TrackHHL: The 1-Bit Quantum Filter for particle trajectory reconstruction

Xenofon Chiotopoulos[1,2,4,*], **Davide Nicotra**[2], **George Scriven**[2,3], **Kurt Driessens**[4], **Marcel Merk**[1,2], **Jochen Schütz**[3], **Jacco de Vries**[2], **and Mark H.M. Winands**[4]

[1]Nikhef National Institute for Subatomic Physics, Science Park 105, 1098 XG Amsterdam, The Netherlands
[2]Maastricht University, Faculty of Science and Engineering, Gravitational Waves and Fundamental Physics department, Duboisdomain 30, 6229 GT Maastricht, The Netherlands
[3]Hasselt University, Faculty of Sciences and Data Science Institute, Agoralaan gebouw D, 3590 Diepenbeek, Belgium
[4]Maastricht University, Faculty of Science and Engineering, Department of Advanced Computing Sciences, Paul-Henri Spaaklaan 1, 6229 EN Maastricht, The Netherlands
[*]xenofon.chiotopoulos@maastrichtuniversity.nl

ABSTRACT

The transition to the High-Luminosity Large Hadron Collider (HL-LHC) presents a computational challenge where particle reconstruction complexity may outpace classical computing resources. While quantum computing offers potential speedups, standard algorithms like Harrow-Hassidim-Lloyd (HHL) require prohibitive circuit depths for near-term hardware. Here, we introduce the 1-Bit Quantum Filter, a domain-specific adaptation of HHL that reformulates tracking from matrix inversion to binary ground-state filtering. By replacing high-precision phase estimation with a single-ancilla spectral threshold and exploiting the Hamiltonian's sparsity, we achieve an asymptotic gate complexity of $\mathscr{O}(\sqrt{N}\log N)$, given Hamiltonian dimension N. We validate this approach by simulating LHCb Vertex Locator events with a toy model, and benchmark performance using the noise models of Quantinuum H2 trapped-ion and IBM Heron superconducting processors. This work establishes a resource-efficient track reconstruction method capable of solving realistic event topologies on noise-free simulators and smaller tracking scenarios within the current constraints of the Noisy Intermediate Scale Quantum (NISQ) era.

Introduction

In the High-Luminosity phase of the Large Hadron Collider (HL-LHC) at CERN, high-energy collisions will result in events where thousands of particles are simultaneously produced. These particles traverse multiple sensitive detection layers where they deposit small amounts of energy, generating discrete signals known as detector hits, from which the particle trajectories are reconstructed for physics analysis. The reconstruction process involves a significant combinatorial pattern recognition task, which must be solved in real-time at the high beam crossing rate of 40 MHz and requires expensive computational resources[1].

The transition to the HL-LHC presents a computational bottleneck that challenges the scalability of traditional track reconstruction algorithms. As the number of simultaneous collisions increases by an order of magnitude, the combinatorial complexity of track reconstruction grows factorially[2], potentially outpacing the growth of the classical computing budgets. To address this, the high-energy physics community is increasing its exploration of disruptive computing paradigms, including machine learning and quantum computing. While Graph Neural Networks (GNNs) have shown great promise, they require extensive training and ensuring their inference latency remains within the strict real-time processing constraints of the trigger represents a significant challenge[3,4]. Quantum computers may offer advantages including potential energy efficiency[5,6] and the ability to leverage high-dimensional Hilbert spaces to address combinatorial problems. Various approaches for particle tracking have been explored, ranging from Quantum Annealing[7,8] to quantum GNNs[9]. In previous work[10,11] we explored the Harrow-Hassidim-Lloyd (HHL[12]) algorithm, achieving excellent performance but requiring circuit depths that exceed the coherence times available on current Noisy Intermediate-Scale Quantum (NISQ) devices.

In this paper we present an update to our global pattern recognition algorithm, TrackHHL[11], which introduces a specialized 1-Bit Quantum Filter to overcome these limitations. Inspired by the Denby-Peterson model[13,14], track reconstruction is formulated as the ground-state search of an Ising-like Hamiltonian. Unlike previous approaches using HHL which rely on resource-intensive high-precision Quantum Phase Estimation (QPE), our approach redefines the problem as a spectral filtering task. By restricting the QPE to a single-bit clock register and exploiting phase aliasing through a problem-informed evolution parameter, we deterministically filter out combinatorial noise while constructing a high-fidelity tracking solution. We take advantage of the Hamiltonian sparsity and structure by implementing a custom Hamiltonian evolution via the proposed Direct Structural Synthesis method, which synthesizes the exact unitary operators, reducing the circuit depth compared to standard Trotterization methods.

We validate the 1-Bit Quantum Filter using a custom toy simulation of the LHCb Vertex Locator (VELO). Our results demonstrate successful track reconstruction in both ideal statevector simulations and small-scale quantum hardware emulations. Analytically, we derive a gate complexity of $\mathscr{O}(\sqrt{N}\log N)$, providing an polynomial speedup over classical inversion methods, and sampling complexity of $\mathscr{O}(N\log N)$ for accurate track reconstruction with state tomography. Finally, we benchmark the algorithm on the noise models of IBM's Heron superconducting processors[15] and Quantinuum's H2 trapped-ion system[16]. We demonstrate that for our approach the all-to-all connectivity and lower error rates of the trapped-ion architecture provide improved solution fidelity for small systems with no error correction, while both systems currently suffer from intractable gate error accumulation beyond a problem size of 4 particle tracks and 3 layers. These findings establish a viable pathway to study quantum particle track reconstruction at the scale of future HL-LHC events.

Methods

Hamiltonian Formulation for Trackfinding

The LHCb **Ve**rtex **Lo**cator (VELO)[17] serves as the innermost tracking detector in the spectrometer, positioned 5 mm from the proton-proton interaction point. It comprises 52 silicon pixel detector modules, divided equally between two retractable halves surrounding the beam line. At the High-Luminosity LHC each beam crossing causes ~ 20 simultaneous collisions, which lead to ~ 1000 produced particles along straight line trajectories due to the absence of a magnetic field. When charged particles traverse the detector modules they deposit energy, allowing the VELO to record precise two-dimensional hit coordinates (x, y) and derive a z coordinate from the detector module position. The subsequent reconstruction process aims to group these hits into distinct particle trajectories (tracks). One such simulated beam crossing event in the LHCb VELO is shown in the left panel of Fig. 1. To convert the task of track reconstruction into a format suitable for quantum processing, we formulate the problem within a Hamiltonian framework, adapting the approach initially proposed by Denby and Peterson[13, 14] and subsequently refined for the LHCb VELO geometry[10, 11], briefly summarized below.

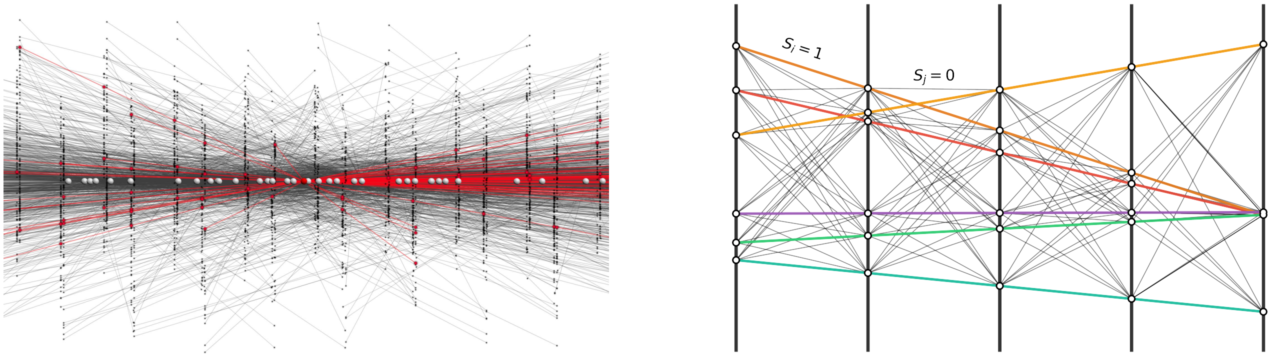

Figure 1. *Left:* A simulated event in the LHCb detector. The larger grey dots represent the primary vertices of multiple simultaneous proton-proton collisions, while the small black dots indicate the detector hits generated by the produced particles. The faint grey lines depict the reconstructed tracks of the produced particles. The red dots and lines indicate the hits and particles produced in a single given collision, represented by the red circle. *Right:* Illustration of the definition of the Hamiltonian graph construction[10]. Colored segments represent active variables $S_i = 1$ forming valid tracks, while grey segments correspond to inactive variables $S_j = 0$.

The system state is parameterized by binary variables $\mathbf{S} = \{S_i, 1 \leq i \leq N, S_i \in \{0, 1\}\}$, representing potential track segments (doublets connecting hits on adjacent detector layers), where $S_i = 1$ indicates an active segment contributing to a track, and $S_i = 0$ indicates inactivity. This graph representation is illustrated in the right panel of Fig. 1. The task of identifying valid tracks is then mapped to finding the ground state of the Hamiltonian $\mathscr{H}(\mathbf{S})$ given by:

$$\mathscr{H}(\mathbf{S}) = \mathscr{H}_{\text{ang}}(\mathbf{S}, \varepsilon) + \alpha \mathscr{H}_{\text{spec}}(\mathbf{S}) + \beta \mathscr{H}_{\text{gap}}(\mathbf{S}) \ . \tag{1}$$

Here, $\mathscr{H}_{\text{ang}}$ rewards straight connections between segments S_i and S_j that satisfy an angular tolerance ε:

$$\mathscr{H}_{\text{ang}}(\mathbf{S},\varepsilon) = -\frac{1}{2}\sum_{i,j} f(\theta_{i,j},\varepsilon)S_iS_j, \quad f(\theta,\varepsilon) = \begin{cases} 1 & \text{if } \cos\theta \geq 1-\varepsilon, \\ 0 & \text{otherwise} . \end{cases} \tag{2}$$

The terms $\mathscr{H}_{\text{spec}}(\mathbf{S}) = \sum_i S_i^2$ and $\mathscr{H}_{\text{gap}}(\mathbf{S}) = \sum_i (1-S_i)^2$ serve as regularization constraints, weighted by parameters $\alpha = 2.0$ and $\beta = 1.0$ following the previous work[10], ensuring the resulting matrix is positive semi-definite. To leverage quantum linear solvers, this discrete optimization problem is transformed into a linear system. By relaxing the discrete segment variables S_i to continuous variables forming a real-valued vector $\mathbf{x} \in \mathbb{R}^N$, the minimum of $\mathscr{H}(\mathbf{x})$ can be found by setting its gradient to zero:

$$\nabla_{\mathbf{x}}\mathscr{H} = \mathbf{A}\mathbf{x} - \mathbf{b} = 0 \quad \Longrightarrow \quad \mathbf{A}\mathbf{x} = \mathbf{b} \quad \Longrightarrow \quad \mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \, . \tag{3}$$

The resulting matrix $\mathbf{A}$ encapsulates the quadratic couplings defined in $\mathscr{H}$. The vector $\mathbf{b}$ originates from the linear terms in $\mathbf{S}$ (arising from $\mathscr{H}_{\text{gap}}$) and takes the form $\mathbf{b} = (1, 1, \ldots, 1)^T$ due to the choice of $\beta = 1$. The solution $\mathbf{x}$ to this linear system represents the relaxed segment amplitudes, which must then be discretized (applying a threshold T via sampling high-amplitude segments) to identify the final set of track segments. This formulation maps the combinatorial track-finding problem onto a linear algebraic structure compatible with algorithms like HHL.

To quantify the computational scale of this approach, we define the system parameters in terms of the detector geometry and event size. The dimension of the linear system, N, corresponds to the total number of candidate segments in the set $\mathbf{S}$, where $N \times N$ defines the size of the $\mathbf{A}$ matrix[10]. For a detector with l layers and m reconstructible particle tracks, the system size scales as $N = m^2(l-1)$. The total number of physical interaction terms, denoted by k, represents the number of angular constraints active in the system, and defines the number of non-zero off-diagonal elements in the Hamiltonian. For full tracks spanning the detector, this scales linearly with the number of particles as $k = m(l-2)$. These parameters will define the gate complexity and sampling cost of our proposed quantum algorithm.

Simulation Tool

To evaluate the performance of the reconstruction algorithm and in particular to explore the spectral and structural properties of the Hamiltonian formulation in Eq. (1), we developed a dedicated simulation tool. This tool captures the essential geometric and detection features of the LHCb VELO, such as multiple scattering, while remaining computationally tractable for hybrid quantum-classical studies.

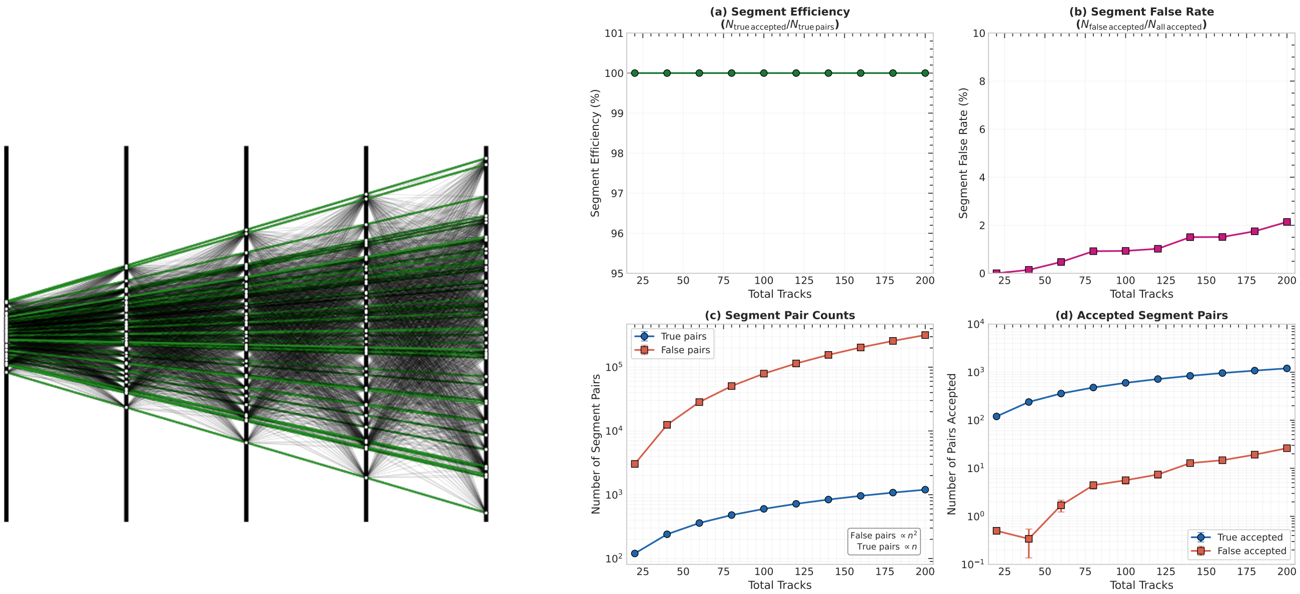

Figure 2. Overview of the simulation tool events and algorithm performance. *Left:* (a) Example of an event with one collision vertex, 5 layers and 64 particles, where green lines indicate correct track segments and black lines indicate false segments. *Right:* (b) The performance of the segment identification algorithm as a function of the number of collision points (with 20 tracks each), showing the correct segment efficiency, fake rate, and the number of segment pairs considered/accepted.

The tool implements a given number of sequential measurement planes in z that register hits of traversing particle tracks produced in events. The events consist of a configurable number of linear particle tracks originating from one or more collision vertices. Given the initial vertex position $(x = 0,\ y = 0,\ z = z_0)$, each track is parametrized by a state vector $(x,\ y,\ t_x,\ t_y,\ p/q)_z$ at given position z, where $t_x = dx/dz$ and $t_y = dy/dz$ denote the slopes of the trajectory in the transverse plane, and p/q is the charge–momentum ratio, included for completeness - in the absence of a magnetic field the particles propagate linearly between the detector layers. At each detector layer $z = z_i$ the track deposits a *hit*, which is registered with an experimental realistic Gaussian measurement resolution. Subsequently, the track undergoes multiple scattering slightly changing its slopes t_x and t_y according to Gaussian multiple scattering σ_{scatt} of the particle in the material. The average scattering angle implemented is given by the material thickness of the detection planes and is inversely proportional to the momentum of the traversing particle, and therefore allows to study the performance as function of particle track momentum. Fig. 2a illustrates an event as generated from a single collision point.

The algorithm performance is evaluated classically as a function of the number of collisions with the track-finding *efficiency*: the fraction of correctly found track segments, and the *fake rate*: the fraction of fake segments in the total sample. Fig. 2b summarizes the performance of the method in terms of segment finding efficiency $\varepsilon_{\text{segment}}$ and fake rate f_{segment} with:

$$\varepsilon_{\text{segment}} = \frac{N_{\text{true accepted}}}{N_{\text{true generated}}} \qquad \text{and} \qquad f_{\text{segment}} = \frac{N_{\text{false accepted}}}{N_{\text{all accepted}}}$$

With the current Hamiltonian formulation, the efficiency remains 100%, independent of problem size, though the fake rate slowly increases with increasing number of tracks, as expected. Future studies with event sizes similar to real LHCb VELO events (~ 1000 tracks) will demonstrate the need for a trade-off between fake rate and efficiency by tuning the acceptance parameter in the Hamiltonian, or adding additional terms from the Denby-Peterson model to suppress the fake rate, at the cost of increased circuit depth in the Hamiltonian simulation.

Quantum Algorithm

As described in the Methods section, this work builds upon the quantum tracking framework established in Ref.[10,11], which uses the HHL[12] algorithm to solve linear systems of equations described by the Hamiltonian track finding formulation in Eq. (1). HHL's core advantage comes from the Quantum Phase Estimation (QPE[18,19]) subroutine to resolve these eigenvalues into a multi-qubit clock register. QPE consists of a Hamiltonian evolution followed by an inverse Quantum Fourier Transform (QFT[20]). The QPE's required precision determines the number of applications of the controlled unitary that implements the time evolution. A controlled rotation is then applied to an ancilla qubit to encode the inverse eigenvalues $(1/\lambda)$ onto the state amplitudes, followed by the uncomputation of the clock register. HHL is theoretically powerful, promising a logarithmic speedup, however, this reliance on high-precision estimation of the spectrum coupled with an expensive tomography, creates significant bottlenecks[21,22]. The algorithm requires prohibitive circuit depths due to iterative application of the Hamiltonian evolution e^{-iAt} and QPE requires many clock register qubits, rendering it unfeasible for particle tracking on near-term hardware.

To address these limitations, we present the 1-Bit Quantum Filter, a formalization and extension of our previous work[11]. We observe that for track reconstruction, exact matrix inversion is not necessary. The tracking problem is inherently binary, the task is to distinguish valid 'signal' segments from 'noise' segments. Consequently, we replace the resource-intensive high-precision QPE and controlled rotation using a 1-bit QPE with a deterministic eigenvalue filter. This reformulates the task from exact inversion to ground-state spectral filtering, drastically reducing the gate complexity and sampling requirements[11]. Additionally, we replace the general purpose approximate Suzuki-Trotter evolution[23] with a proposed Direct Structural Synthesis method, which synthesizes controlled unitary operators without the need for an expensive classical decomposition.

The 1-Bit Quantum Filter

This approach replaces resource-intensive eigenvalue inversion with a binary filtration mechanism that takes advantage of the specific spectral properties of the Hamiltonian track-finding. The algorithm begins with determining the evolution time t, which is used in our 1-bit phase estimation subroutine. In general, the spectrum bounds $(\lambda_{\min}, \lambda_{\max})$ can be estimated classically via the Gershgorin Circle Theorem[24]; however, for the particle tracking case the bounds are known as they converge to $(\lambda_{\min} \approx 1, \lambda_{\max} \approx 5)$[10]. Since the physical length of a track is bounded by the number of detector layers (l), the eigenvalues of the track sub-matrices are bounded by the track length rather than the total number of hits N. Due to this convergence, we can target the central eigenvalue λ_c. The central value associated with combinatorial noise is determined exactly by diagonal coefficients $\alpha + \beta$ from Eq. (1):

$$\lambda_c = \alpha + \beta \quad \text{and} \quad t = \frac{\pi}{\lambda_c}\,. \tag{4}$$

This choice of t is a deliberate deviation from standard QPE, where the evolution time is typically scaled as a function of λ_{max}, to prevent phase aliasing[12,25]. We instead reduce the precision of the QPE to a single time register qubit and take advantage

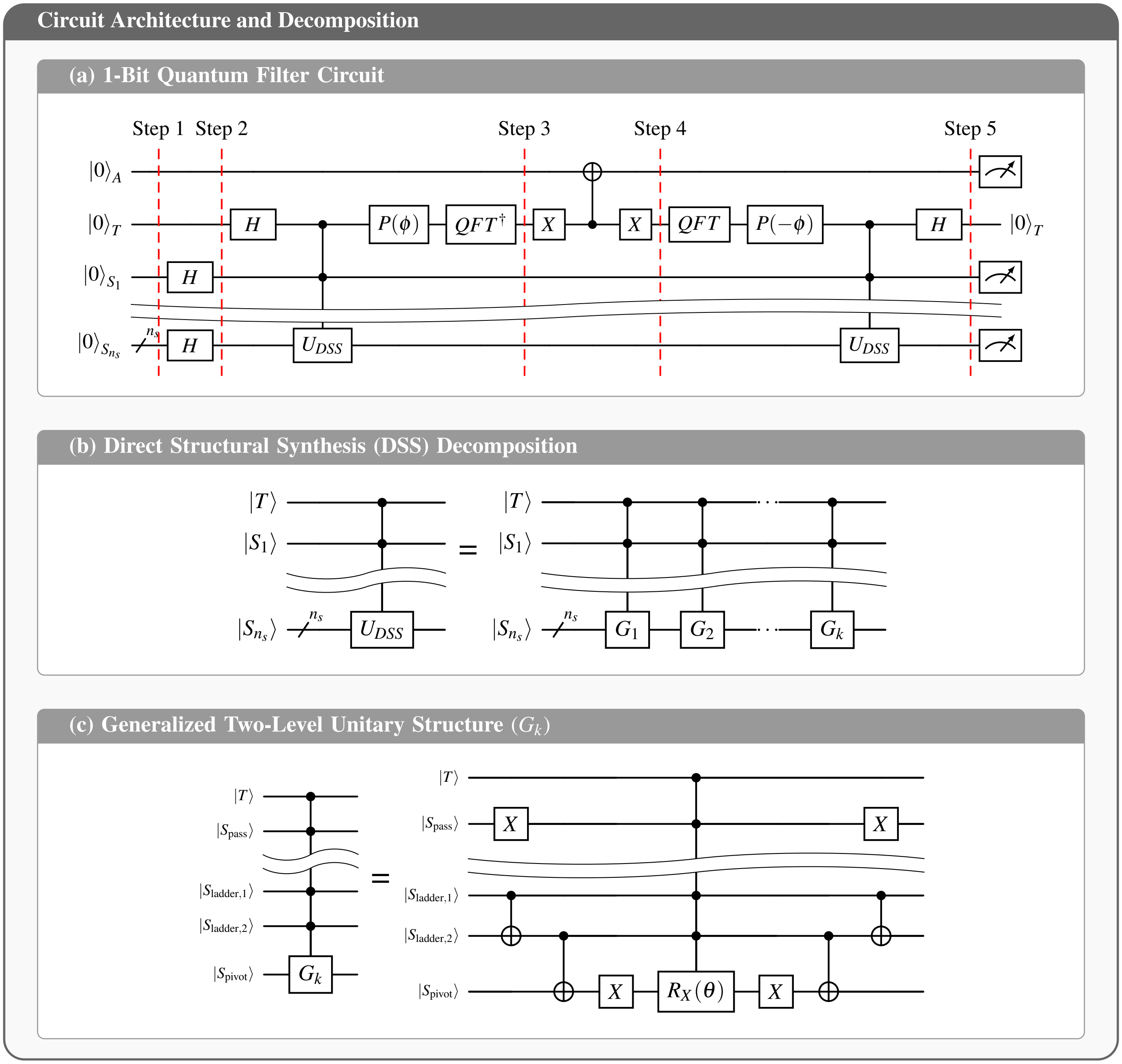

Figure 3. Circuit Architecture and Decomposition: (a) The complete 1-Bit Quantum Filter circuit, featuring state preparation, 1-bit phase estimation with flagging logic, and uncomputation. The additional phase gate $P(\phi)$ on the time register implements the diagonal term of the Hamiltonian. (b) The Direct Structural Synthesis (DSS) decomposition, explicitly showing that the controlled time evolution U_{DSS} is implemented as a sequential application of controlled interaction gates $\prod_k G_k = \prod_k e^{iB_k t}$ corresponding to the non-zero off-diagonal terms of the Hamiltonian matrix. (c) The generalized structure of a single interaction gate $G_k = e^{iB_k t}$. For states with Hamming distance $d_H > 1$, a CNOT ladder (S_{lad}) disentangles the states onto a pivot (S_{piv}), while passive qubits (S_{pass}) act as controls to strictly enforce subspace selection.

of the aliasing due to our choice of t in Eq. (4), which allows us to project the noise to a specific phase. We prepare three registers: the system register $|0\rangle_S$ for encoding the $\mathbf{b}$ vector, the time register used for the QPE $|0\rangle_T$ and the ancilla register used to determine if the algorithm has run successfully $|0\rangle_{\mathbf{A}}$. The system register is initialized in the uniform superposition state $|b\rangle = H^{\otimes n}|0\rangle$, as $\mathbf{b}$ is defined as a uniform vector. To analyze the spectral filtering mechanism, we rewrite the state in the eigenbasis of the Hamiltonian matrix $\mathbf{A}$:

$$|b\rangle = \sum_j \beta_j |u_j\rangle \ , \tag{5}$$

where $|u_j\rangle$ are the orthonormal eigenvectors with eigenvalues λ_j, and $\beta_j = \langle u_j|b\rangle$, represents the overlap amplitude. The 1-Bit Quantum Filter proceeds in five steps, tracking the evolution of the state $|\Psi\rangle$ across the ancilla $|A\rangle$, time $|T\rangle$, and system $|S\rangle$ registers. The following steps match exactly the circuit shown in Fig. 3a.

Step 1: State Preparation We initialize the system register $|0\rangle_S$ into a uniform superposition $|b\rangle = H^{\otimes n}|0\rangle_S$, representing the vector $\mathbf{b} = (1, 1, \ldots, 1)^T$. The initial state is:

$$|\Psi_1\rangle = |0\rangle_A \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)_T \otimes \sum_j \beta_j |u_j\rangle_S \ . \tag{6}$$

The time register is also prepared in superposition via a Hadamard gate.

Step 2: 1-Bit Phase Estimation We apply the controlled time evolution $U = e^{iAt}$ conditioned on the $|1\rangle_T$ state. This unitary is implemented by combining the Direct Structural Synthesis (DSS) method applied to non-diagonal terms, with a Phase Gate $P(\phi)$ (where $\phi = -(\alpha+\beta)t$) that accounts for the diagonal contributions of the Hamiltonian. Together, these components realize the time evolution of A, resulting in the phase $e^{i\lambda_j t}$ being kicked back onto the $|1\rangle_T$ state for each eigen-component:

$$|\Psi_{2_a}\rangle_j = |0\rangle_A \otimes \frac{1}{\sqrt{2}} \sum_j \beta_j \left(|0\rangle_T + e^{i\lambda_j t}|1\rangle_T\right) \otimes |u_j\rangle_S \ . \tag{7}$$

This creates a relative phase between the $|0\rangle_T$ and $|1\rangle_T$ states. Next, the inverse QFT is applied to the time register $|0\rangle_T$. In the 1-Bit case the inverse QFT is implemented using a single Hadamard on the time register to interfere the phases. Using the identity $H(|0\rangle + e^{i\phi}|1\rangle) \propto \cos(\phi/2)|0\rangle - i\sin(\phi/2)|1\rangle$[25], it follows that:

$$|\Psi_{2_b}\rangle_j = |0\rangle_A \otimes \sum_j \beta_j \left[\cos\left(\frac{\lambda_j t}{2}\right)|0\rangle_T - i\sin\left(\frac{\lambda_j t}{2}\right)|1\rangle_T\right] \otimes |u_j\rangle_S \ . \tag{8}$$

The phase information $\lambda_j t$ has been converted into amplitude information in the $|0\rangle_T$ and $|1\rangle_T$ basis states.

Step 3: Conditional Logic We apply a Zero-Controlled NOT gate on the ancilla, flipping $|0\rangle_A \to |1\rangle_A$ only when the time register is $|0\rangle_T$. The relative phase relationship is preserved, but now entangled with the ancilla register:

$$|\Psi_3\rangle = \sum_j \beta_j \left[\underbrace{\cos\left(\frac{\lambda_j t}{2}\right)|1\rangle_A|0\rangle_T}_{\text{Signal}} - \underbrace{i\sin\left(\frac{\lambda_j t}{2}\right)|0\rangle_A|1\rangle_T}_{\text{Noise}}\right] \otimes |u_j\rangle_S \ . \tag{9}$$

Step 4: Uncomputation To disentangle the time register, we apply the inverse of the phase estimation circuit (Inverse QPE). This restores the time register to $|0\rangle_T$, leaving the system state entangled only with the ancilla. We can now group the terms based on the state of the ancilla register:

$$|\Psi_4\rangle = \left(|1\rangle_A \otimes \sum_{j\in\text{Signal}} \beta_j |u_j\rangle_S\right) + \left(|0\rangle_A \otimes \sum_{j\in\text{Noise}} \beta_j |u_j\rangle_S\right) \otimes |0\rangle_T \ . \tag{10}$$

Step 5: Measurement & Filtration Finally, we measure the ancilla. Our specific choice of $t = \pi/\lambda_c$ ensures that for noise eigenvalues ($\lambda = \lambda_c$), the probability of triggering the signal flag vanishes:

$$P(1|\lambda_c) = \cos^2\left(\frac{\lambda_c t}{2}\right) = \cos^2\left(\frac{\pi}{2}\right) = 0 \ . \tag{11}$$

As a result, post-selecting on the measurement projecting onto the signal subspace $|1\rangle_A$ deterministically filters out the combinatorial noise. The collapsed state must then be renormalized by the probability of success P_{succ}, yielding the solution state:

$$|\Psi_5\rangle = \frac{1}{\sqrt{P_{\text{succ}}}} \sum_{j\in\text{Signal}} \beta_j |u_j\rangle_S \ . \tag{12}$$

This spectral separation is guaranteed by the Hamiltonian structure defined in Eq. (1). Since noise segments correspond to isolated nodes in the graph, their eigenvalues are fixed exactly at λ_c, while valid track segments are spectrally shifted away. To exploit this, we replace the standard HHL rotation with a Zero-Controlled NOT gate (Step 3, Eq. 9). This gate targets the ancilla conditioned on the $|0\rangle_T$ state, flipping it to $|1\rangle_A$ only for signal components. Consequently, noise components (which map to $|1\rangle_T$) fail to trigger the gate and remain in $|0\rangle_A$. Following the uncomputation step (Step 4, Eq. 10), we measure the ancilla and post-select on the $|1\rangle_A$ outcome. This deterministically filters the noise contribution ($P(1|\lambda_c) = 0$), leaving a final state composed strictly of valid track segments.

Time Evolution through Direct Structural Synthesis (DSS)

A central component of our 1-bit Quantum Filter is the implementation of the controlled time evolution $C(U) = C(e^{-iAt})$. A general-purpose Suzuki-Trotter time evolution of A is resource intensive[11], as it requires the A matrix to be represented as a sum of Pauli strings. An alternative approach is the Direct Structural Synthesis (DSS) method that exploits the specific structure of our Hamiltonian, placing it within the class of exact gate synthesis algorithms[26]. The DSS method is based on the fact that we can express $A = cI - B$. Since the diagonal matrix term cI commutes with the off-diagonal interaction matrix B, the unitary evolution operator $U(t) = e^{-iAt}$ allows for an exact factorization:

$$U(t) = e^{-i(cI-B)t} = e^{-icIt} \cdot e^{iBt} \ . \tag{13}$$

Implementing the controlled version of this evolution, $C(U)$, separates into a controlled global phase rotation ($C(e^{-icIt})$) and the controlled interaction terms ($C(e^{iBt})$).

The first term $C(e^{-icIt})$ is a controlled global phase rotation, which can be efficiently implemented with a single $P(\phi)$ gate on the control qubit, with $\phi = -ct$. The phase rotation can be implemented with a single gate as cI is a uniform identity matrix, its phase is kicked back deterministically from the target qubit to the control[25]. The matrix $B = \sum_k B_k$ is a sum of its individual interaction components, where the index k represents a unique interaction term $\{i, j\}$, and the total number of interaction terms that pass the $\mathscr{H}_{\text{ang}}(\mathbf{S}, \varepsilon)$ angular function, and can be written as $k = m(l-2)$ for m tracks. Each B_k represents a σ_x-like interaction between two basis states. The algorithm implements $C(e^{iBt})$ by iterating through k components and applying the gate for each one sequentially:

$$G_{\text{DSS}} = \prod_k G_k \quad \text{where} \quad G_k = C(e^{iB_k t}) \ . \tag{14}$$

A circuit based illustration of Eq. (14) is visualized in Fig. 3b. To strictly enforce the graph topology of B, each gate G_k is implemented as an exact Two-Level Unitary, often referred to as a Givens rotation[25]. For an interaction between basis states $|i\rangle$ and $|j\rangle$, the operator G_k performs a rotation strictly within the subspace spanned by $\{|i\rangle, |j\rangle\}$ and acts as the identity on all other states. This is synthesized using a ladder of CNOT gates to disentangle the basis states[27], followed by a multi-controlled rotation $C^{n-1}(R_X(\theta))$, and a subsequent uncomputation step.

The exactness of DSS time evolution hinges on whether the individual interaction terms B_k commute. While this holds for terms acting on disjoint subspaces, the general case for a d-sparse Hamiltonian involves interaction terms sharing basis states. For these non-commuting terms, the factorization $e^{iBt} \neq \prod_k e^{iB_k t}$ is not an equality, but rather a first-order Suzuki-Trotter product formula[23]:

$$e^{iBt} = \prod_k e^{iB_k t} + \mathscr{O}(t^2) \ . \tag{15}$$

While this approximation introduces an error $\mathscr{O}(t^2)$ scaling quadratically with time, it is crucial to note that this error is subspace dependent. Specifically, the primary function of our filter remains robust because the noise eigenvectors $|\psi_c\rangle$ lie in the null space of the interaction matrix B. Since noise segments correspond to isolated vertices in the graph, they have no off-diagonal couplings, meaning $B|\psi_c\rangle = 0$ exactly. Consequently, the time evolution is exact for the noise subspace, ensuring that combinatorial background suppression is unaffected by time-evolution errors. The Trotter error $\mathscr{O}(t^2)$ acts non-trivially only on the signal eigenspace, introducing small phase perturbations that may slightly affect signal acceptance but not noise rejection.

While the noise subspace is immune to this error, the Trotter error acts non-trivially on the solution eigenspace (λ_j). The first-order simulation $U_{\text{sim}}(t)$ returns a perturbed phase $\phi_{\text{sim}} = \phi_{\text{exact}} + \delta\phi$, where $\phi_{\text{exact}} = \lambda_j t$ and $\delta\phi$ represents the $\mathscr{O}(t^2)$ phase error. Consequently, the measured probability for the solution state is shifted:

$$P(0|\lambda_j) = \cos^2\left(\frac{(\lambda_j t + \delta\phi)}{2}\right) \quad \text{and} \quad P(1|\lambda_j) = \sin^2\left(\frac{(\lambda_j t + \delta\phi)}{2}\right) . \tag{16}$$

The phase shift $\delta\phi$ perturbs the interference condition. While $\delta\phi$ does not cause spectral leakage for the critical λ_c state, a large enough $\delta\phi$ can blur the signal phases λ_j. This can cause a portion of the solution states to be projected onto the reject bin ($|1\rangle_T$) by mistake, or conversely, shift the acceptance probability $P(0)$ closer to 1 or 0 depending on the sign of the error term. For the precision required in this application, the first-order approximation is sufficient; however, if higher precision QPE is critical, a higher-order product formula (e.g., 2nd-order Suzuki-Trotter with $\mathscr{O}(t^3)$ scaling) could be employed to suppress the error scaling.

Complexity Analysis

To quantify the impact of the changes to HHL we study the computational complexity of the 1-Bit Quantum Filter and derive the asymptotic complexity for both two-qubit gate count and sampling cost. We demonstrate that the total gate and sampling complexity scales as $\mathscr{O}(\sqrt{N}\log N)$ and $\mathscr{O}(N\log N)$, respectively.

Gate Complexity

We derive the asymptotic gate complexity for a single execution of the circuit. Based on the system parameters defined in Hamiltonian construction, the number of interaction terms scales as $k \approx \sqrt{N}$. Consequently, the total complexity scales as $\mathscr{O}(\sqrt{N}\log N)$, dominated by the sequential implementation of these off-diagonal terms in the time evolution. The total gate complexity C_{gate} is the sum of the complexity in steps 1-4 in Fig. 3a:

1. **Step 1, State Preparation:** Since the $|\mathbf{b}\rangle$ vector is a perfect superposition, we apply a single Hadamard gate to all n_s system qubits. While this can be executed in $\mathscr{O}(1)$ depth, the total gate count is $\mathscr{O}(n_s) = \mathscr{O}(\log N)$.
2. **Step 2, 1-Bit Quantum Phase Estimation:** The QPE is implemented with the DSS method and a single phase rotation gate. The DSS method iterates k times. Each iteration synthesizes an exact two-level unitary G_k using three sub-routines:
    (a) **Basis Transformation:** A ladder of `CNOT` gates is applied to map the interacting states $|i\rangle$ and $|j\rangle$ such that they differ at only a single qubit index (the pivot), seen in Fig 3c. The number of two-qubit gates scales with the Hamming distance d_H as $\mathscr{O}(d_H)$. Since $d_H \leq n_s$, this is upper-bounded by $\mathscr{O}(n_s)$.

    (b) **Control Logic:** To trigger the multi-controlled rotation, X gates are applied to any $n_s - 1$ non-pivot qubits that are currently in the $|0\rangle$ state. This ensures that the multi-controlled rotation will target the subspace defined by $\{|i\rangle, |j\rangle\}$ correctly. This step contributes $\mathscr{O}(n_s)$ single-qubit gates.

    (c) **Rotation:** A multi-controlled $R_X(\theta)$ rotation is applied to the pivot qubit, controlled by the $n_s - 1$ other qubits. Using the decomposition for multi-controlled rotations without ancilla[28], this contributes $\mathscr{O}(n_s)$ gates.

    (d) **Uncomputation:** The X gates and `CNOT` ladder are applied in reverse.

    Summing these components, a single DSS iteration scales linearly with the register size $\mathscr{O}(n_s)$. For k iterations, the total complexity for Step 2 is $\mathscr{O}(k \cdot n_s)$.
3. **Step 3, Controlled Inversion:** The controlled inversion consists of two X gates and a single `CNOT` gate. This is constant $\mathscr{O}(1)$ complexity.
4. **Step 4, Uncomputation:** This is identical to Step 2, thus doubling the complexity contribution of the phase estimation.

The complexity is primarily determined by the k repetitions required to encode every valid interaction term. Substituting $n_s = \log N$ and the sparsity factor $k \approx \sqrt{N}$, the total gate complexity is:

$$C_{total} \approx k \cdot n_s \approx \mathscr{O}(\sqrt{N}\log N) . \tag{17}$$

This scaling represents a polynomial speedup in N compared to classical exhaustive search methods, providing a computational advantage analogous to Grover's algorithm[29].

Sampling Complexity

Unlike the standard HHL algorithm, where the probability of measuring the solution state is suppressed by the condition number κ as $\mathscr{O}(1/\kappa^2)$[12], the performance of the 1-Bit Quantum Filter is governed by the geometric overlap of the initial state with the solution subspace. The algorithm succeeds when the ancilla is measured in the $|1\rangle_A$ state, which requires the time register to collapse to $|0\rangle_T$. The total success probability is the sum of the conditional probabilities for each eigenvector $|u_j\rangle$ weighted by its overlap with the input vector $|\mathbf{b}\rangle$:

$$P_{\text{succ}} = P(T=0) = \sum_j P(0|\lambda_j) \cdot |\beta_j|^2 \,. \tag{18}$$

The Hamiltonian spectrum is partitioned into two orthogonal subspaces: the Signal subspace (valid tracks, $\lambda \neq \lambda_c$) and the Noise subspace (inactive segments, $\lambda = \lambda_c$). Substituting the filter response from Eq. (11), the success probability becomes:

$$P_{\text{succ}} \approx \underbrace{\frac{1}{2}\cos^2\left(\frac{\lambda_j \pi}{2\lambda_c}\right)}_{\text{Signal Pass Rate}} \sum_{j\in\text{Signal}} |\beta_j|^2 + \underbrace{\frac{1}{2}\sin^2\left(\frac{\lambda_c \pi}{2\lambda_c}\right)}_{\text{Noise Nulling}} \sum_{c\in\text{Noise}} |\beta_c|^2 \,. \tag{19}$$

The filter deterministically eliminates the noise subspace (as the second term in Eq. (19) vanishes), while passing the signal component with an effective probability $c_{filter} = \cos^2\left(\frac{\lambda_j \pi}{2\lambda_c}\right)$. The summation term $\sum_{j\in\text{Signal}} |\beta_j|^2$ represents the overlap of the uniform superposition of $|b\rangle$ with the valid track configurations. For a system with N total segments and k valid interaction terms, this overlap is the ratio of the subspace dimensions. Thus, the single-shot success probability scales linearly with the density of valid tracks, given $M = m(l-1)$ valid track segments:

$$P_{succ} \approx c_{filter} \cdot \sum_{j\in\text{Signal}} |\beta_j|^2 \approx \frac{M \cdot c_{filter}}{N} \,. \tag{20}$$

To derive the full sampling complexity we account for the reconstruction of the entire event. Since the output is probabilistic, this process can be modeled by the Coupon Collector's Problem[30], where we sample with replacement from M distinct outcomes. To observe all distinct segments with high probability, the required number of successful measurements scales as $S_{\text{dist}} \approx M \log M$[30]. The total computational cost, defined as the total number of circuit executions S_{total}, is the required number of successes normalized by the single-shot probability P_{succ}:

$$S_{\text{total}} = \frac{S_{\text{dist}}}{P_{\text{succ}}} \approx \frac{M \log M}{c_{\text{filter}}(M/N)} \approx N \log M \approx \mathscr{O}(N \log \sqrt{N}) \approx \mathscr{O}(N \log N) \,. \tag{21}$$

By expressing the track density in terms of the total system size N, we observe that the complexity scales as $\mathscr{O}(N \log N)$. This represents an advantage over HHL, as we eliminate the polynomial dependence on the condition number κ[12] and the precision overhead of the QPE. However, this does not overcome the fundamental challenge of quantum state tomography. To address this we would need to project onto some information dense physics observable or employ some form of amplitude amplification on the $|1\rangle_A$ subspace.

Results

Performance Evaluation

To strictly evaluate the efficacy of the proposed 1-Bit Quantum Filter, we applied the algorithm to simulated event data generated with the LHCb VELO simulation tool. This study centers on two critical performance metrics: (1) the ability to find the correct segments under realistic quantum noise conditions, with low fake rates, and (2) the empirical validation of the theoretically derived sampling and two-qubit gate complexities.

For these benchmarks, we employed a controlled version of the simulation tool. To study the scaling of the quantum algorithm we restricted the toy model to a single primary vertex with no detector inefficiencies or multiple scattering. We then varied the detector geometry between 3 and 5 layers to study the behavior and scaling of the c_{filter} coefficient.

To study the 1-Bit Quantum Filter we utilized Quantinuum's TKET compilation stack[31] and the Qnexus platform[32] to perform both noiseless and noisy simulations targeting the H2 Quantum Processing Unit (QPU), which utilizes high-connectivity trapped-ion hardware. These results are compared against IBM's Qiskit framework[33] and associated noise models for the IBM Torino, IBM Fez, and IBM Pittsburgh devices. These systems represent the Heron r1, r2, and r3 superconducting architectures[15], respectively, allowing for a comparative analysis of algorithmic performance across evolving hardware generations.

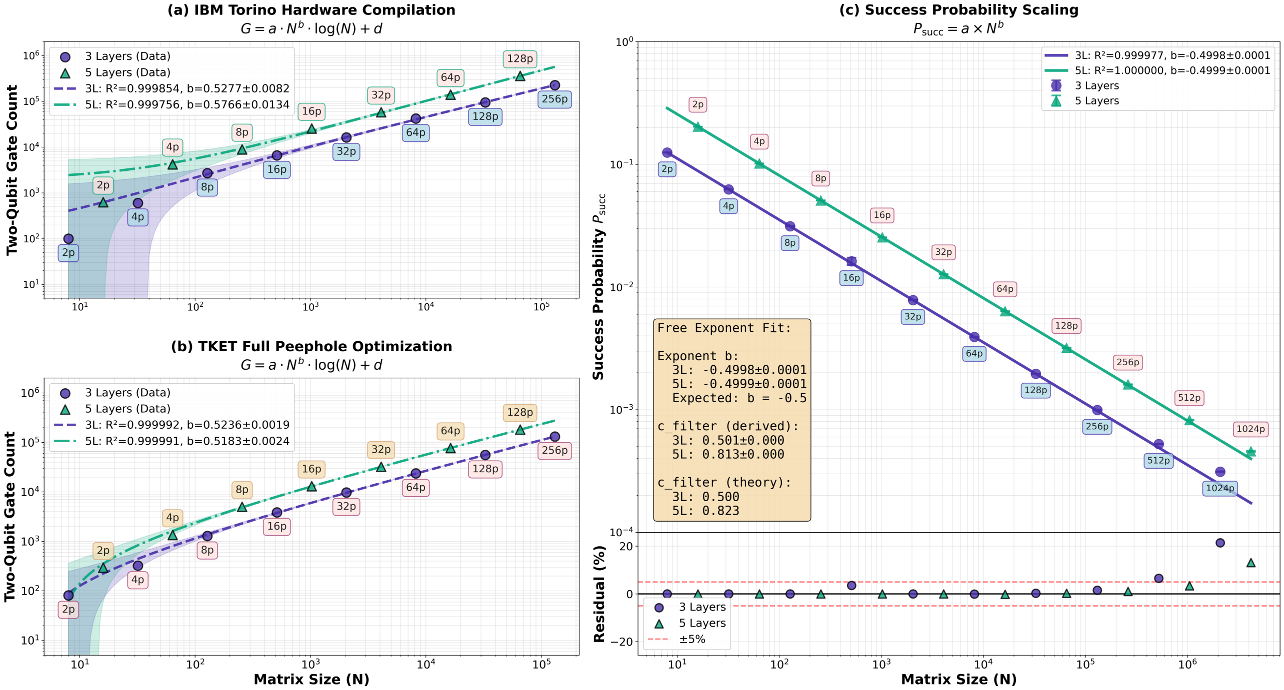

Figure 4. *Left:* Gate count complexity scaling for 1-Bit quantum filter implementations under (a) IBM Torino hardware compilation and (b) TKET optimization. Power-law fits with constrained logarithmic scaling ($c = 1.0$) yield $b \approx 0.53$ (Qiskit) and $b \approx 0.52$ (TKET), consistent with $O(\sqrt{N}\log N)$ theoretical complexity ($R^2 > 0.999$). *Right (c):* Success probability P_{succ} versus matrix size N for 3-layer and 5-layer tracking scenarios. Fitted exponents exactly match the theoretical $N^{-1/2}$ scaling up to $N = 10^6$ where we see an upward deviation in the residuals.

Validation of Derived Complexities

To verify the theoretical two-qubit gate complexity of $\mathcal{O}(\sqrt{N}\log N)$ (Eq. 17), we analyzed the post-compilation two-qubit gate counts for circuits targeting both IBM (superconducting) and Quantinuum (trapped-ion) architectures. We generated Qiskit circuits using noiseless statevector simulations using Qiskit's Aer simulator[33], accelerated via an Nvidia L40S GPU and proceeded with two distinct compilation pipelines: (1) Transpilation to the IBM Torino backend using Qiskit's optimization level 3, and (2) conversion to the TKET framework followed by 'full peephole' optimization[31] targeting Quantinuum H2 hardware. The gate count analysis is restricted to the range $N \in [2^3, 2^{17}]$, as the classical optimization of circuits exceeding 10^6 two-qubit gates becomes computationally intractable beyond this point.

The compiled gate counts are fitted to the theoretical ansatz $C_{total} = aN^b \log(N) + c$. The logarithmic term is fixed to reflect the dependence on register size ($n_s = \log N$), allowing us to isolate the polynomial scaling exponent b. This constraint minimizes the degrees of freedom relative to the dataset size preventing overfitting, in the asymptotic regime.

Figures 4a and 4b display the results with 1σ confidence bands. We observe excellent agreement between the fitted coefficient b and the theoretical prediction of 0.5 in the asymptotic regime ($N > 2^{10}$), where the confidence bands narrow significantly. The divergence observed at smaller system sizes is attributed to the constant-depth overhead not captured by the asymptotic model.

Comparing the two compilation pipelines reveals a distinct scaling behavior driven by hardware topology. The TKET results (Fig. 4b), which target the all-to-all connectivity of trapped-ion systems, align closely with the theoretical lower bound, as the hardware supports direct interaction between arbitrary qubit pairs. In contrast, Qiskit results (Fig. 4a) exhibit larger overhead and higher variance. This discrepancy arises from the limited connectivity of the IBM superconducting architecture, which requires the insertion of SWAP chains during the routing process to implement non-local Hamiltonian terms, causing an inflation in the pre-factor a.

Having established the theoretical total sampling complexity in Eq. (20), we see that the validation of our model is entirely dependent on the behavior of the success probability P_{succ} with system size N, as S_{dist} follows the standard coupon collector scaling. Rewriting Eq. (19) in terms of the system size N, the theoretical success probability is expected to scale as $P_{succ} \approx (c_{filter}\sqrt{l-1})/\sqrt{N}$. As such, we fit the simulation data to a power-law model of the form $P_{succ} = a \cdot N^b$. Validation of the theoretical derivation requires that the fitted exponent b converges to -0.5 and the coefficient a aligns with the derived factor $c_{filter}\sqrt{l-1}$.

The dataset for this scaling analysis was generated using statevector simulations. Since determining P_{succ} does not require the computationally expensive hardware specific compilation step used in the gate analysis, we extended the range from the minimal configuration of 3 layers and 2 tracks ($N = 2^3$) up to a denser environment of 5 layers and 1024 tracks ($N = 2^{22}$). Experimentally, P_{succ} is quantified as the probability of the ancilla flag qubit projecting onto the $|1\rangle$ subspace. Figure 4c presents the power-law fits for 3-layer and 5-layer detector configurations. The fitted coefficients a and exponents b exhibit excellent agreement with theoretical predictions for both configurations ($R^2 > 0.999$), confirming the validity of the derived scaling laws.

A deviation is observed for large system sizes ($N > 2^{18}$), where the measured P_{succ} exceeds the theoretical prediction. This enhancement is not a computational artifact but a consequence of the combinatorial density. In regions of high track occupancy, the density of hits increases the probability of fake segments satisfying the angular constraints of the Hamiltonian Eq. (2). Because the current implementation relies on a simple binary threshold defined in Eq. (2), random geometric alignments can satisfy the angular constraints. These false positives satisfy the Hamiltonian constraints and contribute to the final solution state, increasing the P_{succ} even though they do not correspond to any real tracks.

Testing Solution Fidelity

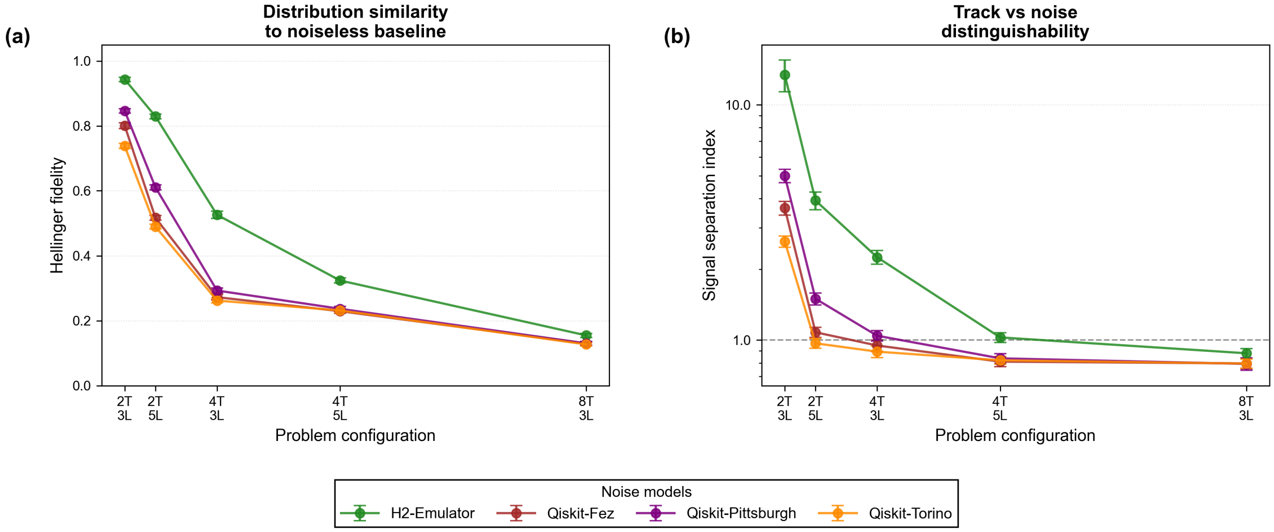

Figure 5. Noise resilience benchmarks. (a) Hellinger fidelity (F_H) relative to the noiseless baseline across five problem configurations, indicating the preservation of the probability distribution structure. **(b)** Signal Separation Index (SSI) quantifying the contrast between valid tracks and the dominant noise floor. The dashed line at SSI $= 1$ represents the distinguishability limit where signal amplitudes drop below the background noise. Error bars denote 1σ confidence intervals derived via Monte Carlo resampling of the measurement counts.

The 1-Bit Quantum Filter resilience to noise is assessed by utilizing the noise models of the IBM Torino, IBM Fez, and IBM Pittsburgh, as well as the Quantinuum H2 emulator. These models are derived from regular device calibration data, accurately reflecting the coherence times, gate fidelities, and readout error rates of the physical hardware. We employ the same compilation pipeline established for the validation of our theoretical complexities, but the circuits are executed with device specific noise models under realistic conditions. For the IBM devices we modify the pipeline compilation step to target specific IBM hardware backends. For Quantinuum we took the optimized circuit and used the Qnexus platform for compilation and execution of the circuit. The analysis is limited to $N \in [2^3, 2^7]$ to ensure the accumulated gate errors remain within a regime where valid tracks can be resolved from the background.

To quantify the impact of noise on track reconstruction performance we construct the following two metrics:

1. **Hellinger Fidelity (F_H)**: Measures the geometric overlap between the noisy experimental distribution P_{exp} and the ideal noiseless baseline P_{ideal}:

$$F_H(P_{exp}, P_{ideal}) = \left(\sum_i \sqrt{P_{exp}(x_i) \cdot P_{ideal}(x_i)} \right)^2 \tag{22}$$

2. **Signal Separation Index (SSI)**: This metric quantifies the distinguishability of valid tracks from the background. We define it as the ratio of the total probability mass of the M valid track segments (S) to that of the M highest-amplitude noise segments (N^*):

$$\mathrm{SSI} = \frac{\sum_{i \in S} P_i}{\sum_{j \in N^*} P_j} \tag{23}$$

The Hellinger Fidelity (F_H) results, presented in Fig. 5a, demonstrate a distinct performance hierarchy among the evaluated backends. The Quantinuum H2 Emulator consistently achieves the highest fidelity across all problem configurations, followed by the noise models for IBM Pittsburgh, IBM Fez, and IBM Torino, respectively. This divergence is primarily driven by the topological differences between the architectures: the all-to-all connectivity of the trapped-ion H2 device allows the compiler (TKET) to synthesize the unitary with approximately half the circuit depth required for the fixed-topology superconducting chips. Since error accumulation is exponential with depth, this topological advantage and higher qubit fidelity[16] significantly enhances the solution quality. Within the superconducting results, we observe a clear progression in performance corresponding to the maturity of the IBM Heron processor generations ($r3 > r2 > r1$).

The Signal Separation Index (SSI), shown in Fig. 5b, corroborates these findings by mapping the distinguishability of valid tracks from fake segment combinations. The dashed horizontal line at $\mathrm{SSI} = 1$ defines the critical threshold below which the quantum filter fails to separate signal from noise. Consistent with the fidelity data, the H2 platform maintains a resolvable signal ($\mathrm{SSI} > 1$) up to the denser 4-track, 5-layer configuration. In contrast, the superconducting baselines approach the noise floor more rapidly, exhibiting the same $r3 > r2 > r1$ ordering. These results indicate that for current NISQ hardware, the effective connectivity and its direct reduction of compilation overhead is a dominant factor in extending the reach of quantum track reconstruction to larger detector geometries. However, we are limited to problem sizes of 5 layers and 4 tracks. For larger N, the increased circuit depth leads to decoherence, causing $\mathrm{SSI} < 1$, rendering the solution indistinguishable from noise.

Discussion

In this paper, we presented the 1-Bit Quantum Filter, a domain-specific adaptation of the HHL algorithm tailored for the combinatorial challenge of particle track reconstruction. Our results demonstrate that by replacing standard quantum linear algebra routines in favour of problem-informed filtering, we can achieve a polynomial gate complexity $\mathcal{O}(\sqrt{N}\log N)$.

A key finding of this study is the computational advantage gained by relaxing the requirements of the HHL algorithm, originally designed to perform precise matrix inversion $A^{-1}b$. This precision preserves the relative amplitudes of the solution vector for any well conditioned matrix, however, for the particle tracking task this is excessive. The physics problem is inherently binary: we aim to distinguish the "signal" subspace (valid track segments) from the "noise" subspace (combinatorial background segments), rather than computing a continuous amplitude distribution. The 1-Bit Quantum Filter exploits this relaxation by replacing the resource intensive multi-qubit QPE with a single-ancilla projection. The modification effectively replaces the expensive arithmetic inversion of the eigenvalues ($1/\lambda$) with a simpler spectral step function, significantly reducing the circuit depth and the number of required ancilla qubits. The shift from calculating the exact amplitudes of the solution spectrum to filtering only track segments is what renders the algorithm feasible for the problem sizes required by the HL-LHC, where the overhead of full QPE would be prohibitive.

While the gate complexity of our approach scales favorably, the total runtime remains constrained by the sampling complexity, which scales linearly with the system size $\mathcal{O}(N)$. To reconstruct the full event, the quantum state must be repeatedly prepared and measured until all M valid track segments are found. For large events on the order of the HL-LHC the linear readout cost is expensive. Future research must therefore pivot from full state tomography to the direct estimation of global properties. Instead of reading out the full distribution, one could construct quantum observables that can extract higher level physics features such as the z-coordinate location of the Primary Vertices, or possibly using amplitude amplification to boost the success probability. By extracting this compressed information directly from the wavefunction it may be possible to achieve sub-linear readout costs.

Our analysis of the Quantinuum H2 emulator and the IBM Heron noise models highlights the critical role of qubit connectivity in the NISQ era. The results in Fig. 5b indicate that the all-to-all connectivity of trapped-ion architectures provides an advantage over the fixed topology of the superconducting circuits for TrackHHL. The fixed topology requires many SWAP operations to route the non-local interaction terms of the tracking Hamiltonian on superconducting devices. This introduces a circuit depth overhead that scales unfavorably with detector density. While the IBM Heron r3 architecture shows clear improvement over its predecessors, the H2 platform currently achieves superior fidelity (F_H) and Signal Separation Indices (SSI) for higher track multiplicity events. However, both types of systems are currently limited to problem sizes of 4 particle tracks and 3 layers due to a buildup of gate errors.

It is important to understand that the results presented here rely on noise models that emulate the behavior of the physical qubits without Quantum Error Correction (QEC). While these noise models accurately capture machine errors, differences between simulators and quantum hardware are expected (1) due to the differences between the noise simulation and hardware noise, and (2) since different hardware platforms employ fundamentally different strategies to combat noise. Future work will focus on benchmarking the 1-Bit Quantum Filter within these logical error-correction layers, and execution on quantum hardware to experimentally validate the simulated noise model results on NISQ era hardware.

Conclusion

The 1-Bit Quantum Filter represents a shift for quantum tracking algorithms, moving away from generic quantum algorithms toward specialized, problem-specific spectral filtering. We have validated the theoretical two-qubit gate complexity $\mathscr{O}(\sqrt{N}\log N)$, simulated event sizes previously inaccessible by quantum algorithms, and demonstrated that for small systems the algorithm is robust against noise profiles of current generation hardware, particularly on architectures with high connectivity. As quantum hardware matures from noisy physical qubits to logically coherent systems, the 1-Bit Quantum Filter demonstrates potential to reconstruct events at the scale of the future HL-LHC.

Data availability

The raw data that support the findings of this study are available at https://github.com/Xenofon-Chiotopoulos/OneBQF

Code availability

The code and the dataset used in this work are open source and available at https://github.com/Xenofon-Chiotopoulos/OneBQF

References

1. Albrecht, J. *et al.* A roadmap for hep software and computing r&d for the 2020s. *Comput. Softw. for Big Sci.* **3**, DOI: 10.1007/s41781-018-0018-8 (2019).

2. Cámpora Pérez, D. H., Neufeld, N. & Riscos Núñez, A. Search by triplet: An efficient local track reconstruction algorithm for parallel architectures. *J. Comput. Sci.* **54**, 101422, DOI: 10.1016/j.jocs.2021.101422 (2021).

3. Ju, X. *et al.* Performance of a geometric deep learning pipeline for hl-lhc particle tracking. *The Eur. Phys. J. C* **81**, DOI: 10.1140/epjc/s10052-021-09675-8 (2021).

4. Correia, A., Giasemis, F. I., Garroum, N., Gligorov, V. V. & Granado, B. Graph neural network-based track finding in the lhcb vertex detector. *J. Instrumentation* **19**, P12022, DOI: 10.1088/1748-0221/19/12/p12022 (2024).

5. Jaschke, D. & Montangero, S. Is quantum computing green? an estimate for an energy-efficiency quantum advantage. *Quantum Sci. Technol.* **8**, 025001, DOI: 10.1088/2058-9565/acae3e (2023).

6. Auffèves, A. Quantum technologies need a quantum energy initiative. *PRX Quantum* **3**, 020101, DOI: 10.1103/PRXQuantum.3.020101 (2022).

7. Zlokapa, A. *et al.* Charged particle tracking with quantum annealing optimization. *Quantum Mach. Intell.* **3**, DOI: 10.1007/s42484-021-00054-w (2021).

8. Saito, Masahiko *et al.* Quantum annealing algorithms for track pattern recognition. *EPJ Web Conf.* **245**, 10006, DOI: 10.1051/epjconf/202024510006 (2020).

9. Tüysüz, C. *et al.* Hybrid quantum classical graph neural networks for particle track reconstruction. *Quantum Mach. Intell.* **3**, DOI: 10.1007/s42484-021-00055-9 (2021).

10. Nicotra, D. *et al.* A quantum algorithm for track reconstruction in the lhcb vertex detector. *J. Instrumentation* **18**, P11028, DOI: 10.1088/1748-0221/18/11/P11028 (2023).

11. Chiotopoulos, Xenofon *et al.* Trackhhl: A quantum computing algorithm for track reconstruction at the lhcb. *EPJ Web Conf.* **337**, 01181, DOI: 10.1051/epjconf/202533701181 (2025).

12. Harrow, A. W., Hassidim, A. & Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **103**, DOI: 10.1103/physrevlett.103.150502 (2009).

13. Denby, B. Neural networks and cellular automata in experimental high energy physics. *Comput. Phys. Commun.* **49**, 429–448, DOI: https://doi.org/10.1016/0010-4655(88)90004-5 (1988).

14. Peterson, C. Track finding with neural networks. *Nucl. Instruments Methods Phys. Res. Sect. A: Accel. Spectrometers, Detect. Assoc. Equip.* **279**, 537–545, DOI: https://doi.org/10.1016/0168-9002(89)91300-4 (1989).

15. AbuGhanem, M. Ibm quantum computers: evolution, performance, and future directions. *The J. Supercomput.* **81**, DOI: 10.1007/s11227-025-07047-7 (2025).

16. Moses, S. *et al.* A race-track trapped-ion quantum processor. *Phys. Rev. X* **13**, DOI: 10.1103/physrevx.13.041052 (2023).

17. Aaij, R. *et al.*. The lhcb upgrade i. *J. Instrumentation* **19**, P05065, DOI: 10.1088/1748-0221/19/05/P05065 (2024).

18. Cleve, R., Ekert, A., Macchiavello, C. & Mosca, M. Quantum algorithms revisited. *Proc. Royal Soc. London. Ser. A: Math. Phys. Eng. Sci.* **454**, 339–354, DOI: 10.1098/rspa.1998.0164 (1998).

19. Kitaev, A. Y. Quantum measurements and the abelian stabilizer problem (1995). quant-ph/9511026.

20. Coppersmith, D. An approximate fourier transform useful in quantum factoring (2002). quant-ph/0201067.

21. Aaronson, S. Read the fine print. *Nat. Phys.* **11**, 291–293, DOI: 10.1038/nphys3272 (2015).

22. Dervovic, D. *et al.* Quantum linear systems algorithms: a primer (2018). 1802.08227.

23. Suzuki, M. Generalized trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems. *Commun. Math. Phys.* **51**, 183–190 (1976).

24. Horn, R. A. & Johnson, C. R. *Matrix Analysis* (Cambridge University Press, 1985).

25. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, 2010).

26. Cowtan, A., Dilkes, S., Duncan, R., Simmons, W. & Sivarajah, S. Phase gadget synthesis for shallow circuits. *Electron. Proc. Theor. Comput. Sci.* **318**, 213–228, DOI: 10.4204/eptcs.318.13 (2020).

27. Di Matteo, O. *et al.* Improving hamiltonian encodings with the gray code. *Phys. Rev. A* **103**, 042405, DOI: 10.1103/PhysRevA.103.042405 (2021).

28. Barenco, A. *et al.* Elementary gates for quantum computation. *Phys. Rev. A* **52**, 3457–3467, DOI: 10.1103/physreva.52.3457 (1995).

29. Grover, L. K. A fast quantum mechanical algorithm for database search (1996). quant-ph/9605043.

30. Motwani, R. & Raghavan, P. *Randomized Algorithms* (Cambridge University Press, 1995).

31. Sivarajah, S. *et al.* t|ket⟩: a retargetable compiler for nisq devices. *Quantum Sci. Technol.* **6**, 014003, DOI: 10.1088/2058-9565/ab8e92 (2020).

32. Quantinuum. Quantinuum nexus: A cloud-based quantum computing platform. https://docs.quantinuum.com/nexus (2024). Accessed: 2024-12-16.

33. Javadi-Abhari, A. *et al.* Quantum computing with qiskit (2024). 2405.08810.

Acknowledgements

This study was supported by the Special Research Fund (BOF24DOCUM01) of Hasselt University and is part of the project *Fast sensors and Algorithms for Space-time Tracking* and Event Reconstruction (FASTER) with project number OCENW.XL21.XL21.076 of the research programme ENW - XL which is (partly) financed by the Dutch Research Council (NWO).

The authors wish to thank Jan Blommaart, Alain Chancé, Taha Selim for stimulating discussions and suggestions. We are thankful to IBM Research through the CERN Quantum Technology Initiative (QTI) and Quantinuum for providing access to their quantum platforms, resulting in the studies reported in this paper. We also thank the Coherence and Quantum Technology group of Eindhoven University for initial tests on their simulation platform. Finally, we are grateful for the support from the LHCb Data Processing and Analysis (DPA) project.

Author contribution

Authors and Affiliations

Nikhef National Institute for Subatomic Physics, Science Park 105, 1098 XG Amsterdam, The Netherlands

Xenofon Chiotopoulos, Marcel Merk

Maastricht University, Faculty of Science and Engineering, Gravitational Waves and Fundamental Physics department, Duboisdomain 30, 6229 GT Maastricht, The Netherlands

Xenofon Chiotopoulos, George Scriven, Davide Nicotra, Jacco de Vries, Marcel Merk

Hasselt University, Faculty of Sciences and Data Science Institute, Agoralaan gebouw D, 3590 Diepenbeek, Belgium

George Scriven, Jochen Schütz

Maastricht University, Faculty of Science and Engineering, Department of Advanced Computing Sciences, Paul-Henri Spaaklaan 1, 6229 EN Maastricht, The Netherlands

Xenofon Chiotopoulos, Kurt Driessens, Mark H.M. Winands

Contributions

X.C. derived and implemented the 1-Bit Quantum Filter and the Direct Structural Synthesis (DSS) method, and performed the theoretical complexity analysis. X.C. also conducted the empirical data fitting for the complexity scaling and executed the noise emulation benchmarks on both hardware platforms. G.S. developed the simulation framework based on original work by D.N. and updated the HHL implementation to the latest Qiskit version. D.N. and G.S. strongly contributed to the theoretical discussions regarding the quantum algorithm, the DSS method, and the complexity derivations. All authors were involved in the development of the algorithm, discussed the results and contributed to the final manuscript.

Corresponding author

Correspondence to Xenofon Chiotopoulos (xenofon.chiotopoulos@maastrichtuniversity.nl)

Competing interests

The authors declare no competing interests.