

Quantum Science and Technology



PAPER

A general approach to quantum integration of cross sections in high-energy physics

OPEN ACCESS

RECEIVED
5 March 2025

REVISED
18 July 2025

ACCEPTED FOR PUBLICATION
4 August 2025

PUBLISHED
14 August 2025

Original Content from
this work may be used
under the terms of the
[Creative Commons
Attribution 4.0 licence](#).

Any further distribution
of this work must
maintain attribution to
the author(s) and the title
of the work, journal
citation and DOI.



Ifan Williams^{1,*} and Mathieu Pellen²

¹ Quantinuum, Terrington House, 13–15 Hills Road, CB2 1NL Cambridge, United Kingdom

² Universität Freiburg, Physikalisches Institut, Hermann-Herder-Str. 3, 79104 Freiburg, Germany

* Author to whom any correspondence should be addressed.

E-mail: ifan.williams@quantinuum.com and mathieu.pellen@physik.uni-freiburg.de

Keywords: quantum Monte Carlo integration (QMCI), quantum computing (QC), quantum algorithms, high-energy physics (HEP), computational physics

Abstract

We present universal *building blocks* for the quantum integration of generic cross sections in high-energy physics. We make use of Fourier quantum Monte Carlo integration (MCI) as implemented in QUANTINUUM's Quantum MCI engine to provide an extendable methodology for generating efficient circuits that can implement generic cross-section calculations, providing a quadratic speed-up in root mean-squared error convergence with respect to classical MCI. We focus on a concrete example of a $1 \rightarrow 3$ decay process to illustrate our work.

1. Introduction

Research in high-energy physics (HEP) aims at exploring fundamental physics at tiny scales. To probe these small scales, elementary particles are collided with very high energies in order to produce other particles. The prime example of such an experiment is the Large Hadron Collider (LHC), where protons are currently collided at a center-of-mass (COM) energy of $\sqrt{s} = 13.6$ TeV. In this way, properties of known particles can be studied and new particles, such as the Higgs boson [1, 2], can be discovered.

In order to extract useful information from experimental data, theoretical predictions are needed for comparison. At the heart of theoretical predictions is the concept of a *cross section*. Cross sections relate to the probability of a certain scattering process occurring in some collider experiment. This means that all theoretical predictions are obtained through the computation of cross sections, which consist of either analytical or numerical integration. As explained later, the method of choice for integrating a cross section in HEP is numerical Monte Carlo (MC) integration (MCI). The computational resources needed to provide theoretical predictions for collider experiments are enormous; for the LHC, they are of the order of billions of CPU hours per year [3–5].

It is therefore particularly interesting to look at whether quantum processing units can provide an alternative to CPU or GPU technology for computing theoretical predictions in HEP. In the last few years, several works have been devoted to exploring the potential of quantum computing for computing theoretical predictions in perturbative theory [6–17]. The quantum analogue of classical MCI is known as *Quantum MCI* (QMCI) [18, 19], and this quantum algorithm utilises the *Quantum Amplitude Estimation* [20] (QAE) algorithm as a key subroutine, providing a quadratic speed-up in root mean-squared error (RMSE) convergence with respect to classical MCI.

In [21], Mathieu Pellen (M.P.) investigated the use of QMCI to integrate elementary scattering processes. In this reference, a variant of the original QAE algorithm was used, namely *Iterative QAE* [22] (IQAE). This seminal work focused on simplified versions of scattering cross sections and explored both one- and two-dimensional integrals. Following this, [23, 24] explored the same idea but using *Quantum Fourier Iterative Amplitude Estimation* [25] for the quantum integration, which relies on the *Fourier QMCI* [19] (FQMCI) method (described later in section 2.3). In [25], a small-scale (small number of qubits) parametrised quantum circuit (PQC) is trained to prepare a quantum state representing a specific cross-section integrand, and decompose this state into its Fourier series. Each term in the Fourier series is

then integrated separately using QAE, in the same manner as in FQMCI. A potential drawback of this method is that it may not scale efficiently to systems with larger numbers of qubits (as would be required for multi-dimensional integrands to be calculated, for example), due to the trainability issues associated with variational methods in quantum machine learning [26–31]. Indeed, preparing arbitrary probability distributions on a quantum computer is thought to be computationally hard in the general case, and the topic remains an active field of research.

In the present work, we go significantly beyond what has been done to date by presenting a methodology for the computation of generic cross sections in HEP, applicable to arbitrary dimensionality. In particular, the method of quantum integration makes use of the FQMCI method implemented in QUANTINUUM’s QMCI engine [32] (developed in part by Ifan Williams [I.W.]). While research using the QMCI engine has so far been focused on financial applications [33], it is particularly interesting to consider other potential fields such as HEP, where the integrals studied possess different challenges to those in finance e.g. high dimensionality, more complex integrands, and non-trivial integral limits.

More concretely, in this article we investigate the computation of one- and two-dimensional cross-section integrations for scattering processes using the QMCI engine when run on a noiseless state-vector simulator. We demonstrate how one can decompose the integrand for an arbitrary, multi-dimensional cross section into products of constituent *building blocks*, which are single-variable terms. We describe methods for implementing these individual building blocks on a quantum computer, using the in-built functionality of the QMCI engine. We also discuss how kinematical constraints can be applied to the integrals using this in-built functionality. Our findings demonstrate that this important future application of quantum computing is likely to require fully fault-tolerant hardware before it becomes practically useful.

In section 2, we begin by defining the problem and explain why numerical MCI is the preferred method for classical cross-section integration. After this, we introduce FQMCI, QUANTINUUM’s QMCI engine, and discuss how we can use the engine to perform QMCI for generic cross-section calculations. We then discuss in detail some potential state-preparation methods for generating relativistic Breit–Wigner (BW) distributions—one of the key building blocks discussed—in section 3. Some example applications demonstrating the applicability of our methodology are then discussed in section 4. Finally, section 5 contains a discussion of the work, an outlook, and concluding remarks.

2. Definition of the problem and approaches

2.1. Cross sections in HEP

The cross section, typically denoted by σ , is a central concept in HEP. It relates to the probability of a given scattering process $a + b \rightarrow c + d + \dots$ occurring, where the letters represent elementary particles. This value can be compared to experimental measurements. In a general, abstract way, a cross section can be expressed as

$$\sigma = \frac{1}{F} \int d\Phi |\mathcal{M}|^2. \quad (1)$$

It is worth mentioning that for cross-section calculations, and thus for the calculations in this article, the integration is always performed over the real domain. The *flux factor*, F , characterises the rate at which the initial-state particles (a and b) collide, and is a real number. It is defined as

$$F = 4E_a E_b v_{\text{rel}}, \quad (2)$$

where E_a and E_b denote the energy of the initial particles, respectively, while v_{rel} is the relative velocity between them. For fixed energies in the initial state, as is the case in this article, F is a real constant, and thus can simply be factored out. The phase-space term $d\Phi$ encompasses four-momentum conservation ($p_a + p_b = p_c + p_d + \dots$) as well as the real integration over the kinematic variables of the final-state particles (c, d, \dots). Finally, the *matrix element* \mathcal{M} encodes the transition probabilities between the states a, b and c, d, \dots . The matrix element itself is constructed from complex numbers and spinors, but when it is complex-conjugate squared (as in equation (1)) it is always a real number. The underlying theory of interactions between the particles is encoded in the matrix element. In particular, when the interaction between the initial and final state is mediated via *virtual* particles (not necessarily identical to the external ones), so-called *propagators* will be present in the matrix element. Their general form, arising in $|\mathcal{M}|^2$, reads

$$p(x) = \frac{1}{(x - M^2)^2 + M^2\Gamma^2}, \quad (3)$$

where M and Γ represent the mass and the decay width of the massive particle, respectively. It is worth mentioning that the decay width of an unstable particle is a measure of its decay probability per unit time. It is the inverse of the particle's lifetime, and in that respect, an important parameter for characterising a particle's dynamics. For massless particles, the propagator in equation (3) reduces to $p(x) = 1/x^2$, by setting $M = 0$. In this case, the variable x is a kinematic invariant which has units of energy/mass squared. This means that the general form of the integral to be performed in a cross-section calculation (as in equation (1)) reads

$$\int \prod_{i=1}^{N_I} dx_i \frac{\sum_{S_k \in I} \alpha_k \prod_{j \in S_k} x_j^{n_j}}{\prod_{p=1}^{N_p} (x_p - M_{op}^2)^2 + M_{op}^2 \Gamma_{op}^2}, \quad (4)$$

where N_I and N_p are the number of integration variables and propagators, respectively, so that $N_I \geq N_p$. In the present notation, the variables of integration x_i can either be kinematic invariants or angles related to the final-state particles. The number of integration variables scales as $3n - 4$ for a $2 \rightarrow n$ scattering process. The number of propagator terms depends on the process considered, which is defined by the initial- and final-state particles. The index set I contains all combinations of all the N_I indices, which are denoted by the subset of indices S_k . In this way, all possible monomials can be represented. The index, n_j , represents the power of each term, and is a positive integer. The labels op denote all possible internal massive particles i.e. all particles that are neither in the initial nor final state. For example, for the process that will be studied later (see figure 8), the W boson is an internal particle. Finally, $\alpha_{j\dots k}$ are real constants to ensure full generality. There are as many constants as there are monomials. The challenge is thus to integrate a multi-dimensional integral that has a highly peaked structure due to the propagators³. Note that the form of equation (4) can be made even more general by allowing sums/differences of random variables instead of simply x_i in the denominator. However, given that this case only arises at very high multiplicities (at least $1 \rightarrow 5$ or $2 \rightarrow 5$), we defer the treatment of this even more general case for future work.

To be more concrete, a $2 \rightarrow 8$ process, such as the production of four leptons in association with four bottom quarks (which is dominated by the production of two top quarks in association with two bottom quarks) [34], will have 20 integration variables, and $\mathcal{O}(1000)$ different propagators. Without going into the details of further complications related to considering higher orders in perturbation theory, this example shows the level of complexity for state-of-the-art theoretical computations in HEP⁴. It also explains why the computational resources needed for the LHC's physics programme is so large⁵.

A further complication when computing cross sections is the appearance of event selections in experimental analyses. Typically, the detectors of an experiment cannot physically cover the entire phase space. Therefore, the integration in equation (1) has to be restricted to a smaller domain of integration, corresponding to that physically accessible by the experiment. This is done by setting minimal and maximal values (setting the cut values) to observable quantities, such as the rapidity or the transverse momentum of the final particles⁶. Events with particles that do not pass these requirements are not detected. From the theoretical side, these observable quantities depend on the final-state momenta, which themselves depend on the integration variables. The cross section therefore becomes

$$\sigma = \frac{1}{F} \int d\Phi |\mathcal{M}|^2 \Theta(C[\Phi] - C[\Phi_c]), \quad (5)$$

where the *cut function*, C , represents these restricted observables. Formally, the function C depends on the variables of integration, as it depends on Φ (the final-state particles), which is itself built from the variables of integration. However an analytical form for C as a function of the variables of integration may not necessarily exist; we will not investigate this aspect in detail in the rest of the article, however we will briefly discuss it.

We now move on to discuss classical approaches for calculating such integrals.

2.2. Classical approach

The most common approach to compute integrals such as the ones presented in the previous section is by reverting to numerical MCI techniques. There are several reasons for this: first, as eluded to above, closed

³ The propagators describe intermediate particles that can become resonant when the invariant mass of the four momentum is close to the mass of the particle. They are therefore sometimes referred to as *resonances*.

⁴ The interested reader can look at [35] and references therein to get an overview of the current frontier in theoretical predictions for HEP.

⁵ To be exact, the computational time quoted in the introduction is for the generation of so-called *events*, representing the underlying distribution to be integrated. These events are thus closely related to the integration of the cross section.

⁶ The rapidity of a particle is defined as $y = \frac{1}{2} \log \left(\frac{E+p_L}{E-p_L} \right)$, where E is the energy of the particle and p_L its momentum along the beam axis, typically taken as the z axis by convention. With this convention, the transverse momentum is defined as $p_T = \sqrt{p_x^2 + p_y^2}$.

forms for the cut functions do not always exist. This means that numerical techniques must be used. Second, while analytical integration is possible for simple scattering processes (e.g. $2 \rightarrow 2$ or $2 \rightarrow 3$), with increasing multiplicity the calculations rapidly become intractable. Third, reverting to analytical calculations make things more difficult to automate, as there are typically many processes that are relevant for a given experiment, e.g. $\mathcal{O}(100)$ different processes for the LHC. The LHC in particular also has the drawback that it is a proton-proton collider, meaning that calculations rely on knowledge of the *parton distribution functions* for the proton, functions that provide the probability for a parton (an elementary particle such as a quark or gluon) to be emitted. These functions are not known from first principles, and must be extracted from data. As they are typically defined in terms of numerical grids, this makes them more naturally embeddable within a numerical framework. Finally, while one is primarily interested in computing the cross section, this quantity is also measured as a function of other experimental observables such as angles between particles or observables related to energies. When using MCI, computing the cross section as a function of other observables comes at no extra cost, as at each evaluation, the weight used for the cross-section evaluation can also be binned in a histogram according to the value of the observable of interest. In this case, while the binning of the histograms is a computationally cheap operation, for analytical calculations this would require additional, more complicated calculations.

The RMSE of classical MCI scales as $\mathcal{O}(1/\sqrt{S})$, where S is the number of samples. Improvements on basic MC techniques aim therefore to reduce the overall variance. This is done either by modifying the original integrand so that the new function has a smaller variance, or by sampling the integration such that the distribution of sampled points is no longer randomly uniform across the entire domain of integration [36].

The first type of technique is known as *importance sampling*. For HEP, propagators are mapped to functions with lower variance⁷. Given that the propagator structure is known *a priori* (thanks to knowledge of the Feynman diagrams that contribute to the process), it is possible to setup an integration routine such that each of the resonant structures are singled out. This leads to the so-called *multi-channelling* integration technique.

The second type of approach, is typically referred to as the *adaptive* MC technique [37]. The idea is that, through an iterative procedure, points are sampled according to the distribution to be integrated. This means that the peak structure is probed more efficiently.

For state-of-the-art integration in HEP, in reality combinations of both approaches are utilised. Nonetheless, the scaling of the overall RMSE is still $\mathcal{O}(1/\sqrt{S})$, motivating the exploration of quantum algorithms for sampling, which are known to have better scaling⁸.

In order to fix notation for later use and clarity, we here define the problem of MCI in a more rigorous way. MCI calculates the expectation of a function, $g(x)$, of a continuous random variable, X , with probability density function (PDF), $f_X(x)$, defined as

$$\mathbb{E}[g(X)] = \int g(x)f_X(x) dx. \quad (6)$$

We will hereafter refer to $g(\cdot)$ as the *function applied*. We can straightforwardly extend this to functions, $g(x, y)$, of two continuous random variables, X and Y , with joint PDF, $f_{XY}(x, y)$,

$$\mathbb{E}[g(X, Y)] = \iint g(x, y)f_{XY}(x, y) dx dy. \quad (7)$$

Indeed, this can be extended further to an arbitrary number of dimensions, and with the integration performed over either all or subsets of the given variables. However, for the purposes of this article it will only be necessary to focus on expectations of separable functions of at most both X and Y i.e. products of functions, $g(x, y) = h(x)l(y)$, with expectation

$$\mathbb{E}[h(X)l(Y)] = \iint h(x)l(y)f_{XY}(x, y) dx dy. \quad (8)$$

We now move on to discuss the equivalent approach in quantum computing.

⁷ See [36] for a pedagogical explanation in the case of a single propagator.

⁸ It is fair to say that *Markov-chain MC* (MCMC) and *quasi-MC* approaches have so far received little attention as methods for cross-section integration in HEP. While the former seems promising [38–41], the latter is known to be more appropriate for calculations at higher orders of perturbation theory [42–44]. The reason is that quasi-MC methods are not appropriate for integrands with highly peaked structures, and where there are asymmetric restrictions on the phase space—as is the case for cross-section calculations.

2.3. Quantum approach

The quantum analogue of classical MCI, QMCI, is a quantum algorithm that returns a numerical estimate for the value of some (possibly multidimensional) integral, in the same manner as for classical MCI.

QMCI provides a quadratic speed up in the convergence of the RMSE of the estimate as a function of the number of samples, as compared to classical MCI. Therefore, the RMSE scales as $\mathcal{O}(1/S)$, compared to $\mathcal{O}(1/\sqrt{S})$ classically. The source of the quantum advantage in QMCI arises from the use of QAE [20]—a generalisation of *Grover's Search* algorithm [45]—which is the key subroutine of QMCI.

The QMCI algorithm comprises three main steps, as described in [32], where for demonstration purposes we only consider a single random variable, X , (however this is straightforwardly extended to multiple variables).

1. A quantum state, $|p\rangle$, encoding a multivariate probability distribution, $f(\cdot)$, in the values of its complex amplitudes is prepared by applying a quantum circuit, P , such that

$$|p\rangle = P|0\rangle_n = \sum_x \sqrt{f_X(x)}|x\rangle. \quad (9)$$

2. An observable function, $g(\cdot)$, is applied coherently to the state based on a quantum arithmetic circuit, R , with the expectation value of interest, $\mathbb{E}[g(X)] = \sum_x g(x)f_X(x)$, then encoded in the amplitude of an additional ancilla qubit as

$$R|p\rangle|0\rangle = \sum_x \sqrt{f_X(x)}|x\rangle \left(\sqrt{1-g(x)}|0\rangle + \sqrt{g(x)}|1\rangle \right). \quad (10)$$

3. The expectation value of interest—which is equal to the probability of measuring the 1 state—is estimated using QAE.

At this stage, it is also worth briefly discussing the fact that the first step of QMCI necessarily entails some state-preparation protocol, which has an associated computational cost. A natural question to ask, then, is how the state-preparation overhead affects the overall quadratic advantage of QMCI, and under what assumptions this advantage is fully retained. Herbert [46] discusses this in detail, and notes that the computational advantage is retained if the operational cost of preparing the quantum state encoding the probability distribution is on par with that of generating a classical sample—an efficiency achieved by the Q -marginal construction, which allows such quantum states to be prepared directly from reversible classical sampling circuits using only a single layer of Hadamard gates.

2.3.1. FQMCI

In [19], Herbert proposed the FQMCI methodology as an efficient (low depth) means of performing QMCI that retains the full quadratic advantage without requiring costly quantum arithmetic. The idea stems from the fact that $\mathbb{E}[\sin^2(X)]$ is a straightforward quantity to estimate on a quantum computer (merely requiring a simple circuit corresponding to a bank of R_y rotation gates, rather than any complicated quantum arithmetic). In FQMCI the function applied—under the assumption that it obeys certain smoothness conditions (continuous in value and first derivative, and second and third derivatives piecewise continuous and bounded)—is extended as a periodic, piecewise function, and this is then decomposed as a Fourier series. Then, because each term in the series is a cosine or sine, their expectations can be easily estimated individually, and these then recombined.

This methodology is also easily extended to the bivariate case, where there are expectations of products of functions applied of the form $\mathbb{E}[h(X)l(Y)]$. In this case the individual Fourier series for h and l are multiplied together. One then ends up with univariate sine and cosine terms, alongside bivariate products of sines and cosines, which in the latter case can be reduced to sums of single bivariate sines or cosines using trigonometric identities. Thus the expectation can again be estimated by calculating expectations of sines and cosines, and then recombining. However, we note that this differs from the univariate case as multiple expectations are required to be calculated, and thus the resources required to perform the calculation are greater. In addition, the circuits are larger, because the expectations are for bivariate trigonometric functions (e.g. $\mathbb{E}[\sin^2(X - Y)]$), requiring an additional bank of R_y rotation gates to represent the Y variable.

It is important to note for what follows that this technique can also be extended to an arbitrary multivariate case i.e. to expectations of products of functions applied of the form $\mathbb{E}[h(X)l(Y)m(Z)\dots]$. However, for products of more than two this technique is unlikely to be particularly efficient due to the polynomial increase in the numbers of terms required to compute, and the additional increase in size of the circuits involved.

2.3.2. QUANTINUUM's QMCI engine

QUANTINUUM's QMCI engine [32] is the world's first fully integrated platform for performing QMCI, with each of the steps discussed previously implemented within the engine based on state-of-the-art methods. There are a number of optimised, in-built features that will become important for the later discussion of how to implement generic cross-section calculations.

First, the engine can take as input any circuit, P , provided by the user, and then this circuit is straightforwardly passed along the rest of the QMCI pipeline. It also contains an in-built *state-preparation library* (the *distribution loader*) containing circuits that represent some common distributions, such as the uniform and Gaussian distributions (see section 7 of [32] for a detailed description of the distribution loader). Second, based on the FQMCI method, the engine provides protocols to implement efficient Fourier-decomposition circuits, R , for a variety of functions applied, particularly corresponding to moments or products of moments of random variables, such as for $g(x) = x^2$, $g(x, y) = xy$ etc (see section 8 of [32] for a detailed description of the types of quantities that can be estimated using the QMCI engine). Third, the engine contains a number of in-built algorithms for performing QAE, in general optimised for performance⁹ (see section 9 of [32] for a detailed description of the in-built QAE algorithms). Fourth, the engine is able to implement limits on integral calculations, based on the use of *thresholding operations*, implemented via the *enhanced P-builder* functionality. Thresholding operations correspond to binary operations based on some threshold value, V_{th} , that enable the estimation of $\mathbb{E}[X\Theta(X \geq V_{\text{th}})]$, where the indicator function, $\Theta(z)$, with z either a random variable or an expression, is defined as¹⁰

$$\Theta(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

Fifth, the user is able to perform a QMCI calculation such that the RMSE of the final estimate is upper bounded by a chosen value (see section 8 of [32] for a description of this). Finally, the engine is equipped with a *resource mode* that can determine the quantum resources (in terms of qubit numbers and gate counts/depths) required to perform a given QMCI calculation to a desired RMSE. The resource calculator outputs either: with a view to running calculations in the noisy intermediate-scale quantum (NISQ) era—where applying two-qubit gates are the main bottleneck for hardware—the total number and depth of CX gates across all circuits, alongside the same information for the largest circuit that is run, and the same information for the total number and depth of all gates; or, with a view to running calculations in the fault-tolerant era with full quantum error correction—where applying T gates will be the main bottleneck for hardware—the total number and depth of T gates across all circuits, alongside the same information for the largest circuit that is run. In both cases, the total number of qubits required for the largest circuit is also output (see section 10 of [32] for a detailed description of exactly how the resource mode is implemented). Overall, these features will be key to later discussions in this article.

We note that how to split the integrand for the expectation value into the product of function applied, $g(x)$, and probability distribution, $f_X(x)$, is in general arbitrary. For QMCI applications in finance, which is so far the only application that has been previously studied using the QMCI engine [33], there is a clear notion of calculating some expectation of the random variable for use in a financial calculation, such as the mean, $g(x) = x$, or the second moment, $g(x) = x^2$. However, for general applications (i.e. integrands) there is freedom in how to perform this split, and we take advantage of this notion in the next section.

2.4. Implementing a generic cross-section calculation

In the introduction, we have mentioned so called generic *building blocks* for the computation of cross sections. To illustrate this, we provide here two concrete examples. From equations (1) and (4), a cross section with only a single integration variable reduces to

$$\sigma_1 \propto \int_0^s \frac{dx x^n}{(x - M_0^2)^2 + M_0^2 \Gamma_0^2}, \quad (12)$$

where s is the maximal value of x . In order to cast the above equation into the more rigorous one of the MCI problem as in equation (6), noting that we are free to choose how to split the integrand for the expectation

⁹ This optimisation generally involves setting the number of shots such that the RMSE of the estimator is minimised.

¹⁰ The threshold is enforced by adding the one's complement of the classical threshold value, ensuring that the most significant result qubit is set when the condition is met. A subsequent step resets the remaining result qubits to the zero state while preserving the most significant qubit. This makes the cleared qubits available as ancillas for future operations.

value into the product of function applied, $g(x)$, and probability distribution, $f_X(x)$, it follows that we can choose the function applied to be

$$g(x) = x^n, \quad (13)$$

(i.e. the numerator). The probability distribution to sample from can then be associated to the propagator function (i.e. the denominator), such that

$$f_X(x) = \frac{1}{(x - M_0^2)^2 + M_0^2 \Gamma_0^2}. \quad (14)$$

The reason we chose this particular factorisation is twofold. First, as mentioned previously, implementing moments or products of moments of random variables as the function applied is straightforward to implement based on the QMCI engine's existing functionality, and the RMSE scaling of the corresponding Fourier series' are likely favourable as compared to those of more complex functions. Second, because the probability distribution corresponds to a real distribution that has physical significance (being used to model resonance behaviour in general), we can envision that future development of, for example, specific state-preparation methods tailored to this distribution¹¹ would not only be useful for this particular application, but also for future research directions in HEP (e.g. for quantum-computing applications where resonance behaviours are required to be modelled).

We can identify the general form of equation (14) as an (unnormalised) relativistic BW distribution [47] (hereafter simply denoted by BW distribution). As mentioned previously, state preparation of arbitrary probability distributions is a difficult problem. However, in this case we need only prepare one specific distribution, and as will be discussed in the following section, it will suffice to prepare this distribution a single time (for a given COM energy) for each of the (small number of) relevant massive particles in the Standard Model, i.e. the W and Z bosons, the top (t) quark, and the Higgs boson. In the present example, the two building blocks are the function, g , which is a monomial, and the BW distribution, f_X .

While not the focus of this article, it is also worth briefly discussing the case of massless particles where, as discussed previously, the propagator term then reduces to

$$p(x) = \frac{1}{x^2}. \quad (15)$$

Such terms would then be absorbed in the monomials in the numerator, possibly with negative exponents. For this case of negative exponents, because the functions applied for FQMCI must obey certain smoothness conditions, as detailed previously in section 2.3.1, then in this case one would not be able to exploit FQMCI. Instead, an approach similar to those of prior works would be required, where the entire integrand is prepared as the probability distribution and then QAE run directly on this state (equivalent in our language to the 'function applied' being constant).

In the same way, a general expression for a two-dimensional integration reads

$$\sigma_2 \propto \iint \frac{dx dy x^n y^m}{\left[(x - M_{01}^2)^2 + M_{01}^2 \Gamma_{01}^2 \right] \left[(y - M_{02}^2)^2 + M_{02}^2 \Gamma_{02}^2 \right]}, \quad (16)$$

where the term $x^n y^m$ means that the numerator can be made of any monomials which are a product of x and y raised to various powers. It implies that in order to match the form of equation (8), the separable functions applied should be

$$h(x) = x^m, \quad l(y) = y^m. \quad (17)$$

Then, the probability distribution becomes

$$f_{XY}(x, y) = \frac{1}{\left[(x - M_{01}^2)^2 + M_{01}^2 \Gamma_{01}^2 \right] \left[(y - M_{02}^2)^2 + M_{02}^2 \Gamma_{02}^2 \right]}. \quad (18)$$

In the case that we consider, it should be noted that the two BW distributions factorise in the sense that each propagator is a function of a single random variable. This property is particularly useful as it means that each propagator can be associated to one particular random variable, allowing the propagator product to be cast

¹¹ Note that in this article we only consider general state-preparation methods for preparing the BW distribution, and we defer research into more tailored methods for future work.

as a simple product of orthogonal propagator terms. This is key, as it means that our approach is easily extendable to multiple dimensions, as would be required in the general case.

The strategy is thus to consider as the function applied the monomials in the numerator, while the propagators terms are treated as the probability distribution to be sampled from. Analogously, the basic building blocks are simply monomials, and BW distributions. As highlighted here, this approach can be extended to arbitrary dimensions.

The QMCI engine provides efficient Fourier decomposition circuits for moments or products of moment of random variables, as discussed previously, and allows the user to provide the probability distribution to be sampled from as an input. In addition, when constraints on the phase space are required to be imposed on the integration, as given by the cut function, C , in equation (5), then for simple constraints such as sums, maximum/minimum etc the engine's built-in thresholding operations can be used to achieve this. Thus—providing there are efficient ways to prepare BW distributions as quantum states, something discussed in the next section—in principle the QMCI engine can be used to perform quantum integration of a generic cross section in HEP.

It is worth noting that the cut function, C , is a function of the integration variables, and as discussed previously, possibly non-analytical. This means that in order to implement the restriction of phase space, the value $C(\Phi)$ should be numerically evaluated at each phase-space point. Such an approach has been followed in section 4 to set the boundaries of integration of a given variable that depends on another variable of integration; however, in this case the expression was analytical, and the quantum arithmetic implemented via the QMCI engine's enhanced P -builder. For the case of some non-analytical form, one possibility is to implement the required quantum arithmetic by constructing the equivalent classical circuit, and then making it a reversible circuit. However, we note that this approach is unlikely to be efficient, and indeed, in the future there may be better ways of constructing such circuits. We leave this aspect for future work.

3. State preparation of relativistic BW distributions

We see that to be able to perform generic cross-section calculations using the QMCI engine, we require a methodology for preparing on a quantum register BW distributions for the massive propagators W , Z and t ,¹² as these constitute one of the building blocks of the method. As discussed briefly in the introduction, state preparation of probability distributions for QMCI is an active research topic, and several methods have been proposed in the literature (see, e.g. [48–54]); however, there is currently no standard methodology that is commonly used.

In this section, we propose two different methods for preparing BW distributions, and discuss the benefits and drawbacks of each. We note that this is by no means meant to be a definitive nor exhaustive list, as the purpose of this section is not to demonstrate the *best possible* solution for preparing BW distributions, but rather to give some examples to illustrate that this can be done in different ways. Indeed, in the future, as new state-preparation methods are developed, these can be utilised to generate larger-scale circuits, improve on the accuracy of the generated distributions, and reduce the quantum resources required to prepare them.

Before we continue the discussion, however, we must first discuss some of the challenges inherent to state preparation for QMCI. These challenges generally arise due to the limited number of qubits, n , that are available (due to the limitations of current hardware) to represent the true continuous distribution as a discretised, truncated distribution. In the QMCI engine, the amplitudes of the state, $|p\rangle$, are interpreted as the support points of the discretised distribution, with the number of support points, $N = 2^n$. These are uniformly spaced with spacing size, Δ , spanning the range of the support of the distribution, $[x_l, x_u]$. In practice, the limited number of points alongside the uniform binning leads to limited resolution on the distribution of interest, as well as having to truncate what could be an infinite distribution. For a finite number of qubits, these limitations introduce a number of systematic errors into a QMCI calculation¹³. [33] discusses these systematic errors in detail, and so here we merely reproduce some of the definitions of the relevant ones for this discussion:

- **Discretisation error:** this arises from the finite resolution of sampling for the random variable, due to the limited number of discrete support points used to encode the continuous random variable. This is negligible for classical MCI with (effectively) unlimited numbers of bits, but significant for QMCI due to the limited

¹² In practice, completeness would also require a propagator for the Higgs boson. However, due to the narrow width of the resonance, generating this peak with sufficient resolution is highly challenging, given current qubit counts and methods. At high-energy colliders, all leptons are typically considered as massless, alongside all quarks (apart from the top quark), as their masses are negligible at these energy scales. In certain cases, the mass of the bottom quark is also included in theoretical calculations, but with the width assumed to be zero.

¹³ It is important to note that many of these systematic errors are also applicable to classical MCI, as will be touched upon.

number of qubits available, therefore

$$\epsilon_d = \left| \int_{x_l}^{x_u} g(x) f_X(x) dx - \sum_{i=0}^{N-1} g(x_i) f_X(x_i) \Delta \right|, \quad (19)$$

where $x_0 = x_l$, $x_i = x_l + i\Delta$, $x_{N-1} = x_u$, and $\Delta = x_{i+1} - x_i = \frac{x_u - x_l}{N}$.

- **Normalisation error:** this arises because truncation and discretisation distort the probability distribution, meaning the mass of the PDF that is loaded as a quantum state may not be equal to unity, therefore

$$\epsilon_n = \sum_{i=0}^{N-1} \left| g(x_i) \left(f_X(x_i) \Delta - \tilde{f}_X(x_i) \right) \right|, \quad (20)$$

where $\tilde{f}_X(x_i)$ are the true normalised probabilities.

- **Thresholding error:** this occurs specifically for the QMCI engine when considering the thresholding operations used to impose limits of integration. To estimate $\mathbb{E}[X\Theta(X \geq V_{\text{Th}})]$, the QMCI engine defines a threshold with an inclusive upper bound, but if $V_{\text{Th}} \in (x_i, x_{i+1})$, it instead approximates the threshold using x_i , therefore

$$\epsilon_{\text{th}} = |\mathbb{E}[X\Theta(X \geq V_{\text{Th}})] - \mathbb{E}[X\Theta(X \geq x_i)]|. \quad (21)$$

- **State-preparation error:** this error is distinct from the other errors discussed as it does not arise from the limited number of qubits, but rather the imperfect method of preparing the distribution as a quantum state i.e. there is a difference between the ideal quantum state and the approximated state that is actually prepared. The state-preparation error can be defined in a variety of different ways. In this article, we consider the *cumulative-distribution-function (CDF) mean-squared error (CMSE)* between the probabilities of the prepared state, $f_X^p(x_i)$, and the true probabilities, $\tilde{f}_X(x_i)$, as the metric representing the state-preparation error

$$\epsilon_s^{\text{CMSE}} = \frac{1}{N} \sum_{i=1}^N (\tilde{F}_X(x_i) - F_X^s(x_i))^2, \quad (22)$$

where \tilde{F}_X, F_X^s are the empirical CDFs of \tilde{f}_X and f_X^s , respectively. In addition, we also consider the *Jensen-Shannon divergence (JSD)* as a metric for determining how accurately the state is prepared

$$\text{JSD} = \frac{1}{2} \left(\sum_i \tilde{f}_X(x_i) \log \frac{\tilde{f}_X(x_i)}{\mathcal{F}(x_i)} + \sum_i f_X^s(x_i) \log \frac{f_X^s(x_i)}{\mathcal{F}(x_i)} \right), \quad (23)$$

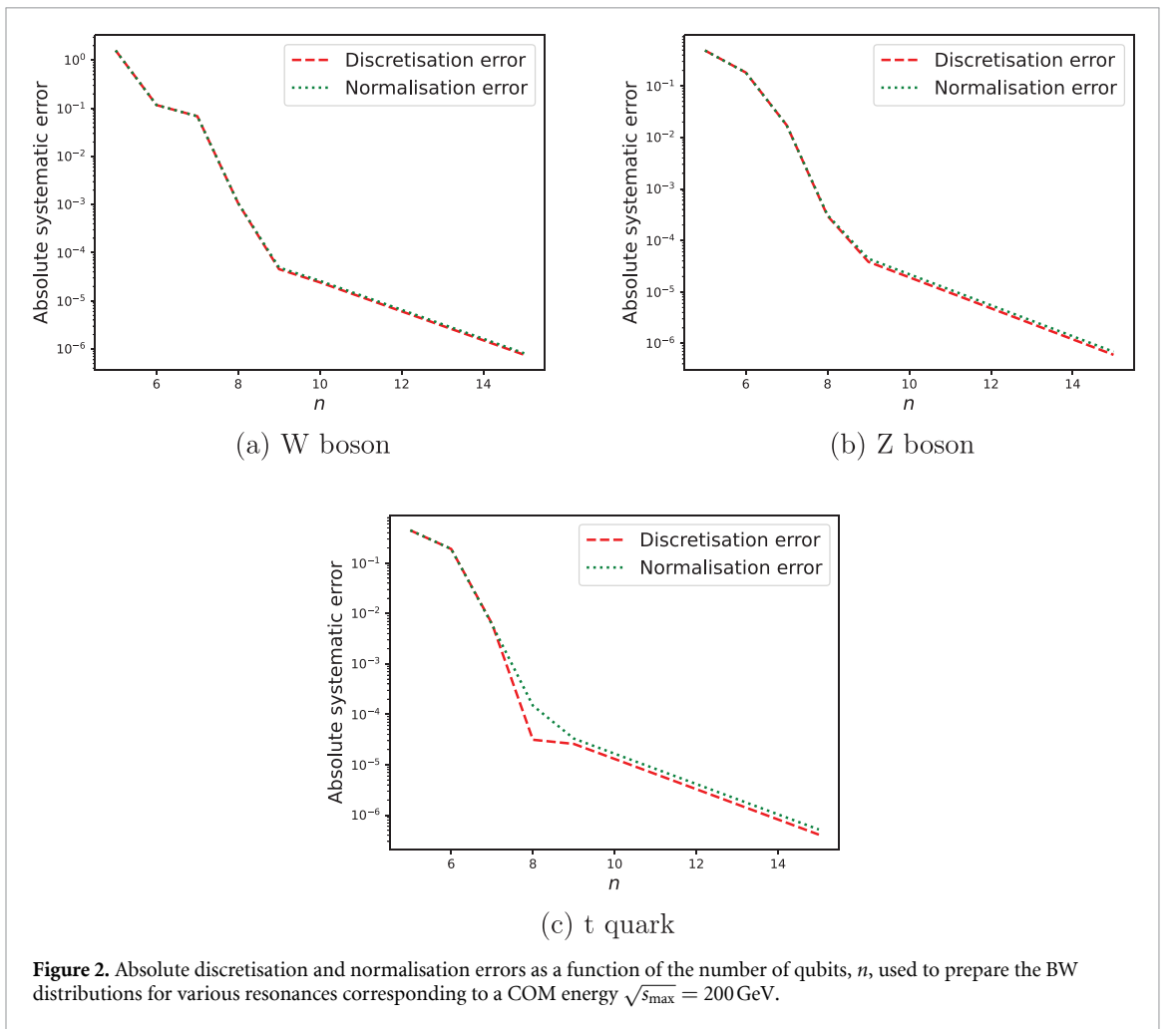
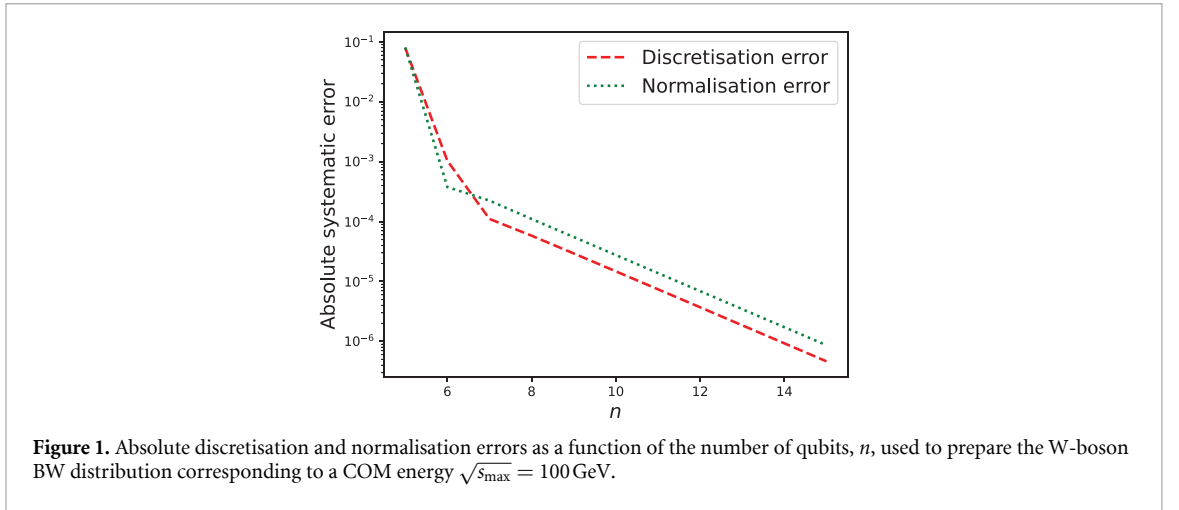
where $\mathcal{F}(x_i) \equiv \frac{1}{2}(\tilde{f}_X(x_i) + f_X^s(x_i))$.

All of these systematic errors can be considered relevant for the cross-section calculations considered later in section 4.

Considering the discretisation and normalisation errors in particular, for a given number of qubits, n , there will be a sub-range of the full potential support of the distribution where these systematic errors are sufficiently minimised (below a chosen bound) to have a negligible impact on a QMCI calculation. When considering cross-section calculations for a given experiment, the integration is only ever performed from $s_{\text{min}} = 0 \text{ GeV}^2$ to some $s_{\text{max}} = S \text{ GeV}^2$. Therefore, in practice, it will not be necessary to prepare the BW distribution to span the full potential support range [$s_{\text{min}} = 0 \text{ GeV}^2$ to $s_{\text{max}} = +\infty \text{ GeV}^2$]—which would require an infinite number of qubits in order to sufficiently suppress these errors. Thus, whilst qubit numbers are still limited, for practical applications it will instead suffice to generate a range of circuits that prepare BW distributions with supports spanning a range of different COM energies squared, with each prepared using a sufficient number of qubits as to sufficiently suppress these systematic errors.

It is also worth mentioning that while hadron colliders collide particles at energies of up to tens of TeV (13.6 TeV for the LHC), due to the suppression of the parton distribution functions at high energies, the accessible energy range is effectively reduced to a few hundreds of GeV.

To this end, and for illustration purposes, in this article we consider preparing BW distributions for the W, Z, and t resonances, corresponding to a COM energy, $\sqrt{s_{\text{max}}} = 200 \text{ GeV}$, and for just the W to $\sqrt{s_{\text{max}}} = 100 \text{ GeV}$. The value, $\sqrt{s_{\text{max}}} = 100 \text{ GeV}$, was chosen such that the circuits for the cross-section calculations in sections 4.2 and 4.3 (which only use the W-boson propagator) were sufficiently small to be



classically simulable using state-vector methods, while $\sqrt{s_{\max}} = 200$ GeV was chosen in order to be able to generate distributions covering all of the resonances.

Figures 1 and 2 plot the discretisation and normalisation errors as a function of the number of qubits used to prepare the distributions for the various propagators. Considering the $\sqrt{s_{\max}} = 100$ GeV distribution first, with the aim of allowing for classical simulations later, while keeping systematic errors relatively suppressed (i.e. minimising the number of qubits in the state-preparation circuit), we selected $n = 6$, based on figure 1, which corresponds to errors in the range $\sim(10^{-3}-10^{-4})$. For the $\sqrt{s_{\max}} = 200$ GeV distributions we chose $n = 9$, such that errors are in the $\sim(10^{-4}-10^{-5})$ range.

We now move on to discuss the two different methodologies investigated for preparing BW distributions on quantum computers.

3.1. Variational method

The first methodology is based on a variational, quantum machine-learning approach, as motivated by [32]. Here a PQC [55] is trained to generate an approximation of the distribution of interest.

Briefly summarising the method, the loss function that is minimised in the training using a classical optimiser is the L_α norm (for $\alpha = 1, 2, \infty$) between the target state, $|\text{tar}\rangle$, and the generated state, $|\text{gen}\rangle$,

$$\| |\text{tar}\rangle - |\text{gen}\rangle \|_\alpha = \left(\sum_{i=1}^N (| |\text{tar}\rangle - |\text{gen}\rangle |_i)^\alpha \right)^{1/\alpha}. \quad (24)$$

The PQC ansatz chosen, $U(\vec{\theta})$, is a maximally expressive hardware-efficient ansatz [56] consisting of an n -qubit circuit formed of $L + 1$ layers, with all angles initialised to $\pi/2$, giving $n(L + 1)$ variational parameters as

$$U(\vec{\theta}) = U_R(\vec{\theta}^{L+1}) \overbrace{U_{CX}U_R(\vec{\theta}^L) \dots U_{CX}U_R(\vec{\theta}^1)}^{L\text{-times}}, \quad (25)$$

where U_{CX} are fixed blocks of CX gates. The training parameters to optimise are the angles of the R_y rotations across k layers

$$U_R(\vec{\theta}^k) = \bigotimes_{i=0}^{n-1} R_y(\theta_i^k). \quad (26)$$

Thus the aim is to learn a $U(\vec{\theta})$ such that

$$|\text{gen}\rangle = |\psi(\vec{\theta})\rangle = U(\vec{\theta}) |0^n\rangle. \quad (27)$$

As is well understood from the literature (see, e.g. [26–31]), variational methods are plagued by issues of trainability, specifically the gradients of the cost function vanishing exponentially in the size of the system (number of qubits), known as *barren plateaus*. Barren plateaus have been shown to be directly related to the expressivity of the circuit ansätze used in the training [57]. In general, there is therefore a balance between allowing the circuit to be expressive enough to contain the desired solution, whilst also limiting the effects of barren plateaus. Given this, it is thought that variational approaches for state preparation will likely not be scalable for large system sizes. However, for the BW distributions considered in this article—which are all small-scale circuits—variational methods should give good results in practice. Indeed, no particular effort is made here to limit the expressivity of the circuit ansatz (for example by using some alternative ansätze such as the ones discussed in [58, 59]), as it was not considered necessary for these small systems.

3.2. Fourier expansion method

The second methodology that we investigate for preparing BW distributions involves a promising recent technique by Rosenkranz *et al* [60], where a multivariate probability distribution is decomposed into a Fourier series. In our case we need only consider univariate state preparation.

To again briefly summarise the method, the target function, f , is approximated by mapping to the interval $[-1, 1]$ ¹⁴ with a finite Fourier series of the form

$$f_d(x) = \sum_{k=-d}^d c_k e^{i\pi kx}, \quad (28)$$

where d is the degree of the expansion and c_k are the Fourier coefficients. An interpolation is used such that f_d matches f for a number of interpolation points, and the coefficients c_k for the interpolant are calculated using the fast Fourier transform. The basis functions of the Fourier expansion up to a chosen d are then prepared efficiently on a quantum register using a block encoding. The weighted sum of basis functions is prepared using a *linear combination of unitary* (LCU) operations [61], leading to a block-encoding circuit for f_d .

¹⁴ Specifically, it is represented in the interval $[0, 1]$ and then a periodic extension is applied to map onto the interval $[-1, 0)$.

The potential benefits of this method is that, in contrast to the previous method, it should scale well to larger system sizes. However, one important consideration is that the method relies on a considerable number of ancilla qubits, and is a probabilistic state preparation; this is because the LCU method requires $\log_2(2d + 1)$ additional ancilla qubits, and to then prepare the correct state, all of these ancilla qubits must be post selected to give the zero state, which has success probability $p_{\text{success}} = \frac{\sum_x |f(x)|^2}{2^n (\sum_k |c_k|)^2}$. In practice, p_{success} can be increased using amplitude amplification, although this of course means additional quantum resources. It is also worth noting that, in practice, the chosen value of d dictates the final accuracy of the prepared distribution (i.e. larger d means a smaller state-preparation error and JSD).

3.3. Results and comparison

Both the methods discussed in the previous sections were used in order to produce circuits that prepared BW distributions for the W, Z, and t resonances based on the $\sqrt{s_{\text{max}}}$ ranges and corresponding numbers of qubits in the circuit ansatz. For comparing the different methods, the metrics that are considered are the resources required for the circuit, in terms of the total number of qubits, n , alongside the number of one- and two-qubit gates, g_{1q} and g_{2q} , respectively, and the accuracy of the prepared distribution, in terms of the state-preparation error, ϵ_s^{CMSE} , and the JSD. In this section we will explicitly discuss the nine-qubit circuits corresponding to $\sqrt{s_{\text{max}}} = 200 \text{ GeV}$, in order to perform a comparison of the two methods for all resonances. However, results for the other COM energy range are also briefly discussed later.

PQCs were trained based on the variational method, as discussed previously. In all cases, the metric used in the cost function was the L_2 norm between the target and generated state. The *basin-hopping* algorithm [62] implementing the *Broyden–Fletcher–Goldfarb–Shanno* method [63] was used for the minimisation. A coarse hyperparameter search was performed during training in order to optimise the performance, based on varying the number of layers (parameters) in the ansätze alongside another hyperparameter related to the cost function (the overall scaling factor).

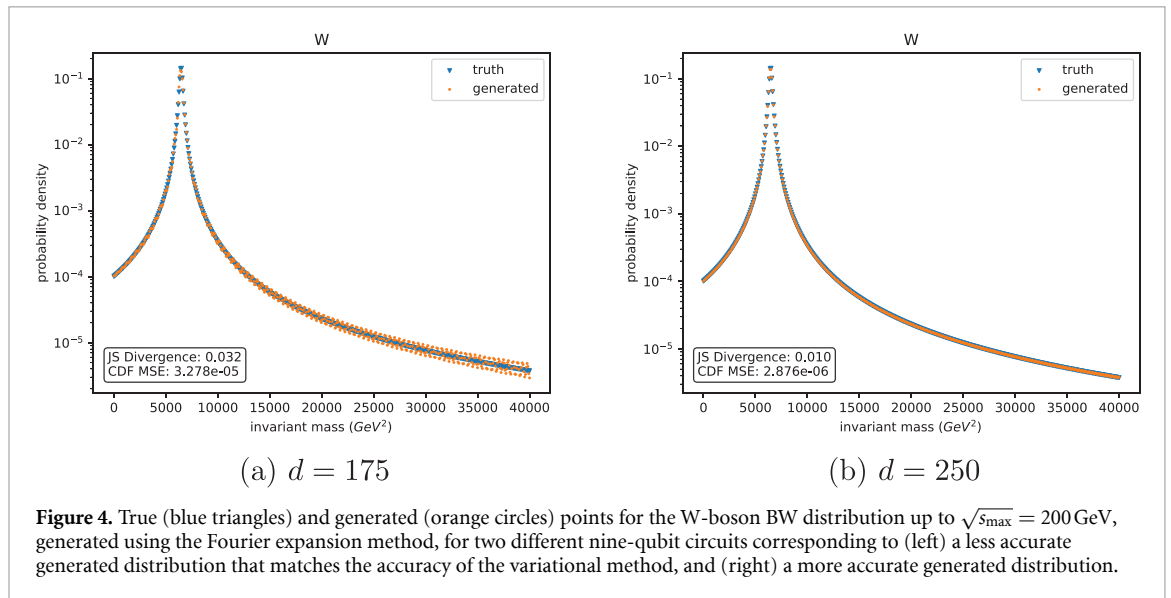
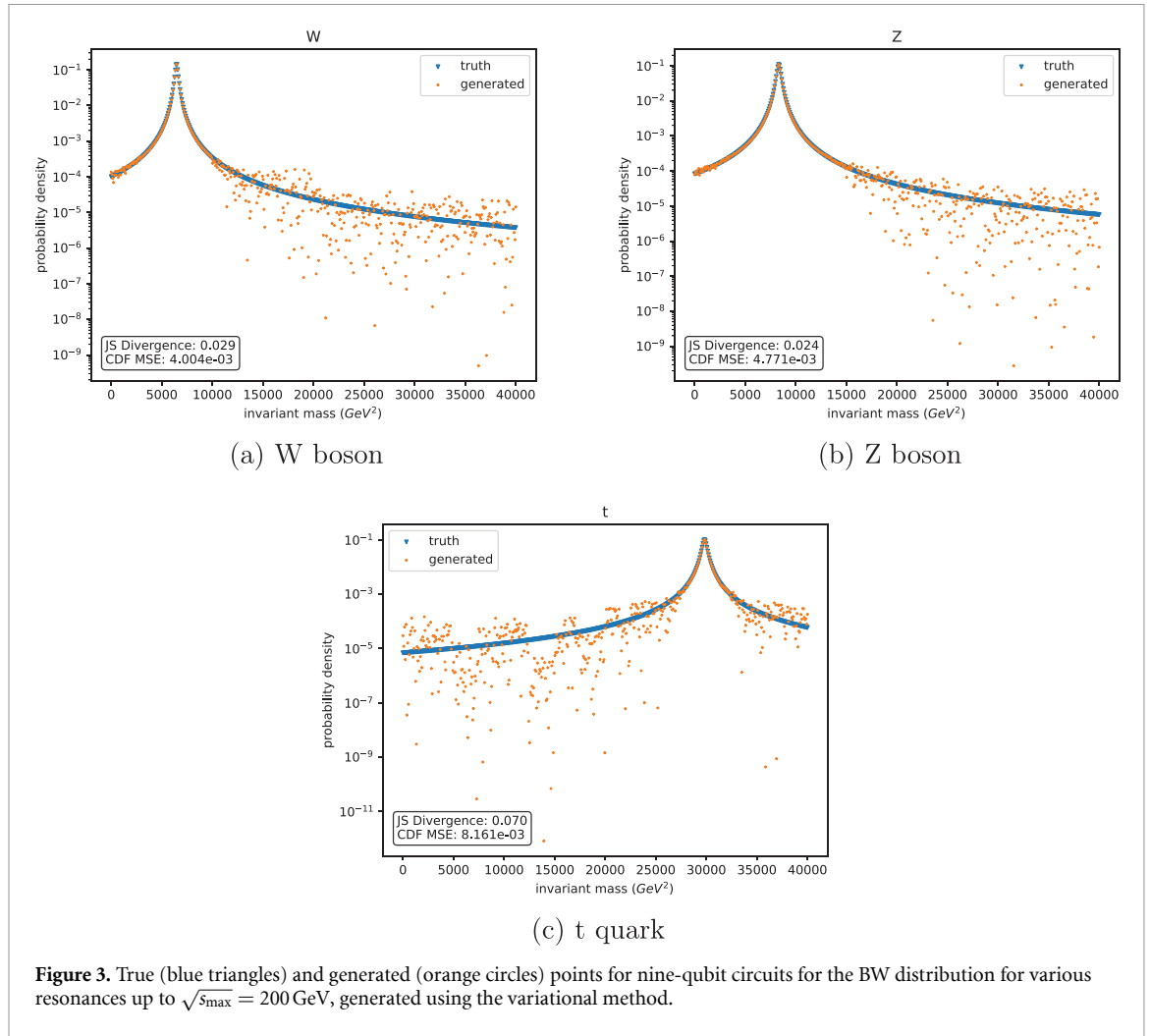
The Fourier expansion method was then analysed. For each of the resonances we produced a circuit that was roughly equivalent to the best one produced by the variational method in terms of the accuracy of the prepared distribution. In addition, we also produced another circuit where the accuracy of the prepared distribution was significantly greater than in the variational case (i.e. with an increased value of d). This was done to demonstrate that the method enables the preparation of larger-scale circuits while maintaining accuracy.

Figure 3 gives the generated distributions for each resonance using the variational method, while figures 4–6 give the generated distributions using the Fourier method for each of the resonances, respectively. Table 1 lists the metrics for the distributions produced for all COM ranges for both methods. Figure 12 in appendix A.1 also plots the distributions generated by the variational and Fourier expansion methods for the $n = 6$, $\sqrt{s_{\text{max}}} = 100 \text{ GeV}$ circuits.

In order to compare the two approaches, we will only discuss the W-boson results, as the same arguments apply to the others. We note that the Fourier expansion method allows for much more accurate states to be prepared than the variational method, and as mentioned previously, the real benefit is that it is scalable to larger numbers of qubits. However, we found that the resources required were significantly larger when considering the equivalent accuracy circuits to those produced using the variational method, due to the large number of additional ancilla qubits required. In addition, because the Fourier expansion method is a probabilistic state-preparation method, then in practice, without using techniques such as amplitude amplification, such small success probabilities will require the circuit to be run several times before the desired state is actually prepared (successful post selection).

It is worth also discussing that, for the $n = 6$ circuits (with distributions plotted in figure 12 in appendix A.1), we find that the variational method produces slightly more accurate results than the Fourier expansion method, and again requires significantly less resources. This indeed makes sense, given that we expect variational methods to perform well for small-scale circuits; however, one would expect the performance to decrease as the system size increases, which is indeed what we see when considering the $n = 9$ circuits discussed previously.

As an aside, if we were to consider an example of a potential more realistic use case corresponding to an energy scale comparable to a modern collider experiment—where in reality it would be essential to really suppress systematic errors—of $\sqrt{s_{\text{max}}} = 1 \text{ TeV}$, for example, then we can see from figure 7 that we would need around $n = 16$ for systematic errors to be suppressed to around $\sim(10^{-5}-10^{-6})$. For circuits this size, variational methods are likely not to be performant due to the trainability issues previously discussed. On the contrary, the Fourier expansion method should still be effective for generating accurate distributions at this circuit size, although the resources required would likely be considerable.



As discussed previously, state-preparation for probability distributions is an active research topic, and future developments should hopefully allow for the preparation of accurate BW distributions with reduced circuit sizes, paving the way for less resource-intensive QMCI for HEP. We now move on to discuss an example application of the QMCI engine for calculating a cross-section in HEP, making use of the six-qubit, $\sqrt{s_{\max}} = 100$ GeV, variationally-trained circuit for the W-boson BW distribution discussed in this section.

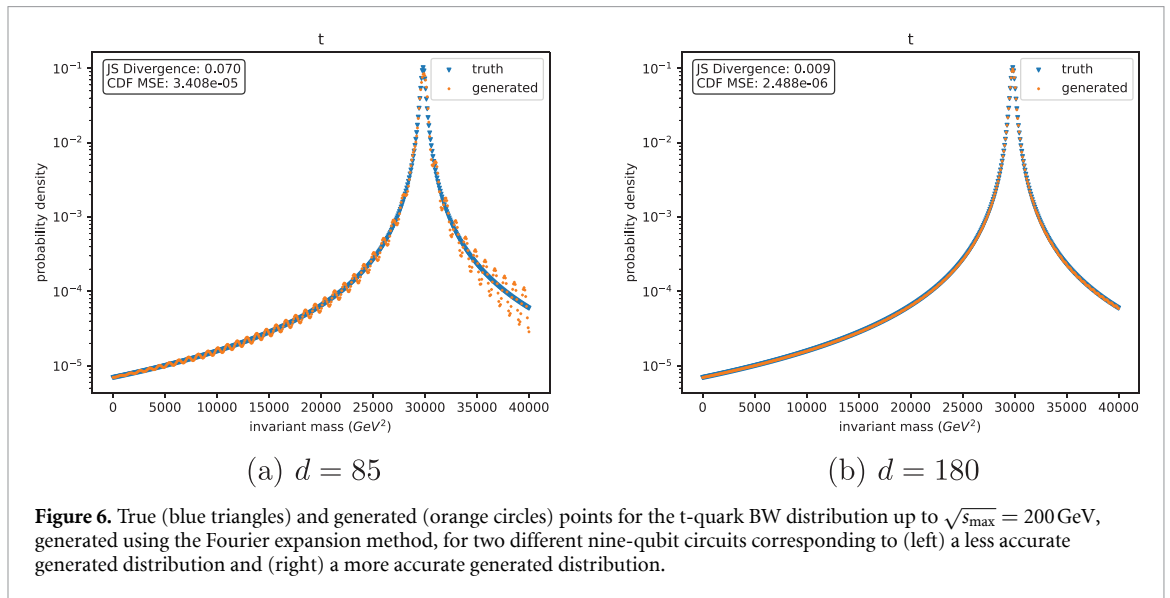
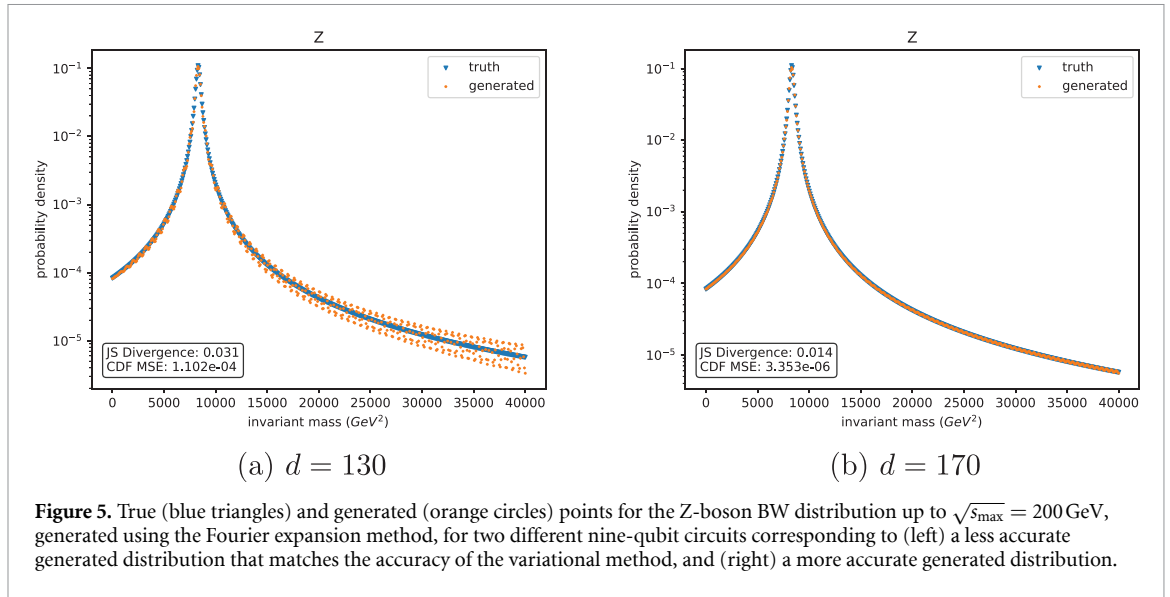
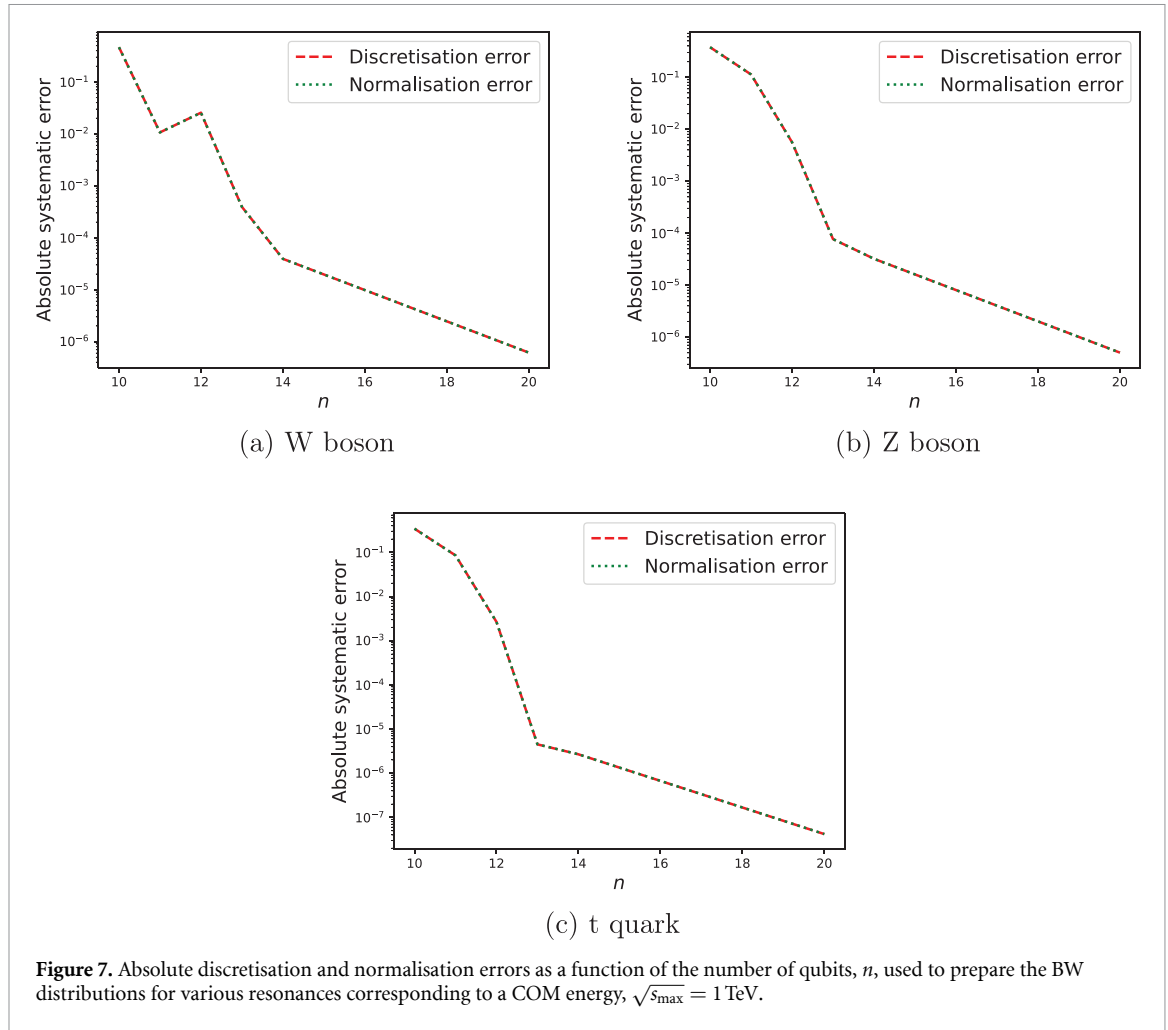


Table 1. Comparison of the metrics for the prepared distributions for each resonance for both COM energies, using either variational or Fourier expansion methods. ‘Res’ stands for ‘Resonance’. For ‘Accuracy’, ‘Optimised’ refers to the variationally-trained circuits with optimised hyperparameters, and ‘Matched’ and ‘More’ refer to the Fourier-expansion-method circuits, with either similar, or much greater accuracy, to the equivalent variationally trained circuits, respectively.

$\sqrt{s_{\max}}$	Method	Res	Accuracy	n	g_{1q}	g_{2q}	ϵ_s^{CMSE}	JSD	p_{success}
100 GeV	Variational	W	Optimised	6	186	150	1.22×10^{-4}	3.31×10^{-6}	N/A
	Fourier	W	Matched ($d = 250$)	15	1592	1638	1.22×10^{-4}	6.73×10^{-5}	3.12%
200 GeV	Variational	W	Optimised	9	180	152	4.01×10^{-4}	0.029	N/A
		Z	Optimised	9	234	200	4.77×10^{-3}	0.024	N/A
		t	Optimised	9	126	104	8.16×10^{-3}	0.070	N/A
	Fourier	W	Matched ($d = 175$)	18	1576	1684	3.28×10^{-5}	0.032	0.9%
			More ($d = 250$)	18	1602	1686	2.13×10^{-6}	0.003	0.9%
		Z	Matched ($d = 130$)	18	1591	1692	3.28×10^{-5}	0.031	1.1%
			More ($d = 170$)	18	1601	1686	2.58×10^{-6}	0.014	1.2%
		t	Matched ($d = 85$)	17	825	906	3.41×10^{-5}	0.070	1.4%
More ($d = 180$)	18	1593	1692	2.45×10^{-6}	0.010	1.2%			



4. Example applications

To illustrate the generality of the approach sketched above, we will focus on one particular example, the decay of the tau lepton into three fermions

$$\begin{aligned}\tau^- &\rightarrow \nu_\tau e^- \bar{\nu}_e, \\ p &= k_1 + k_2 + k_3,\end{aligned}\tag{29}$$

where p and k_i are the initial and final state four momenta, respectively. In this case, the matrix element squared can be written as¹⁵

$$|\mathcal{M}|^2 = -\frac{\alpha^2 \pi^2}{\sin^4 \theta_w} \frac{s_1^2 - M_\tau^2 s_1}{(s_2 - M_W^2)^2 + \Gamma_W M_W},\tag{30}$$

where the invariants, s_1 and s_2 , are equal to $(p_1 + p_3)^2$ and $(p_2 + p_3)^2$, respectively. The electroweak coupling is denoted by α , while the weak mixing angle is θ_w . The mass and width of the W bosons are denoted by m_W and Γ_W , respectively, while M_τ is the mass of the tau lepton¹⁶. It is worth pointing out that the expression in equation (30) contains a W-boson propagator term (see equation (3)). This originates from the fact that the electron and anti-electron neutrino are produced through the decay of an intermediate W boson, as depicted in the Feynman diagram in figure 8.

¹⁵ The expression has been obtained with the help of the FEYNARTS [64, 65] and FORMCALC [66] packages to the lowest order in perturbation theory.

¹⁶ It is worth noting that the expression for the matrix element in equation (30) assumes a massless electron, which is a common approximation in HEP. This is because the mass of the electron (0.5 MeV) is negligible with respect to a collision energy (13.6 TeV for the LHC, for example).

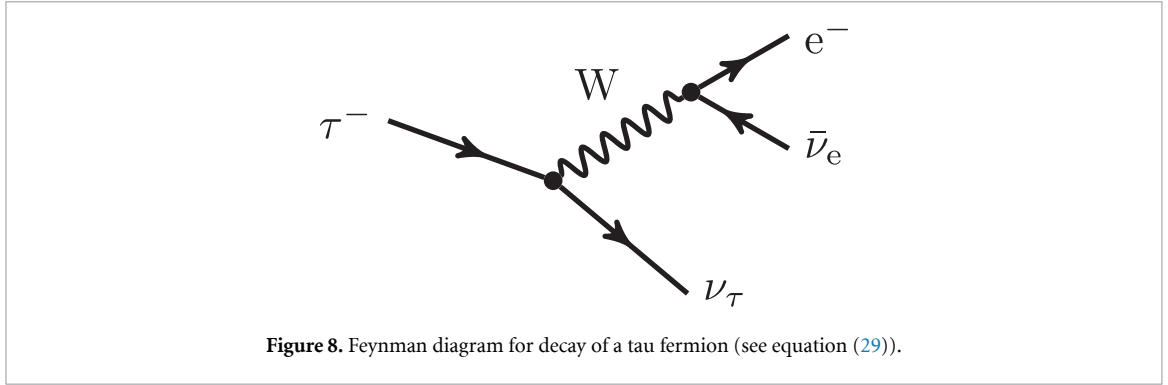


Figure 8. Feynman diagram for decay of a tau fermion (see equation (29)).

For a three-particle phase space, the integration measure reads [36]

$$\begin{aligned} d\Phi_3 &= \prod_{i=1}^3 \frac{d^3 p_i}{2E_i} \delta^4(p - k_1 - k_2 - k_3) \\ &= \frac{1}{32s} ds_1 ds_2 d\Omega_1 d\phi_3, \end{aligned} \quad (31)$$

where Ω_1 is a solid angle, and ϕ a rotation angle. The integration boundaries are $s_1 \in [0, s - s_2]$ and $s_2 \in [0, s]$, respectively, where s is the available energy of the system, i.e. $s = M_\tau^2$ for a tau decay. For what follows, we will discuss this particular example and some modifications of it (either by simplifying it or rendering it more complex).

4.1. One-dimensional integration

To demonstrate the methodology, for simplicity, we start by considering only the numerator of equation (30). By neglecting the constant factors and integrating only over the two invariants s_1 and s_2 , one obtains the following integral

$$\sigma = \int_0^s ds_2 \int_0^{s-s_2} ds_1 (s_1^2 - s_1 M_\tau^2), \quad (32)$$

which has analytical value $\sigma = \frac{s^4}{12} - M_\tau^2 \frac{s^3}{6}$

In order to implement such a calculation using the QMCI engine, noting that the integral over s_2 trivially gives s , then we encode the calculation as the following one-dimensional integration with a two-dimensional cut function, $C(s_1, s_2)$,

$$\sigma = s \left(\int_0^s ds_1 s_1^2 C(s_1, s_2) - M_\tau^2 \int_0^s ds_1 s_1 C(s_1, s_2) \right), \quad (33)$$

where $C(s_1, s_2) = 1$, if $s_1 + s_2 < s$, and $C(s_1, s_2) = 0$, otherwise. If we compare this to the general form of the expectation calculated using the QMCI engine in equation (7), and identify $s_1 = x$ and $s_2 = y$ (this identification will remain for all subsequent examples), we observe that there is freedom in the integrand separation, in terms of what to define as the probability distribution, $f_{s_1, s_2}(s_1, s_2)$, and the function applied $g(s_1, s_2)$. In this case, for simplicity, we choose to set $f_{s_1, s_2}(s_1, s_2) = U(s_1)U(s_2)$, where $U(\cdot)$ is the uniform distribution, and $g(s_1, s_2) = s_1^2$ or $g(s_1, s_2) = s_1$ ¹⁷.

As discussed previously in section 2.3.2, the QMCI engine contains powerful, in-built functionality for performing such a calculation; the cut function can be straightforwardly implemented using thresholding operations, whilst the engine contains efficient methods for calculating both $g(s_1, s_2) = s_1^2$ and $g(s_1, s_2) = s_1$. The QMCI engine can thus be used to build efficient, low depth circuits that give estimates of both expectation values on the RHS of equation (33).

For this example, and the ones described later in sections 4.2 and 4.3, we considered three different levels of expected precision, corresponding to upper bounds on the expected RMSE of the final estimator of the order 10%, 1%, and 0.1%,¹⁸ respectively (not accounting for the systematic errors related to the state

¹⁷ Note that this choice does not leverage the capabilities of the QMCI engine in terms of decomposing the integrand into building blocks, as discussed previously—however, that is not the purpose of this initial example.

¹⁸ The per-mille precision represents the typical accuracy of classical MCI calculation in HEP.

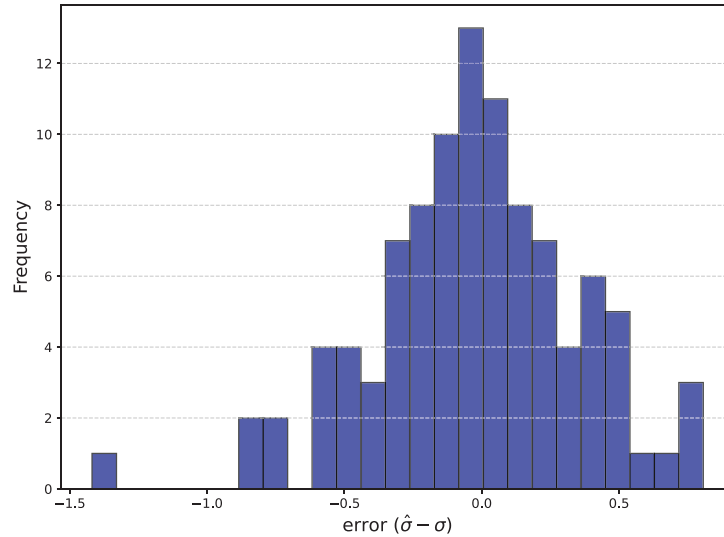


Figure 9. Histogram of the error for 100 runs of QMCI estimating the RHS of equation (33), for an expected RMSE upper-bound precision of 10%.

preparation discussed in section 3). For illustration purposes, we run noiseless simulations of the circuits for the case of 10% precision.

We considered a decay-like process where $\sqrt{s_{\max}} = M_{\tau} = 1.776 \text{ GeV}$, giving an analytical value $\sigma = -8.248$.¹⁹ We set $n = 5$ for each of the dimensions, s_1 and s_2 , respectively. This results in a final QMCI circuit containing 23 qubits in total. The circuits to prepare the uniform distributions used to model the probability distributions for each dimension were loaded directly in the QMCI engine using the in-built state-preparation library, and these then combined to construct the bivariate probability distribution, $f_{S_1, S_2}(s_1, s_2)$.

In order to demonstrate the validity of the approach, we ran noiseless simulations using the QMCI engine to numerically estimate the RHS of equation (33), which we denote as $\hat{\sigma}$, 100 different times, based on an expected precision of 10%. The numerical experiments, and the ones described later in sections 4.2 and 4.3, were carried out using QULACS' noiseless state-vector simulator [67]. The QAE algorithm used in all cases was an in-built implementation of the *maximum-likelihood* QAE algorithm [68], that is optimised to be able to run for any given number of samples, thereby maximising performance.

Figure 9 gives a histogram of the error, $\hat{\sigma} - \sigma$, where we can see that the results are as expected, with nearly all (except for a single outlier) values falling within an expected precision of 10%.

Using the in-built resource mode of the QMCI engine (described in section 2.3.2), we also calculated the resources required to perform this calculation to the three different levels of expected precision. Table 2 gives these resources for both NISQ and fault-tolerant compilation. We note that the number of qubits required is in general small [$\mathcal{O}(10)$], and in particular is well within the reach of current quantum hardware (this is assuming that the systematic errors based on using 5-qubit circuits to prepare the uniform distributions do not excessively affect calculations—something that is not observed for this case for 10% precision). However, the number of operations required for NISQ compilation almost certainly means that logical qubits would be required; considering the largest circuit for an expected precision of 10% has 1.80×10^6 CX gates, then a simple rule-of-thumb calculation suggests one would need a machine with a two-qubit gate fidelity of approximately $1 - 1/(1.80 \times 10^6) = 99.99994\%$ —well out of reach of current quantum hardware. Given the substantial two-qubit gate counts, even the application of targeted error mitigation strategies—such as noise-aware QAE techniques [69]—is unlikely to suffice. There is only so much error mitigation can do in the presence of such high noise levels, and the accumulation of errors is likely to compromise the accuracy of the results, regardless. We note that the number of T-gate operations for fault-tolerant execution are also high; for example, for a 0.1% precision, as would be required to compete with current classical MCI methods, the largest circuit requires 1.71×10^9 T gates. However, it is worth noting these gate counts are likely to reduce significantly in future as more research into optimised synthesis is carried out (see section 10.3 of [32] for a detailed discussion of this topic), and, therefore, these values should be regarded as loose upper bounds, and treated with a degree of uncertainty.

¹⁹ Given that this quantity is a proxy of a cross section, it is not physical, and thus not necessarily positive.

Table 2. Resources required to estimate the RHS of equation (33) using QMCI for various expected precisions, for both NISQ and fault-tolerant resource mode.

Compilation	Resource	Metric	Precision		
			10%	1%	0.1%
NISQ	Number of qubits	Largest across circuits	24	24	24
	CX gates	Total number across circuits	3.99×10^6	4.68×10^7	6.26×10^8
		Total depth across circuits	2.52×10^6	2.94×10^7	3.93×10^8
		Number in largest circuit	1.80×10^6	1.14×10^7	1.69×10^8
		Depth of largest circuit	1.13×10^6	7.18×10^6	1.06×10^8
	All gates	Total number across circuits	7.97×10^6	9.34×10^7	1.25×10^9
		Total depth across circuits	4.71×10^6	5.51×10^7	7.37×10^8
		Number in largest circuit	3.59×10^6	2.28×10^7	3.37×10^8
		Depth of largest circuit	2.12×10^6	1.35×10^7	1.99×10^8
	Fault tolerant	Number of qubits	Largest across circuits	35	35
T gates		Total number across circuits	8.62×10^6	3.49×10^8	5.85×10^9
		Total depth across circuits	7.21×10^6	2.92×10^8	4.89×10^9
		Number in largest circuit	3.83×10^6	9.62×10^7	1.71×10^9
		Depth of largest circuit	3.21×10^6	8.06×10^7	1.43×10^9

4.2. Separable two-dimensional integration

We now move on to the expression for the actual integration of the tau decay given in equation (30), by also considering the denominator (i.e. the propagator). We again omit the integration over the angles. The integral then becomes

$$\sigma \propto \int_0^s \int_0^{s-s_2} ds_1 ds_2 \frac{s_1^2 - s_1 M_\tau^2}{(s_2 - M_W^2)^2 + (M_W \Gamma_W)^2}. \quad (34)$$

The analytical solution to this integral is given in the [appendix](#).

In order to implement such a calculation using the QMCI engine, we first note that we can express this integral in the following way

$$\begin{aligned} \sigma \propto & \int_0^s \int_0^s ds_1 ds_2 \frac{1}{(s_2 - M_W^2)^2 + (M_W \Gamma_W)^2} s_1^2 C(s_1, s_2) \\ & - M_\tau^2 \int_0^s \int_0^s ds_1 ds_2 \frac{1}{(s_2 - M_W^2)^2 + (M_W \Gamma_W)^2} s_1 C(s_1, s_2), \end{aligned} \quad (35)$$

where the cut function, $C(s_1, s_2)$, is the same as previously. Then, recalling the general form of an expectation calculated using the QMCI engine in equation (7), and following the discussion in section 2.4, we see that each individual integral has the canonical form, in terms of products of building blocks, that we require for performing a generic cross-section calculation using the QMCI engine. We can identify the probability distribution, $f_{s_1 s_2}$, as a product of univariate BW distributions [or rather, in this case, the product of an uniform distribution for the dimension s_1 , and an univariate W-boson BW distribution for the dimension s_2 , i.e. $f_{s_1 s_2}(s_1, s_2) = U(s_1)BW_W(s_2)$]. Then, the functions applied, $g(\cdot)$, are just products of moments of the integration variables [or rather in this case simply the univariate products $g(s_1, s_2) = s_1^2$ or $g(s_1, s_2) = s_1$].

For this example, for demonstration purposes, it is important that we integrate across the range of the BW peak, and therefore we set $\sqrt{s_{\max}} = 100 \text{ GeV}^{20}$. The analytical value in this case is $\sigma = 3.162 \times 10^8$. As discussed in section 3, we make use of a trained PQC with $n = 6$ (specifically the optimal circuit given in table 1) to prepare the BW distribution for the W-boson propagator used to model the probability distributions for dimension s_2 , and use the QMCI engine's state-preparation library to load a $n = 6$ circuit preparing the uniform distribution used to model the probability distribution for dimension s_1 . This results in a QMCI circuit containing 27 qubits in total. We again construct $f_{s_1 s_2}(s_1, s_2)$ by combining the circuits.

In order to demonstrate the validity of our generic approach, we carried out numerical experiments by running noiseless simulations to numerically estimate the RHS of equation (35), 24 different times, for an expected precision of 10%.

²⁰ Note that this calculation does not correspond to a physical tau decay (or indeed a physical process).

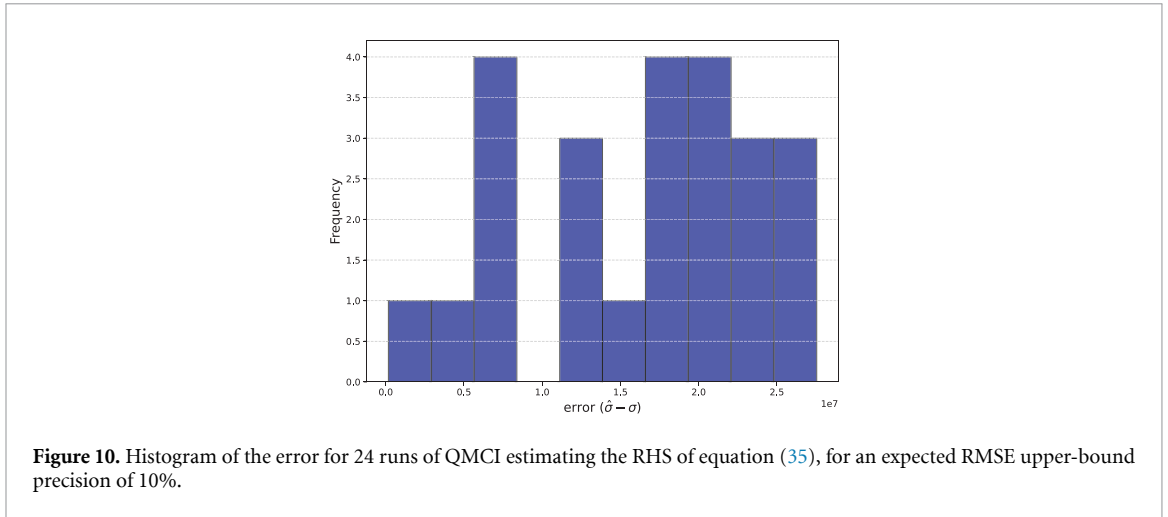


Figure 10 gives a histogram of the error, $\hat{\sigma} - \sigma$, where we again see all values within the expected precision of 10%. However, in this case there appears to be a bias in the results (as they are not centred around 0). Some bias is likely expected given the discussion of systematic errors regarding the state preparation of BW distributions given in section 3—a source of error that was notably not applicable to the previous example. In this case, the observed bias is approximately an order of magnitude smaller than the required precision, and is therefore negligible. However, for higher levels of precision—such as for 1% and 0.1%—one would likely require state-preparation circuits with smaller corresponding systematic errors, in order to still obtain accurate results. In particular, a quantitative analysis of the effect of such systematic errors on the final estimation error is needed to be able to fully characterise these QMCI calculations. However, such a detailed analysis would require a significant extension to the error analyses detailed in section 3 (which are derived from [33]). This is because, for this particular example, systematic errors from the state preparation only indirectly affect the final estimation error via the thresholding condition; the s_2 dimension is not directly integrated over, but is summed with the s_1 dimension to form a new dimension, and this new dimension used for the threshold condition, thereby indirectly propagating the error in the state preparation to the final estimation. An extension of the error analyses to be able to provide a quantitative treatment for such cases is well beyond the scope of this article, constituting an independent research direction, and is therefore left for future work.

We used the resource mode to calculate the resources for the three expected levels of precision. Table 3 gives the resources for NISQ and fault-tolerant compilation. The results are similar to those found in section 4.1, and therefore a similar analysis to that described there applies. However, in this case it is worth noting that due to the more complicated circuit used to prepare the BW distribution for the s_2 dimension, then the qubit counts are slightly larger, and the gate counts are also approximately an order of magnitude larger.

4.3. Non-separable two-dimensional integration

Finally, in order to increase the complexity of the problem and mimic the more general case of equation (4), we compute the following

$$\sigma \propto \int_0^s \int_0^{s-s_2} ds_1 ds_2 \frac{s_1^2 s - s_1 M_\tau^2 s + s_1 M_\tau^2 s_2}{(s_2 - M_W^2)^2 + (M_W \Gamma_W)^2}, \quad (36)$$

which amounts in practice to, in addition to calculating the same terms as in equation (35) (multiplied by the constant, s), calculating the following, non-separable, two-dimensional integral

$$I_1 = \int_0^s \int_0^{s-s_2} ds_1 ds_2 \frac{s_1 M_\tau^2 s_2}{(s_2 - M_W^2)^2 + (M_W \Gamma_W)^2}. \quad (37)$$

The analytical solution to this additional integral is also provided in the [appendix](#). The analytical value of equation (36) is then $\sigma = 3.179 \times 10^{12}$.

Table 3. Resources required to estimate the RHS of equation (35) using QMCI for various expected precisions, for both NISQ and fault-tolerant resource mode.

Compilation	Resource	Metric	Precision		
			10%	1%	0.1%
NISQ	Number of qubits	Largest across circuits	28	28	28
	CX gates	Total number across circuits	1.34×10^7	1.44×10^8	1.49×10^9
		Total depth across circuits	7.88×10^6	8.43×10^7	8.74×10^8
		Number in largest circuit	4.86×10^6	6.32×10^7	7.48×10^8
		Depth of largest circuit	2.85×10^6	3.71×10^7	4.39×10^8
	All gates	Total number across circuits	2.72×10^7	2.91×10^8	3.02×10^9
		Total depth across circuits	1.45×10^7	1.56×10^8	1.62×10^9
		Number in largest circuit	9.84×10^6	1.28×10^8	1.51×10^9
		Depth of largest circuit	5.27×10^6	6.85×10^7	8.11×10^8
	Fault tolerant	Number of qubits	Largest across circuits	41	41
T gates		Total number across circuits	5.37×10^8	6.97×10^9	8.23×10^{10}
		Total depth across circuits	5.21×10^8	6.75×10^9	7.98×10^{10}
		Number in largest circuit	2.18×10^8	3.08×10^9	4.21×10^{10}
		Depth of largest circuit	2.11×10^8	2.99×10^9	4.08×10^{10}

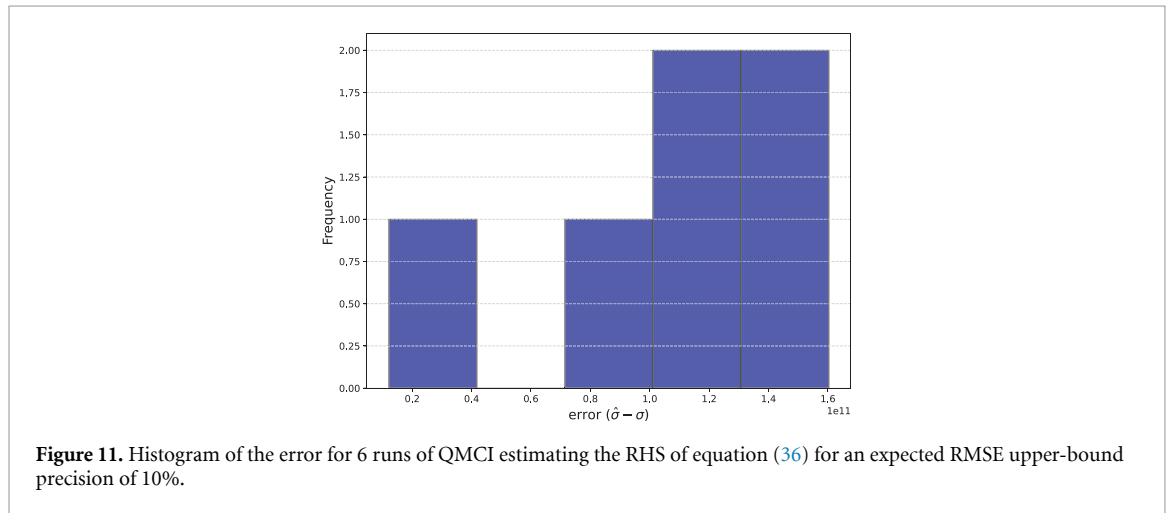


Figure 11. Histogram of the error for 6 runs of QMCI estimating the RHS of equation (36) for an expected RMSE upper-bound precision of 10%.

For this example, the setup is the same as in section 4.2. In order to demonstrate the validity of our approach, we carried out numerical experiments by running noiseless simulations to numerically estimate the RHS of equation (36), 6 times, for an expected precision of 10%.

Figure 11 gives a histogram of the error, $\hat{\sigma} - \sigma$. The results are broadly similar to those in section 4.2, with all values within the expected precision, and with a bias again observed (this is not surprising, as the calculations only differ by the additional term, which is subdominant).

We analysed the resources required to run the circuits at the nominal precisions using the resource mode. Table 4 gives these for NISQ and fault-tolerant compilation. Once again the results are similar to the previous sections, and therefore a similar analysis again applies. In this case, as compared to the example in section 4.2—which itself required greater resources than the example in section 4.1—both the qubit counts and number of gates are larger, by approximately an additional order of magnitude in the latter case. This increase is due to the extra 2D integration that is performed (i.e. calculating $\mathbb{E}[s_1 s_2]$), which as discussed in section 2.3.1 requires greater resources than when calculating univariate expectations. Thus, we see that as the dimensionality or complexity of the integrals increases, so too do the required resources.

Based upon the resource analysis here, and that in the previous sections, it is clear that to be able to perform state-of-the-art HEP calculations using QMCI in the future—whereby the dimensionality and complexity of the calculations will be much greater than for these examples—we will require significantly

Table 4. Resources required to estimate the RHS of equation (36) using QMCI for various expected precisions, for both NISQ and fault-tolerant resource mode.

Compilation	Resource	Metric	Precision		
			10%	1%	0.1%
NISQ	Number of qubits	Largest across circuits	28	28	28
	CX gates	Total number across circuits	7.39×10^7	6.15×10^8	5.09×10^9
		Total depth across circuits	4.34×10^7	3.61×10^8	2.99×10^9
		Number in largest circuit	3.39×10^7	2.71×10^8	1.20×10^9
		Depth of largest circuit	1.99×10^7	1.59×10^8	7.03×10^8
	All gates	Total number across circuits	1.50×10^8	1.24×10^9	1.03×10^{10}
		Total depth across circuits	8.02×10^7	6.67×10^8	5.52×10^9
		Number in largest circuit	6.86×10^7	5.49×10^8	2.42×10^9
		Depth of largest circuit	3.68×10^7	2.94×10^8	1.30×10^9
	Fault tolerant	Number of qubits	Largest across circuits	41	41
T gates		Total number across circuits	3.39×10^9	4.08×10^{10}	2.72×10^{11}
		Total depth across circuits	3.29×10^9	3.95×10^{10}	2.63×10^{11}
		Number in largest circuit	1.65×10^9	2.14×10^{10}	6.97×10^{10}
		Depth of largest circuit	1.60×10^9	2.07×10^{10}	6.75×10^{10}

greater resources than are available with near-term hardware. Thus, this clearly remains an application for the future, fault-tolerant era of quantum computing.

5. Conclusion

Quantum technology holds the promise of solving complex problems that are classically intractable, or providing more efficient methods for tackling existing challenges than classical approaches allow. Meanwhile, the field of HEP—which seeks to uncover the fundamental laws of nature at the smallest scales—demands immense computational resources.

In this work, we investigated whether quantum computing could, in the future, help address key computational bottlenecks in HEP. Building on, and going far beyond, the work presented in [21], where M.P. first introduced the idea of using QMCI for cross-section calculations in HEP, we have made significant advancements in this direction.

We have developed a general approach for integrating non-trivial cross sections in HEP, in terms of decomposing the general integrand into products of constituent building blocks. This work leverages the Fourier QMCI method implemented in QUANTINUUM's QMCI engine [32] (developed in part by I.W.) along with several other key proprietary features of the engine. Specifically, we introduced two distinct approaches for preparing relativistic BW distributions on quantum registers—functions that appear as one of the key building blocks of generic integrands in cross-section calculations.

To demonstrate the method's capabilities, we performed two-dimensional integrations for several examples that arguably represent some of the most challenging integrands investigated to date for quantum integration. However, it is still worth remarking that these examples remain significantly less complex than the high-dimensional integrands encountered in state-of-the-art classical calculations.

Our findings suggest that such applications are unlikely to be practical during the NISQ era. Instead, they are expected to become viable when fault-tolerant quantum devices are available. Notably, the resource estimates we provide for fault tolerance represent loose upper bounds. We anticipate that significant improvements will be achieved in the future—and thus significantly reduced resource requirements—as the field of fault-tolerant compilation matures beyond its infancy.

While this work represents a significant first step toward the quantum integration of cross sections in HEP, there are several areas for improvement. One limitation is the use of uniform spacing for representing underlying probability distributions with qubits, which is an inherent requirement of the QMCI engine. Extending the engine's functionality to allow for non-uniform spacing could significantly reduce both the number of qubits and the quantum gates required for integration. Additionally, while the method is general and theoretically applicable to integrals of arbitrary dimensionality (beyond the two-dimensional examples studied here), its efficiency will decrease for high-dimensional problems due to the polynomial scaling in the number of terms to calculate. Adapting the method to achieve more favourable scaling would enhance its

practicality, and studying how circuit size scales heuristically with integration dimensionality would be particularly valuable.

Another avenue for improvement concerns the handling of experimental constraints, as discussed in section 2.1, as these could be considered in further generality. Restricted integration domains based on constraints such as sums, maximum/minimum etc can be easily implemented via the QMCI engine's enhanced P-builder functionality. However, in more general cases, the cut functions may be much more complex, or even non-analytical, and this thus requires further exploration to ensure efficient integration for these cases. Similarly, extending the method to accommodate more general forms of cross-section integrands would broaden its applicability. Accurate preparation of relativistic BW distributions, which is crucial to this method, represents another area for further investigation. Alternative state-preparation methods that are tailored to the particular distribution could enhance both accuracy and resource efficiency, and also could be of use in general to the HEP community for other future quantum computing applications where resonances are required to be modelled. In particular, understanding in detail the systematic errors associated with state preparation on the final estimation error is critical for performing precise integrations, and this issue warrants more detailed exploration. Finally, event sampling based on underlying distributions is a ubiquitous task in HEP. Exploring whether quantum computing and quantum integration could enhance efficiency or accuracy in this area would be of particular interest, and could provide significant advancements for the field. In summary, while this work lays a solid foundation for quantum integration in HEP, addressing these challenges will significantly expand its potential and applicability.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

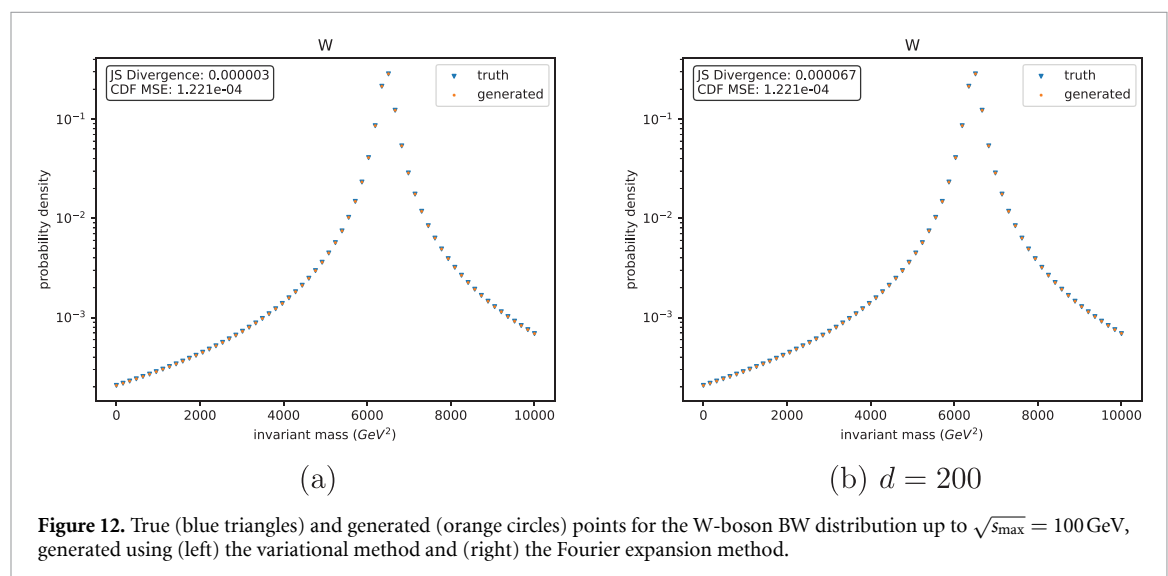
Acknowledgments

The authors would like to thank Matthew DeCross and Julien Sorci for their careful reviewing of the manuscript. The authors thank Steven Herbert and Alexandre Krajenbrink for their support in this work and interesting discussions (and the latter for also reviewing the manuscript). The authors also want to thank Matthias Rosenkranz for interesting discussions. M.P. acknowledges support by the German Research Foundation (DFG) through the Research Training Group RTG2044.

Appendix

A.1. State preparation of relativistic BW distributions

In this section, we have some additional plots related to state preparation of BW distributions.



A.2. Analytical expressions

In this section, we provide the analytical expressions for several integrals required in the main text.

First, the analytical expression for the integral in equation (34) in section 4.2 reads

$$\begin{aligned}
 I &= \int_0^s \int_0^{s-s_2} ds_1 ds_2 \frac{s_1^2 - s_1 M_\tau^2}{(s_2 - M_W^2)^2 + (M_W \Gamma_W)^2} \\
 &= \frac{1}{6\Gamma_W M_W} \left[\Gamma_W M_W (\Gamma_W^2 M_W^2 - 3(M_W^2 - s)(M_\tau^2 + M_W^2 - s)) \log \left[\frac{\Gamma_W^2 M_W^2 + (M_W^2 - s)^2}{M_W^2 (\Gamma_W^2 + M_W^2)} \right] \right. \\
 &\quad + \tan^{-1} \left(\frac{M_W}{\Gamma_W} \right) \left(3\Gamma_W^2 M_W^2 (M_\tau^2 + 2M_W^2 - 2s) - (M_W^2 - s)^2 (3M_\tau^2 + 2M_W^2 - 2s) \right) \\
 &\quad + \left((M_W^2 - s)^2 (3M_\tau^2 + 2M_W^2 - 2s) - 3\Gamma_W^2 M_W^2 (M_\tau^2 + 2M_W^2 - 2s) \right) \cot^{-1} \left(\frac{\Gamma_W M_W}{M_W^2 - s} \right) \\
 &\quad \left. + \Gamma_W M_W s (-3M_\tau^2 - 4M_W^2 + 5s) \right]. \tag{38}
 \end{aligned}$$

Then, the analytical result for the additional integral in equation (37) in section 4.3 reads

$$\begin{aligned}
 I_1 &= \int_0^s \int_0^{s-s_2} ds_1 ds_2 \frac{s_1 M_\tau^2 s_2}{(s_2 - M_W^2)^2 + (M_W \Gamma_W)^2} \\
 &= \frac{M_\tau^2}{4\Gamma_W} \left[-2M_W \left(\Gamma_W^2 (2s - 3M_W^2) + (M_W^2 - s)^2 \right) \left(\cot^{-1} \left(\frac{\Gamma_W M_W}{M_W^2 - s} \right) - \tan^{-1} \left(\frac{M_W}{\Gamma_W} \right) \right) \right. \\
 &\quad \left. + \Gamma_W \left(\Gamma_W^2 M_W^2 - 3M_W^4 + 4M_W^2 s - s^2 \right) \log \left(\frac{M_W^2 (\Gamma_W^2 + M_W^2)}{\Gamma_W^2 M_W^2 + (M_W^2 - s)^2} \right) + \Gamma_W s (4M_W^2 - 3s) \right]. \tag{39}
 \end{aligned}$$

ORCID iDs

Ifan Williams  0009-0000-3033-5619

Mathieu Pellen  0000-0001-5324-2765

References

- [1] Aad G *et al* (ATLAS Collaboration) 2012 Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC *Phys. Lett. B* **716** 1–29
- [2] Chatrchyan S *et al* (CMS Collaboration) 2012 Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC *Phys. Lett. B* **716** 30–61
- [3] Buckley A 2020 Computational challenges for MC event generation *J. Phys.: Conf. Ser.* **1525** 012023
- [4] Valassi A *et al* (HSF Physics Event Generator WG Collaboration) 2021 Challenges in Monte Carlo event generator software for high-luminosity LHC *Comput. Softw. Big Sci.* **5** 12
- [5] ATLAS Collaboration 2022 ATLAS software and computing HL-LHC roadmap *Technical Report* (CERN) (available at: <https://cds.cern.ch/record/2802918>)
- [6] Nachman B, Provasoli D, de Jong W A and Bauer C W 2021 Quantum algorithm for high energy physics simulations *Phys. Rev. Lett.* **126** 062001
- [7] Bepari K, Malik S, Spannowsky M and Williams S 2021 Towards a quantum computing algorithm for helicity amplitudes and parton showers *Phys. Rev. D* **103** 076020
- [8] Pérez-Salinas A, Cruz-Martinez J, Alhajri A A and Carrazza S 2021 Determining the proton content with a quantum computer *Phys. Rev. D* **103** 034027
- [9] Bepari K, Malik S, Spannowsky M and Williams S 2022 Quantum walk approach to simulating parton showers *Phys. Rev. D* **106** 056002
- [10] Bravo-Prieto C, Baglio J, Cè M, Francis A, Grabowska D M and Carrazza S 2022 Style-based quantum generative adversarial networks for Monte Carlo events *Quantum* **6** 777
- [11] Ramírez-Uribe S, Rentería-Olivo A E, Rodrigo G, Sborlini G F R and Silva L V 2022 Quantum algorithm for Feynman loop integrals *J. High Energy Phys.* **JHEP05(2022)100**
- [12] Deliyannis P, Sud J, Chamaki D, Webb-Mack Z, Bauer C W and Nachman B 2022 Improving quantum simulation efficiency of final state radiation with dynamic quantum circuits *Phys. Rev. D* **106** 036007
- [13] Gustafson G, Prestel S, Spannowsky M and Williams S 2022 Collider events on a quantum computer *J. High Energy Phys.* **JHEP11(2022)035**
- [14] Clemente G, Crippa A, Jansen K, Ramírez-Uribe S, Rentería-Olivo A E, Rodrigo G, Sborlini G F R and Silva L V 2023 Variational quantum eigensolver for causal loop Feynman diagrams and directed acyclic graphs *Phys. Rev. D* **108** 096035
- [15] Chawdhry H A and Pellen M 2023 Quantum simulation of colour in perturbative quantum chromodynamics *SciPost Phys.* **15** 205
- [16] Bauer C W, Chigusa S and Yamazaki M 2024 Quantum parton shower with kinematics *Phys. Rev. A* **109** 032432
- [17] Ramírez-Uribe S, Rentería-Olivo A E and Rodrigo G 2024 Quantum querying based on multicontrolled Toffoli gates for causal Feynman loop configurations and directed acyclic graphs (arXiv:2404.03544 [quant-ph])
- [18] Montanaro A 2015 Quantum speedup of Monte Carlo methods *Proc. R. Soc. A* **471** 20150301
- [19] Herbert S 2022 Quantum Monte Carlo integration: the full advantage in minimal circuit depth *Quantum* **6** 823

- [20] Brassard G, Mosca M and Tapp A 2002 Quantum amplitude amplification and estimation *Quantum Comput. Inf.* **305** 53–74
- [21] Agliardi G, Grossi M, Pellen M and Prati E 2022 Quantum integration of elementary particle processes *Phys. Lett. B* **832** 137228
- [22] Grinko D, Gacon J, Zoufal C and Woerner S 2021 Iterative quantum amplitude estimation *npj Quantum Inf.* **7** 52
- [23] de Lejarza J J M, Cieri L, Grossi M, Vallecorsa S and Rodrigo G 2024 Loop Feynman integration on a quantum computer *Phys. Rev. D* **110** 074031
- [24] de Lejarza J J M, Rentería-Estrada D F, Grossi M and Rodrigo G 2025 Quantum integration of decay rates at second order in perturbation theory *Quantum Sci. Technol.* **10** 025026
- [25] de Lejarza J J M, Grossi M, Cieri L and Rodrigo G 2023 Quantum Fourier iterative amplitude estimation *2023 Int. Conf. on Quantum Computing and Engineering* (IEEE) (<https://doi.org/10.1109/QCE57702.2023.00071>)
- [26] McClean J R, Boixo S, Smelyanskiy V N, Babbush R and Neven H 2018 Barren plateaus in quantum neural network training landscapes *Nat. Commun.* **9** 4812
- [27] Cerezo M, Sone A, Volkoff T, Cincio L and Coles P J 2021 Cost function dependent barren plateaus in shallow parametrized quantum circuits *Nat. Commun.* **12** 1791
- [28] Sharma K, Cerezo M, Cincio L and Coles P J 2022 Trainability of dissipative perceptron-based quantum neural networks *Phys. Rev. Lett.* **128** 180505
- [29] Wang S, Fontana E, Cerezo M, Sharma K, Sone A, Cincio L and Coles P J 2021 Noise-induced barren plateaus in variational quantum algorithms *Nat. Commun.* **12** 6961
- [30] Marrero C O, Kieferová M and Wiebe N 2020 Entanglement induced barren plateaus (arXiv:2010.15968 [quant-ph])
- [31] Arrasmith A, Cerezo M, Czarnik P, Cincio L and Coles P J 2021 Effect of barren plateaus on gradient-free optimization *Quantum* **5** 558
- [32] Akhalwaya I Y *et al* 2023 A modular engine for quantum Monte Carlo integration (arXiv:2308.06081 [quant-ph])
- [33] Cui J, de Brouwer P J S, Herbert S, Intallura P, Kargi C, Korpas G, Krajenbrink A, Shoosmith W, Williams I and Zheng B 2024 Quantum Monte Carlo integration for simulation-based optimisation (arXiv:2410.03926 [quant-ph])
- [34] Denner A, Lang J-N and Pellen M 2021 Full NLO QCD corrections to off-shell $t\bar{t}b\bar{b}$ production *Phys. Rev. D* **104** 056018
- [35] Huss A, Huston J, Jones S and Pellen M 2023 Les Houches 2021—physics at TeV colliders: report on the standard model precision wishlist *J. Phys. G: Nucl. Part. Phys.* **50** 043001
- [36] Byckling E and Kajantie K 1971 *Particle Kinematics: (Chapters I–VI, X)* (University of Jyväskylä)
- [37] Lepage G P 1978 A new algorithm for adaptive multidimensional integration *J. Comput. Phys.* **27** 192
- [38] Kharraziha H and Moretti S 2000 The Metropolis algorithm for on-shell four momentum phase space *Comput. Phys. Commun.* **127** 242–60
- [39] Kröninger K, Schumann S and Willenberg B 2015 (MC)**3—a multi-channel Markov chain Monte Carlo algorithm for phase-space sampling *Comput. Phys. Commun.* **186** 1–10
- [40] Yallup D, Janßen T, Schumann S and Handley W 2022 Exploring phase space with nested sampling *Eur. Phys. J. C* **82** 8
- [41] Cagnina S L, Grunwald C, Janßen T, Kröninger K and Schumann S 2024 Phase space sampling with Markov chain Monte Carlo methods (arXiv:2412.12963 [hep-ph])
- [42] Li Z, Wang J, Yan Q-S and Zhao X 2016 Efficient numerical evaluation of Feynman integrals *Chin. Phys. C* **40** 033103
- [43] de Doncker E, Almulihi A and Yuasa F 2018 High-speed evaluation of loop integrals using lattice rules *J. Phys.: Conf. Ser.* **1085** 052005
- [44] Borowka S, Heinrich G, Jahn S, Jones S P, Kerner M and Schlenk J 2019 A GPU compatible quasi-Monte Carlo integrator interfaced to pySecDec *Comput. Phys. Commun.* **240** 120–37
- [45] Grover L K 1996 A fast quantum mechanical algorithm for database search *Proc. 28th Annual ACM Symp. on Theory of Computing (STOC '96)* (Association for Computing Machinery) pp 212–9 (<http://doi.org/10.1145/237814.237866>)
- [46] Herbert S 2021 Every classical sampling circuit is a quantum sampling circuit (arXiv:2109.04842 [quant-ph])
- [47] Breit G and Wigner E 1936 Capture of slow neutrons *Phys. Rev.* **49** 519–31
- [48] Plesch M and Brukner Č 2011 Quantum-state preparation with universal gate decompositions *Phys. Rev. A* **83** 032302
- [49] Sanders Y R, Low G H, Scherer A and Berry D W 2019 Black-box quantum state preparation without arithmetic *Phys. Rev. Lett.* **122** 020502
- [50] Zoufal C, Lucchi A and Woerner S 2019 Quantum generative adversarial networks for learning and loading random distributions *npj Quantum Inf.* **5** 103
- [51] Zhang X-M, Li T and Yuan X 2022 Quantum state preparation with optimal circuit depth: implementations and applications *Phys. Rev. Lett.* **129** 230504
- [52] Bausch J 2022 Fast black-box quantum state preparation *Quantum* **6** 773
- [53] Rattew A G and Koczor B 2022 Preparing arbitrary continuous functions in quantum registers with logarithmic complexity (arXiv:2205.00519 [quant-ph])
- [54] Wodecki A, Marecek J, Kungurtsev V, Eichler P, Korpas G and Intallura P 2024 Spectral methods for quantum optimal control: artificial boundary conditions (arXiv:2403.14436 [quant-ph])
- [55] Benedetti M, Lloyd E, Sack S and Fiorentini M 2019 Parameterized quantum circuits as machine learning models *Quantum Sci. Technol.* **4** 043001
- [56] Kandala A, Mezzacapo A, Temme K, Takita M, Brink M, Chow J M and Gambetta J M 2017 Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets *Nature* **549** 242–6
- [57] Holmes Z, Sharma K, Cerezo M and Coles P J 2022 Connecting ansatz expressibility to gradient magnitudes and barren plateaus *PRX Quantum* **3** 010313
- [58] Grant E, Wossnig L, Ostaszewski M and Benedetti M 2019 An initialization strategy for addressing barren plateaus in parametrized quantum circuits *Quantum* **3** 214
- [59] Volkoff T and Coles P J 2021 Large gradients via correlation in random parameterized quantum circuits *Quantum Sci. Technol.* **6** 025008
- [60] Rosenkranz M, Brunner E, Marin-Sanchez G, Fitzpatrick N, Dilkes S, Tang Y, Kikuchi Y and Benedetti M 2024 Quantum state preparation for multivariate functions (arXiv:2405.21058 [quant-ph])
- [61] Childs A M and Wiebe N 2012 Hamiltonian simulation using linear combinations of unitary operations *Quantum Inf. Comput.* **12** 0901–24
- [62] Olson B, Hashmi I, Molloy K and Shehu A 2012 Basin hopping as a general and versatile optimization framework for the characterization of biological macromolecules *Adv. Artif. Intell.* **2012** 1–19

- [63] Nocedal J and Wright S 2006 *Numerical Optimization (Springer Series in Operations Research and Financial Engineering)* (Springer) pp 1–664
- [64] Kublbeck J, Bohm M and Denner A 1990 Feyn arts: computer algebraic generation of Feynman graphs and amplitudes *Comput. Phys. Commun.* **60** 165–80
- [65] Hahn T 2001 Generating Feynman diagrams and amplitudes with FeynArts 3 *Comput. Phys. Commun.* **140** 418–31
- [66] Hahn T and Perez-Victoria M 1999 Automatized one loop calculations in four-dimensions and D-dimensions *Comput. Phys. Commun.* **118** 153–65
- [67] Suzuki Y *et al* 2021 Qulacs: a fast and versatile quantum circuit simulator for research purpose *Quantum* **5** 559
- [68] Suzuki Y, Uno S, Raymond R, Tanaka T, Onodera T and Yamamoto N 2020 Amplitude estimation without phase estimation *Quantum Inf. Process.* **19** 1–17
- [69] Herbert S, Williams I, Guichard R and Ng D 2024 Noise-aware quantum amplitude estimation *IEEE Trans. Quantum Eng.* **5** 1–23