

Quantum simulation of scattering amplitudes and interferences in perturbative QCD

Herschel A. Chawdhry¹*, Mathieu Pellen², and Simon Williams³

¹ *Department of Physics, Florida State University,
77 Chieftan Way, Tallahassee FL 32306, USA*

² *Albert-Ludwigs-Universität Freiburg, Physikalisches Institut,
Hermann-Herder-Straße 3, D-79104 Freiburg, Germany*

³ *Institute for Particle Physics Phenomenology,
Durham University, Durham DH1 3LE, UK*

Abstract

A flagship application of quantum computers is the simulation of other quantum systems, including quantum field theories. In this article, we show how quantum computers can be employed to naturally calculate Feynman diagrams and their interferences in Quantum Chromodynamics (QCD). We simulate the colour parts of the interactions directly on the quantum computer, while the kinematic parts are for now pre-computed classically. For processes where some of the external particles are identical, we find the first hints of a potential quantum advantage. We validate our techniques using simulated quantum computers. Furthermore, for toy examples we also demonstrate our algorithms on a 56-qubit trapped-ion quantum computer. The work constitutes a further key step towards a full quantum simulation of generic perturbative QCD processes.

*E-mail: hchawdhry@fsu.edu

[†]E-mail: mathieu.pellen@physik.uni-freiburg.de

[‡]E-mail: simon.j.williams@durham.ac.uk

Contents

1	Introduction	2
2	Amplitude-level colour simulation	3
3	Interference of multiple Feynman diagrams	8
4	Feynman diagrams with identical particles	12
5	Validation on physical quantum devices	18
6	Conclusion	19

1 Introduction

Scattering amplitudes are of core importance in high-energy physics, due to their central role in predicting the scattering, production, and decay of fundamental particles. These amplitudes are defined in the framework of Quantum Field Theory (QFT), and their precise calculation is vital for relating our fundamental models of Nature to the experimental measurements performed at particle colliders such as the Large Hadron Collider (LHC). This enables stringent validations of the Standard Model of particle physics, as well as increasing our ability to detect potential new particles or phenomena beyond the Standard Model. Since the LHC collides protons, composed of quarks and gluons, one relies in large part on calculations in Quantum Chromodynamics (QCD), the QFT that describes the strong interactions of quarks and gluons.

The high-energy regime of QCD is of special interest, because it is extensively probed by LHC events and is therefore of most direct relevance for studying the properties of Nature at the shortest distance scales presently accessible to experiments. For these energies, far above the fundamental QCD scale $\Lambda_{\text{QCD}} \sim 200\text{MeV}$, a direct lattice-based simulation of QCD is likely to remain well beyond reach for the foreseeable future due to the large number of lattice sites that it would require. At the same time, at these energies the QCD coupling constant α_s becomes small, allowing scattering amplitudes and their associated physical observables to be calculated as perturbative series in α_s . Perturbative QCD calculations thus play a central role in predicting the outcomes of high-energy collisions [1].

Perturbative QCD calculations are computationally challenging, which limits our ability to predict the results of experiments, infer the properties of known fundamental particles, or detect the existence of new ones. New computational techniques are thus constantly being sought. In view of recent advances in quantum computing hardware technology [2–7], it is natural to consider whether quantum computers could be beneficial for calculating perturbative QCD scattering amplitudes. The investigation of this question is the purpose of this article, and indeed we find the first hints of a potential quantum advantage, *i.e.* indications that quantum computers could outperform classical computers in this domain.

Perturbative QCD observables, such as scattering cross-sections and particle decay rates, are calculated from the modulus-square $|\mathcal{A}|^2$ of the amplitude \mathcal{A} , which can in turn be written as a sum over Feynman diagrams:

$$\mathcal{A} = \sum_n C_n K_n. \tag{1}$$

Here for later convenience, each Feynman diagram F_n has been decomposed into a product of a colour factor C_n and a kinematic factor K_n .¹ Since quantum computers operate by manipulating superpositions of states, they are natural candidates for simulating perturbative QCD processes. Indeed, when performing a measurement of the quantum computer, the probability of a given outcome is determined by the modulus-square of the relevant component of the wavefunction of the quantum computer. These commonalities will be exploited in several ways in this article: first to directly simulate the colour parts C_n of individual Feynman diagrams, and later to obtain the quantum-coherent sums and interferences of multiple Feynman diagrams.

The possibility of using quantum computers to simulate quantum systems has previously been studied in various contexts, with examples ranging from chemistry [8, 9] and condensed matter [10–13] to non-perturbative QFT simulations [14–26]. However, there is a significant difference between those applications and ours: most such proposals seek to use a known Hamiltonian to perform the unitary evolution of a quantum system, usually by means of Trotterisation [27, 28], whereas in perturbative QFT calculations one seeks to calculate the (Hermitian, but not unitary) transition matrix between specified external states.

The first step towards a quantum simulation of generic perturbative QCD processes was taken in Ref. [29], which introduced two fundamental quantum circuits, denoted Q and G , to simulate the colour parts of the quark-gluon and triple-gluon interactions, respectively. In this way, the circuits provide building blocks for the colour factors C_n appearing in Eq. (1). In the present article we start by employing Q and G gates to simulate the colour parts of individual Feynman diagrams (Sec. 2), in a refinement of the colour-tracing algorithm of Ref. [29]. Next, in Sec. 3 we present techniques to simulate multiple Feynman diagrams on a quantum computer and obtain their interference via the measurement process. We then in Sec. 4 consider the special case of Feynman diagrams related to one another by the interchange of identical external particles, and show efficient ways to obtain the sums and interferences of these diagrams. To validate these new algorithms, we implement them on idealised noiseless quantum emulators as well as on the physical 56-qubit trapped-ion quantum computer H2-1 made by QUANTINUUM. The results from the noiseless emulators are presented alongside the relevant methods in Secs. 3 and 4, while for the demonstrations on the physical device we simplify to a 2-colour toy model of QCD and so we present the latter results separately in Sec. 5.

Finally, we highlight that our work fits into a broader context of proposed quantum-computing applications to high-energy physics which have emerged in the last few years [30–34], ranging from quantum integration [35–39] and quantum Monte Carlo [40–42], to parton showers including interference effects [43–48], and also parton-distribution functions [49, 50] and loop-integrals [51–54]. The present article thus continues the growing effort to harness the potential of quantum computing in addressing complex problems in high-energy physics, aiming to uncover new applications and computational strategies within this domain.

2 Amplitude-level colour simulation

In general, the amplitude \mathcal{A} and colour factors C_n in Eq. (1) will carry free colour indices representing the colours of external particles, and so we can explicitly write

$$\mathcal{A}_{i_1 i_2 \dots j_1 j_2 \dots}^{a_1 a_2 \dots} = \sum_n C_{n, i_1 i_2 \dots j_1 j_2 \dots}^{a_1 a_2 \dots} K_n. \quad (2)$$

¹Quartic gluon interactions cannot be factorised in this way, but their contributions can be expanded as a sum of terms, each factorised into a product of a colour factor and a kinematic factor.

Here for each external gluon in the process there is an index a_m in the adjoint representation of the QCD gauge group algebra $\mathfrak{su}(N_c)$, where $N_c = 3$ is the number of quark colours, and for each external quark or external antiquark there is a fundamental-representation index i_m or antifundamental-representation index j_m as appropriate. Some of the colour factors C_n may have additional colour indices corresponding to internal particles, but such indices are contracted and so are not displayed in Eq. (2). Our aim in this section is to encode the colour factor $C_{n,i_1 i_2 \dots, j_1 j_2 \dots}^{a_1 a_2 \dots}$ of a single diagram F_n into the wavefunction of the quantum computer. Such a representation will allow us to directly obtain the colour-summed squared colour factor C_n^2 for our chosen diagram by performing a measurement on the quantum computer, but it will also lay the groundwork for obtaining interferences of multiple diagrams in the later sections.

We begin by summarising the behaviour of the Q and G gates from Ref. [29], and then present a refined algorithm to combine these gates to produce the colour factor C_n of any chosen Feynman diagram. We represent the colour state of a particle by using a quantum computer register comprising several qubits. For example, the $N_c^2 - 1 = 8$ colour basis states of a gluon can be represented by the $2^3 = 8$ basis states of a 3-qubit register. The Q and G gates act upon these registers to simulate, respectively, the fundamental-representation generators T_{ij}^a and the structure functions f^{abc} of $\mathfrak{su}(N_c)$ in the following way.² Given a register q representing a quark in colour basis state $|j\rangle$, with $j \in \{1, \dots, N_c\}$, and a register g representing a gluon in colour basis state $|a\rangle$, with $a \in \{1, \dots, N_c^2 - 1\}$, the Q gate performs the transformation

$$Q |a\rangle_g |j\rangle_q |\Omega\rangle_{\mathcal{U}} = \sum_{i=1}^{N_c} T_{ij}^a |a\rangle_g |i\rangle_q |\Omega\rangle_{\mathcal{U}} + (\text{terms orthogonal to } |\Omega\rangle_{\mathcal{U}}). \quad (3)$$

Here \mathcal{U} is a small ancilla register whose inclusion allows for the simulation of non-unitary operators like T_{ij}^a and f^{abc} , as further explained in Ref. [29]. We use the notation $|\Omega\rangle_r$ to represent a reference state of register r in which all qubits are in the $|0\rangle$ state. Similarly to Eq. (3), the G gate acts on three gluon registers as follows:

$$G |a\rangle_{g_1} |b\rangle_{g_2} |c\rangle_{g_3} |\Omega\rangle_{\mathcal{U}} = f^{abc} |a\rangle_{g_1} |b\rangle_{g_2} |c\rangle_{g_3} |\Omega\rangle_{\mathcal{U}} + (\text{terms orthogonal to } |\Omega\rangle_{\mathcal{U}}). \quad (4)$$

Full details regarding the construction and implementation of the Q and G gates can be found in Ref. [29]. We highlight that when combining multiple Q and G gates to simulate one or more Feynman diagrams, a single \mathcal{U} register suffices, and the number of qubits it must contain is logarithmic in the number of vertices of any one Feynman diagram.

By applying a suitable sequence of Q and G gates, it is straight-forward to implement the colour part of a Feynman diagram as specified by the Feynman rules of QCD. Indeed, in Ref. [29] we provided an explicit algorithm to do so and hence obtain the colour trace for a particular Feynman diagram. However, that trace algorithm required all colour indices to be contracted, and therefore implicitly performed colour traces at the squared-amplitude level. In the remainder of this section, we introduce an improved tracing algorithm that instead operates at the amplitude level. The advantages of this approach are three-fold. Firstly, it halves the circuit depth (*i.e.* the number of sequential operations in the circuit), since an unsquared Feynman diagram has half as many factors of T_{ij}^a and f^{abc} as a squared diagram, and therefore requires half as many Q and G gates to simulate it. Secondly, it reduces the number of shots (*i.e.* repeated runs) required on the quantum device to attain a given level of precision. Thirdly, and most importantly, it

²As is common practice, the normalisation of the fundamental-representation generators in this article is chosen such that $T_F = \frac{1}{2}$ in the identity $\text{Tr}(T^a T^b) = T_F \delta^{ab}$.

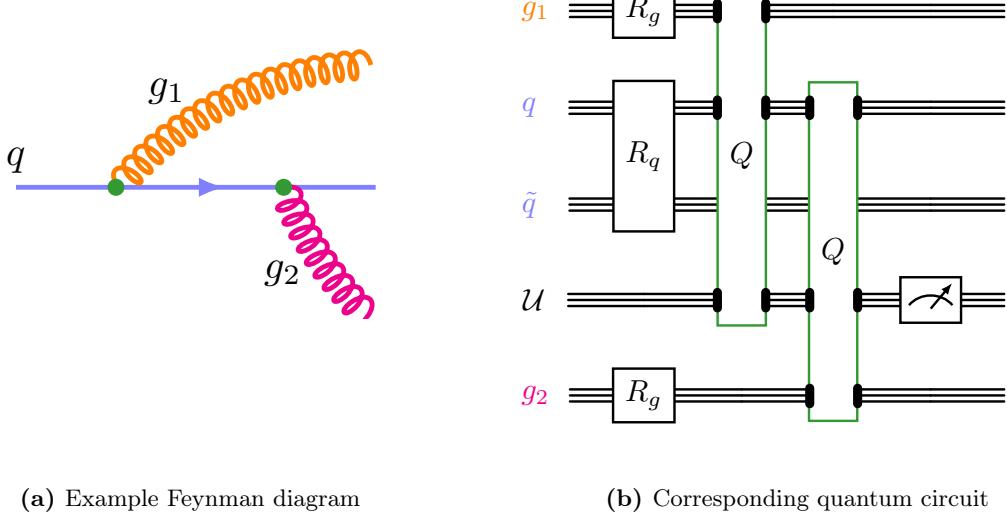


Figure 1: Illustrative example Feynman diagram (left) and a quantum computing circuit to simulate it (right). Qubits are grouped into registers, shown as triple horizontal lines. Each gluon is represented by a register as indicated. The quark line is represented by a pair of registers q and \tilde{q} to encode its fundamental and antifundamental colour index respectively. There is also an ancilla register \mathcal{U} for reasons explained in the main text. Boxes represent operations acting on one or more registers. Each Q operation simulates a quark-gluon vertex by acting on a subset of registers identified by black dots. The meter icon represents measurement.

lays the groundwork for allowing multiple Feynman diagrams to be coherently summed at the amplitude level by the quantum computer, which we will exploit in Sections 3 and 4 to calculate interferences of diagrams.

Before laying out our refined colour-tracing algorithm in full generality, we will start with an illustrative example. We will consider the Feynman diagram in Fig. 1a. Its colour factor, according to the Feynman rules of QCD, is

$$\sum_l T_{il}^b T_{lj}^a, \quad (5)$$

where the adjoint indices a and b are associated, respectively, to the gluons g_1 and g_2 , while the indices j, l, i are associated, respectively, to the initial, intermediate, and final colour states of the quark. This colour factor can be obtained by simulating the Feynman diagram using the quantum computer circuit in Fig. 1b, whose construction and operation will now be explained step by step. The circuit has 2 gluon registers, one for each gluon in Fig 1a. There is also a pair of quark registers q and \tilde{q} , to represent the fundamental and antifundamental external index of the quark line, respectively. In addition, there is a register \mathcal{U} as discussed above. Initially, each register r is in the state $|\Omega\rangle_r$, and so the state of the quantum computer is

$$|\Omega\rangle_{g_1} |\Omega\rangle_{g_2} |\Omega\rangle_q |\Omega\rangle_{\tilde{q}} |\Omega\rangle_{\mathcal{U}}. \quad (6)$$

We start by placing each gluon register g into an equal superposition of colour states by applying a rotation operator R_g designed such that

$$R_g |\Omega\rangle_g = \sum_{a=1}^8 \frac{1}{\sqrt{8}} |a\rangle_g. \quad (7)$$

We also apply a rotation to the q and \tilde{q} registers using an operator R_q designed such that

$$R_q |\Omega\rangle_q |\Omega\rangle_{\tilde{q}} = \sum_{j=1}^3 \frac{1}{\sqrt{3}} |j\rangle_q |j\rangle_{\tilde{q}}. \quad (8)$$

Explicit constructions for R_g and R_q are given in Appendix A of Ref. [29]. Thus, after applying the R_g and R_q gates appearing in Fig. 1b, the state of the quantum computer is

$$\frac{1}{8\sqrt{3}} \sum_{\substack{a,b \in \{1,\dots,8\} \\ j \in \{1,2,3\}}} |a\rangle_{g_1} |b\rangle_{g_2} |j\rangle_q |j\rangle_{\tilde{q}} |\Omega\rangle_{\mathcal{U}}. \quad (9)$$

We then apply the Q gate in Eq. (3) to registers q , g_1 , and \mathcal{U} to simulate the first gluon emission vertex in Fig. 1a. This places the quantum computer into a state

$$\frac{1}{8\sqrt{3}} \sum_{\substack{a,b \in \{1,\dots,8\} \\ j,l \in \{1,2,3\}}} T_{lj}^a |a\rangle_{g_1} |b\rangle_{g_2} |l\rangle_q |j\rangle_{\tilde{q}} |\Omega\rangle_{\mathcal{U}} + (\text{terms orthogonal to } |\Omega\rangle_{\mathcal{U}}). \quad (10)$$

Next, we apply another Q gate, this time to registers q , g_2 , and \mathcal{U} , to simulate the second gluon emission vertex in Fig. 1a. This places the quantum computer into a state

$$\frac{1}{8\sqrt{3}} \sum_{\substack{a,b \in \{1,\dots,8\} \\ i,j,l \in \{1,2,3\}}} T_{il}^b T_{lj}^a |a\rangle_{g_1} |b\rangle_{g_2} |i\rangle_q |j\rangle_{\tilde{q}} |\Omega\rangle_{\mathcal{U}} + (\text{terms orthogonal to } |\Omega\rangle_{\mathcal{U}}). \quad (11)$$

It can be seen that in this state, the coefficient of $|\Omega\rangle_{\mathcal{U}}$ neatly encodes the colour factor of the Feynman diagram. We highlight that in this state, just as in Eq. (5), the dummy index l has been summed over. The free indices i, j, a, b , on the other hand, remain encoded in the states of the registers q, \tilde{q}, g_1, g_2 respectively, which will turn out to be convenient in Secs. 3 and 4 when we sum multiple Feynman diagrams and obtain their interferences. It can also be seen that if the \mathcal{U} register is measured, the probability of obtaining the reference state $|\Omega\rangle_{\mathcal{U}}$ is

$$P(|\Omega\rangle_{\mathcal{U}}) = \frac{1}{192} \sum_{\substack{a,b \in \{1,\dots,8\} \\ i,j,l,m \in \{1,2,3\}}} T_{il}^a T_{lj}^b T_{jm}^b T_{mi}^a, \quad (12)$$

which is, up to a normalisation, the colour-averaged squared colour factor for the diagram in Fig. 1a.

We can now generalise to a generic Feynman diagram F_n containing N_{g_I} internal gluons, N_{g_E} external gluons, N_{q_I} closed quark loops, and N_{q_E} open quark lines. For each gluon, we require a register containing 3 qubits, as explained above. For each quark line, we require a pair of 2-qubit registers q and \tilde{q} , which will encode respectively the fundamental and antifundamental index of the quark line. We also require an ancilla register \mathcal{U} comprising $\lceil \log_2(N_V) \rceil$ qubits, where N_V is the number of vertices in the Feynman diagram. The colour part of F_n can then be simulated using the following procedure:

1. Initialise each register r into the state $|\Omega\rangle_r \equiv |0\dots0\rangle_r$.
2. Apply R_g to each gluon register.

3. For each quark line, apply R_q to the associated q and \tilde{q} registers.
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.
5. For each triple-gluon interaction, apply a G gate to the 3 corresponding gluon registers.
6. For each internal gluon, apply an R_g^{-1} gate to the corresponding gluon register.
7. For each internal quark line, apply an R_q^{-1} gate to the corresponding pair of quark registers, q and \tilde{q} .
8. Measure the unitarisation register \mathcal{U} and the registers associated with internal particles I , but not the registers associated with external particles E .

This procedure is similar to that of Ref. [29], but has three important refinements so that it operates at the amplitude level and can be used later for calculating interferences. Firstly, steps 4 and 5 are performed for a Feynman diagram instead of its square. Secondly, steps 6 and 7 apply only to internal particles. Thirdly, the qubits associated with external particles are not measured.

Defining a shorthand $|\Omega\rangle_{r_1+r_2+\dots}$ to indicate a state where registers r_1, r_2, \dots are each in a reference state $|0\dots0\rangle$, the final state $|\Psi\rangle$ of the quantum computer at the time of the measurement is, analogously to Eq. (11),

$$|\Psi\rangle = \frac{1}{\mathcal{N}} |\xi_n\rangle_E |\Omega\rangle_{I+\mathcal{U}} + (\text{terms orthogonal to } |\Omega\rangle_{I+\mathcal{U}}), \quad (13)$$

where $|\xi_n\rangle$ encodes the desired colour factor C_n of the chosen Feynman diagram F_n as:

$$|\xi_n\rangle = \sum_{\substack{a_1, a_2, \dots \in \{1, \dots, N_c^2 - 1\} \\ i_1, i_2, \dots \in \{1, \dots, N_c\} \\ j_1, j_2, \dots \in \{1, \dots, N_c\}}} C_{n, i_1 i_2 \dots, j_1 j_2 \dots}^{a_1 a_2 \dots} (|a_1\rangle |a_2\rangle \dots) (|i_1\rangle |i_2\rangle \dots) (|j_1\rangle |j_2\rangle \dots). \quad (14)$$

Here, just as in Eq. (2), the indices a_k, i_k, j_k are associated only with external particles. In Eq. (13), we have introduced a normalisation factor

$$\mathcal{N} = N_c^{(2N_{q_I}+N_{q_E})/2} (N_c^2 - 1)^{(2N_{g_I}+N_{g_E})/2}. \quad (15)$$

Note that $|\xi_n\rangle$ does not necessarily have unit norm. In fact, we can see from Eq. (14) that $\langle \xi_n | \xi_n \rangle$ is the fully-contracted squared colour factor C_n^2 for our chosen Feynman diagram F_n :

$$\langle \xi_n | \xi_n \rangle = \sum_{\substack{a_1, a_2, \dots \in \{1, \dots, N_c^2 - 1\} \\ i_1, i_2, \dots \in \{1, \dots, N_c\} \\ j_1, j_2, \dots \in \{1, \dots, N_c\}}} \left(C_{n, i_1 i_2 \dots, j_1 j_2 \dots}^{a_1 a_2 \dots} \right)^2. \quad (16)$$

Equation (13) implies that the probability that the measurement in step 8 yields $|\Omega\rangle_{I+\mathcal{U}}$ is:

$$P(|\Omega\rangle_{I+\mathcal{U}}) = \frac{1}{\mathcal{N}^2} \langle \xi_n | \xi_n \rangle. \quad (17)$$

Thus, if C_n^2 is known beforehand, we can use Eq. (17) to verify that the output state $|\Psi\rangle$ is correctly being produced by performing multiple shots of the circuit, whereby we run the quantum

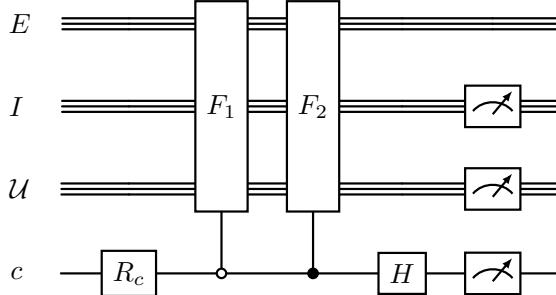


Figure 2: Quantum circuit to compute the sum and interference of two Feynman diagrams F_1 and F_2 . Qubits are grouped into registers, shown as triple horizontal lines. Registers E and I encode the states of external and internal particles respectively, while \mathcal{U} is the unitarisation register and c is a single-qubit register (shown as a single horizontal line) used for combining the two Feynman diagrams. A meter icon represents measurement. All other boxes depict unitary operations defined in the main text. White circles or black circles indicate, respectively, controlling on a $|0\rangle$ or $|1\rangle$ state.

circuit multiple times and count the fraction of times that the measurement yields $|\Omega\rangle_{I+\mathcal{U}}$. Conversely, Eq. (17) can also be used to estimate C_n^2 from the outcome of multiple shots, but while this provides a straight-forward and transparent way to validate the simulation, we emphasise that more sophisticated measurement schemes are available such as quantum amplitude estimation [55–58], which compared to a repeated-measurement strategy could provide a quadratic speedup for obtaining C_n^2 at a given level of precision.

3 Interference of multiple Feynman diagrams

Now that we have shown how to simulate the colour part of a single Feynman diagram, in this section we present a method to use the quantum computer to calculate interferences of multiple Feynman diagrams. Let us start by considering the case where we wish to calculate the sum (and hence the interference) of two diagrams, F_1 and F_2 . The external particles in the two diagrams are of course the same. To ensure that the two diagrams require the same number of qubits and share the same normalisation factor Eq. (15), we will for this example also assume that the number of closed quark-loops and internal gluons does not differ between the diagrams, although later in this section we will lift this restriction. We introduce a new ancilla qubit c and define the following single-qubit rotation operation R_c :

$$R_c = \frac{1}{\sqrt{|K_1|^2 + |K_2|^2}} \begin{pmatrix} K_1 & -K_2 \\ K_2 & K_1 \end{pmatrix}. \quad (18)$$

In the present algorithm, the kinematic factors K_1 and K_2 need to have been classically computed in advance for a chosen kinematic point and spin configuration, but in the future one could incorporate a full quantum simulation of the kinematics. By running the circuit shown in Fig. 2, the amplitude in Eq. (1) is obtained via a sequence of states that we will now define in detail.

At the start of the circuit, each register r is in the reference state $|\Omega\rangle_r \equiv |0\dots 0\rangle_r$ and so the state of the quantum computer is

$$|\Omega\rangle_E |\Omega\rangle_I |\Omega\rangle_{\mathcal{U}} |\Omega\rangle_c. \quad (19)$$

After applying the R_c gate defined in Eq. (18) to the ancilla qubit c , the state of the quantum computer becomes

$$|\Omega\rangle_E |\Omega\rangle_I |\Omega\rangle_{\mathcal{U}} \left(\frac{K_1 |0\rangle_c + K_2 |1\rangle_c}{\sqrt{|K_1|^2 + |K_2|^2}} \right). \quad (20)$$

A controlled- F_1 operation is then performed on registers E, I, \mathcal{U} , where the control is on qubit c being in the $|0\rangle$ state. Here for brevity we will abuse notation by using F_n not only to denote a particular diagram, but also to denote a quantum circuit constructed according to the instructions of Sec. 2 (excluding measurement) to simulate that diagram. Since the operation is controlled, the state specified in Eq. (13) is produced for the $|0\rangle_c$ component of wavefunction while leaving the $|1\rangle_c$ component unchanged, thus creating the following state:

$$\frac{|\Omega\rangle_I |\Omega\rangle_{\mathcal{U}}}{\sqrt{|K_1|^2 + |K_2|^2}} \left(K_1 \frac{|\xi_1\rangle_E}{\mathcal{N}} |0\rangle_c + K_2 |\Omega\rangle_E |1\rangle_c \right) + (\text{terms orthogonal to } |\Omega\rangle_I |\Omega\rangle_{\mathcal{U}}). \quad (21)$$

Next, a controlled- F_2 operation is performed, this time controlled on qubit c being in the $|1\rangle$ state, thereby producing the following state:

$$|\Omega\rangle_I |\Omega\rangle_{\mathcal{U}} \left(\frac{K_1 |\xi_1\rangle_E |0\rangle_c + K_2 |\xi_2\rangle_E |1\rangle_c}{\mathcal{N} \sqrt{|K_1|^2 + |K_2|^2}} \right) + (\text{terms orthogonal to } |\Omega\rangle_I |\Omega\rangle_{\mathcal{U}}). \quad (22)$$

Finally, a Hadamard gate

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (23)$$

is applied to qubit c , causing the state of the quantum computer to become

$$\frac{1}{\mathcal{N} \sqrt{2}} \left(\frac{K_1 |\xi_1\rangle_E + K_2 |\xi_2\rangle_E}{\sqrt{|K_1|^2 + |K_2|^2}} \right) |\Omega\rangle_I |\Omega\rangle_{\mathcal{U}} |\Omega\rangle_c + (\text{terms orthogonal to } |\Omega\rangle_I |\Omega\rangle_{\mathcal{U}} |\Omega\rangle_c). \quad (24)$$

The $|\Omega\rangle_I |\Omega\rangle_{\mathcal{U}} |\Omega\rangle_c$ component of this output state encodes the amplitude \mathcal{A} , as can be seen by comparing to Eq. (1) and recalling that $|\xi_n\rangle$ encodes the colour factor C_n of diagram F_n via Eq. (14). It follows from Eq. (24) that if the I , \mathcal{U} , and c registers are measured, the probability of obtaining $|\Omega\rangle_I |\Omega\rangle_{\mathcal{U}} |\Omega\rangle_c$ will be

$$P(|\Omega\rangle_I |\Omega\rangle_{\mathcal{U}} |\Omega\rangle_c) = \frac{|K_1|^2 \langle \xi_1 | \xi_1 \rangle + K_1^* K_2 \langle \xi_1 | \xi_2 \rangle + K_2^* K_1 \langle \xi_2 | \xi_1 \rangle + |K_2|^2 \langle \xi_2 | \xi_2 \rangle}{2 (|K_1|^2 + |K_2|^2) \mathcal{N}^2}. \quad (25)$$

By examining Eqs. (1) and (14), it can be seen that the numerator of the RHS of Eq. (25) is in fact the squared amplitude $|\mathcal{A}|^2$, including all interference effects between the two diagrams.

The above procedure can be generalised to an arbitrary number N_D of Feynman diagrams F_0, \dots, F_{N_D-1} . To do so, the single ancilla qubit c is replaced by a larger ancilla register D of sufficient size to have at least N_D basis states. This means D contains $\lceil \log_2(N_D) \rceil$ qubits, with basis states $|0\rangle, |1\rangle, \dots, |2^{\lceil \log_2(N_D) \rceil} - 1\rangle$. For simplicity, let us for now continue to assume that each Feynman diagram has the same numbers of internal particles, so that the internal particle register I required for each diagram is the same. The R_c gate is replaced by a gate R_D defined such that

$$R_D |\Omega\rangle_D = \frac{1}{(\sum_m |K_m|^2)^{1/2}} \sum_{n=0}^{N_D-1} K_n |n\rangle_D. \quad (26)$$

In the special case where N_D is a power of 2 and where all $K_n = 1$, the R_D operation can be implemented by simply applying an H gate to each qubit of the D register, while in general an R_D gate achieving Eq. (26) can straight-forwardly be constructed by classically pre-computing all K_n at a chosen kinematic point and then employing a sequence of single-qubit rotations and CNOT operations. The two controlled- F_n operations in Fig. 2, are generalised to a sequence of N_D controlled- F_n operations, with the k^{th} operation applying F_k to registers E, I, \mathcal{U} , conditioned on the D register being in the state $|k\rangle$:

$$\prod_{k=0}^{N_D-1} \left[(|k\rangle\langle k|)_D F_n + (\mathbb{1}_D - (|k\rangle\langle k|)_D) \mathbb{1}_E \mathbb{1}_I \mathbb{1}_{\mathcal{U}} \right], \quad (27)$$

where $\mathbb{1}_r$ denotes the identity operator acting on register r . Finally, the H gate in Fig. 2 is replaced by a gate S_D acting on a general state $|\psi\rangle_D$ of the D register in the following way:

$$S_D |\psi\rangle_D = \frac{1}{\sqrt{N_D}} \left(\sum_{n=0}^{N_D-1} \langle n|\psi\rangle \right) |\Omega\rangle_D + (\text{terms orthogonal to } |\Omega\rangle_D). \quad (28)$$

In matrix form, the S_D gate is simply a $2^{\lceil \log_2(N_D) \rceil}$ -by- $2^{\lceil \log_2(N_D) \rceil}$ matrix in which the first N_D entries of the first row are $1/\sqrt{N_D}$, any remaining entries of the first row are zero, and the remaining rows can be arbitrary. For the special case where N_D is a power of 2, Eq. (28) can be achieved by simply applying an H gate to each qubit of the D register, whereas again the general case is implemented by a suitable sequence rotations and CNOT gates. Note that unlike R_D , the S_D gate is independent of kinematics.

The output state from running this circuit is

$$\frac{1}{\mathcal{N}_{N_D}} \left(\sum_{n=0}^{N_D-1} K_n |\xi_n\rangle_E \right) |\Omega\rangle_{I+\mathcal{U}+D} + (\text{terms orthogonal to } |\Omega\rangle_{I+\mathcal{U}+D}). \quad (29)$$

where we have defined $\mathcal{N}_{N_D} = \mathcal{N} \sqrt{N_D} \sqrt{\sum_m |K_m|^2}$. This state, analogously to Eq. (24), encodes the scattering amplitude from Eq. (1). It follows from Eq. (29) that if the I , \mathcal{U} , and D registers are measured, the state $|\Omega\rangle_I |\Omega\rangle_{\mathcal{U}} |\Omega\rangle_D$ will be obtained with probability

$$P(|\Omega\rangle_I |\Omega\rangle_{\mathcal{U}} |\Omega\rangle_D) = \frac{1}{\mathcal{N}_{N_D}^2} \left| \sum_n K_n |\xi_n\rangle_E \right|^2. \quad (30)$$

By examining Eqs. (1) and (14), it can be seen that up to an overall normalisation factor, the RHS of Eq. (30) is the squared amplitude $|\mathcal{A}|^2$ for the chosen kinematic point and spin configuration, with all interference effects between the diagrams included.

So far we have assumed that the number of internal gluons and closed quark lines does not vary between diagrams, so that all diagrams can be implemented using the same internal-particle register I . If this assumption does not hold, the above algorithm to sum and interfere Feynman diagrams can still be employed, but one must be aware that the normalisation factor \mathcal{N} varies with the number of internal particles in the diagram according to Eq. (15), which would introduce a diagram-dependent weighting in Eq. (29). This unphysical weighting can, however, easily be removed by re-scaling the diagram's kinematic factor K_n by the amount necessary to cancel it.

It is interesting to notice that the RHS of Eq. (30) does not necessarily have any particular scaling with the number of Feynman diagrams, since the growth of the denominator factor on

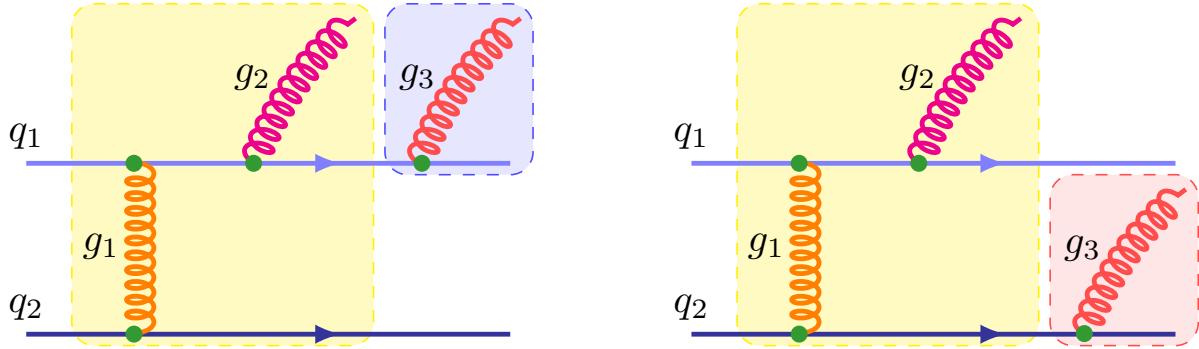


Figure 3: Feynman diagrams F_A (left) and F_B (right).

the RHS of Eq. (30) can get compensated, partly or entirely, by the growth of the numerator. Indeed, it is easy to see that if all the Feynman diagrams had identical values, the numerator and denominator would both scale as $|K|^2 N_D^2$, where K is the kinematic factor of each diagram, and so Eq. (30) would be invariant under changes of the number of diagrams. In an opposite extreme case where one diagram is non-zero and all remaining diagrams turn out to be zero, the probability in Eq. (30) would scale as $1/N_D$, and so the number of shots required to obtain the amplitude at a given level of precision would scale as N_D , which is the same scaling as a classical computation (assuming that neither the classical computation nor the quantum computation exploits the knowledge that most diagrams are zero). Even in this second extreme case, quantum amplitude estimation [55–58] techniques could produce a quadratic speedup. Realistic processes would lie in between these two cases, and so in general a quantum simulation would scale favourably with N_D in comparison to a classical computation of the diagrams.

The circuit depth is also expected to scale well with the number of Feynman diagrams, especially if one exploits shared structures between diagrams. In the present work we explore two ways to do this, firstly by factoring out shared sub-diagrams, which is illustrated in the example shortly to be discussed, and secondly by exploiting exchange symmetries, which is the subject of Sec. 4.

To validate our method for calculating interferences, we now apply it to two Feynman diagrams, F_A and F_B , which are shown in Fig. 3. For simplicity, we set both kinematic factors $K_A = K_B = 1$. The sum and interference of F_A and F_B can be obtained using the quantum circuit shown in Fig. 4. We employ a convenient optimisation to reduce the circuit depth: noting that the two diagrams in Fig. 3 share a sub-diagram, which is highlighted in yellow, the circuit in Fig. 4 factors the shared sub-diagram out of the controlled- F_n operations that are prescribed for generic pairs of diagrams in Fig. 2. The shared sub-diagram is therefore only implemented once in the circuit, and is not controlled by the c qubit. This circuit has been implemented in QISKit and run on a noiseless quantum emulator for 10^7 shots. By inverting Eq. (25) and inserting $K_1 = K_2 = 1$ as well as $\mathcal{N}^2 = N_c^2 (N_c^2 - 1)^4$, with $N_c = 3$, we can infer from the simulation that the squared sum of the diagrams, summed over colours, is:

$$|F_A + F_B|^2 = 4\mathcal{N}^2 P(|\Omega\rangle_I |\Omega\rangle_{\mathcal{U}} |\Omega\rangle_c) = 7.65 \pm 0.34. \quad (31)$$

This is consistent with the theoretical value for the squared sum of the Feynman diagrams with K_A and K_B both set to 1:

$$|F_A + F_B|^2 = \frac{2(N_c^6 - 3N_c^4 + 5N_c^2 - 3) T_F^4}{N_c^2} = \frac{22}{3} \approx 7.33. \quad (32)$$

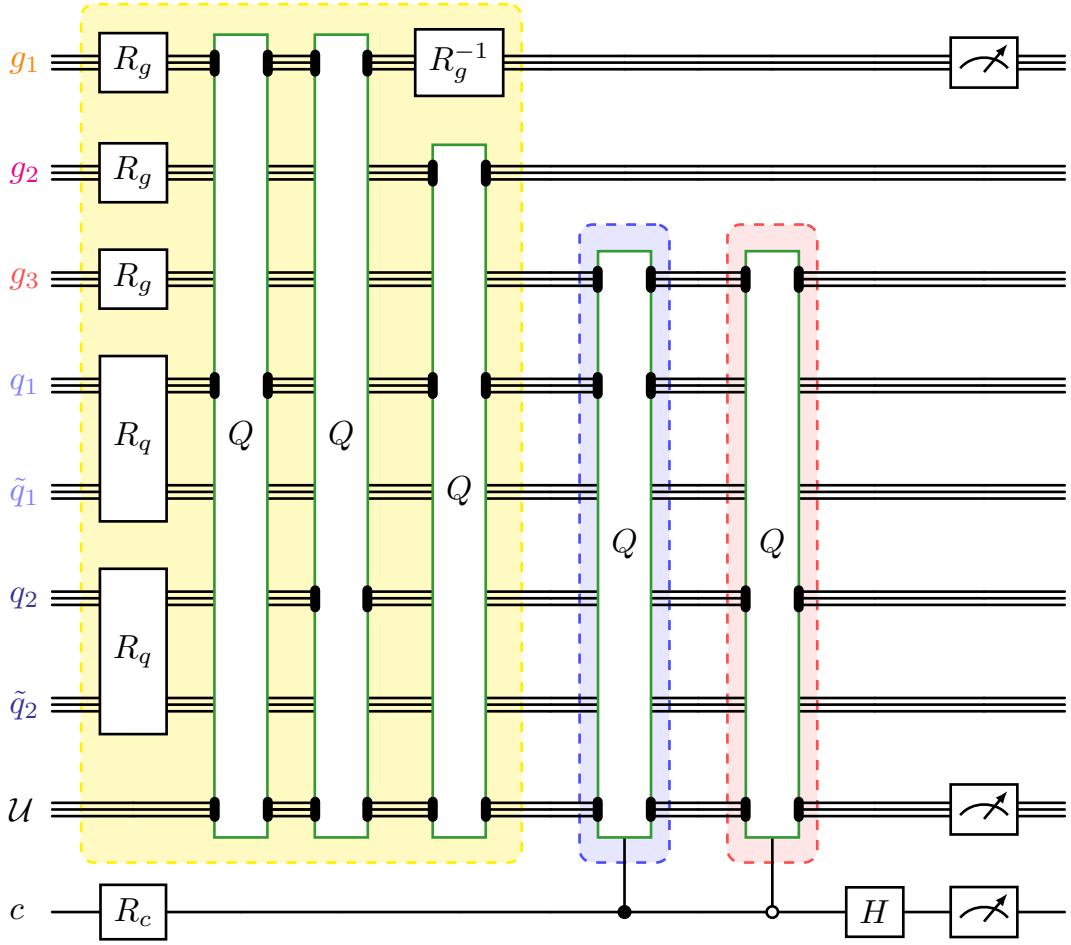


Figure 4: Quantum circuit to calculate the sum and interference of the two Feynman diagrams F_A and F_B shown in Fig. 3 and obtain their sum and interference. The parts shared between the two Feynman diagrams are implemented once, by the section of the circuit shaded in yellow. The part where the diagrams differ, *i.e.* the emission of g_3 , is implemented by the blue shaded section for diagram F_A and the red shaded section for diagram F_B . The output of the circuit encodes the fully-interfered squared sum $|F_A + F_B|^2$ of the two diagrams.

4 Feynman diagrams with identical particles

Having shown in Sec. 3 how to obtain the sums and interferences of generic Feynman diagrams, in this section we showcase the value of our methodology by presenting a particularly efficient way to obtain the sums and interferences of Feynman diagrams where some of the external particles are identical. In considering this class of processes, we find hints of a potential quantum advantage.

4.1 2-gluon example

As a simple example, let us start by considering in this sub-section the emission of two gluons from a single quark line. We will consider two of the Feynman diagrams contributing to this process, which are shown in Fig. 5. In principle we could directly apply the method from Sec. 3 to implement the sum (and hence the interference) of these diagrams. However, since the second

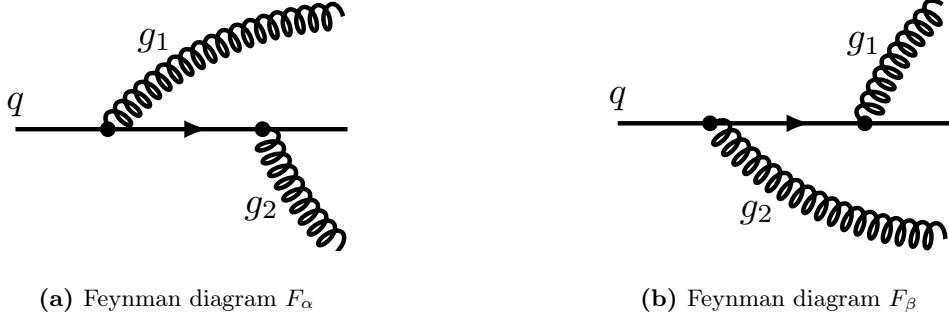


Figure 5: Feynman diagrams for two sequential gluon emissions from a quark line.

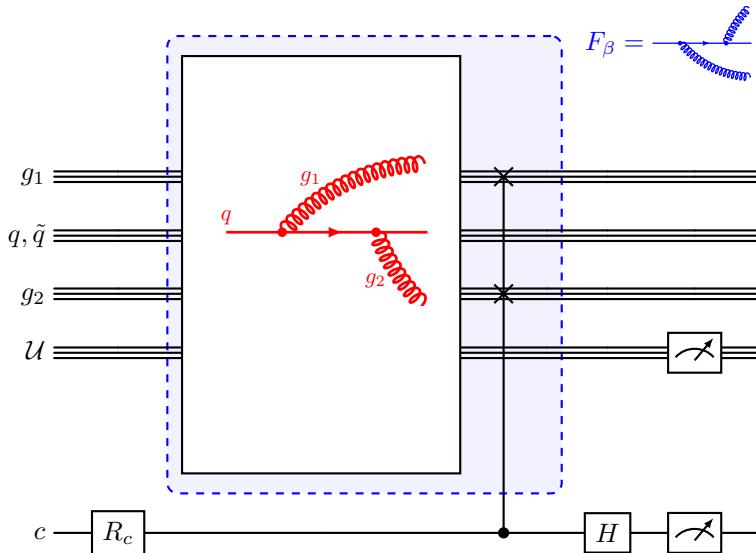


Figure 6: Efficient quantum circuit to simulate two Feynman diagrams F_α (red) and F_β (blue) and obtain their sum and interference. F_α is simulated directly whereas F_β is obtained from F_α by swapping the states of the two gluons g_1 and g_2 . Registers q and \tilde{q} encode the outgoing and incoming state of the quark line, respectively. The solid vertical line with two \times symbols represents a swap of the gluon states, controlled on qubit c being in the state $|1\rangle$. The output of the circuit encodes $|F_\alpha + F_\beta|^2$ with full interference effects.

diagram F_β in Fig. 5 is related to the first diagram F_α by simply relabelling the two gluons, a far more efficient implementation can be achieved, which is shown in Fig. 6 and will be explained step-by-step below. The key observation is that F_β can be simulated by implementing F_α followed by a swap of the two gluon registers. This means F_α can be factored out of the two controlled-swap operations required in Fig. 2 and performed without being controlled by the c qubit, so that only the swap operation needs to be controlled by the c qubit. Exploiting the exchange symmetry of the gluons in this way has several benefits: the resulting circuit contains only 2 Q gates instead of 4, and requires far fewer controlled operations, both of which properties reduce the circuit depth. By including more registers to encode the states of additional particles, the circuit in Fig. 6 could be used to calculate the sum and interference of any two Feynman diagrams in which the final state contains a pair of gluons.

We will now provide a step-by-step account of the detailed operation of the circuit in Fig. 6.

Just as in Sec. 3, the wavefunction is initially as shown in Eq. (19), although in this particular example there is no I register since the diagrams in Fig. 5 contain no internal gluons or closed quark loops and so the initial state of the quantum computer is simply:

$$|\Omega\rangle_E |\Omega\rangle_{\mathcal{U}} |\Omega\rangle_c. \quad (33)$$

For this example, the E register is composed of two registers g_1 and g_2 which represent the states of the two gluons, and a pair of registers q and \tilde{q} to represent, respectively, the fundamental and antifundamental $\mathfrak{su}(N_c)$ indices encoding the colour of the quark line, as specified in Sec. 2. We can therefore explicitly write the initial state of the E register as follows:

$$|\Omega\rangle_E = |\Omega\rangle_{g_1} |\Omega\rangle_{g_2} |\Omega\rangle_q |\Omega\rangle_{\tilde{q}}. \quad (34)$$

After applying the R_c gate from Eq. (18) with $K_1 = K_\alpha$ and $K_2 = K_\beta$, we obtain the state specified in Eq. (20), or more explicitly for the present example:

$$|\Omega\rangle_{g_1} |\Omega\rangle_{g_2} |\Omega\rangle_q |\Omega\rangle_{\tilde{q}} |\Omega\rangle_{\mathcal{U}} \left(\frac{K_\alpha |0\rangle_c + K_\beta |1\rangle_c}{\sqrt{|K_\alpha|^2 + |K_\beta|^2}} \right). \quad (35)$$

After simulating F_α , we obtain according to Eq. (13) the state

$$\frac{|\xi_\alpha\rangle_E}{\mathcal{N}} |\Omega\rangle_{\mathcal{U}} \left(\frac{K_\alpha |0\rangle_c + K_\beta |1\rangle_c}{\sqrt{|K_\alpha|^2 + |K_\beta|^2}} \right) + (\text{terms orthogonal to } |\Omega\rangle_{\mathcal{U}}), \quad (36)$$

where

$$|\xi_\alpha\rangle_E = \sum_{a,b,i,j,l} T_{il}^b T_{lj}^a |a\rangle_{g_1} |b\rangle_{g_2} |i\rangle_q |j\rangle_{\tilde{q}} \quad (37)$$

encodes the colour factor $C_\alpha = \sum_l T_{il}^b T_{lj}^a$ of F_α . We next apply a controlled-swap gate **CSWAP**, which in general acts on a control qubit q_0 and two equal-sized particle registers r_1 and r_2 in the following way:

$$\text{CSWAP} \left[\left(\alpha |0\rangle_{q_0} + \beta |1\rangle_{q_0} \right) |\psi_1\rangle_{r_1} |\psi_2\rangle_{r_2} \right] = \alpha |0\rangle_{q_0} |\psi_1\rangle_{r_1} |\psi_2\rangle_{r_2} + \beta |1\rangle_{q_0} |\psi_2\rangle_{r_1} |\psi_1\rangle_{r_2}. \quad (38)$$

Noting that swapping the state of the g_1 and g_2 registers in $|\xi_\alpha\rangle$ produces the state

$$|\xi_\beta\rangle_E = \sum_{a,b,i,j,l} T_{il}^b T_{lj}^a |b\rangle_{g_1} |a\rangle_{g_2} |i\rangle_q |j\rangle_{\tilde{q}} \quad (39)$$

$$= \sum_{a,b,i,j,l} T_{il}^a T_{lj}^b |a\rangle_{g_1} |b\rangle_{g_2} |i\rangle_q |j\rangle_{\tilde{q}}, \quad (40)$$

which encodes the colour factor $C_\beta = \sum_l T_{il}^a T_{lj}^b$ of the Feynman diagram F_β , it can be seen that applying the controlled-swap operation specified in Fig. 6 to the state in Eq. (36) produces the state

$$\frac{1}{\mathcal{N}} \sum_{a,b,i,j,l} \left(\frac{K_\alpha T_{il}^b T_{lj}^a |0\rangle_c + K_\beta T_{il}^a T_{lj}^b |1\rangle_c}{\sqrt{|K_\alpha|^2 + |K_\beta|^2}} \right) |a\rangle_{g_1} |b\rangle_{g_2} |i\rangle_q |j\rangle_{\tilde{q}} |\Omega\rangle_{\mathcal{U}} + (\text{terms orthogonal to } |\Omega\rangle_{\mathcal{U}}). \quad (41)$$

Lastly, applying the H gate defined in Eq. (23) to the c qubit, the final output state of the circuit in Fig. 6 is found to be

$$\frac{1}{\mathcal{N}\sqrt{2}} \sum_{a,b,i,j,l} \left(\frac{K_\alpha T_{il}^b T_{lj}^a + K_\beta T_{il}^a T_{lj}^b}{\sqrt{|K_\alpha|^2 + |K_\beta|^2}} \right) |a\rangle_{g_1} |b\rangle_{g_2} |i\rangle_q |j\rangle_{\tilde{q}} |\Omega\rangle_{\mathcal{U}} |\Omega\rangle_c + (\text{terms orthogonal to } |\Omega\rangle_{\mathcal{U}} |\Omega\rangle_c), \quad (42)$$

which encodes the sum $K_\alpha C_\alpha + K_\beta C_\beta$ of the two Feynman diagrams as desired. It follows from Eq. (42) that if the \mathcal{U} and c registers are then measured, the probability of obtaining $|\Omega\rangle_{\mathcal{U}} |\Omega\rangle_c$ is

$$P(|\Omega\rangle_{\mathcal{U}} |\Omega\rangle_c) = \frac{1}{2\mathcal{N}^2} \frac{\sum_{a,b} (|K_\alpha|^2 + |K_\beta|^2) \text{Tr}(T^a T^b T^b T^a) + 2\text{Re}(K_\alpha^* K_\beta) \text{Tr}(T^a T^b T^a T^b)}{|K_\alpha|^2 + |K_\beta|^2} \quad (43)$$

where the numerator is the squared sum $|F_\alpha + F_\beta|^2$ of the two Feynman diagrams, including all interference effects.

As a validation, we implemented this example using QISKit and ran 10^7 shots of the circuit on a noiseless quantum emulator, setting $K_\alpha = K_\beta = 1$ for simplicity. By inverting Eq. (43) and setting $\mathcal{N}^2 = N_c (N_c^2 - 1)^2$ where $N_c = 3$, our simulation gives the following estimate for the squared sum of the Feynman diagrams at the chosen kinematic point:

$$|F_\alpha + F_\beta|^2 = 4\mathcal{N}^2 P(|\Omega\rangle_{\mathcal{U}} |\Omega\rangle_c) = 9.357 \pm 0.027. \quad (44)$$

Noting that the theoretical values for the colour-summed colour parts of the squared Feynman diagrams and their interference read

$$|C_\alpha|^2 = |C_\beta|^2 = \frac{(N_c^2 - 1)^2}{4N_c} \quad \text{and} \quad C_\alpha C_\beta^* = \frac{1 - N_c^2}{4N_c}, \quad (45)$$

we see that the results of the simulation at the chosen kinematic point are well in agreement with the analytic expectation of

$$|F_\alpha + F_\beta|^2 = \frac{N_c^4 - 3N_c^2 + 2}{2N_c} = \frac{28}{3} \approx 9.333. \quad (46)$$

In general, such a swap-based technique can be employed to more efficiently implement the interference of Feynman diagrams whenever the external states of a process include a pair of identical particles.

4.2 Generalisation to many identical particles

The technique from Section 4.1 can be generalised to the case of N identical particles. Given a Feynman diagram F containing N identical external particles, there exists a set of $N!$ diagrams related to F by permutations of the identical particles. We focus on the general case where F itself has no graph symmetries under such permutations, and so the $N!$ permuted diagrams are mutually distinct. Remarkably, the sum (and hence interference) of these $N!$ diagrams can be obtained by implementing a single diagram followed by a small number N_{swaps} of **CSWAP** operations, where N_{swaps} will be quantified below after we have described our quantum circuit construction. The single c qubit in Fig. 6 is replaced by a larger register \mathcal{P} comprising N_{swaps} qubits, each acting as the control for a distinct **CSWAP**. In a procedure building in part on Ref. [59],

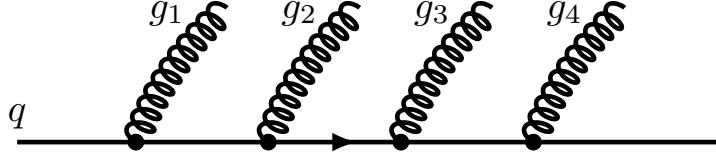


Figure 7: Example Feynman diagram for the sequential emission of four gluons from a quark line.

we apply **CSWAP** operations to the N registers encoding the states of the particles, these operations being applied in a layout that corresponds to an N -wire sorting network [60]. Since the swap operations form a sorting network, any particular permutation σ of the N particles can be generated by an appropriate subset of the swap operations, although often there will exist several distinct subsets that generate the same permutation. To ensure that each permutation is obtained exactly once, the R_c and H gates in Fig. 6 must be replaced by generalised gates that rotate the control qubits between the reference state $|\Omega\rangle$ and a superposition of $N!$ basis states judiciously chosen to ensure that the **CSWAP** operations in a particular sorting network generate all $N!$ permutations with no double counting. As in the previous sections, the kinematic factors K_n can be pre-computed and encoded in the first of the two generalised gates. Having implemented the circuit for F together with a sorting network and generalised rotations, the final state of the quantum computer will, similarly to Eq. (29), be

$$\frac{1}{N_D} \left(\sum_{\sigma} K_{\sigma} |\xi_{\sigma}\rangle_E \right) |\Omega\rangle_{I+U+P} + (\text{terms orthogonal to } |\Omega\rangle_{I+U+P}), \quad (47)$$

where $N_D = N!$, and where K_{σ} and $|\xi_{\sigma}\rangle$ encode respectively the kinematic and colour factor of the Feynman diagram F_{σ} obtained by applying the permutation σ to the N identical particles in the original Feynman diagram F .

Sorting networks are a well-studied topic in computer science and so there are many choices of networks available. Formally, an AKS network [61] could allow the permutations to be performed with $N_{\text{swaps}} \sim \mathcal{O}(N \log N)$ **CSWAP** operations arranged in a circuit of depth $\mathcal{O}(\log(N))$, which is doubly-exponentially shorter than implementing each of the $N!$ diagrams sequentially. AKS networks have a large constant overhead, but more practical algorithms exist requiring a depth of $\mathcal{O}(N \log(N))$ operations, which is still exponentially shorter than implementing each diagram sequentially. For small values of N , optimal circuits are explicitly known. For example, given a circuit to implement a single Feynman diagram containing $N = 10$ identical particles, one can use 31 **CSWAP** operations, parallelised over a depth of 7, to generate the sum and interference of all $10! = 3,628,800$ permuted diagrams. Depth-optimal networks are known for $N \leq 17$ (e.g. depth 10 for $N = 17$ as shown in Ref. [62]) and size-optimal networks are known for $N \leq 12$ (e.g. $N_{\text{swaps}} = 39$ for $N = 12$ as shown in Ref. [63]).

As a concrete example of our procedure for generating and interfering permuted diagrams, in Fig. 8 we explicitly show a quantum circuit for calculating the squared sum, with full interference effects, of all $(4!) = 24$ Feynman diagrams corresponding to the sequential emission of 4 gluons from a quark line. One representative diagram, shown in Fig. 7, is directly implemented in the circuit, whereas the remaining 23 diagrams are generated by the 5-swap sorting network that follows it. The sorting network has depth 3 (since the first two swaps can be performed simultaneously, as can the next two) and size $N_{\text{swaps}} = 5$, both of which are optimal when the number of identical particles is 4. The generalised rotation gates have been labelled R_P and R'_P .

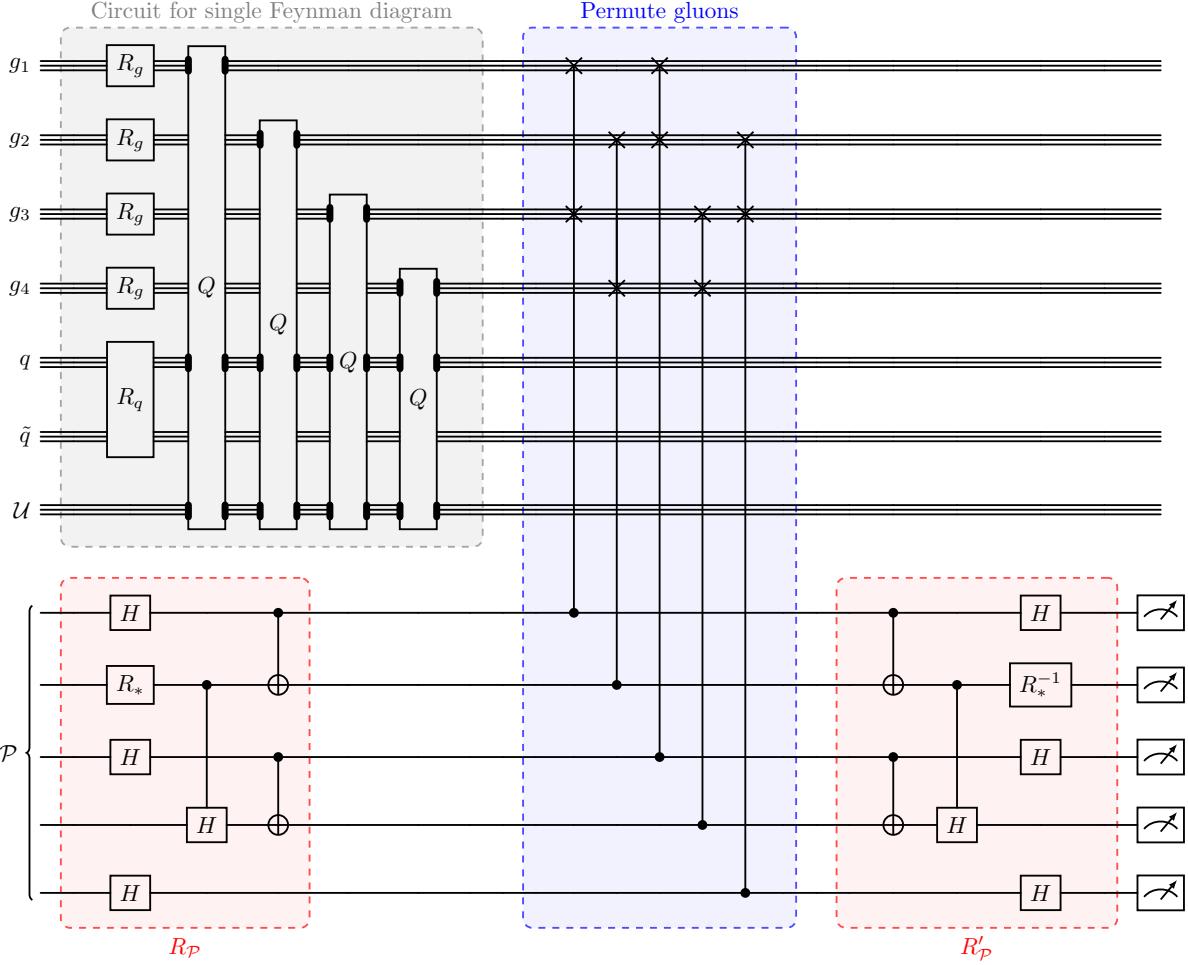


Figure 8: Quantum circuit to obtain the sum and interference of 24 Feynman diagrams describing the sequential emission of 4 gluons from a quark line. The part highlighted in grey implements a single representative Feynman diagram shown in Fig. 7. The parts highlighted in red and blue generate all 24 permutations of the original diagram. The fully-interfered squared sum $|\mathcal{A}|^2 = |\sum_{\sigma} F_{\sigma}|^2$ of the diagrams is obtained from the output of the circuit.

Their definition includes a single-qubit rotation gate

$$R_* = \begin{pmatrix} \sqrt{1/3} & -\sqrt{2/3} \\ \sqrt{2/3} & \sqrt{1/3} \end{pmatrix}. \quad (48)$$

For simplicity, all 24 kinematic factors have been set to be 1. If desired, R_P could instead be replaced by a rotation gate that incorporates kinematic factors pre-computed at a chosen kinematic point. In Sec. 5 we will implement this circuit on a present-day quantum computer in order to validate our techniques.

It is helpful to distinguish the way the permutation symmetries of diagrams are exploited in this quantum algorithm in comparison to the way they are exploited in classical computing algorithms. Classical calculations can be performed either analytically or numerically. In analytic

calculations, it is straight-forward to exploit permutation symmetries by calculating a symbolic expression for one Feynman diagram and then permuting its variables to generate other related diagrams. However, analytically calculating Feynman diagrams requires symbolic manipulations that can be computationally expensive and so direct numerical calculations are often preferable. In such numerical calculations, however, each of the permuted diagrams must be independently evaluated at the desired phase-space points, and so the permutation symmetries between the diagrams are of limited benefit. Our direct quantum-computer-based simulation of the Feynman diagrams thus provides additional capabilities to exploit symmetries, going beyond what can be achieved in a classical numerical simulation. For processes with several identical external particles, such as in the production of multiple hard jets, the factorial reduction in the number of Feynman diagrams needing to be implemented could provide a sizeable benefit.

5 Validation on physical quantum devices

In Sections 3 and 4 we included validations of the techniques presented in those sections, by implementing various examples on classically-emulated quantum computers. In addition, we have also performed validations on a physical quantum computer, as we will present in this section. Specifically, we ran our two most complicated examples on the 56-qubit trapped-ion quantum computer Quantinuum H2-1. Details about the properties and performance of the device can be found in Ref. [2]. Quantum computers available today are constrained primarily by noise and decoherence, thus limiting the circuit depths that can be achieved. In order to operate within these constraints, we perform our physical-device demonstrations using a toy model of QCD where $N_c = 2$. The Lie algebra of the gauge group is thus $\mathfrak{su}(2)$, whose fundamental-representation generators T^a (with $a \in \{1, 2, 3\}$) can be chosen to be unitary³ by defining $T^a = \sigma^a$. Here σ^a are the Pauli matrices, which are readily available as single-qubit gates on many quantum devices. In $\mathfrak{su}(2)$ we can represent fundamental, antifundamental, and adjoint indices by using registers containing, respectively, 1, 1, and 2 qubits. We can then construct an $\mathfrak{su}(2)$ version of the Q gate, denoted $Q_{\mathfrak{su}(2)}$, to act on a gluon register g in a colour basis state $|a\rangle$ and a quark register q in a colour basis state $|k\rangle$ in the following way:

$$Q_{\mathfrak{su}(2)} |a\rangle_g |j\rangle_q = \sum_{i=1}^2 T_{ij}^a |a\rangle_g |i\rangle_q, \quad (49)$$

where T_{ij}^a now denotes the components of the fundamental-representation generators of $\mathfrak{su}(2)$ instead of $\mathfrak{su}(3)$. Equation (49) is simpler than Eq. (3) because here no unitarisation register is required. An explicit construction of $Q_{\mathfrak{su}(2)}$ is shown in Fig. 9. We also constructed $\mathfrak{su}(2)$ versions of the R_g and R_q gates.

Using this setup, we first implemented the two-diagram interference example from Figs. 3 and 4. We set the values of the kinematic factors to be $K_A = K_B = 1$. The circuit was compiled using PYTKET [64] version 1.4.0, extension `pytket-quantinuum 0.44.0`, with setting `optimization_level=0`. By running the circuit for 1,000 shots on the device H2-1, we obtained the result

$$|F_A + F_B|^2 = 1.099 \pm 0.075. \quad (50)$$

³This choice would ordinarily imply $T_F = 2$ instead of the conventional $T_F = \frac{1}{2}$ used in the previous sections, but we have corrected for the differences in convention in order to consistently use $T_F = \frac{1}{2}$ throughout this article.

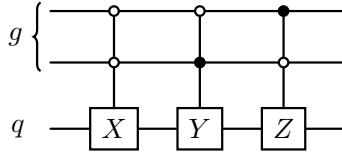


Figure 9: Circuit diagram for the $Q_{\mathfrak{su}(2)}$ gate acting on a 2-qubit gluon register g and 1-qubit quark register q . Boxes labelled X , Y , and Z represent the three elementary Pauli gates. White and black circles indicate, respectively, controlling on a $|0\rangle$ or $|1\rangle$ state.

This is consistent with the analytic expectation

$$|F_A + F_B|^2 = \frac{2(N_c^6 - 3N_c^4 + 5N_c^2 - 3)T_F^4}{N_c^2} = \frac{33}{32} \approx 1.031, \quad (51)$$

where we have set $N_c = 2$ and $T_F = \frac{1}{2}$.

To further validate and demonstrate our techniques, we also implemented the 24-diagram interference from Figs. 7 and 8. As in the previous example, we work in $\mathfrak{su}(2)$ and set all kinematic factors $K_n = 1$. We compiled the circuit using the PYTKET version 2.4.1, extension `pytket-quantinuum` 0.48.0, with the setting `optimization_level=0`. By running the circuit for 1,000 shots on the device H2-1, we obtained the result

$$\left| \sum_{m=1}^{24} F_{\sigma_m} \right|^2 = 11.12 \pm 1.42, \quad (52)$$

where $F_{\sigma_1}, F_{\sigma_2}, \dots, F_{\sigma_{24}}$ are the 24 Feynman diagrams obtained by permuting the ordering of the gluon emissions in Fig. 7. This result is consistent with the analytic expectation of

$$\left| \sum_{m=1}^{24} F_{\sigma_m} \right|^2 = \frac{24(N_c^8 - N_c^6 - 24N_c^2 + 24)T_F^4}{N_c^4} = \frac{45}{4} = 11.25, \quad (53)$$

where we have set $N_c = 2$ and $T_F = \frac{1}{2}$.

6 Conclusion

High-energy physics presents a promising domain of application for quantum computers. In this article we have presented quantum computing techniques to simulate scattering amplitudes in perturbative QCD, which describes the high-energy regime of QCD probed by colliders such as the LHC. We first showed how the colour parts of individual Feynman diagrams can be encoded into the wavefunction of a quantum computer. We then showed how the sum of multiple Feynman diagrams can be implemented, providing a natural way to obtain their interferences. Our methods are well suited to exploiting relations between Feynman diagrams, which we explored in two ways: firstly with diagrams sharing a common sub-diagram, and secondly with diagrams related to one another by permutations of the external particles. The latter case shows the first hints of a potential quantum advantage, *i.e.* indications that quantum computers could outperform classical computers for such calculations. We validated our techniques by building simulation circuits to obtain the fully-interfered squared sums of Feynman diagrams for several example

cases. The circuits were run on quantum emulators as well as a state-of-the-art 56-qubit trapped-ion quantum computer, and the results were consistent with analytic expectations.

This work opens up several possibilities to run more ambitious applications on present and future physical quantum computers. This would benefit from tailoring the fundamental implementation of the Q and G gates to suit the architectures of specific present-day devices and could also incorporate a direct quantum simulation of the kinematic parts of the amplitudes. At the same time, this work also constitutes a conceptual milestone for further developments, with potential future applications ranging from improved parton-shower algorithms to a quantum-accelerated Monte Carlo simulation of scattering cross-sections.

Acknowledgements

We thank Zohreh Davoudi, Aida El-Khadra, and Fernando Febres Cordero for helpful discussions. The work of HC was supported by the U.S. Department of Energy under grant DE-SC0010102. MP acknowledges support by the German Research Foundation (DFG) through the Research Training Group RTG2044. This research was supported by the Munich Institute for Astro-, Particle and BioPhysics (MIAPbP), which is funded by the DFG under Germany’s Excellence Strategy - EXC-2094 - 390783311. This work was performed in part at the Aspen Center for Physics, which is supported by National Science Foundation grant PHY-2210452. This research was supported in part by grant NSF PHY-2309135 to the Kavli Institute for Theoretical Physics (KITP). This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725. The Feynman diagrams were produced using the `TIKZ` package [65]. The circuit diagrams were produced using the `QUANTIKZ` package [66].

References

- [1] A. Huss, J. Huston, S. Jones, M. Pellen, and R. Röntsch, *Les Houches 2023 – Physics at TeV Colliders: Report on the Standard Model Precision Wishlist*. [arXiv:2504.06689 \[hep-ph\]](https://arxiv.org/abs/2504.06689).
- [2] S. A. e. a. Moses, *A Race-Track Trapped-Ion Quantum Processor*. Phys. Rev. X **13** (Dec, 2023) 041052.
- [3] Y. Kim *et al.*, *Evidence for the utility of quantum computing before fault tolerance*. Nature **618** (2023) no. 7965, 500–505.
- [4] **Google Quantum AI and Collaborators** Collaboration, R. Acharya *et al.*, *Quantum error correction below the surface code threshold*. Nature **638** (2025) no. 8052, 920–926, [arXiv:2408.13687 \[quant-ph\]](https://arxiv.org/abs/2408.13687).
- [5] B. Grinkemeyer, E. Guardado-Sánchez, I. Dimitrova, D. Shchepanovich, G. E. Mandopoulou, J. Borregaard, V. Vuletić, and M. D. Lukin, *Error-detected quantum operations with neutral atoms mediated by an optical cavity*. Science **387** (2025) no. 6740, adr7075, [arXiv:2410.10787 \[quant-ph\]](https://arxiv.org/abs/2410.10787).
- [6] H. Aghaee Rad *et al.*, *Scaling and networking a modular photonic quantum computer*. Nature **638** (2025) no. 8052, 912–919.

- [7] M. Larsen, J. Bourassa, S. Kocsis, J. Tasker, R. Chadwick, C. González-Arciniegas, J. Hastrup, C. Lopetegui-González, F. Miatio, A. Motamed, *et al.*, *Integrated photonic source of Gottesman–Kitaev–Preskill qubits*. *Nature* (2025) 1–5.
- [8] 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.
- [9] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, *Quantum computational chemistry*. *Rev. Mod. Phys.* **92** (2020) 015003.
- [10] I. M. Georgescu, S. Ashhab, and F. Nori, *Quantum simulation*. *Rev. Mod. Phys.* **86** (2014) 153–185.
- [11] 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.
- [12] Z.-Y. Zhou, G.-X. Su, J. C. Halimeh, R. Ott, H. Sun, P. Hauke, B. Yang, Z.-S. Yuan, J. Berges, and J.-W. Pan, *Thermalization dynamics of a gauge theory on a quantum simulator*. *Science* **377** (2022) no. 6603, abl6277, [arXiv:2107.13563 \[cond-mat.quant-gas\]](https://arxiv.org/abs/2107.13563).
- [13] A. Mallick, M. Lewenstein, J. Zakrzewski, and M. Płodzień, *String breaking dynamics in Ising chain with local vibrations*. [arXiv:2501.00604 \[quant-ph\]](https://arxiv.org/abs/2501.00604).
- [14] S. P. Jordan, K. S. M. Lee, and J. Preskill, *Quantum Computation of Scattering in Scalar Quantum Field Theories*. *Quant. Inf. Comput.* **14** (2014) 1014–1080, [arXiv:1112.4833 \[hep-th\]](https://arxiv.org/abs/1112.4833).
- [15] S. P. Jordan, K. S. M. Lee, and J. Preskill, *Quantum Algorithms for Quantum Field Theories*. *Science* **336** (2012) 1130–1133, [arXiv:1111.3633 \[quant-ph\]](https://arxiv.org/abs/1111.3633).
- [16] E. A. Martinez *et al.*, *Real-time dynamics of lattice gauge theories with a few-qubit quantum computer*. *Nature* **534** (2016) 516–519, [arXiv:1605.04570 \[quant-ph\]](https://arxiv.org/abs/1605.04570).
- [17] N. Klco and M. J. Savage, *Digitization of scalar fields for quantum computing*. *Phys. Rev. A* **99** (2019) no. 5, 052335, [arXiv:1808.10378 \[quant-ph\]](https://arxiv.org/abs/1808.10378).
- [18] M. C. Bañuls *et al.*, *Simulating Lattice Gauge Theories within Quantum Technologies*. *Eur. Phys. J. D* **74** (2020) no. 8, 165, [arXiv:1911.00003 \[quant-ph\]](https://arxiv.org/abs/1911.00003).
- [19] Z. Davoudi, I. Raychowdhury, and A. Shaw, *Search for efficient formulations for Hamiltonian simulation of non-Abelian lattice gauge theories*. *Phys. Rev. D* **104** (2021) no. 7, 074505, [arXiv:2009.11802 \[hep-lat\]](https://arxiv.org/abs/2009.11802).
- [20] 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 \[quant-ph\]](https://arxiv.org/abs/2107.04769).

- [21] M. Carena, H. Lamm, Y.-Y. Li, and W. Liu, *Improved Hamiltonians for Quantum Simulations of Gauge Theories*. Phys. Rev. Lett. **129** (2022) no. 5, 051601, arXiv:2203.02823 [hep-lat].
- [22] M. Rigobello, G. Magnifico, P. Silvi, and S. Montangero, *Hadrons in (1+1)D Hamiltonian hardcore lattice QCD*. arXiv:2308.04488 [hep-lat].
- [23] N. A. Zemlevskiy, *Scalable Quantum Simulations of Scattering in Scalar Field Theory on 120 Qubits*. arXiv:2411.02486 [quant-ph].
- [24] L. W. Anderson, M. Kiffner, T. O’Leary, J. Crain, and D. Jaksch, *Solving lattice gauge theories using the quantum Krylov algorithm and qubitization*. Quantum **9** (2025) 1669, arXiv:2403.08859 [quant-ph].
- [25] T. A. Cochran *et al.*, *Visualizing dynamics of charges and strings in (2 + 1)D lattice gauge theories*. Nature **642** (2025) no. 8067, 315–320, arXiv:2409.17142 [quant-ph].
- [26] D. Gonzalez-Cuadra *et al.*, *Observation of string breaking on a (2 + 1)D Rydberg quantum simulator*. Nature **642** (2025) no. 8067, 321–326, arXiv:2410.16558 [quant-ph].
- [27] H. F. Trotter, *On the product of semi-groups of operators*. Proceedings of the American Mathematical Society **10** (1959) no. 4, 545–551.
- [28] M. Suzuki, *Generalized Trotter’s formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems*. Communications in Mathematical Physics **51** (1976) no. 2, 183–190.
- [29] H. A. Chawdhry and M. Pellen, *Quantum simulation of colour in perturbative quantum chromodynamics*. SciPost Phys. **15** (2023) no. 5, 205, arXiv:2303.04818 [hep-ph].
- [30] C. W. Bauer *et al.*, *Quantum Simulation for High-Energy Physics*. PRX Quantum **4** (2023) no. 2, 027001, arXiv:2204.03381 [quant-ph].
- [31] H. M. Gray and K. Terashi, *Quantum Computing Applications in Future Colliders*. Front. in Phys. **10** (2022) 864823.
- [32] 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].
- [33] C. Brown, M. Spannowsky, A. Tapper, S. Williams, and I. Xiotidis, *Quantum pathways for charged track finding in high-energy collisions*. Front. Artif. Intell. **7** (2024) 1339785, arXiv:2311.00766 [hep-ph].
- [34] A. Di Meglio *et al.*, *Quantum Computing for High-Energy Physics: State of the Art and Challenges*. PRX Quantum **5** (2024) no. 3, 037001, arXiv:2307.03236 [quant-ph].
- [35] 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].
- [36] J. J. M. de Lejarza, M. Grossi, L. Cieri, and G. Rodrigo, “*Quantum Fourier Iterative Amplitude Estimation*,” in *2023 International Conference on Quantum Computing and Engineering*. IEEE, 5, 2023. arXiv:2305.01686 [quant-ph].

- [37] J. M. Cruz-Martinez, M. Robbiati, and S. Carrazza, *Multi-variable integration with a variational quantum circuit*. *Quantum Sci. Technol.* **9** (2024) no. 3, 035053, [arXiv:2308.05657 \[quant-ph\]](#).
- [38] J. J. M. de Lejarza, D. F. Rentería-Estrada, M. Grossi, and G. Rodrigo, *Quantum integration of decay rates at second order in perturbation theory*. *Quantum Sci. Technol.* **10** (2025) no. 2, 025026, [arXiv:2409.12236 \[quant-ph\]](#).
- [39] I. Williams and M. Pellen, *A general approach to quantum integration of cross sections in high-energy physics*. [arXiv:2502.14647 \[quant-ph\]](#).
- [40] 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 \[quant-ph\]](#).
- [41] M. Robbiati, J. M. Cruz-Martinez, and S. Carrazza, *Determining probability density functions with adiabatic quantum computing*. *Quantum Machine Intelligence* **7** (2025) no. 1, 5, [arXiv:2303.11346 \[quant-ph\]](#).
- [42] E. Lee, S. Lee, and S. Kim, *Quantum walk based Monte Carlo simulation for photon interaction cross sections*. *Phys. Rev. D* **111** (2025) no. 11, 116001, [arXiv:2502.14374 \[quant-ph\]](#).
- [43] 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 \[hep-ph\]](#).
- [44] 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 \[hep-ph\]](#).
- [45] 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 \[hep-ph\]](#).
- [46] P. Deliyannis, J. Sud, D. Chamaki, Z. Webb-Mack, C. W. Bauer, and B. Nachman, *Improving quantum simulation efficiency of final state radiation with dynamic quantum circuits*. *Phys. Rev. D* **106** (2022) no. 3, 036007, [arXiv:2203.10018 \[hep-ph\]](#).
- [47] G. Gustafson, S. Prestel, M. Spannowsky, and S. Williams, *Collider events on a quantum computer*. *JHEP* **11** (2022) 035, [arXiv:2207.10694 \[hep-ph\]](#).
- [48] C. W. Bauer, S. Chigusa, and M. Yamazaki, *Quantum parton shower with kinematics*. *Phys. Rev. A* **109** (2024) no. 3, 032432, [arXiv:2310.19881 \[hep-ph\]](#).
- [49] 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 \[hep-ph\]](#).
- [50] 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 \[hep-ph\]](#).

- [51] 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 \[hep-ph\]](https://arxiv.org/abs/2105.08703).
- [52] 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 directed acyclic graphs*. *Phys. Rev. D* **108** (2023) no. 9, 096035, [arXiv:2210.13240 \[hep-ph\]](https://arxiv.org/abs/2210.13240).
- [53] J. J. M. de Lejarza, L. Cieri, M. Grossi, S. Vallecorsa, and G. Rodrigo, *Loop Feynman integration on a quantum computer*. *Phys. Rev. D* **110** (2024) no. 7, 074031, [arXiv:2401.03023 \[hep-ph\]](https://arxiv.org/abs/2401.03023).
- [54] S. Ramírez-Uribe, A. E. Rentería-Olivo, and G. Rodrigo, *Quantum querying based on multicontrolled Toffoli gates for causal Feynman loop configurations and directed acyclic graphs*. [arXiv:2404.03544 \[quant-ph\]](https://arxiv.org/abs/2404.03544).
- [55] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, *Quantum Amplitude Amplification and Estimation*. *Quantum Computation and Information* **305** (2002) , [arXiv:quant-ph/0005055 \[quant-ph\]](https://arxiv.org/abs/quant-ph/0005055).
- [56] D. Grinko, J. Gacon, C. Zoufal, and S. Woerner, *Iterative Quantum Amplitude Estimation*. *npj Quantum Inf* **7** (2021) no. 52, , [arXiv:1912.05559 \[quant-ph\]](https://arxiv.org/abs/1912.05559).
- [57] 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 \[quant-ph\]](https://arxiv.org/abs/1904.10246).
- [58] K. Nakaji, *Faster Amplitude Estimation*. *Quantum Information & Computation* **20** (2020) no. 13&14, , [arXiv:2003.02417 \[quant-ph\]](https://arxiv.org/abs/2003.02417).
- [59] D. S. Abrams and S. Lloyd, *Simulation of many body Fermi systems on a universal quantum computer*. *Phys. Rev. Lett.* **79** (1997) 2586–2589, [arXiv:quant-ph/9703054](https://arxiv.org/abs/quant-ph/9703054).
- [60] R. J. Nelson and D. G. O’Connor, *Sorting Systems with N-Line Sorting Switch*, February, 1957.
- [61] M. Ajtai, J. Komlós, and E. Szemerédi, “An $o(n \log n)$ sorting network,” in *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, STOC ’83, p. 1–9. Association for Computing Machinery, New York, NY, USA, 1983. <https://doi.org/10.1145/800061.808726>.
- [62] M. Codish, L. Cruz-Filipe, T. Ehlers, M. Müller, and P. Schneider-Kamp, *Sorting networks: To the end and back again*. *Journal of Computer and System Sciences* **104** (2019) 184–201. <https://www.sciencedirect.com/science/article/pii/S0022000016300162>. Language and Automata Theory and Applications - LATA 2015.
- [63] J. Harder, *An Answer to the Bose-Nelson Sorting Problem for 11 and 12 Channels*, 2022. [arXiv:2012.04400 \[cs.DS\]](https://arxiv.org/abs/2012.04400), <https://arxiv.org/abs/2012.04400>.

- [64] S. Sivarajah, S. Dilkes, A. Cowtan, W. Simmons, A. Edgington, and R. Duncan, $t/ket\rangle$: a retargetable compiler for NISQ devices. *Quantum Sci. Technol.* **6** (2020) no. 1, 014003, [arXiv:2003.10611 \[quant-ph\]](https://arxiv.org/abs/2003.10611).
- [65] T. Tantau, *The TikZ and PGF Packages: Manual for version 3.1.10*, June, 2025. <https://github.com/pgf-tikz/pgf>.
- [66] A. Kay, *Tutorial on the Quantikz Package*. [arXiv:1809.03842 \[quant-ph\]](https://arxiv.org/abs/1809.03842).