar quantum field theory, *Quantum* **2**, 44 (2018).
- [25] J. Preskill, Simulating quantum field theory with a quantum computer, *Proc. Sci.*, LATTICE2018 (2018) 024 [[arXiv:1811.10085](https://arxiv.org/abs/1811.10085)].
- [26] C. W. Bauer, W. A. De Jong, B. Nachman, and D. Provasoli, A Quantum Algorithm for High Energy Physics Simulations, *Phys. Rev. Lett.* **126**, 062001 (2021).
- [27] A. H. Moosavian, J. R. Garrison, and S. P. Jordan, Site-by-site quantum state preparation algorithm for preparing vacua of fermionic lattice field theories, [arXiv:1911.03505](https://arxiv.org/abs/1911.03505).
- [28] A. Alexandru, P. F. Bedaque, H. Lamm, and S. Lawrence (NuQS Collaboration), σ Models on Quantum Computers, *Phys. Rev. Lett.* **123**, 090501 (2019).
- [29] A. Alexandru, P. F. Bedaque, S. Harmalkar, H. Lamm, S. Lawrence, and N. C. Warrington (NuQS Collaboration), Gluon field digitization for quantum computers, *Phys. Rev. D* **100**, 114501 (2019).
- [30] H. Lamm, S. Lawrence, and Y. Yamauchi (NuQS Collaboration), Parton physics on a quantum computer, *Phys. Rev. Research* **2**, 013272 (2020).
- [31] H. Lamm, S. Lawrence, and Y. Yamauchi (NuQS Collaboration), Suppressing coherent gauge drift in quantum simulations, [arXiv:2005.12688](https://arxiv.org/abs/2005.12688).
- [32] I. Márquez-Mártin, P. Arnault, G. D. Molfetta, and A. Pérez, Electromagnetic lattice gauge invariance in two-dimensional discrete-time quantum walks, *Phys. Rev. A* **98**, 032333 (2018).
- [33] P. Arrighi, G. D. Molfetta, I. Márquez-Mártin, and A. Pérez, The Dirac equation as a quantum walk over the honeycomb and triangular lattices, *Phys. Rev. A* **97**, 062111 (2018).
- [34] G. Jay, F. Debbasch, and J. B. Wang, Dirac quantum walks on triangular and honeycomb lattices, *Phys. Rev. A* **99**, 032113 (2019).
- [35] G. D. Molfetta and P. Arrighi, A quantum walk with both a continuous-time limit and a continuous-spacetime limit, *Quantum Inf. Process.* **19**, 47 (2020).
- [36] H. Lamm and S. Lawrence, Simulation of Nonequilibrium Dynamics on a Quantum Computer, *Phys. Rev. Lett.* **121**, 170501 (2018).
- [37] S. Harmalkar, H. Lamm, and S. Lawrence (NuQS Collaboration), Quantum simulation of field theories without state preparation, [arXiv:2001.11490](https://arxiv.org/abs/2001.11490).
- [38] A. Y. Wei, P. Naik, A. W. Harrow, and J. Thaler, Quantum algorithms for jet clustering, *Phys. Rev. D* **101**, 094015 (2020).
- [39] K. T. Matchev, P. Shyamsundar, and J. Smolinsky, A quantum algorithm for model independent searches for new physics, [arXiv:2003.02181](https://arxiv.org/abs/2003.02181).
- [40] IBM Research, Qiskit, an open source computing framework.
- [41] IBM Q Team, IBM Q 32 Simulator v0.1.547.
- [42] H. Elvang and Y.-T. Huang, *Scattering Amplitudes in Gauge Theory and Gravity* (Cambridge University Press, Cambridge, England, 2015).
- [43] IBM Q Team, IBM Q Santiago 5 Qubit Quantum Computer v1.0.3.
- [44] D. Maître and P. Mastrolia, S@M, a Mathematica implementation of the spinor-helicity formalism, *Comput. Phys. Commun.* **179**, 501 (2008).
- [45] F. Cachazo, P. Svrcek, and E. Witten, MHV vertices and tree amplitudes in gauge theory, *J. High Energy Phys.* **09** (2004) 006.
- [46] R. Britto, F. Cachazo, B. Feng, and E. Witten, Direct Proof of Tree-Level Recursion Relation in Yang-Mills Theory, *Phys. Rev. Lett.* **94**, 181602 (2005).
- [47] B. Yirka, Honeywell claims to have built the highest-performing quantum computer available (2020), <https://phys.org/news/2020-06-honeywell-built-highest-performing-quantum.html>.
- [48] P. Jurcevic *et al.*, Demonstration of quantum volume 64 on a superconducting quantum computing system, [arXiv:2008.08571](https://arxiv.org/abs/2008.08571).

- [49] Ibm's roadmap for scaling quantum technology, 2020.
- [50] T.R. Taylor, A course in amplitudes, *Phys. Rep.* **691**, 1 (2017).
- [51] Y.L. Dokshitzer, Calculation of the structure functions for deep inelastic scattering and e^+e^- annihilation by perturbation theory in quantum chromodynamics, *Sov. Phys. JETP* **46**, 641 (1977), http://www.jetp.ac.ru/cgi-bin/dn/e_046_04_0641.pdf.
- [52] V. Gribov and L. Lipatov, Deep inelastic e p scattering in perturbation theory, *Sov. J. Nucl. Phys.* **15**, 438 (1972), <https://inspirehep.net/literature/73449>.
- [53] G. Altarelli and G. Parisi, Asymptotic freedom in parton language, *Nucl. Phys.* **B126**, 298 (1977).
- [54] S. Marzani, G. Soyez, and M. Spannowsky, Looking inside jets, *Lect. Notes Phys.* **19** (2019).
- [55] M. G. Echevarria, I. L. Egusquiza, E. Rico, and G. Schnell, Quantum simulation of light-front parton correlators, [arXiv:2011.01275](https://arxiv.org/abs/2011.01275).
- [56] M. Kreshchuk, W. M. Kirby, G. Goldstein, H. Beauchemin, and P. J. Love, Quantum simulation of quantum field theory in the light-front formulation, [arXiv:2002.04016](https://arxiv.org/abs/2002.04016).
- [57] J. Wentz, I. M. Brancus, A. Bercuci, D. Heck, J. Oehlschlager, H. Rebel, and B. Vulpescu, Simulation of atmospheric muon and neutrino fluxes with CORSIKA, *Phys. Rev. D* **67**, 073020 (2003).
- [58] P. Schichtel, M. Spannowsky, and P. Waite, Constraining strongly coupled new physics from cosmic rays with machine learning techniques, *Europhys. Lett.* **127**, 61002 (2019).
- [59] IBM Q Team, IBM Q Valencia 5 Qubit Quantum Computer v1.3.1.