Quantum simulation of colour
in perturbative quantum chromodynamics

Herschel A. Chawdhry[1][*] and Mathieu Pellen[2][†]

[1] *Rudolf Peierls Centre for Theoretical Physics, University of Oxford,*
Clarendon Laboratory, Parks Road, Oxford, OX1 3PU, United Kingdom
[2] *Albert-Ludwigs-Universität Freiburg, Physikalisches Institut,*
Hermann-Herder-Straße 3, D-79104 Freiburg, Germany

Abstract

Quantum computers are expected to give major speed-ups for the simulation of quantum systems. In this work, we present quantum gates that simulate the colour part of the interactions of quarks and gluons in perturbative quantum chromodynamics (QCD). As a first application, we implement these circuits on a simulated noiseless quantum computer and use them to calculate colour factors for various examples of Feynman diagrams. This work constitutes a first key step towards a quantum simulation of generic scattering processes in perturbative QCD.

Contents

---

[*]E-mail: herschel.chawdhry@physics.ox.ac.uk
[†]E-mail: mathieu.pellen@physik.uni-freiburg.de

1 Introduction

Quantum computing is of widespread interest because it offers exponential or polynomial speedups for a variety of problems ranging from prime factorisation [1] to unstructured searching [2]. A natural use of quantum computers is the simulation of other quantum systems, with well-known applications in computational chemistry [3, 4] and condensed matter physics [5, 6].

Recent years have seen proposed applications of quantum computers to lattice-based Quantum Field Theory (QFT) simulations (see Refs. [7, 8] and references therein) including the simulation of Quantum Chromodynamics (QCD), the theory describing the fundamental interactions of quarks and gluons. Lattice QCD is well-suited to studying the lower-energy (sub-GeV) behaviour of QCD, but the rapid increase in computational cost with lattice size makes lattice QCD exceedingly challenging to use for simulating collisions at the shortest length scales probed in high-energy colliders such as the Large Hadron Collider (LHC). At these energies, the QCD coupling constant α_s becomes small, and so perturbative calculations become the method of choice.

The use of quantum computers to simulate hard scattering processes in perturbative QCD has largely remained unexplored to date. This may be in part because the aims of perturbative QFT calculations differ from the aims of most quantum simulations: most quantum simulations (including lattice QCD) aim to take a known Hamiltonian and use it to perform the (unitary) evolution of a quantum system, whereas perturbative QFT calculations aim to calculate the (Hermitian, but not unitary) transition matrix describing the scattering of specified external states and hence study the production or decay of elementary particles.

A method to simulate generic perturbative QCD processes on a quantum computer is still missing but is desirable for several reasons. Firstly, perturbative QCD calculations require the evaluation—and quantum coherent combination—of contributions from many unobservable intermediate states, which makes such calculations natural candidates to benefit from the ability of quantum computers to manipulate superpositions of quantum states. Secondly, this ability also means quantum simulations could be well-suited to performing calculations with full interference effects for processes with high-multiplicity final states. Thirdly, a quantum simulation of generic perturbative QCD processes could improve the speed and precision of perturbative QCD predictions by exploiting speedups provided by known quantum algorithms such as quantum amplitude estimation [9–12].

The object of this article is to take steps towards using quantum computers to simulate generic perturbative QCD processes. Calculations in perturbative QCD can be performed by summing contributions from Feynman diagrams. Each contribution can be factorised into a colour part and a kinematic part. The colour part is simpler to compute than the kinematic part, and indeed there exist efficient programs [13–18] for calculating colour factors on a classical computer. Nonetheless, the colour part still presents some of the generic challenges of simulating perturbative QCD processes on a quantum computer. For example, the quantum gates that form a quantum computer must always be unitary whereas the Feynman rules (colour and kinematic parts alike) describing components of a Feynman diagram are not generally unitary. This means the colour parts provide a useful simplified setup with which to begin developing a framework for the quantum computation of Feynman diagrams, and they will therefore be the focus of the present work.

The main results in this article are two quantum gates, Q and G, which represent the colour part of the Feynman rules describing the quark-gluon and the triple-gluon interaction vertices, respectively. To implement these gates, we introduce the new concept of a *unitarisation register*

$\mathcal{U}$, which enables the simulation of the non-unitary interactions of quarks and gluons.[1] As an example of the use of Q and G, we use the QISKIT [24] quantum-computing framework to build quantum circuits that calculate the colour factors of various Feynman diagrams. This is by no means the only use case for Q and G; they could also be used, for example, to simulate emissions, absorption, and exchanges of quarks and gluons, either in a scattering amplitude or in a parton shower. These quantum circuits could also form a component of a quantum Monte Carlo program which, as highlighted in Ref. [25], would offer a quadratic speedup for the calculation of cross sections.

Finally, let us mention that this work forms part of a broader exploration of quantum-technology applications in high-energy physics (see *e.g.* existing reviews [7, 8, 26, 27] and references therein). While most of that exploration has focused on the experimental side of high-energy physics, the last few years have also seen the emergence of applications on various topics in high-energy theory. These range from parton distribution functions (PDFs) [28, 29] to amplitudes [30–34], effective field theory [35], cross-section computations [25], parton showers [30, 36–38], and event generation [38–40].

This article is organised as follows: sec. 2 begins with a high-level overview of the use of our quantum circuits to calculate the colour factor for a simple Feynman diagram (sec. 2.1). The rest of sec. 2 provides details of the methods and algorithms employed. In particular, sec. 2.2 explains the functioning of the unitarisation register mentioned above, and sec. 2.3 presents the quantum circuits implementing quark and gluon interactions. In sec. 3, we generalise our methods to simulate more complicated processes and validate this by using a simulated noiseless quantum computer to calculate colour factors for various Feynman diagrams. Finally, sec. 4 contains a summary of our findings and concluding remarks. In Appendix A, a few miscellaneous quantum gates related to the calculation of traces are presented.

2 Methods

2.1 Illustrative example

The main results in this article are two quantum gates, Q and G, which simulate, respectively, the colour factors T_{ij}^a of the quark-gluon vertex and f^{abc} of the triple-gluon vertex. We defer a description of the explicit construction of Q and G to sec. 2.3. As we will see in sec. 3, these gates can be used to calculate the colour factor of any Feynman diagram. In the present section, we will illustrate how our method works by applying it to calculate the colour factor of the simple example Feynman diagram shown on the left-hand side of fig. 1.

In general, any Feynman diagrams involving quark-gluon or gluon-gluon interactions will carry colour information from the SU(3) symmetry group of QCD. When squaring the diagrams to obtain the cross section, the colour algebra has to be carried out, resulting in so-called colour factors. The latter contain a generator T^a for each quark-gluon vertex and a structure constant

---

[1]While the use of ancillary qubits to enforce unitarity is by now well established with methods such as *block encoding* [19–22] or *qubitisation* [23], to the best of our knowledge our implementation is original and has the advantage of allowing multiple independent non-unitary operations to be carried out sequentially while only requiring a small number of ancilla qubits, as will be explained in sec. 2.2.
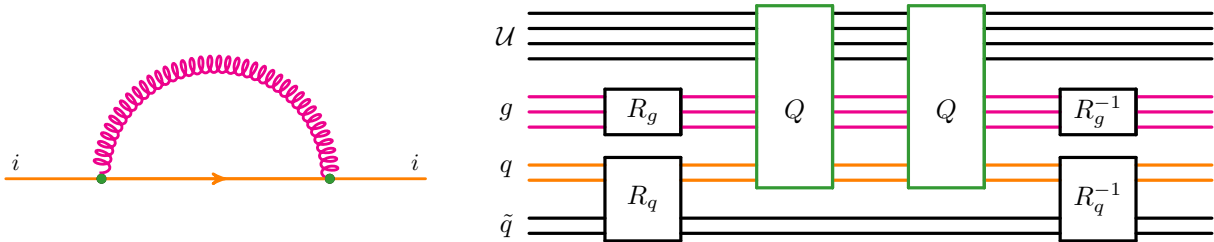
Figure 1: Example Feynman diagram (left) and a graphical representation of its corresponding circuit (right).

f^{abc} for each triple-gluon vertex.[2] In the present example, the colour factor reads

$$\mathcal{C} = \sum_{\substack{a\in\{1,...,8\}\\i,j,k\in\{1,2,3\}}} T^a_{ij}T^a_{jk}\delta_{ik}, \tag{1}$$

where

$$T^a = \frac{1}{2}\lambda^a \tag{2}$$

and λ^a are the Gell-Mann matrices:

$$\lambda^1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \lambda^2 = \begin{pmatrix} 0 & -i & 0 \\ i & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \lambda^3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$\lambda^4 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \lambda^5 = \begin{pmatrix} 0 & 0 & -i \\ 0 & 0 & 0 \\ i & 0 & 0 \end{pmatrix}, \quad \lambda^6 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \tag{3}$$

$$\lambda^7 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -i \\ 0 & i & 0 \end{pmatrix}, \quad \lambda^8 = \frac{1}{\sqrt{3}}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{pmatrix}.$$

This colour factor can be computed using the quantum circuit shown on the right-hand side of fig. 1, whose workings will now be explained. The circuit contains several qubits, which we combine into groups called registers. Each register r is initially in some reference state $|\Omega\rangle_r$.[3] In this particular example, there is a gluon register, labelled g, and a pair of quark registers, labelled q and $\tilde{q}$. In general there will be a gluon register for each gluon in a Feynman diagram, and a pair of quark registers for each quark line in the diagram. We will see that the state of the q register is altered by the simulated emission and absorption of gluons, while the $\tilde{q}$ register is left unaffected and serves only to help implement the δ_{ik} term in eq. (1). Each quark register is made of 2 qubits, with the states $|00\rangle, |01\rangle, |10\rangle$ representing the $N_c = 3$ quark colours $|1\rangle, |2\rangle, |3\rangle$, while the state $|11\rangle$ is unused. The gluon register is composed of 3 qubits, whose 8 states $|000\rangle, |001\rangle, \ldots, |111\rangle$ represent the $N_c^2 - 1 = 8$ colours $|1\rangle, |2\rangle, \ldots, |8\rangle$ of the gluon. There is also a unitarisation register, labelled $\mathcal{U}$, whose purpose will be explained in section 2.2. The initial state of the circuit is thus $|\Omega\rangle_g |\Omega\rangle_q |\Omega\rangle_{\tilde{q}} |\Omega\rangle_{\mathcal{U}}$.

---

[2]More references and information on colour algebra can be found in Refs. [41, 42].
[3]In practice, in this work we always choose $|\Omega\rangle_r$ to be the state where each qubit of r is in the state $|0\rangle$.

First, a gate R_g is applied to the gluon register to put it into an equal superposition of colour states. A detailed definition of R_g will be given in Appendix A and its effect reads as follows:

$$R_g \left|\Omega\right\rangle_g = \sum_{a=1}^{8} \frac{1}{\sqrt{8}} \left|a\right\rangle_g .$$ (4)

The gate R_q (also to be defined in Appendix A) is now applied to the quark registers with the following effect:

$$R_q \left|\Omega\right\rangle_q \left|\Omega\right\rangle_{\tilde{q}} = \sum_{k=1}^{3} \frac{1}{\sqrt{3}} \left|k\right\rangle_q \left|k\right\rangle_{\tilde{q}} .$$ (5)

Thus, after applying the R_g and R_q gates, the quantum computer is in the state

$$\frac{1}{\sqrt{24}} \sum_{a=1}^{8} \sum_{k=1}^{3} \left|a\right\rangle_g \left|k\right\rangle_q \left|k\right\rangle_{\tilde{q}} \left|\Omega\right\rangle_{\mathcal{U}} .$$ (6)

Next, the Q gate is applied to the g and q registers. Q represents the quark-gluon interaction and is designed (see sec. 2.3.1) such that for a gluon colour basis state $\left|a\right\rangle_g$ and quark colour basis state $\left|k\right\rangle_q$, where $a \in \{1, \ldots, 8\}$ and $k \in \{1, 2, 3\}$, the following equation holds:

$$Q \left|a\right\rangle_g \left|k\right\rangle_q \left|\Omega\right\rangle_{\mathcal{U}} = \sum_{j=1}^{3} T_{jk}^a \left|a\right\rangle_g \left|j\right\rangle_q \left|\Omega\right\rangle_{\mathcal{U}} + (\text{terms orthogonal to } \left|\Omega\right\rangle_{\mathcal{U}}) .$$ (7)

Note that Q does not act on the $\tilde{q}$ register. The linearity of quantum gates ensures that Q models the quark-gluon interaction correctly even if the quark or gluon registers are in superpositions of colour basis states or are entangled with other registers. Thus, after applying the Q gate once, the quantum computer is in the state

$$\frac{1}{\sqrt{24}} \sum_{\substack{a \in \{1, \ldots, 8\} \\ j,k \in \{1,2,3\}}} T_{jk}^a \left|a\right\rangle_g \left|j\right\rangle_q \left|k\right\rangle_{\tilde{q}} \left|\Omega\right\rangle_{\mathcal{U}} + (\text{terms orthogonal to } \left|\Omega\right\rangle_{\mathcal{U}}) .$$ (8)

The Q gate is now applied a second time to the g and q registers, to simulate the second quark-gluon vertex. This puts the quantum computer into the state

$$\frac{1}{\sqrt{24}} \sum_{\substack{a \in \{1, \ldots, 8\} \\ i,j,k \in \{1,2,3\}}} T_{ij}^a T_{jk}^a \left|a\right\rangle_g \left|i\right\rangle_q \left|k\right\rangle_{\tilde{q}} \left|\Omega\right\rangle_{\mathcal{U}} + (\text{terms orthogonal to } \left|\Omega\right\rangle_{\mathcal{U}}) .$$ (9)

Since R_g is unitary, one can see by inverting eq. (4) that R_g^{-1} acting on any state $\sum_{a=1}^{8} c_a \left|a\right\rangle_g$ of the gluon register would produce the state

$$R_g^{-1} \sum_{a=1}^{8} c_a \left|a\right\rangle_g = \left(\frac{1}{\sqrt{8}} \sum_{a=1}^{8} c_a \right) \left|\Omega\right\rangle_g + \left(\text{terms orthogonal to } \left|\Omega\right\rangle_g \right) .$$ (10)

Similarly, it can be seen by inverting eq. (5) that R_q^{-1} acting on any state $\sum_{i,k \in \{1,2,3\}} c_{ik} \left|i\right\rangle_q \left|k\right\rangle_{\tilde{q}}$ of the q and $\tilde{q}$ registers would produce the state

$$R_q^{-1} \sum_{i,k \in \{1,2,3\}} c_{ik} \left|i\right\rangle_q \left|k\right\rangle_{\tilde{q}} = \left(\frac{1}{\sqrt{3}} \sum_{i=1}^{3} c_{ii} \right) \left|\Omega\right\rangle_q \left|\Omega\right\rangle_{\tilde{q}} + \left(\text{terms orthogonal to } \left|\Omega\right\rangle_q \left|\Omega\right\rangle_{\tilde{q}} \right) .$$ (11)

Therefore, taking the state produced in eq. (9) and applying the gates R_g^{-1} and R_q^{-1}, the state of the quantum circuit becomes

$$\frac{1}{24} \left(\sum_{\substack{a \in \{1,...,8\} \\ i,j \in \{1,2,3\}}} T_{ij}^a T_{ji}^a \right) |\Omega\rangle_g |\Omega\rangle_q |\Omega\rangle_{\tilde{q}} |\Omega\rangle_{\mathcal{U}} + \left(\text{terms orthogonal to } |\Omega\rangle_g |\Omega\rangle_q |\Omega\rangle_{\tilde{q}} |\Omega\rangle_{\mathcal{U}} \right). \quad (12)$$

Thus, in the final quantum state (12) of the quantum circuit, the colour trace (1) to be computed is found encoded in the coefficient of the reference state $|\Omega\rangle_g |\Omega\rangle_q |\Omega\rangle_{\tilde{q}} |\Omega\rangle_{\mathcal{U}}$. As we will now discuss, this circuitry can either be used as part of a higher-level algorithm, or the information in the output state (12) can be extracted directly.

A simple way to verify this result is to perform many independent runs of the circuit, where after each run the final state of all the registers is measured and then reset to the initial state $|\Omega\rangle_g |\Omega\rangle_q |\Omega\rangle_{\tilde{q}} |\Omega\rangle_{\mathcal{U}}$. One can then count the number of runs where the final state is measured to be $|\Omega\rangle_g |\Omega\rangle_q |\Omega\rangle_{\tilde{q}} |\Omega\rangle_{\mathcal{U}}$, and compare against the prediction of eq. (12) which is that for each run the measurement of the final state will yield $|\Omega\rangle_g |\Omega\rangle_q |\Omega\rangle_{\tilde{q}} |\Omega\rangle_{\mathcal{U}}$ with probability $\frac{\mathcal{C}^2}{24^2}$, where $\mathcal{C}$ was defined in eq. (1).[4] We note that this measurement strategy, while relatively simple to understand, is only being presented as a transparent way to verify that the state (12) is correctly produced. More advanced methods exist for examining, measuring, and exploiting quantum states. For example, Quantum Amplitude Estimation [9–12] could be employed in order to achieve a quadratic improvement in speed. Alternatively, to go beyond calculating the colour factor of a single Feynman diagram, the output (12) of the circuit can be directly used as a component of a future algorithm, such as one that calculates the kinematic factors and then sums over Feynman diagrams, or one that performs Monte Carlo integration to compute cross-sections.

Although not required for this simple example Feynman diagram, let us mention that in addition to the Q gate used above, we have also designed a gate G which represents triple-gluon interactions. The detailed definition of G will be given in sec. 2.3.2, but for now we will simply note that, similarly to eq. (7), G has been designed to act on 3 gluon registers (g_1, g_2, and g_3) and $\mathcal{U}$ such that

$$G |a\rangle_{g_1} |b\rangle_{g_2} |c\rangle_{g_3} |\Omega\rangle_{\mathcal{U}} = f^{abc} |a\rangle_{g_1} |b\rangle_{g_2} |c\rangle_{g_3} |\Omega\rangle_{\mathcal{U}} + (\text{terms orthogonal to } |\Omega\rangle_{\mathcal{U}}), \quad (13)$$

where f^{abc} are the SU(3) structure constants mentioned above. Note that to avoid artificially distinguishing between "emitted" and "emitter" gluons, we have a separate register for each of the 3 gluons at a triple-gluon vertex and so the G gate in eq. (13) only rescales the amplitude (projected onto $|\Omega\rangle_{\mathcal{U}}$) by f^{abc}, without rotating the gluon colour states. In contrast, the q register represents an entire quark line, whose state (projected onto $|\Omega\rangle_{\mathcal{U}}$) is rotated by the Q gate at each interaction. Note also that we do not construct a specific gate for the four-gluon vertex since that vertex can decomposed into a linear combination of products of three-gluon vertices, each product having an independent kinematic coefficient.

This concludes our example computation of a colour trace. In sec. 3, we will generalise this to calculate the colour traces of more complicated processes. Before that, however, we will describe the details that we have so far deferred: in sec. 2.2 we will describe the purpose and functioning of the unitarisation register $\mathcal{U}$, and in sec. 2.3 we will present the explicit construction of the Q and G gates.

---

[4]Such a verification will not yield information about the complex phase of the trace, but that can be obtained by instead implementing the well-known Hadamard test [43].

2.2 Unitarisation register

To simulate perturbative QCD processes, we would like to construct quantum gates for the 8 linear operators

$$|j\rangle_q \to \sum_i T_{ij}^a |i\rangle_q \tag{14}$$

and the (diagonal) linear operator

$$|a\rangle_{g_1} |b\rangle_{g_2} |c\rangle_{g_3} \to f^{abc} |a\rangle_{g_1} |b\rangle_{g_2} |c\rangle_{g_3}, \tag{15}$$

where q is a quark register and g_1, g_2, g_3 are gluon registers. However, quantum gates can only be constructed for unitary operators. A linear operator is unitary if and only if the rows of its matrix representation are orthonormal. The matrix representations of eqs. (14) and (15) consist of rows which are mutually orthogonal but not necessarily of unit norm[5]; indeed, many rows are zero. In this section we will present a way to circumvent this problem.

Let L be a linear operator acting on a Hilbert space $\mathcal{H}_1$. If L is non-unitary, it cannot be implemented as a quantum circuit. The Gell-Mann matrices in eq. (3) are examples of such non-unitary operations. However, it may still be possible to define a unitary operator $\hat{L}$ acting on a larger Hilbert space $\mathcal{H}_1 \otimes \mathcal{H}_\mathcal{U}$ such that for some state $|\Omega\rangle_\mathcal{U} \in \mathcal{H}_\mathcal{U}$ and for any states $|\chi_1\rangle, |\chi_2\rangle \in \mathcal{H}_1$ the following equation holds:

$$\langle\Omega|_\mathcal{U} \langle\chi_2|\hat{L}|\chi_1\rangle |\Omega\rangle_\mathcal{U} = \langle\chi_2|L|\chi_1\rangle. \tag{16}$$

Clearly, there are many ways to achieve eq. (16), each with different advantages. In this work, we have sought a way that firstly allows multiple independent non-unitary operations to be performed sequentially, secondly keeps the size of $\mathcal{H}_\mathcal{U}$ small, and thirdly maintains quantum coherence without intermediate measurements so that these circuits can be used as building blocks in a higher-level algorithm. Specifically, regardless of the complexity of the Feynman diagram, we introduce a single additional register $\mathcal{U}$, whose size is small: it contains $N_\mathcal{U} = \lceil\log_2(N_V + 1)\rceil$ qubits, where N_V is the number of vertices in the Feynman diagram. More generally, N_V would be the number of non-unitary operations to be performed. We call $\mathcal{U}$ the *unitarisation register*, and denote its $2^{N_\mathcal{U}}$ basis states $|k\rangle_\mathcal{U}$ with $k \in \{0, \ldots, 2^{N_\mathcal{U}} - 1\}$. Later in this section we will define two gates A and $B(\alpha)$, where $\alpha \in \mathbb{C}$ and $|\alpha|^2 \leq 1$, which are designed to act on $\mathcal{U}$ in the following way:

$$B(\alpha)A|k\rangle = \begin{cases} \alpha|0\rangle + \sqrt{1-|\alpha|^2}|1\rangle & \text{if } k = 0 \\ |k+1\rangle & \text{if } 0 < k < 2^{N_\mathcal{U}} - 1 \\ \sqrt{1-|\alpha|^2}|0\rangle - \alpha|1\rangle & \text{if } k = 2^{N_\mathcal{U}} - 1. \end{cases} \tag{17}$$

The state $|0\rangle_\mathcal{U}$ is special and we interchangeably denote it as $|\Omega\rangle_\mathcal{U}$. Equation (17) implies two key properties: firstly,

$$\langle\Omega|_\mathcal{U} B(\alpha)A|\Omega\rangle_\mathcal{U} = \alpha \tag{18}$$

and secondly, we can apply the A and B gates[6] repeatedly up to $2^{N_\mathcal{U}} - 1$ times and obtain

$$\langle\Omega|_\mathcal{U} \prod_{i=1}^{N_{ops}} \{B(\alpha_i)A\} |\Omega\rangle_\mathcal{U} = \prod_{i=1}^{N_{ops}} \alpha_i, \tag{19}$$

---

[5]Helpfully for what follows later, the factor of $\frac{1}{2}$ in eq. (2) ensures that all rows have norm ≤ 1.

[6]It will turn out that $B(0) = \mathbb{1}$, and so the separation of the A and $B(\alpha)$ gates will allow the $B(\alpha)$ gate to be omitted if $\alpha = 0$.

where the number of operations $N_{ops} \leq 2^{N_\mathcal{U}} - 1$.

For a given operator L, our general strategy for implementing an operator $\hat{L}$ which satisfies eq. (16) comprises two steps. One step is to act on $\mathcal{H}_1$ with a unitary operator whose rows differ from the rows of L by only a (row-dependent) normalisation. The other step is to act on $\mathcal{H}_1 \otimes \mathcal{H}_\mathcal{U}$ with controlled[7] versions of the A and $B(\alpha)$ gates in a way that, thanks to eq. (18), corrects for the normalisation changes.[8] Equation (16) follows as a direct consequence of these two steps. To apply a sequence of non-unitary operations, we simply repeat these two steps. In general, this will place $\mathcal{U}$ into a superposition of states, with the component proportional to $|\Omega\rangle_\mathcal{U}$ containing the information of interest due to eq. (16). Since the step acting on $\mathcal{H}_1$ does not affect $\mathcal{U}$, and the step acting on $\mathcal{H}_1 \otimes \mathcal{H}_\mathcal{U}$ only increments the state of $\mathcal{U}$ by at most 1 according to eq. (17) and never decrements it as long as $k < 2^{N_\mathcal{U}} - 1$, it can be seen that the two steps can be repeated up to $2^{N_\mathcal{U}} - 1$ times before the unitarisation register overflows. Therefore, as desired, the required size of $\mathcal{U}$ is small: it is logarithmic in the number of sequential non-unitary operations that we wish to perform. Furthermore, our unitarisation strategy maintains the quantum coherence of states produced by non-unitary operations, thus allowing the circuits in this paper to be used as building blocks for higher-level algorithms. The explicit implementation of eq. (16) for the linear operators in eqs. (14) and (15) is left to sec. 2.3. In the remainder of this section, we will introduce notation for controlled quantum gates, and then give explicit definitions for the A and $B(\alpha)$ gates.

For convenience, let us define some notation for controlled quantum gates, which will be used in this section and the next one. For any quantum gate, a controlled version of it can loosely be understood as applying that gate to one register, designated as the *target* register, if one or more other registers, designated as the *control* registers, are in a particular specified state. For example, as we will see in sec. 2.3.1, the Q gate implements the quark-gluon interaction by applying a rotation which is targeted at the quark register, with the choice of rotation controlled by the state of the gluon register. More precisely, let U (not to be confused with $\mathcal{U}$) be any quantum gate acting on a Hilbert space $\mathcal{H}_\mathrm{trgt}$ and let $|\psi\rangle$ be a normalised state in another Hilbert space $\mathcal{H}_\mathrm{ctrl}$. Then we define the $|\psi\rangle$-controlled-U gate $C_{|\psi\rangle}[U]$ acting on the Hilbert space $\mathcal{H}_\mathrm{ctrl} \otimes \mathcal{H}_\mathrm{trgt}$ as follows:

$$C_{|\psi\rangle}[U] = |\psi\rangle\langle\psi| \otimes U + (\mathbb{1}_{\mathcal{H}_\mathrm{ctrl}} - |\psi\rangle\langle\psi|) \otimes \mathbb{1}_{\mathcal{H}_\mathrm{trgt}}. \tag{20}$$

Here $\mathbb{1}_{\mathcal{H}_i}$ is the identity operator acting on the Hilbert space $\mathcal{H}_i$. In the context of eq. (20) we will call the qubits represented by $\mathcal{H}_\mathrm{ctrl}$ the control qubits, and call the qubits represented by $\mathcal{H}_\mathrm{trgt}$ the target qubits. Equation (20) implies in particular that for any state $|\phi\rangle \in \mathcal{H}_\mathrm{trgt}$,

$$C_{|\psi\rangle}[U]\{|\psi\rangle \otimes |\phi\rangle\} = |\psi\rangle \otimes U|\phi\rangle \tag{21}$$

and furthermore that given any state $|\psi'\rangle \in \mathcal{H}_\mathrm{ctrl}$ satisfying $\langle\psi'|\psi\rangle = 0$,

$$C_{|\psi\rangle}[U]\{|\psi'\rangle \otimes |\phi\rangle\} = |\psi'\rangle \otimes |\phi\rangle. \tag{22}$$

In this article, any controlled gate $C_{|\psi\rangle}[U]$ will be depicted in the manner shown in fig. 2. In our QISKIT program, the function `qiskit.extensions.UnitaryGate.control` has been used to implement eq. (20).

---

[7] to be explained shortly

[8] By applying suitable rotations, our method could be extended to apply to cases where the rows of L are not mutually orthogonal, but that is beyond the scope of this article.
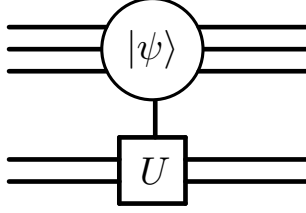
Figure 2: Graphical representation of $|\psi\rangle$-controlled-U gate $C_{|\psi\rangle}[U]$ defined in eq. (20). Here, $|\psi\rangle$ is any specified state of the upper register, and U is any operator defined on Hilbert space of the lower register.

Let us proceed by defining the well-known increment operator (see for example Refs. [44, 45]), which we denote A and which is depicted in fig. 3. Formally, the gate can be defined as follows. Let X be the Pauli-X single-qubit gate:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{23}$$

Let X_n be the gate $C_{|\underbrace{1\ldots1}_{n-1}\rangle}[X]$ acting on the first n qubits of the unitarisation register, with qubits 1 through $n-1$ serving as control qubits and qubit n serving as target qubit. Let A_1 be the Pauli-X gate acting on qubit 1, and define recursively[9] for $n > 1$:

$$A_n = A_{n-1}X_n. \tag{24}$$

For a unitarisation register with $N_{\mathcal{U}}$ qubits, the increment operator of fig. 3 can now be formally defined as $A = A_{N_{\mathcal{U}}}$. Interpreting each basis state of the unitarisation register as a binary representation of a number $|k = \sum_{i=1}^{N_{\mathcal{U}}} u_i 2^{i-1}\rangle$, where $u_n \in \{0,1\}$ is the state of the nth qubit, it can be verified that

$$A|k\rangle_{\mathcal{U}} = |k+1 \pmod{2^{N_{\mathcal{U}}}}\rangle_{\mathcal{U}} \quad \forall k. \tag{25}$$

In this article, A should be assumed to always act on the unitarisation register.
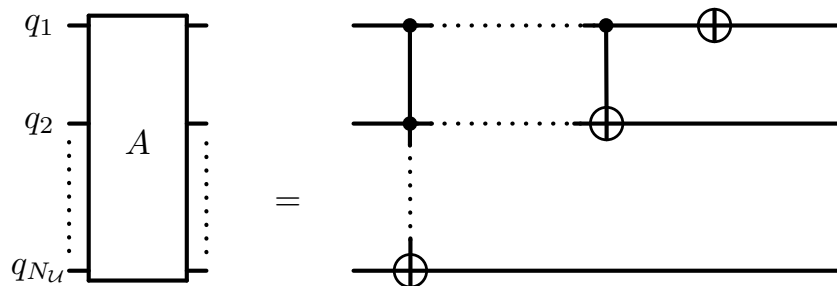
Figure 3: Graphical representation of the circuit for the A operator, which increments the unitarisation register $\mathcal{U}$. Here, $\oplus$ represents the Pauli-X gate and $|q_1\rangle, \ldots, |q_{N_{\mathcal{U}}}\rangle$ are the qubits that form $\mathcal{U}$.

Next let us define a single-qubit gate

$$B_1(\alpha) = \begin{pmatrix} \sqrt{1-|\alpha|^2} & \alpha \\ -\alpha & \sqrt{1-|\alpha|^2} \end{pmatrix}, \tag{26}$$

---

[9]One sometimes encounters [45] a different (but equivalent) recursive definition, which involves a controlled-A_{n-1} gate, but we have chosen not to adopt it here.
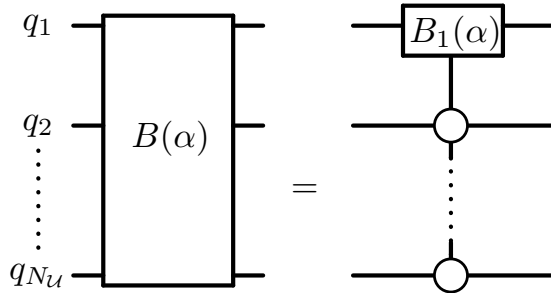
Figure 4: Graphical representation of the circuit for the partial-decrement operator $B(\alpha)$ defined formally in eq. (27).

which should be understood to always act on qubit 1 of the unitarisation register.

We now define a partial-decrement operator $B(\alpha)$ as follows:

$$B(\alpha) = C_{|\underbrace{0\ldots0}_{N_{\mathcal{U}}-1}\rangle}[B_1(\alpha)], \tag{27}$$

which should be understood to act on the unitarisation register with qubits 2 through $N_{\mathcal{U}}$ serving as control qubits and qubit 1 serving as target qubit, as shown in fig. 4. The desired behaviour of the A and $B(\alpha)$ gates shown in eq. (17) follows immediately from eqs. (25–27).

This concludes our explanation of the unitarisation register $\mathcal{U}$ and the A and $B(\alpha)$ gates. We will now proceed to use them to implement eq. (16) for the quark-gluon (14) and triple-gluon (15) interactions.

2.3 Construction of interaction gates

In this section, we will give details of the construction of the Q and G gates which simulate quark-gluon and triple-gluon interactions, respectively. As mentioned earlier, 2-qubit registers are used to represent the $N_{\mathrm{c}} = 3$ colour states of a quark as $|00\rangle$, $|01\rangle$, and $|10\rangle$. Note that the $|11\rangle$ state of each quark register is unused: all operators $\mathcal{O}$ acting on a quark register satisfy $\langle 11|\mathcal{O}|11\rangle = 1$, which means the last row and last column of the 4×4 matrix representing $\mathcal{O}$ are both $(0 \quad 0 \quad 0 \quad 1)$, ensuring that the register never enters the $|11\rangle$ state. For brevity we will omit the fourth row and fourth column when representing these operators as matrices.

2.3.1 Quark-gluon interaction gate Q

We wish to construct a gate Q which will implement the quark-gluon interaction. This interaction is described by the non-unitary operator shown in eq. (14), and we therefore wish unitarise it—see eq. (16)—by constructing a suitable $\hat{L}$ for it. Following the general strategy explained in

sec. 2.2, we start by defining the following unitary matrices $\overline{\lambda}_a$:

$$\overline{\lambda}_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \overline{\lambda}_2 = \begin{pmatrix} 0 & -i & 0 \\ i & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \overline{\lambda}_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\overline{\lambda}_4 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \overline{\lambda}_5 = \begin{pmatrix} 0 & 0 & -i \\ 0 & 1 & 0 \\ i & 0 & 0 \end{pmatrix}, \quad \overline{\lambda}_6 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \tag{28}$$

$$\overline{\lambda}_7 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -i \\ 0 & i & 0 \end{pmatrix}, \quad \overline{\lambda}_8 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

It can be observed that these matrices are similar to the Gell-Mann matrices eq. (3) but have been adjusted to make them unitary, and therefore implementable as quantum gates acting on a 3-state quark register. We combine the $\overline{\lambda}_a$ gates into a new gate Λ acting on a gluon register g and quark register q in the following way

$$\Lambda = \left[\prod_{a=1}^{8} C_{|a\rangle_g} \left[\overline{\lambda}_a \right] \right], \tag{29}$$

where g serves as control register and q serves as target register, as shown in fig. 5. Thus, depending on the colour a of the gluon register, Λ applies the gate $\overline{\lambda}_a$ to the quark register.
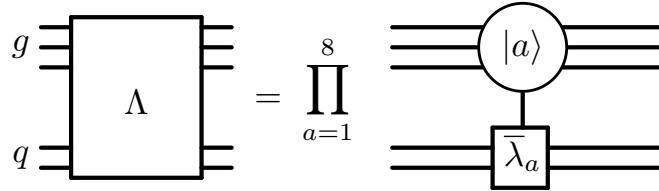
Figure 5: Graphical representation of the circuit of the Λ gate defined in eq. (29).

We proceed to construct the quark-gluon interaction gate Q, which is based on Λ but uses A and B gates to account for the differences between the matrices λ_a and $\overline{\lambda}_a$, as well as for the factor of $\frac{1}{2}$ in eq. (2). In brief, the Q gate will increment $\mathcal{U}$ using the A gate and then conditionally decrement it again using B gates, before finally applying the Λ gate to the state thus produced. More formally, as depicted in fig. 6, the Q gate is defined to act on the state
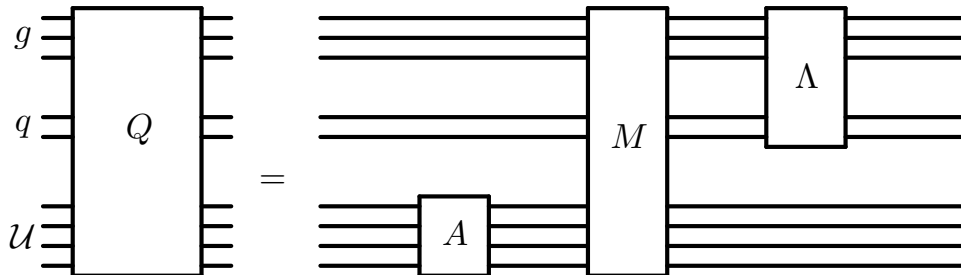

Figure 6: Graphical representation of the circuit of the quark gate Q defined in eq. (30).

$|\Psi\rangle_{g \otimes q \otimes \mathcal{U}}$ of any gluon register g, any quark register q, and the unitarisation-register $\mathcal{U}$ in the following way:

$$Q \, |\Psi\rangle_{g \otimes q \otimes \mathcal{U}} = (\Lambda \otimes \mathbb{1}_{\mathcal{U}}) \, M (\mathbb{1}_g \otimes \mathbb{1}_q \otimes A) \, |\Psi\rangle_{g \otimes q \otimes \mathcal{U}} \, , \tag{30}$$

where

$$M = \prod_{a,i \, : \, \mu(a,i) \neq 0} C_{|a\rangle_g |i\rangle_q} \left[B \left(\mu(a,i) \right) \right] , \tag{31}$$

and

$$\mu(a,i) = \begin{cases} \frac{1}{2} & \text{if } \left(\overline{\lambda}_a \right)_{ij} - (\lambda_a)_{ij} = 0 \quad \forall j \\ \frac{1}{2\sqrt{3}} & \text{if } a = 8 \text{ and } i \in \{1, 2\} \\ \frac{-1}{\sqrt{3}} & \text{if } a = 8 \text{ and } i = 3 \\ 0 & \text{otherwise.} \end{cases} \tag{32}$$

A graphical representation of the operator M is shown in fig. 7.
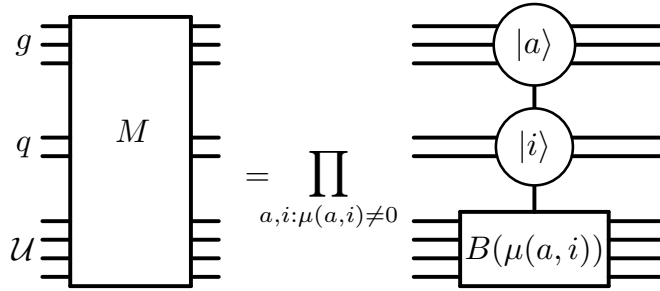
Figure 7: Graphical representation of the circuit of the M operator defined in eq. (31).

The definitions of $\overline{\lambda}_a$ and $\mu(a,i)$ in eqs. (28) and (32) have been chosen so that the following equation holds

$$\mu(a,i) \overline{\lambda}_a \, |i\rangle = \frac{1}{2} \lambda_a \, |i\rangle \, , \tag{33}$$

where the λ_a are defined in eq. (3) and the factor of $\frac{1}{2}$ originates from eq. (2). Note that since the control states in eq. (31) are mutually orthogonal, applying M does not decrement the state of $\mathcal{U}$ by more than 1. By using the properties of $\mathcal{U}$ in eq. (18) and the definitions of Q, M, and Λ in eqs. (29–31), it can be seen that if $|\psi_1\rangle = |a\rangle_g |k\rangle_q |\Omega\rangle_{\mathcal{U}}$ and $|\psi_2\rangle = |b\rangle_g |l\rangle_q |\Omega\rangle_{\mathcal{U}}$ then

$$\langle \psi_2 | Q | \psi_1 \rangle = \langle b | a \rangle \, \langle l | \tfrac{1}{2} \lambda_a | k \rangle \, . \tag{34}$$

The desired property of Q shown in eq. (7) follows immediately from this. Thus, a sequence of emissions and absorptions of gluons by a quark line can be simulated on a quantum computer by chaining a corresponding sequence of Q gates.

2.3.2 Triple-gluon interaction gate G

We shall now proceed to the description of a quantum gate for the triple-gluon interaction. This interaction is described by the non-unitary operator shown in eq. (15), and so we wish to unitarise it—see eq. (16)—by finding a suitable $\hat{L}$ for it, following the general strategy explained in sec. 2.2. In contrast to the quark-gluon operator (14), the triple-gluon operator (15) is diagonal and so

its corresponding $\hat{L}$ operator, which we call G, can be constructed using only A and controlled $B(\alpha)$ gates.

We define the gate G acting on any state $|\Psi\rangle_{g_1 \otimes g_2 \otimes g_3 \otimes \mathcal{U}}$ of gluon registers g_1, g_2, and g_3, and the unitarisation register $\mathcal{U}$ in the following way:

$$G |\Psi\rangle_{g_1 \otimes g_2 \otimes g_3 \otimes \mathcal{U}} = G' A |\Psi\rangle_{g_1 \otimes g_2 \otimes g_3 \otimes \mathcal{U}} \quad , \tag{35}$$

where

$$G' |\Psi\rangle_{g_1 \otimes g_2 \otimes g_3 \otimes \mathcal{U}} = \left(\prod_{a,b,c \,:\, f^{abc} \neq 0} C_{|a\rangle |b\rangle |c\rangle} \left[B\left(f^{abc} \right) \right] \right) |\Psi\rangle_{g_1 \otimes g_2 \otimes g_3 \otimes \mathcal{U}} \quad . \tag{36}$$

The gates G and G' are illustrated in figs. 8 and 9, respectively.
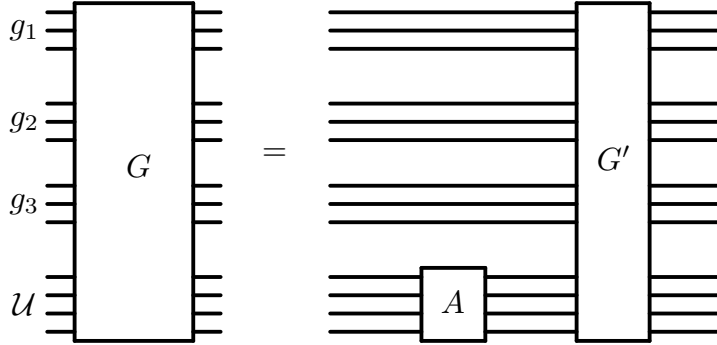
Figure 8: Graphical representation of the gluon gate G defined in eq. (35).
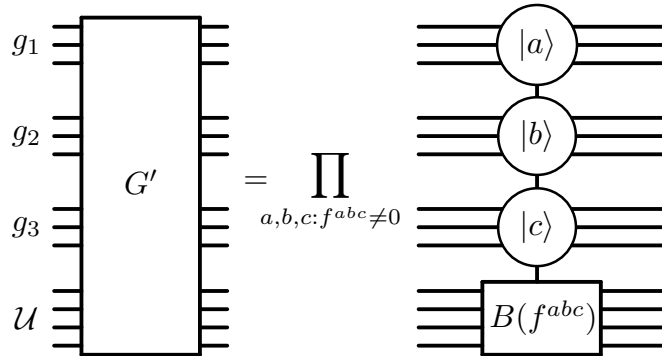
Figure 9: Graphical representation of the circuit of the G' gate defined in eq. (36).

One sees that in eq. (36), by definition (20) of the control gate, each factor $C_{|a\rangle |b\rangle |c\rangle} \left[B\left(f^{abc} \right) \right]$ applies $B(f^{abc})$ to the unitarisation register if the three gluons have colours a, b, c respectively, and leaves the unitarisation register unchanged if the gluons are in a state orthogonal to $|a\rangle_{g_1} |b\rangle_{g_2} |c\rangle_{g_3}$. As was seen for the M gate in the previous section, applying the G' gate does not decrement the state of $\mathcal{U}$ by more than 1, since the control states in eq. (36) are mutually orthogonal. Since $B(0) = \mathbb{1}$ and $C_{|a\rangle |b\rangle |c\rangle} [\mathbb{1}] = \mathbb{1}$, the product in eq. (36) does not need to include any cases where $f^{abc} = 0$. By using the property shown in eq. (18) and the definitions of G and G' in eqs. (35) and (36), it can be verified that eq. (13) indeed holds.

Hence, a triple-gluon interaction can be implemented by applying the gate G to the corresponding gluon registers. It can be observed that the gates G and Q do not rotate the states

of the gluon registers and so in diagrams where several triple-gluon interactions are present, the corresponding G gates can be applied in any order.

3 Results

In this section, the method introduced in sec. 2.1 will be generalised to simulate—and calculate colour factors for—arbitrary Feynman diagrams.

Let n_g be the number of gluons in the diagram and let n_q be the number of quark lines in the diagram. The quantum circuit to be constructed will contain gluon registers, quark registers, and a unitarisation register. There are n_g gluon registers, each with 3 qubits. There are n_q pairs of quark registers, each pair comprising 2 registers labelled q and $\tilde{q}$, with 2 qubits per register. The unitarisation register has $N_{\mathcal{U}} = \lceil \log_2(N_V + 1) \rceil$ qubits, where N_V is the number of vertices in the Feynman diagram. The procedure for calculating colour factors is as follows:

1. Initialise each register r into the state $|\Omega\rangle_r$.

2. Apply R_g, as in eq. (4), to each gluon register separately.

3. For each quark line, apply R_q to the corresponding pair of quark registers, q and $\tilde{q}$, as in eq. (5).

4. For each quark-gluon interaction vertex, apply a Q gate to the quark register q and gluon register g that correspond to the quark and gluon at that vertex. The corresponding $\tilde{q}$ register does not participate here.

5. For each triple-gluon interaction, apply a G gate to the 3 corresponding gluon registers.

6. Apply an R_g^{-1} gate to each gluon register.

7. For each quark line, apply an R_q^{-1} gate to the corresponding pair of quark registers, q and $\tilde{q}$.

In the above procedure, steps 4 and 5 simulate the evolution of the colour states of the particles in the Feynman diagram, while the remaining steps serve to perform the trace over the colours. Note that the Q gates corresponding to a given quark line must be applied in the order in which the corresponding interactions appear on that quark line in the Feynman diagram. Apart from this, there is no restriction on the ordering of the Q and G gates.

Analogously to the result in sec. 2.1, it follows from the Feynman rules and from eqs. (7), (13), and (19) that after step 7 the colour factor $\mathcal{C}$ of the Feynman diagram will be found encoded in the final state of the quantum computer, which is

$$\frac{1}{\mathcal{N}}\mathcal{C} |\Omega\rangle_{all} + (\text{terms orthogonal to } |\Omega\rangle_{all}), \tag{37}$$

where $\mathcal{N} = N_c^{n_q} \left(N_c^2 - 1\right)^{n_g}$ and

$$|\Omega\rangle_{all} = \left(\prod_{m=1}^{n_g} |\Omega\rangle_{g_m}\right) \left(\prod_{l=1}^{n_q} |\Omega\rangle_{q_l} |\Omega\rangle_{\tilde{q}_l}\right) |\Omega\rangle_{\mathcal{U}}. \tag{38}$$

As explained in sec. 2.1, the result in eq. (37) can be verified by repeatedly running the circuit and counting the number of times it is measured[10] to be in the state $|\Omega\rangle_{all}$, optionally

---

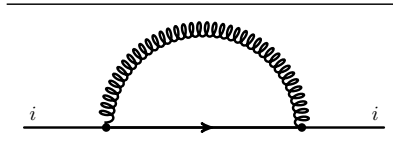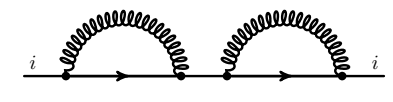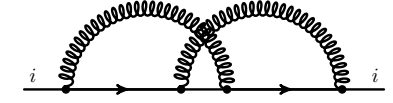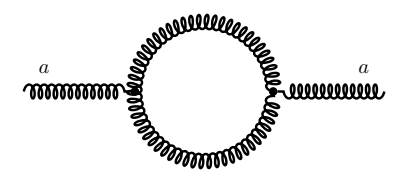[10]The measurements are performed in the computational basis.
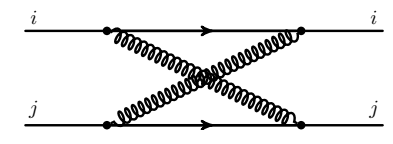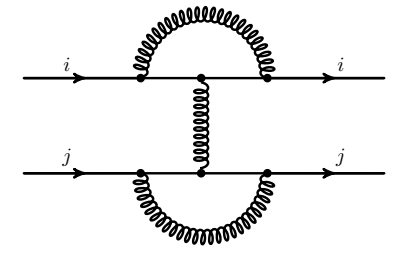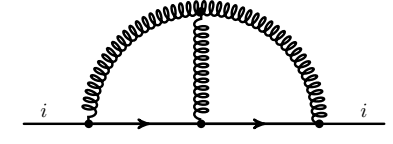
Diagram	Analytical	Numerical
	$C_F N = 4$	3.9988 ± 0.0012
	$C_F{}^2 N = \frac{16}{3}$	5.331 ± 0.010
	$\frac{C_F}{2} = \frac{2}{3}$	0.673 ± 0.010
	$N(N^2 - 1) = 24$	23.95 ± 0.03
	$\frac{(N^2-1)}{4} = 2$	2.00 ± 0.03
	0	$0.0^{+0.5}_{-0.0}$
	$\frac{C_F N^2}{2} = 6$	5.92 ± 0.08

Table 1: Colour factors for example Feynman diagrams. The first column depicts the Feynman diagrams, with indices on external legs indicating identical colours. The central column states the analytical result for the colour factor. The last column displays the numerical result for each colour factor obtained using 100 million runs of the simulated quantum circuit, along with the associated statistical uncertainty.

in conjunction with Hadamard testing (to retain phase information) and Quantum Amplitude Estimation (to obtain a quadratic speedup). Alternatively, the output (37) of the circuit can be directly used as a component of a future algorithm, such as one that calculates the kinematic factors and then sums over Feynman diagrams, or one that performs Monte Carlo integration to compute cross-sections. Since the output (37) is a quantum state, once combined with the corresponding kinematic factor it would be well-suited to future work which calculates the quantum interference of multiple diagrams, possibly by implementing a quantum-computing equivalent of the recursive algorithms [46] that are widely used in modern classical calculations.

To validate the methods and circuits described in this article, we have implemented them in a program named QCOLOUR written in PYTHON using the IBM QISKIT package.[11] Using QCOLOUR we have built quantum circuits to simulate several Feynman diagrams shown in Table 1. For each Feynman diagram, the corresponding circuit was run 10^8 times on a simulated noiseless quantum computer and the number N_Ω of times the output state was measured to be $|\Omega\rangle_{all}$ was counted. It follows from eq. (37) that N_Ω is binomially distributed as $N_\Omega \sim B\left(10^8, \left(\frac{C}{N}\right)^2\right)$. We therefore infer the absolute value of the colour factor to be

$$|\mathcal{C}| = \mathcal{N}\sqrt{\frac{N_\Omega}{10^8}}, \tag{39}$$

with a statistical uncertainty that can be estimated using the Wilson score interval [47]. As can be seen from Table 1, the colour factors obtained using the simulated quantum circuits are fully consistent with the colour factors calculated analytically.

It may be observed that the fractional uncertainty in the inferred colour factors increases with the complexity of the diagram, but we emphasise again that the measurement strategy employed here is only intended as a transparent way to verify that the circuits correctly produce the state (37). As mentioned above and in sec. 2.1, more sophisticated strategies can be employed to examine, measure, and exploit this quantum state, and the state can furthermore be used in a higher-level algorithm rather than being immediately measured. Therefore, while more complicated scattering processes are always likely to have higher computational costs (as in classical Monte Carlo calculations), the examples in Table 1 should not be taken as providing a conclusive indication of the scaling rate.

4 Conclusion

The simulation of quantum systems is a flagship application of quantum computers, with expectations for polynomial or exponential speed-ups over classical computers. In this article, first steps were taken towards a quantum simulation of generic perturbative QCD processes. In particular, quantum circuits were designed to simulate the colour parts of the interactions of quarks and gluons. In order to do so, the concept of a *unitarisation register* was devised to enable a unitary quantum-circuit implementation of the non-unitary Gell-Mann matrices λ_a and structure constants f^{abc} that describe the interaction vertices in Feynman diagrams. It was shown that these quantum circuits can be used to simulate the colour parts of arbitrary Feynman diagrams. Furthermore, these circuits were implemented on a simulated noiseless quantum computer using the QISKIT framework, and colour factors were hence calculated for various examples of Feynman diagrams. It is to be emphasised that besides enabling the calculation of colour factors, the

---

[11]For this implementation, we have used QISKIT version 0.36.1.

quantum circuits presented in this work can in the future be directly used as components of a full quantum simulation of scattering amplitudes.

The work presented here opens several directions for future exploration. Following the present simulation of the colour parts of Feynman diagrams, a natural extension of this work is the simulation of the kinematic parts, since the unitarisation register devised here would also be particularly useful there. Simulating the kinematic parts will require appropriate ways to handle the much larger Hilbert spaces resulting from the continuous nature of kinematic variables. This is likely to require many more qubits than the colour parts do, but since colliders like the LHC probe energy scales that are several orders of magnitude higher than Λ_{QCD}, we expect that the number of qubits required for simulating the kinematic parts can still be competitive against the requirements of lattice-based quantum-computer simulations of the same processes. Another natural extension is to explore the quantum-coherent interference of the contributions from multiple Feynman diagrams, a task to which quantum computers are naturally well-suited. Furthermore, it would also be interesting to explore the application of the quantum circuits from this work to perform calculations with full quantum correlations for high-multiplicity processes that are currently described using parton showers. Finally, although the present work is aimed at error-corrected quantum computers that are envisaged for the medium term, it would meanwhile be interesting to test these circuits against the noise characteristics of specific near-term hardware devices, and then perform custom adaptations (in hardware and software) to mitigate against the noise and its effects. In the long term, all these aspects can be combined with quantum algorithms known to have quadratic (or better) speedups, such as quantum Monte Carlo simulations, and then implemented on future physical quantum computers. This would provide significant improvements in the speed and possibly the reach of perturbative QCD calculations.

Acknowledgements

A   Miscellaneous gates

In this Appendix we give explicit definitions for the gates R_g and R_q. Both of these gates are used in sec. 2.1 and sec. 3. We also provide a circuit diagram for the Λ gate that was defined in eq. (29).

The R_g gate is composed of a Hadamard gate H acting on each qubit of the gluon register. Since H satisfies $H^{-1} = H$, it follows that $R_g^{-1} = R_g$. Its graphical representation is provided in fig. 10.
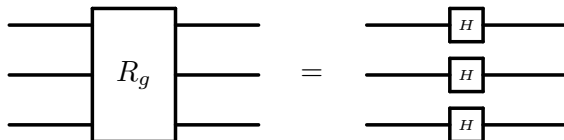

Figure 10: Graphical representation of the circuit of the R_g gate.

The R_q gate is defined by the circuit shown in fig. 11 and is composed of two controlled-X gates and a 2-qubit gate R defined as follows:

$$R = \begin{pmatrix} \sqrt{\frac{1}{3}} & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{6}} & 0 \\ \sqrt{\frac{1}{3}} & -\sqrt{\frac{1}{2}} & \sqrt{\frac{1}{6}} & 0 \\ \sqrt{\frac{1}{3}} & 0 & \sqrt{\frac{2}{3}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{40}$$

The inverse R_q^{-1} of R_q is easily constructed by reversing the order of the 3 gates in fig. 11 and replacing R by its transpose R^T.
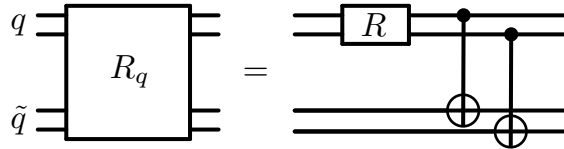

Figure 11: Graphical representation of the circuit of the R_q gate.

Finally, the explicit form of the Λ gate in terms of basic gates can be found in fig. 12. The matrices $\bar{\lambda}_a$ are given in eq. (28), and should be understood to be 4×4 dimensional as explained at the start of sec. 2.3. The M gate is constructed in a similar fashion, and comprises 17 controlled gates corresponding to the 17 combinations of a and i for which $\mu(a,i)$ is non-zero according to eq. (32). However, since μ takes one of only 4 different values, we expect that M could be constructed with fewer controlled operations by choosing a suitable encoding of a and i into qubits.
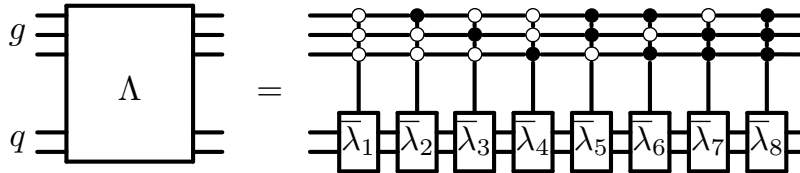

Figure 12: Explicit graphical representation of the circuit of the Λ gate defined in eq. (29). White and black circles represent controlled operations with control states $|0\rangle$ and $|1\rangle$, respectively.

References

[1] P. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134. 1994.

[2] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC 1996, p. 212–219. 1996.

[3] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. D. Sawaya, S. Sim, L. Veis, and

A. Aspuru-Guzik, *Quantum Chemistry in the Age of Quantum Computing*. Chemical Reviews **119** (2019) no. 19, 10856–10915.

[4] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, *Quantum computational chemistry*. Rev. Mod. Phys. **92** (2020) 015003.

[5] I. M. Georgescu, S. Ashhab, and F. Nori, *Quantum simulation*. Rev. Mod. Phys. **86** (2014) 153–185.

[6] F. Tacchino, A. Chiesa, S. Carretta, and D. Gerace, *Quantum Computers as Universal Quantum Simulators: State-of-the-Art and Perspectives*. Advanced Quantum Technologies **3** (2020) no. 3, 1900052.

[7] N. Klco, A. Roggero, and M. J. Savage, *Standard model physics and the digital quantum revolution: thoughts about the interface*. Rept. Prog. Phys. **85** (2022) no. 6, 064301, arXiv:2107.04769 [quant-ph].

[8] C. W. Bauer *et al.*, *Quantum Simulation for High Energy Physics*. arXiv:2204.03381 [quant-ph].

[9] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, *Quantum Amplitude Amplification and Estimation*. Quantum Computation and Information **305** (2002) , arXiv:quant-ph/0005055 [quant-ph].

[10] D. Grinko, J. Gacon, C. Zoufal, and S. Woerner, *Iterative Quantum Amplitude Estimation*. npj Quantum Inf **7** (2021) no. 52, , arXiv:1912.05559 [quant-ph].

[11] Y. Suzuki, S. Uno, R. Raymond, T. Tanaka, T. Onodera, and N. Yamamoto, *Amplitude estimation without phase estimation*. Quantum Information Processing **19** (2020) no. 75, , arXiv:1904.10246 [quant-ph].

[12] K. Nakaji, *Faster Amplitude Estimation*. Quantum Information & Computation 2020 **20** (2020) no. 13&14, , arXiv:2003.02417 [quant-ph].

[13] Z. Bern and D. A. Kosower, *Color decomposition of one loop amplitudes in gauge theories*. Nucl. Phys. B **362** (1991) 389–448.

[14] J. Hakkinen and H. Kharraziha, *COLOR: A Computer program for QCD color factor calculations*. Comput. Phys. Commun. **100** (1997) 311–321, arXiv:hep-ph/9603229.

[15] A. Kanaki and C. G. Papadopoulos, *HELAC: A Package to compute electroweak helicity amplitudes*. Comput. Phys. Commun. **132** (2000) 306–315, arXiv:hep-ph/0002082.

[16] C. G. Papadopoulos and M. Worek, *Multi-parton cross sections at hadron colliders*. Eur. Phys. J. C **50** (2007) 843–856, arXiv:hep-ph/0512150.

[17] M. Sjödahl, *ColorMath - A package for color summed calculations in SU(Nc)*. Eur. Phys. J. C **73** (2013) no. 2, 2310, arXiv:1211.2099 [hep-ph].

[18] S. Actis, A. Denner, L. Hofer, A. Scharf, and S. Uccirati, *Recursive generation of one-loop amplitudes in the Standard Model*. JHEP **04** (2013) 037, arXiv:1211.6316 [hep-ph].

[19] S. Lloyd, M. Mohseni, and P. Rebentrost, *Quantum principal component analysis*. Nature Physics **10** (2014) 631–633, arXiv:1307.0401 [quant-ph].

[20] S. Kimmel, C. Y.-Y. Lin, G. H. Low, M. Ozols, and T. J. Yoder, *Hamiltonian simulation with optimal sample complexity*. npj Quantum Information **3** (2017) no. 13, , arXiv:1608.00281 [quant-ph].

[21] J. van Apeldoorn and A. Gilyén, "Improvements in Quantum SDP-Solving with Applications," in *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, vol. 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, p. 99. 2019. arXiv:1804.05058 [quant-ph].

[22] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, "Quantum Singular Value Transformation and beyond: Exponential Improvements for Quantum Matrix Arithmetics," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, p. 193–204. 2019. arXiv:1806.01838 [quant-ph].

[23] G. H. Low and I. L. Chuang, *Hamiltonian Simulation by Qubitization*. Quantum **3** (2016) 163, arXiv:1610.06546 [quant-ph].

[24] S. Anis *et al.*, *Qiskit: An Open-source Framework for Quantum Computing*, 2021.

[25] G. Agliardi, M. Grossi, M. Pellen, and E. Prati, *Quantum integration of elementary particle processes*. Phys. Lett. B **832** (2022) 137228, arXiv:2201.01547 [hep-ph].

[26] H. M. Gray and K. Terashi, *Quantum Computing Applications in Future Colliders*. Front. in Phys. **10** (2022) 864823.

[27] A. Delgado *et al.*, "Quantum Computing for Data Analysis in High-Energy Physics," in *2022 Snowmass Summer Study*. 3, 2022. arXiv:2203.08805 [physics.data-an].

[28] A. Pérez-Salinas, J. Cruz-Martinez, A. A. Alhajri, and S. Carrazza, *Determining the proton content with a quantum computer*. Phys. Rev. D **103** (2021) no. 3, 034027, arXiv:2011.13934 [hep-ph].

[29] **QuNu** Collaboration, 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*. Phys. Rev. D **105** (2022) no. 11, L111502, arXiv:2106.03865 [hep-ph].

[30] K. Bepari, S. Malik, M. Spannowsky, and S. Williams, *Towards a quantum computing algorithm for helicity amplitudes and parton showers*. Phys. Rev. D **103** (2021) no. 7, 076020, arXiv:2010.00046 [hep-ph].

[31] S. Ramírez-Uribe, A. E. Rentería-Olivo, G. Rodrigo, G. F. R. Sborlini, and L. Vale Silva, *Quantum algorithm for Feynman loop integrals*. JHEP **05** (2022) 100, arXiv:2105.08703 [hep-ph].

[32] A. Cervera-Lierta, J. I. Latorre, J. Rojo, and L. Rottoli, *Maximal Entanglement in High Energy Physics*. SciPost Phys. **3** (2017) no. 5, 036, arXiv:1703.02989 [hep-th].

[33] S. Fedida and A. Serafini, *Tree-level entanglement in Quantum Electrodynamics*. arXiv:2209.01405 [quant-ph].

[34] G. Clemente, A. Crippa, K. Jansen, S. Ramírez-Uribe, A. E. Rentería-Olivo, G. Rodrigo, G. F. R. Sborlini, and L. Vale Silva, *Variational quantum eigensolver for causal loop Feynman diagrams and acyclic directed graphs.* arXiv:2210.13240 [hep-ph].

[35] C. W. Bauer, M. Freytsis, and B. Nachman, *Simulating Collider Physics on Quantum Computers Using Effective Field Theories.* Phys. Rev. Lett. **127** (2021) no. 21, 212001, arXiv:2102.05044 [hep-ph].

[36] C. W. Bauer, W. A. de Jong, B. Nachman, and D. Provasoli, *Quantum Algorithm for High Energy Physics Simulations.* Phys. Rev. Lett. **126** (2021) no. 6, 062001, arXiv:1904.03196 [hep-ph].

[37] K. Bepari, S. Malik, M. Spannowsky, and S. Williams, *Quantum walk approach to simulating parton showers.* Phys. Rev. D **106** (2022) no. 5, 056002, arXiv:2109.13975 [hep-ph].

[38] G. Gustafson, S. Prestel, M. Spannowsky, and S. Williams, *Collider events on a quantum computer.* JHEP **11** (2022) 035, arXiv:2207.10694 [hep-ph].

[39] C. Bravo-Prieto, J. Baglio, M. Cè, A. Francis, D. M. Grabowska, and S. Carrazza, *Style-based quantum generative adversarial networks for Monte Carlo events.* Quantum **6** (2022) 777, arXiv:2110.06933 [quant-ph].

[40] O. Kiss, M. Grossi, E. Kajomovitz, and S. Vallecorsa, *Conditional Born machine for Monte Carlo event generation.* Phys. Rev. A **106** (2022) no. 2, 022612, arXiv:2205.07674 [quant-ph].

[41] L. J. Dixon, "Calculating scattering amplitudes efficiently," in *Theoretical Advanced Study Institute in Elementary Particle Physics (TASI 95): QCD and Beyond*, pp. 539–584. 1, 1996. arXiv:hep-ph/9601359.

[42] M. L. Mangano, "Introduction to QCD," in *1998 European School of High-Energy Physics*, pp. 53–97. 1998.

[43] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, *Quantum algorithms revisited.* Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences **454** (1998) no. 1969, 339–354.

[44] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition.* Cambridge University Press, 2010.

[45] E. Johnston, N. Harrigan, and M. Gimeno-Segovia, *Programming Quantum Computers: Essential Algorithms and Code Samples.* O'Reilly Media, Incorporated, 2019.

[46] F. A. Berends and W. T. Giele, *Recursive Calculations for Processes with n Gluons.* Nucl. Phys. **B306** (1988) 759–808.

[47] E. B. Wilson, *Probable Inference, the Law of Succession, and Statistical Inference.* Journal of the American Statistical Association **22** (1927) no. 158, 209–212.