Noisy simulations of quantum walk and quantum walk search via quantum cellular automata on a semiconducting spin processor emulator

A. Mammola [1,2,*] Q. Schaeverbeke,[1] and G. Di Molfetta [2]
[1] *C12 Quantum Electronics, Paris, France*
[2] *Aix Marseille University, CNRS, LIS, Marseille, France*

(Received 27 May 2025; accepted 8 December 2025; published 2 February 2026)

In this work, we map noisy intermediate-scale quantum (NISQ)-friendly implementations of the noninteracting quantum cellular automata (QCA) to a circuit quantum electrodynamics (cQED) hardware. We perform both noiseless and noisy simulations of the QCA one-particle sector, namely, the quantum walk, on N-cycles and $N \times N$ torus graphs. Moreover, within this framework, we also investigate the search problem and present a circuit for preparing the W state (i.e., the Dicke state with Hamming weight 1) using only $N-1$ $\sqrt{\text{iSWAP}}$ gates and no ancilla qubits. The noiseless simulations are conducted with the Qiskit Aer simulator, while the noisy simulations with C12 Quantum Electronics' in-house noisy emulator, *Callisto*. We benchmark the performance of our implementations by analyzing the simulations via relevant metrics and quantities such as the state count distributions, the Hellinger fidelity, the ℓ^1 distance, the hitting time, and the success probability. Our results demonstrate that the QCA framework, in combination with cQED processors, holds promise as an effective platform for early NISQ implementations of quantum walk and quantum walk search algorithms.

DOI: 10.1103/wc4h-thmg

I. INTRODUCTION

Proving quantum advantage within the noisy intermediate-scale quantum (NISQ) [1] era is among the main challenges currently faced by the quantum computing community. Indeed, crafting a successful NISQ implementation is hard and requires suitable applications and algorithms, NISQ-driven algorithmic formulations, and hardware-tailored implementations. Among the category of suitable algorithms, one can certainly find those rephrased in terms of the quantum walk (QW) [2], the quantum equivalent of the classical random walk, which has been theoretically proven to bring a speedup, depending on the nature of the graph, e.g., exponential speedup on glued trees [3]. Within the quantum formalism, random walks can be formulated within a discrete- or continuous-time framework and exhibit simple properties, featuring a predictable phenomenology, a clear scaling with the system size, and showcasing explicit quantum advantage [4]. Moreover, much like their classical counterparts, quantum walks have diverse applications across fields such as quantum computing, quantum simulation, quantum information, and graph theory [5]. In the latter category, the quantum walk search [6] stands out as a reference application of QWs to graph-based search problems, requiring only a slight modification to the standard QW implementation.

The aforementioned properties have rendered discrete- and continuous-time QW and QW search algorithms popular benchmarking tools, respectively, for digital and analog NISQ implementations, with many examples already present in the literature [4,7–9]. In particular, discrete-time QW (DTQW) [2] implementations provide valuable insight into the quality of the many gate-based NISQ processors and emulators, which are currently being developed within the industry. Even so, such digital implementations are typically translated into circuits with considerable size and depth, thus restricting the number of time steps that can be accurately executed on NISQ devices.

Moreover, QWs are known to be the one-particle sector of quantum cellular automata (QCA) [10]. QCA are lattice-based universal computational models that evolve in discrete time steps under a unitary, local, and translation-invariant operator [11–13], making them *close-to-physics* models. As a result, QCA-based circuit implementations often facilitate the translation of quantum algorithms into straightforward experimental realizations, typically using low-connectivity lattices and two-qubit gates only. Comprehensibly, these features are particularly valuable for early NISQ simulations, as well as for applications in searching and optimization problems, distributed quantum computing [11], quantum field theory simulations [14,15], and quantum error correction [16]. Different QCA designs can give rise to different DTQW dynamics, and among these, we can find schemes where the QCA evolution operator is only built with the two-qubit XY gate [17]. From an experimental point of view, the XY gate is derived from the XY interaction [18], which emerges, among many, in Josephson charge qubits linked by Josephson junctions [19], nuclear spins interacting through a two-dimensional (2D) electron gas [20], and quantum dot spins coupled by a cavity [21]. In particular, hardware based on the latter, such as the circuit quantum electrodynamics (cQED) processor of C12 Quantum Electronics [22], has

*Contact author: andrea@c12qe.com

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

recently attracted significant attention due to its potential to enable highly connected architectures—a rare and valuable feature among NISQ devices. With their high connectivity and the integration of the XY gate into their native gate set, cQED processors like C12's are ideal candidates for testing and being benchmarked by QCA implementations.

In this work, we map QCA implementations of the QW and QW search over the cycle and torus to C12 Quantum Electronics cQED hardware. The QCA implementations are then executed with Qiskit software development kit (SDK) [23], with both Qiskit Aer noiseless simulator and *Callisto* [24], C12 Quantum Electronics in-house noisy emulator. The simulations act as a proof of concept, showcasing the feasibility of efficiently implementing quantum walks and quantum walk search via QCA on cQED NISQ processors. Moreover, via the Hellinger fidelity [25] and suitable QW search quantities [26], we study the impact of noise and benchmark the expected performance of C12's processor and our QCA implementations with respect to other QW implementations over quantum machines or emulators [4,7,27]. To the authors' knowledge, this study provides the first evaluation of QW and QW search implementations via QCA on an NISQ emulator. Clearly, further investigation is required to test such implementations over a real cQED processor, in higher dimensions or over different graphs [28], as algorithm subroutines [29], for state preparation [30], or as a starting point for more complex simulations of quantum physical theories, e.g., the Schwinger model [14] or quantum phases of matter [15].

This document is organized as follows. Section II briefly introduces the noninteracting one-particle sector QCA model simulating the discrete-time QW and QW search. Section III provides a mapping of the QCA models introduced to a cQED hardware, with the specific example of the C12 Quantum Electronics processor. Section IV provides a description of the algorithm and the software used to perform the QW and QW search simulations over cycles and torus graphs, with a discussion of the noisy simulation results performed with C12's in-house emulator *Callisto*. Finally, in Sec. V, we summarize the results obtained in this work and provide an outlook for future directions to be explored.

II. MODEL

A quantum walker moves on a d-dimensional graph under the action of a coin operator, acting on the walker's internal state encoding for its direction, and a coin-conditioned shift operator, acting on the lattice degrees of freedom. The QW dynamics is known to be simulated by the noninteracting one-particle sector QCA. Let us now describe a QCA model implementing the QW and QW search over one- and two-dimensional lattices with periodic boundary conditions, respectively, cycles and torus graphs.

Consider an N-cycle $\mathcal{C}(V_{\mathcal{C}},\ E_{\mathcal{C}})$ and an $N \times N$ torus $\Gamma(V_{\Gamma},\ E_{\Gamma})$ with fixed vertex sets $V_$ and edge sets $E_$, respectively, associated with the Hilbert spaces $\{\mathcal{H}_{V_{\mathcal{C}}},\ \mathcal{H}_{E_{\mathcal{C}}}\}$ and $\{\mathcal{H}_{V_{\Gamma}},\ \mathcal{H}_{E_{\Gamma}}\}$. We label each vertex of the cycle $\mathcal{C}$ with $j \in \mathbb{N}$ and place there a qubit with the state $|1\rangle_j\ (|0\rangle_j) \in \mathcal{H}_{V_{\mathcal{C}}}$, encoding for an (un)occupied site. In the case of the torus Γ, each vertex is labeled with $(j,\ l)$, where $j \in \mathbb{N}$ on the x axis and $l \in \mathbb{N}$ on the y axis, and can be associated with the state $|1\rangle_{jl}\ (|0\rangle_{jl}) \in \mathcal{H}_{V_{\Gamma}}$, encoding for an (un)occupied site. Within the torus, sites are sequentially labeled along each axis. For instance, in a 4×4 torus, the first site of the first row is (0, 0), the first site of the second row is (1, 4), and the last site of the last row is (15, 15). To simulate the QW dynamics, the cycle (torus) QCA has to be restricted to its one-particle sector; namely, for all sites $j(l)$, only states of the likes $|0_{0(0)},\ \ldots,\ 1_{j(l)},\ \ldots,\ 0_{N(N)}\rangle$ are accepted. The graphs $\mathcal{C}(V_{\mathcal{C}},\ E_{\mathcal{C}})$ and $\Gamma(V_{\Gamma},\ E_{\Gamma})$ are divided into tessellations $\{\mathcal{T}_k\}_{k=0}^{K-1}$ [17]. The global time-evolution operator of the QCA is then obtained as the product of unitary operators defined over each tessellation $\mathcal{W} = \prod_{k=0}^{K-1} W_{\mathcal{T}_k}(\theta_k)$, where $W_{\mathcal{T}_k}(\theta_k) = \bigoplus_{\alpha \in \mathcal{T}_k} e^{-i\theta_k H_\alpha}$ and α is a complete subgraph of G. As depicted in Fig. 1, the one- and two-dimensional lattice minimal tessellation covers are, respectively, composed of the following two tessellations $\mathcal{T}_k$:

$$\begin{aligned} \mathcal{T}_0 &= \{2j,\ 2j+1\}, \\ \mathcal{T}_1 &= \{2j+1,\ 2j+2\}, \end{aligned} \tag{1}$$

and the following four tessellations $\mathcal{T}_{k_x k_y}$:

$$\begin{aligned} \mathcal{T}_{0_x 0_y} &= \{(2j, l_1),\ (2j+1, l_2)\}, \\ \mathcal{T}_{1_x 0_y} &= \{(2j+1, l_1),\ (2j+2, l_2)\}, \\ \mathcal{T}_{0_x 1_y} &= \{(j_1, 2l),\ (j_2, 2l+1)\}, \\ \mathcal{T}_{1_x 1_y} &= \{(j_1, 2l+1),\ (j_2, 2l+2)\}, \end{aligned} \tag{2}$$

where j and $l \in \mathbb{N}$. Each tessellation consists of the set of points $\{(j, l)\}$ satisfying a condition on the corresponding axis, while being independent with respect to the other axis. For example, an x-axis tessellation is determined by conditions on the x coordinates, while the y coordinates are left trivial (denoted above by l_1 and l_2); the same holds for tessellations on the y axis. Accordingly, the cycle and torus QCA time-evolution operators can be written as

$$\begin{aligned} \mathcal{W}_{\text{cycle}} &= \mathcal{W}_{\mathcal{T}_1} \cdot \mathcal{W}_{\mathcal{T}_0} = \prod_{k=0}^{1} \left[\bigotimes_{j \in \mathcal{T}_k} W_{j,j+1}(\theta_j) \right] \bigotimes_{j \notin \mathcal{T}_k} \mathbb{1}_j \\ &= \left[\bigotimes_{j=0}^{N-1} W_{2j+1,2j+2}(\theta_j) \right] \cdot \left[\bigotimes_{j=0}^{N-1} W_{2j,2j+1}(\theta_j) \right], \\ \mathcal{W}_{\text{torus}} &= \mathcal{W}_{\mathcal{T}_{1_x 1_y}} \cdot \mathcal{W}_{\mathcal{T}_{1_x 0_y}} \cdot \mathcal{W}_{\mathcal{T}_{0_x 1_y}} \cdot \mathcal{W}_{\mathcal{T}_{0_x 0_y}} \\ &= \prod_{k_x, k_y=0}^{1} \left[\bigotimes_{j \in \mathcal{T}_{k_x k_y}} W_{j,j+1}(\theta_j) \right] \bigotimes_{j \notin \mathcal{T}_{k_x k_y}} \mathbb{1}_j \\ &= \left[\bigotimes_{y=0}^{N-1} W_{2l+1,2l+2}(\theta_y) \right] \cdot \left[\bigotimes_{x=0}^{N-1} W_{2j+1,2j+2}(\theta_x) \right] \\ &\quad \cdot \left[\bigotimes_{y=0}^{N-1} W_{2l,2l+1}(\theta_y) \right] \cdot \left[\bigotimes_{x=0}^{N-1} W_{2j,2j+1}(\theta_x) \right], \end{aligned} \tag{3}$$

where $\bigotimes_{j \notin \mathcal{T}_k} \mathbb{1}_j$ and $\bigotimes_{j \notin \mathcal{T}_{k_x k_y}} \mathbb{1}_j$ indicate that the transition function is acting as the identity on the sites that are not pertaining to the chosen tessellation (i.e., $j \notin \mathcal{T}$). Note that these identities are omitted in the last term of the equations for

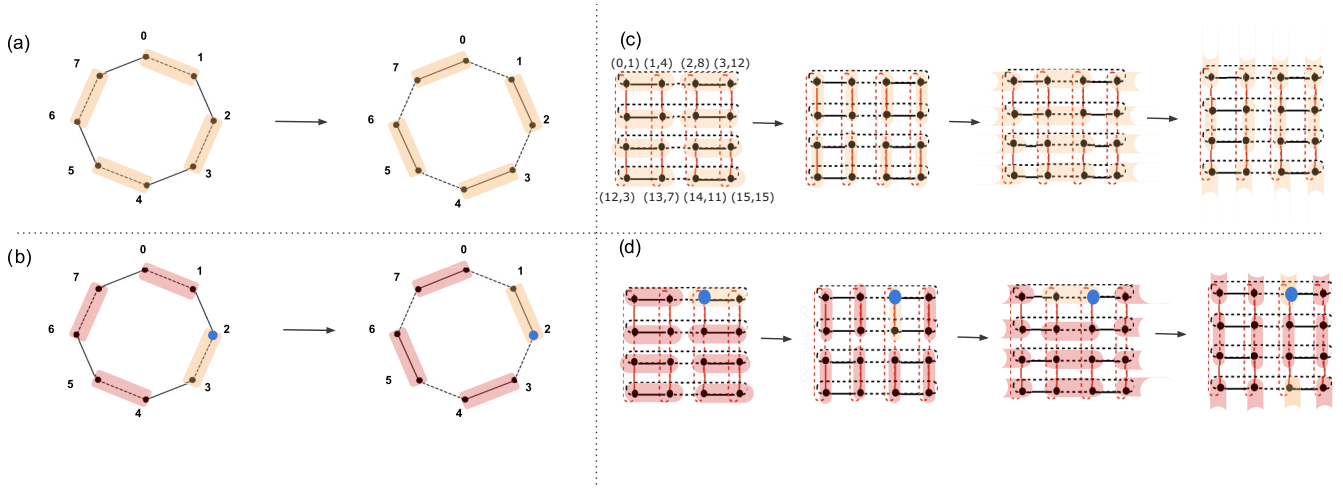


FIG. 1. Left: One time step of an 8-cycle QW [panel (a)] and QW search [panel (b)]. The minimal tessellation cover of cycles is two: $\mathcal{T}_0$ (dashed line) and $\mathcal{T}_1$ (solid line) [see Eq. (1)]. Each time step is divided into two substeps, where two-qubit matrices $W_{j,j+1}(\theta_j = \frac{\pi}{4})$ (orange box) and $W_{j,j+1}(\theta_j = \frac{\pi}{2})$ (red box) are applied via the transition functions $W_{\mathcal{T}_0}$ and $W_{\mathcal{T}_1}$, associated to each tessellation [see Eq. (3)]. Right: One time step of a 4×4 torus QW [panel (c)] and QW search [panel (d)]. The minimal tessellation cover of torus graphs is four: $\mathcal{T}_{0_x 0_y}$ (black solid line), $\mathcal{T}_{1_x 0_y}$ (black dashed line), $\mathcal{T}_{0_x 1_y}$ (red solid line), and $\mathcal{T}_{1_x 1_y}$ (red dashed line) [see Eq. (2)]. For clarity, only the indices of the first and last rows of the first torus in panel (c) are shown. A single time step is divided into four substeps, where two-qubit matrices $W_{j,j+1}(\theta_j = \frac{\pi}{4})$ (orange box) and $W_{j,j+1}(\theta_j = \frac{\pi}{2})$ (red box) are applied via the transition functions $W_{\mathcal{T}_{0_x 0_y}}$, $W_{\mathcal{T}_{1_x 0_y}}$, $W_{\mathcal{T}_{0_x 1_y}}$, and $W_{\mathcal{T}_{1_x 1_y}}$, associated with each tessellation [see Eq. (3)]. The blue dot in panels (b) and (d) indicates the marked vertex in the search.

notational simplicity, and that the subscript of the two-qubit matrix $W_{j,j+1}$ indicates only the qubit coordinates within the chosen tessellation (along either the x or y axis). For example, $W_{2j,2j+1}$ with $x = 0$ acts on the two qubits located at positions $(\mathbf{0}, 0)$ and $(\mathbf{1}, 4)$, respectively. The two-qubit gate

$$W_{j,j+1}(\theta_j) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_j & i\sin\theta_j & 0 \\ 0 & i\sin\theta_j & \cos\theta_j & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{4}$$

is the XY gate [31] and acts upon two neighboring qubits located in the x or y axis of the lattice at positions $(2j,\ 2j+1)$ and $(2j+1,\ 2j+2)$, as depicted in Fig. 1. The full unitary QCA evolution, generating the QW dynamics, for a single time step can be written as $|\psi(t+1)\rangle = \mathcal{W}_{\text{cycle}}\,|\psi(t)\rangle$ for the cycle and $|\psi(t+1)\rangle = \mathcal{W}_{\text{torus}}\,|\psi(t)\rangle$ for the torus. Various QWs can be implemented for different choices of θ_i. We set $\theta_i = \frac{\pi}{4}$, for which the XY gate is equal to $\sqrt{\text{iSWAP}}$ gate [32]. For the QW search instead, given a marked vertex v_m, we set $\theta_i = \frac{\pi}{4}$ for $i = v_m$ and $\theta_i = \frac{\pi}{2}$, $\forall i \neq v_m$, for which the XY gate equals the iSWAP gate.

As a natural consequence of QCA being *physicslike* models, the QW implementations described above require $\mathcal{O}(n)$ qubits and $\mathcal{O}(nt)$ two-qubit gates (i.e., $\sqrt{\text{iSWAP}}$) with a depth of $\mathcal{O}(t)$, where t is the time step. To date, several efficient circuit-based implementations of one-dimensional (1D) QWs have been proposed. All methods require $\mathcal{O}(\log N)$ qubits and differ in how the shift operator is implemented. Shakeel [33] introduced a design using a quantum Fourier transform (QFT)-based *incrementer* and $\mathcal{O}(n^2 t)$ cnot gates with a circuit depth of $\mathcal{O}(nt)$. Razzoli *et al.* [8] later improved this by reducing the QFT usage, lowering the gate count and depth to $\mathcal{O}(n^2)$ and $\mathcal{O}(nt + n^2)$, respectively. Additionally, various efficient incrementers based on generalized cnots have been proposed, including

(1) Gidney's scheme [34] with Toffoli cost of $32n$ ($64n$ cnots), depth of $\mathcal{O}(n)$, and one clean ancilla qubit;

(2) Nie *et al.* [35] with Toffoli cost $\mathcal{O}(n)$, depth $\mathcal{O}(\log^2 n)$, and one clean ancilla qubit;

(3) Khattar and Gidney [36] with Toffoli cost $3n$, depth $\mathcal{O}(n)$, and $\log_2^* n$ clean ancilla qubits.

III. QCA MAPPING TO cQED PROCESSOR

In the following, we describe a mapping of the QCA described in the previous section to a circuit quantum electrodynamics processor, with the specific example of C12 Quantum Electronics hardware [22]. We start with a brief introduction of C12's processor, to then provide the QCA mapping.

cQED processors [37,38] are based on cavity quantum electrodynamics [39], which focuses electric fields in small regions around atoms to control light-matter interactions and generate entanglement. This concept can be applied to quantum circuits made from semi- or superconducting components, where artificial atoms (qubits) are embedded in a microwave cavity. The cavity acts as a quantum bus, enabling long-range entanglement by exchanging photons between qubits and thus, in principle, allowing for all-to-all connectivity. Within such setting, one-qubit gates are implemented with fast electric lines, while two-qubit gates are achieved by coupling qubits synchronously to the microwave cavity [40].

A. cQED semiconducting spin processor

An example of a cQED processor is C12 Quantum Electronics hardware, which stems from the single electron spin-qubit paradigm [41]. The particularities of C12's

technology are twofold with respect to this paradigm. The first one is the usage of a microwave resonator to mediate the coupling between the qubits in a cQED-type architecture [40]. This allows to reach all-to-all connectivity in the processor at the cost of a decoherence channel in the Purcell effect. The second specificity is the usage of carbon nanotube as the host material for the spin qubit. Carbon nanotubes, although less studied, have the advantage of being structurally simpler than the other materials typically used for spin qubits. Indeed, while silicon or germanium will consist of layers deposited onto a substrate in which bidirectional confinement has to be performed, the quasi-1D structure of the tube has a natural confinement in one direction and therefore only requires confining the electron in the remaining direction. On top of that, the tube can be suspended above the electrode used for the confinement, allowing for a molecular structure suspended in vacuum and consequently more isolated from its environment.

This type of cQED-based processor is promising but still requires further development to reach state-of-the-art capabilities. Indeed, while the all-to-all connectivity offered by the resonator should, in principle, provide significant advantages for quantum computation, current chips do not achieve state-of-the-art coherence times [42,43]. Despite this, carbon nanotube-based spin qubits have already shown promise, currently holding the record for coherence times among cQED spin qubits [44]. At present, the primary challenge limiting the scalability of this platform in terms of two-qubit entanglement is achieving sufficient coupling with the resonator to enable high-fidelity gates. Progress on high-impedance circuit design is expected to enhance this coupling. Resonance between qubits constitutes another potential limitation; however, it can be actively controlled via electrostatic tuning of the double quantum dot and is therefore not anticipated to pose a major obstacle. Variations in decoherence rates across the qubit register may still affect gate fidelity depending on the specific configuration. While promising, cQED semiconducting spin platforms still require further development before they can provide a quantum processor suitable for experiments. Accordingly, the simulations presented in this work rely on theoretical projections using a hardware emulator with C12's proprietary noise model, which reflects the expected behavior of the processor under the assumption that the aforementioned technical limitations can be addressed in future experimental implementations. Indeed, two-qubit gates on these platforms have only been demonstrated very recently, with fidelities around 80% [40], highlighting the need for substantial optimization before practical experimental applications are feasible. While precise timelines remain uncertain, it is reasonable to anticipate that quantum walk experiments with a few qubits could become feasible in the near future.

C12's gate set includes the three Pauli rotations $R_X(\theta)$, $R_Y(\theta)$, $R_Z(\theta)$, with θ depending on experimentally controllable parameters and the gate time, and the two-qubit XY gate:

$$U^{(2)}(\theta = \gamma t_g) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & i\sin\theta & 0 \\ 0 & i\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{5}$$

where t_g is the gate time and γ the cavity-mediated qubit-qubit coupling. For $t_g = \frac{\pi}{2\gamma}$, the two-qubit matrix $U^{(2)}$ implements the iSWAP gate, and for $t_g = \frac{\pi}{4\gamma}$, the $\sqrt{\text{iSWAP}}$. On top of this basis gate set, to be operative, C12's multiqubit processor needs on-off gates, which decouple unused qubits from the resonator during an operation and set them within the memory or *idle* state. Experimentally, this can be achieved by forcing the electron to be stuck in one of the dots, e.g., by applying a large bias voltage between the two dots, in a process known as the Landau-Zener transition [45].

1. Decoherence

Decoherence affects the qubit's lifetime and the gate error rates, thus limiting the maximal depth of a quantum circuit executable on the processor. To have a realistic description of the qubit and its operation, it is crucial to account for the sources of decoherence within the qubit's external environment. We separate the sources into two categories based on their effect on the qubit, i.e., relaxation and dephasing, which, respectively, describe processes by which (1) the qubit relaxes toward its ground state over a time T_1 and (2) the qubits lose coherence without effect on the populations with time T_2. The main sources include the microwave cavity, modeled as a single-mode resonator, where qubits are influenced by the Purcell effect [46], leading to energy relaxation. This effect is determined by the cavity's quality factor, which depends on imperfections in the conductors used to build it, the electron-photon coupling, the qubit-cavity frequency shift, and the electronics around the cavity. Another source is the elastic properties of carbon nanotubes, where phonon vibrational modes cause both dephasing and relaxation of the qubit state [47–51]. Charge noise also plays an important role. Defined as fluctuations in the qubit's electrostatic environment, it predominantly causes dephasing and also contributes to relaxation [52–55]. Additionally, crosstalk errors arise from unwanted interactions during computation [56], such as residual idle-active qubit couplings emerging in cavity-mediated two-qubit gates, which could potentially restrict the hardware's connectivity.

Formally, we treat decoherence within the density matrix formalism [57] and write the time evolution of the system via the Lindblad equation [58,59]

$$\dot{\rho} = -i[H_g, \rho] + K\mathcal{L}\left[\sum_i \sigma_-^{(i)}\right](\rho) + \delta\mathcal{L}\left[\sum_i \sigma_3^{(i)}\right](\rho), \tag{6}$$

where ρ is the system's density matrix, H_g is the gate Hamiltonian, i.e., the Hamiltonian describing the system whose unitary is among C12's basis gate set, and $\mathcal{L}[\sum_i \sigma_-^{(i)}](\rho)$ and $\mathcal{L}[\sum_i \sigma_3^{(i)}](\rho)$ are the Lindblad operators acting on the i qubit, respectively, encoding for the relaxation of the states with a rate K and dephasing with a rate δ. The Lindblad operator is defined as $\mathcal{L}[A](\rho(t)) = A\rho(t)A^\dagger - \frac{1}{2}AA^\dagger\rho(t) - \frac{1}{2}\rho(t)AA^\dagger$.

2. C12's emulator Callisto

C12's in-house emulator *Callisto* is built on top of the Qiskit library with the noise model described in the previous

TABLE I. *Callisto* native gate set and gate fidelity.

Gate	Fidelity
$R_X(\theta),\ R_Y(\theta)$	0.9999
$R_Z(\theta)$	1
$U^{(2)}(\frac{\pi}{2}) = \text{iSWAP}$	0.9987
$U^{(2)}(\frac{\pi}{4}) = \sqrt{\text{iSWAP}}$	0.9991

section, allowing to perform realistic noisy simulations. The version of the emulator used for this work is $c12 - callisto - clients == 2.0.1$; it allows for all-to-all connectivity and has the basis gate set and fidelity reported in Table I.

Leveraging the high connectivity of cQED hardware, the QCA described in Sec. III can be easily implemented with C12's qubits of Fig. 2. The QW and QW search QCA dynamics is then reproduced by applying at each time step the native two-qubit gates $U^{(2)}(\theta = \gamma t_g)$, respectively, with gate times $t_g = \frac{\pi}{4\gamma}$ and $t_g = \frac{\pi}{2\gamma}$, as the unitaries $W_{2i,\,2i+1}(\theta_i)$ and $W_{2i+1,\,2i+2}(\theta_i)$ of Eq. (3). We note that the periodic boundary conditions impose that, at the borders of the one- and two-dimensional lattices, the two-qubit matrix $W_{i,\,i+1}(\theta_i)$ acts on the last and first qubits of the row or column.

IV. SIMULATIONS

In the following, we will describe the algorithms and the software used to run noiseless and noisy simulations of the QW and QW search over the cycle and torus via a cQED QCA. In the second part, we will outline the metrics and quantities used to study and benchmark the QW simulations and then discuss the simulation results.

A. Algorithm

Algorithms 1 and 2, respectively, describe the implementation of QW and QW search on cycle and torus graphs via QCA models mapped to cQED hardware. Following the mapping outlined in Sec. III, a system of N qubits is configured with the appropriate connectivity to represent either a cycle $\mathcal{C}(E_\mathcal{C}, V_\mathcal{C})$ or a torus $\Gamma(E_\Gamma, V_\Gamma)$, enforcing periodic boundary conditions in both cases. The qubits can be operated upon with the hardware's native gate set of Table I and the QW dynamics is generated with the time-evolution operator of Eq. (3). The algorithms are divided into initialization, where the qubits are initialized in the required initial state, and propagation, where the QCA transition function is applied to the system N_{stps} times and the final state $|\psi_{N_{\text{stps}}}\rangle$ is returned.

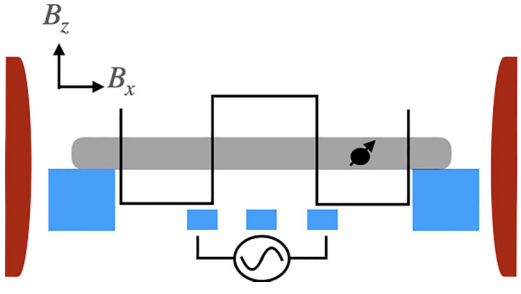


FIG. 2. Schematic of C12's spin qubit. An electron is hosted in a carbon nanotube (gray tube), suspended over five electrodes (blue squares) generating a double quantum dot (black squared well) trapping the particle (black arrowed dot), and immersed in an asymmetric magnetic field. One-qubit gates are then realized by changing the dot's voltage (black wave), and two-qubit gates are realized by embedding two qubits within the same microwave cavity (red resonator plates).

ALGORITHM 1. QW on N-cycles ($N \times N$ torus graphs) via cQED-QCA.

Data: N qubits with all-to-all connectivity mapped to N-cycle $\mathcal{C}(E_\mathcal{C}, V_\mathcal{C})$ ($N \times N$ torus $\Gamma(E_\Gamma, V_\Gamma)$)

Initialization:

if *symmetric == True* **then**

$|\psi_0\rangle = U_{i,i+1}(\frac{\pi}{4})^{(2)} \cdot R_{X_i}(\frac{\pi}{2}) |0_0, \cdots, 0_{N-1}\rangle$

$= \frac{1}{\sqrt{2}} |0_0, \cdots\rangle (|0_i\rangle + |1_i\rangle) |\cdots, 0_{N-1}\rangle$;

else

$|\psi_0\rangle = R_{X_i}(\frac{\pi}{2}) |0_0, \cdots, 0_{N-1}\rangle$;

end

$\theta_i = \frac{\pi}{4}\ \forall i$;

$\mathcal{W}_{\text{cycle(torus)}} = \prod_{k(,l)=0}^{1} \left[\bigotimes_{i \in \mathcal{T}_{k(l)}}^{N-1} W_{i,i+1}(\theta_i) \right]$,

with $W_{i,i+1}(\theta_i) = U^{(2)}(\frac{\pi}{4})$, Eq. (5) ;

Propagation:

for $(n = 1,\ n < N_{stps})$: **do**

$|\psi_n\rangle = \mathcal{W}_{cycle(torus)} |\psi_{n-1}\rangle$;

end

Return $|\psi_{N_{stps}}\rangle$;

ALGORITHM 2. QW search on N-cycles ($N \times N$ torus graphs) via cQED-QCA.

Data: N qubits with all-to-all connectivity mapped to N-cycle $\mathcal{C}(E_\mathcal{C}, V_\mathcal{C})$ ($N \times N$ torus $\Gamma(E_\Gamma, V_\Gamma)$).

Initialization:

$|\psi_0\rangle = Dicke_one_initializer(N)$, see Algorithm 3

$= \frac{1}{\sqrt{2^N}} \sum_{i=0}^{N-1} |\cdots, 0_{i-1}, 1_i, 0_{i+1}, \cdots\rangle$;

mark vertex v_m |

$\theta_i = \frac{\pi}{4}$, for $i = v_m$ and $\theta_i = \frac{\pi}{2}$, $\forall i \neq v_m$;

$W_{i,i+1}(\theta_i) = R_{Z_i}(-\frac{\pi}{2}) \otimes R_{Z_{i+1}}(-\frac{\pi}{2}) \cdot U^{(2)}(\frac{\pi}{2})\ \forall i \neq v_m$,

$W_{i,i+1}(\theta_i) = U^{(2)}(\frac{\pi}{4})$ for $i = v_m$, Eq. (5) ;

$\mathcal{W}_{\text{cycle(torus)}} = \prod_{k(,l)=0}^{1} \left[\bigotimes_{i \in \mathcal{T}_{k(l)}}^{N-1} W_{i,i+1}(\theta_i) \right]$,

Propagation:

for $(n = 1,\ n < N_{stps})$: **do**

$|\psi_n\rangle = \mathcal{W}_{cycle(torus)} |\psi_{n-1}\rangle$;

end

Return $|\psi_{N_{stps}}\rangle$;

In the QW (Algorithm 1), if "symmetric" is true, the system is initialized in a superposition of two chosen sites, whereas if

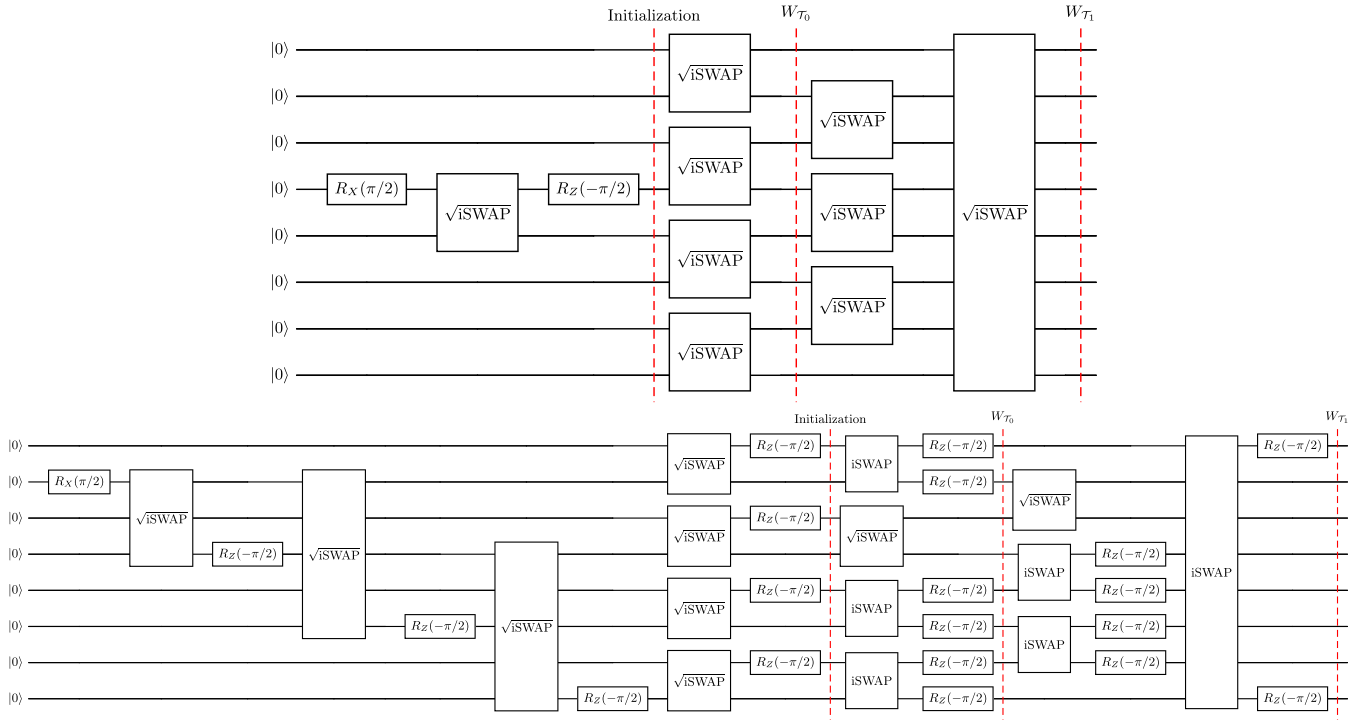


FIG. 3. Circuits implementing the initialization and one time step of the QW (top) and QW search (bottom) via QCA over an 8-cycle. Time steps are divided into $t_{\mathcal{T}_0}$ and $t_{\mathcal{T}_1}$, according to the tessellation cover defined in Eqs. (1) and (3) (see Fig. 1). In the QW, the qubits are initialized in the state $|\psi_0\rangle = \frac{1}{\sqrt{2}}(|3\rangle + |4\rangle)$, while in the QW search they are initialized in an equiprobable superposition of the allowed one-particle sector states (i.e., Dicke state with Hamming weight 1) $\frac{1}{\sqrt{2^N}} \sum_{i=0}^{N-1} |\ldots, 0_{i-1}, 1_i, 0_{i+1}, \ldots\rangle$. In the QW search, the marked vertex is mapped to the third qubit (third line of the circuit).

"symmetric" is false, the system is simply initialized on one chosen site. Propagation is performed using the two-qubit gate $U^{(2)}(\frac{\pi}{4})$ from Eq. (5).

In the QW search (Algorithm 2), the *Dicke_one_initializer(N)* function (Algorithm 3) initializes N qubits in the Dicke $|D_k^N\rangle$ state with Hamming weight $k = 1$, i.e., an equiprobable superposition of the one-particle sector states $|\psi_0\rangle = \sum_{i=0}^{N-1} \frac{1}{\sqrt{N}} |\ldots, 0_{i-1}, 1_i, 0_{i+1}, \ldots\rangle$, via repeated application of the two-qubit gate $U^{(2)}(\frac{\pi}{4}) = \sqrt{\text{iSWAP}}$. Precisely, the circuit we present here requires $N - 1$ $\sqrt{\text{iSWAP}}$ gates. A vertex v_m is then marked: $\theta_i = \frac{\pi}{4}$ for $i = v_m$ and $i = \frac{\pi}{2}$ $\forall i \neq v_m$; namely, the transition function applies $U^{(2)}(\frac{\pi}{4})$ to unmarked sites and $U^{(2)}(\frac{\pi}{2})$ to the marked one.

An example QW and QW search circuit over an 8-cycle is shown in Fig. 3.

Software

Algorithms 1 and 2 are translated into circuits and executed using Qiskit SDK [23]. The noiseless simulations are run with Qiskit Aer backend with the following modules and versions: qiskit==0.41.0, qiskit-aer==0.11.2, qiskit-ibmq-provider==0.20.0, and qiskit-terra==0.23.1. The noisy simulations are run with *Callisto* $c12 - callisto - clients ==$ 2.0.1 [24], with all-to-all connectivity and the basis gate set and fidelity reported in Table I.

ALGORITHM 3. *Dicke_one_initializer(N).*

Data: N qubits initialized in $|0_0, \cdots, 0_{N-1}\rangle$

if $N > 1$ *and* N *is power of* 2 **then**
  $|\psi_0\rangle = R_{X_0}(\frac{\pi}{2}) |0_0, \cdots, 0_{N-1}\rangle$
  $p_list = [2, 4, \cdots, N/2]$
  for $(i = 0,\ i < N/2)$*:* **do**
    if $i \ \% \ 2 \neq 0$ **then**
      for k **in** p_list **: do**
        if $i < k$ **then**
          $|\psi_0\rangle = R_{Z_{i+k}}(-\frac{\pi}{2}) \cdot U^{(2)}_{i,\ i+k}(\frac{\pi}{4}) |\psi_0\rangle$;
        end
      end
    end
  end
  for $(i = 0,\ i < N)$*:* **do**
    if $i \ \% \ 2 = 0$ **then**
      $|\psi_0\rangle = R_{Z_i}(-\frac{\pi}{2}) \cdot U^{(2)}_{i,\ i+1}(\frac{\pi}{4}) |\psi_0\rangle$;
    end
  end
end
Return $|\psi_0\rangle$;

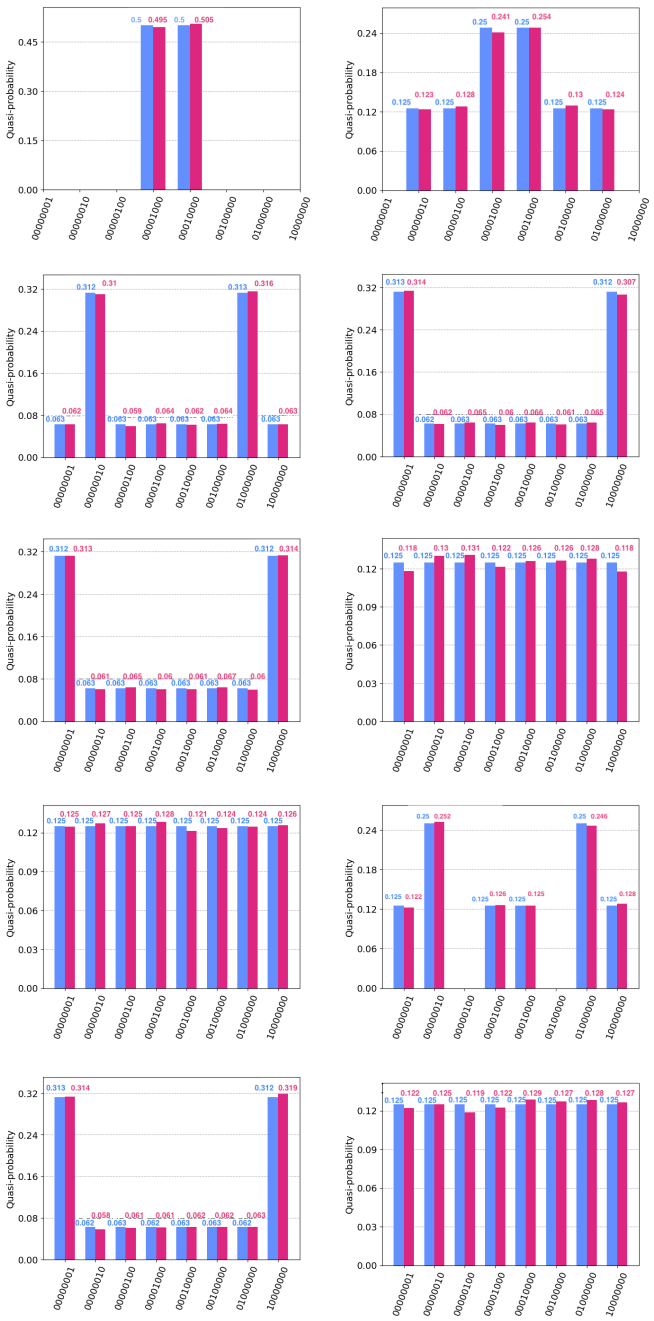


FIG. 4. Eight steps 8-cycle noiseless (blue) and noisy (purple) simulations with 10 000 shots, of QW via QCA for initial state $|\psi_0\rangle = \frac{1}{\sqrt{2}}|000\rangle(|3\rangle + |4\rangle)|000\rangle$. Last bottom right plot shows the 18th step.

B. QW benchmarks

In the following, we outline the quantities and metrics used to analyze the results of the QW and QW search simulations:

(1) *Qiskit state count distribution*: To study the QW (search) algorithm phenomenology, we use Qiskit's state count function. After executing the walk for a set number of time steps, the register is measured and the walker's position distribution over the lattice is extracted. This process is repeated across user-defined simulation shots, averaged, and visualized using histogram plots.

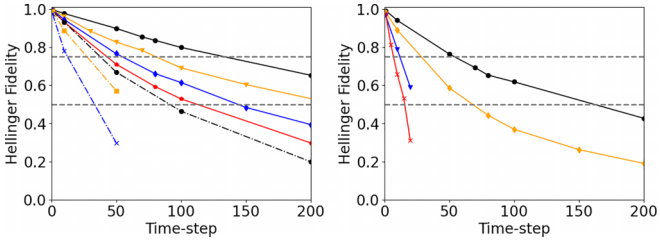


FIG. 5. Hellinger fidelity vs time step of the QW (left) and QW search (right) via QCA over 4-cycle (black dots), 6-cycle (orange triangles), 8-cycle (blue diamonds), 10-cycle (red pyramids), 12-cycle (black dash-dotted line), 16-cycle (orange dashed square line), and 4×4 torus (blue dashed cross line).

(2) *Hellinger fidelity*: To quantify the impact of noise on the simulation, we use the Hellinger fidelity [25], a standard metric for benchmarking quantum algorithms like QW and QW search [4,7,9]. It is defined as

$$\mathcal{F}_{\mathcal{H}}(P, Q) = [1 - H(P, Q)^2]^2,$$
$$H(P, Q) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{K}(\sqrt{p_i} - \sqrt{q_i})^2}, \tag{7}$$

where P and Q are probability distributions over K outcomes, and $H(P, Q)$ is the Hellinger distance [60]. Note that $\mathcal{F}_{\mathcal{H}}$ is equivalent to the classical fidelity [61]. In this work, P and Q, respectively, represent the noiseless (Qiskit Aer) and noisy (*Callisto*) emulator state probability distributions. Studying $\mathcal{F}_{\mathcal{H}}$ over time steps helps reveal how noise affects QW dynamics. Its values range from 0 (no correlation) to 1 (perfect match). While values above 0.5 are often seen as positively correlated, this can be misleading—e.g., two uniform, uncorrelated distributions may yield high fidelity [27]. Thus, the fidelity analysis should be complemented by visual checks of the state count distribution. In general, $\mathcal{F}_{\mathcal{H}} \geqslant 0.7$ with accurate distributions suggests a successful simulation.

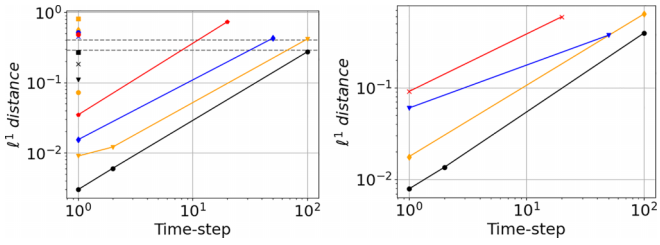


FIG. 6. Left: ℓ^1 distance vs time step for QW via QCA: 4-cycle (black curve), 8-cycle (orange curve), 16-cycle (blue curve), and 4×4 torus (red curve) on *Callisto*; via Shakeel QFT-based QW [33]: 4-cycle (black: crosses on *ibmqx2*, triangles on *ibmqx2*, squares on *ibmq_16_melbourne*), and 8-cycle (orange: crosses on *ibmqx2* computer, diamonds on *ibmqx2* processor, squares on *ibmq_16_melbourne*); and via Portugal [9]: 8-cycle (orange dots on *ourense*), 16-cycle (blue crosses on *vigo* processor, diamonds on *ibmqx2*), and 4×4 torus (red dots on *vigo*). Right: ℓ^1 distance vs time steps for QW search via QCA: 4-cycle (black), 8-cycle (orange), 16-cycle (blue), and 4×4 torus (red) on *Callisto*.

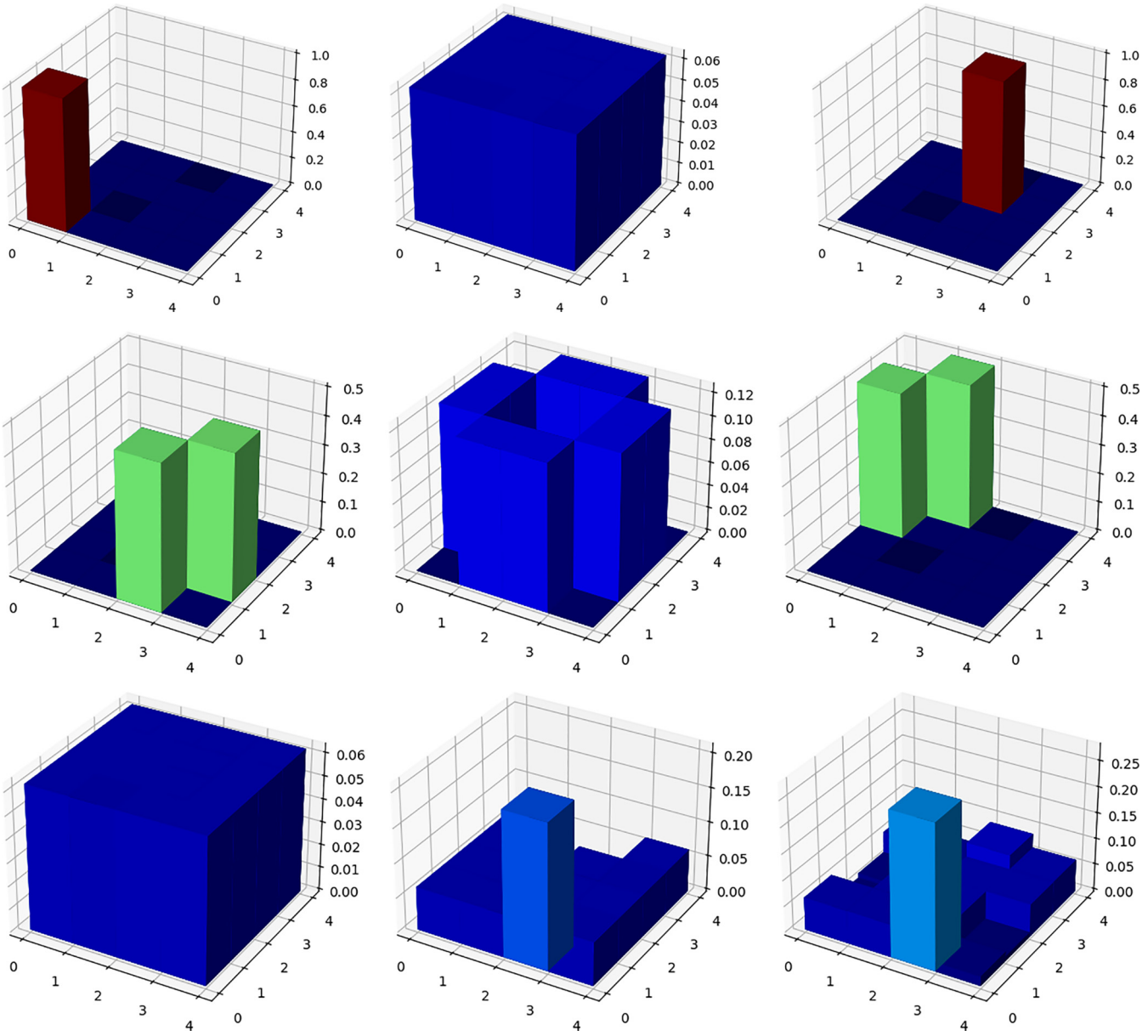


FIG. 7. Three steps of QW (top, middle) and QW search (bottom) over a 4 × 4 torus via QCA. In the QW search, the marked vertex is (3, 0).

(3) ℓ^1 *(Taxicab or Manhattan) distance*: Defined as $\ell^1(P, Q) = \sum_i |p_i - q_i|$ similarly to the Hellinger fidelity, this metric quantifies the impact of noise over the QW dynamics by measuring the distance between the noiseless and noisy probability distributions P and Q. Such metric is used in Refs. [27,33].

We now introduce some quantities and metrics specifically used to analyze the QW search [26,62]:

(1) *Hitting time*: It quantifies the number of QW time steps before the marked vertex probability reaches its maximum. QW search algorithms find the marked vertex in $\mathcal{O}(N)$ in one dimension, $\mathcal{O}(\sqrt{N \log N})$ steps in two dimensions, and $\mathcal{O}(\sqrt{N})$ steps for three or more dimensions [63,64].

(2) *Success probability*: It is the maximum probability peak of the marked vertex. The success probability of the QW search scales as $\mathcal{O}(\frac{1}{N})$ in one dimension and $\mathcal{O}(\frac{1}{\log_2 N})$ in two dimensions [62–64].

(3) *Degraded ratio*: It is defined as the ratio between the noiseless (or theoretical) and noisy success probabilities, $R = \frac{P_{s,\text{noisy}}}{P_{s,\text{ideal}}}$.

(4) *Selectivity*: It is defined as $S = \ln \frac{P_{v_m}}{\max\{P_{\tilde{v}}\}}$ [65,66], where P_{v_m} is the success probability and $\max\{P_{\tilde{v}}\}$ is the maximum of the nonmarked vertex probabilities. The higher the selectivity, the more robust to noise a circuit is.

C. Results

We simulate QW and QW search on the cycle and torus using the cQED QCA models from Sec. III. Simulations

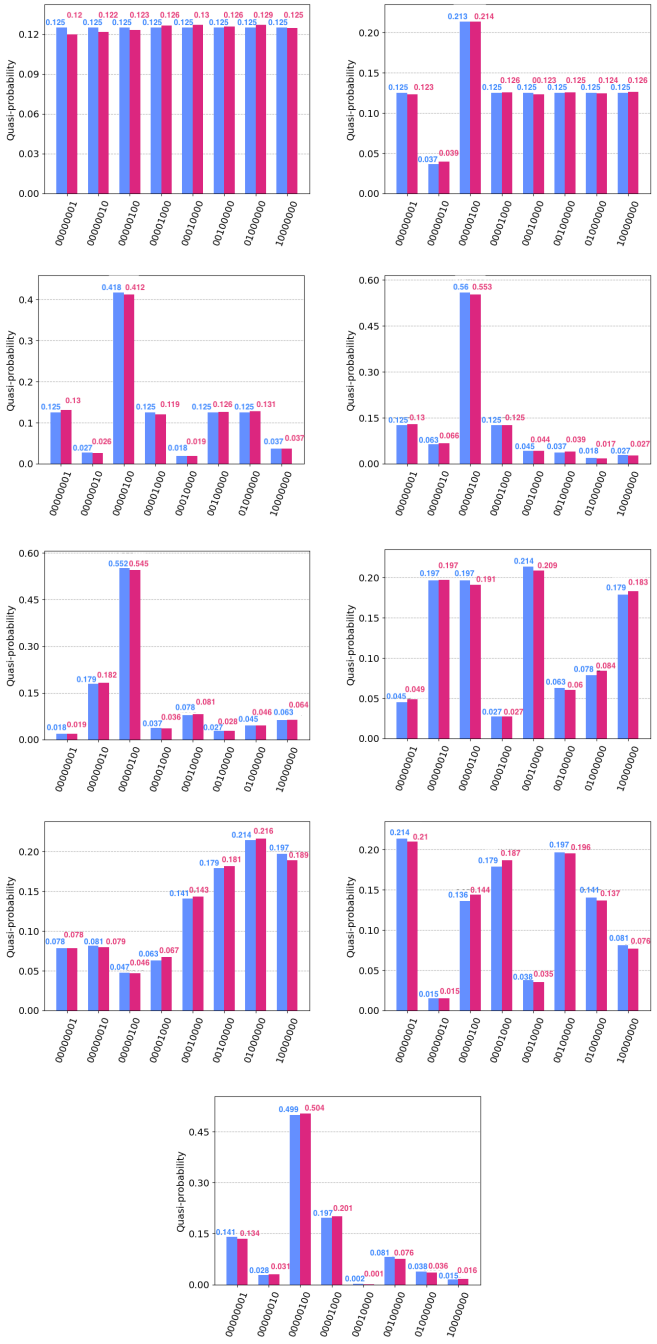


FIG. 8. Eight steps of QW search via QCA over an 8-cycle with marked vertex equal to 2. Noiseless Qiskit Aer (blue) and noisy *Callisto* (purple) simulations with 10 000 shots.

are run with Qiskit SDK [23] using both the noiseless Aer simulator and C12's noisy emulator, *Callisto* [24]. Each simulation runs an "experiment" (i.e., Algorithms 1 or 2 with fixed qubits and time steps) 10 000 times ("shots") to obtain a state distribution.

1. Quantum walk

N-*cycles.* We simulate QWs on cycles of sizes 4, 6, 8, 10, 12, and 16 using our cQED QCA implementation. To validate the dynamics, we analyze state count distributions over time; an example for the 8-cycle is shown in Fig. 4, displaying typical QW behavior, consistent with QW patterns [27,67]. Noiseless and noisy results align closely in early steps. To quantify noise effects, we track the Hellinger fidelity and the ℓ^1 distance over time steps (Figs. 5 and 6). The fidelity decreases with time due to cumulative gate errors and more rapidly with larger cycles, as the gate count scales with system size (Sec. III). For a very high number of time steps, we expect all the curves to reach an asymptote, corresponding to a completely random output due to noise. Our results compare well with prior work [4,7,27]. In Ref. [27], the authors report the following: (1) For a 16-cycle QW, after one step, fidelities of ≈0.7170 with *ibmqx2* quantum computer and ≈0.7358 with *vigo* [68]; in contrast, our simulation achieves ≈0.6380 after 50 steps (Fig. 5). (2) For an 8-cycle, *Ourense* reaches ≈0.90 after eight steps, though the corresponding state count distribution shows the noisy output to be wrong. In our case, the fidelity reaches the same value at step 18, with state count distributions showing good agreement between noisy and noiseless outputs. Similar considerations hold for the ℓ^1 distance analyses. The bottom plot of Fig. 6 reports ℓ^1 against the QW time step for different cycle sizes, circuit implementations (via QCA [9,33]), and quantum computers or emulators. The QCA implementation presented in this work scores a consistently lower ℓ^1 distance with respect to other implementations, i.e., around one order of magnitude lower for the same time step and one order of magnitude later (in time steps) for the same ℓ^1 value. The positive performance of our QCA implementation highlights its potential: Despite higher qubit requirements, it benefits from efficient linear gate scaling. Obviously, we stress that our results are based on emulator runs, unlike the hardware-based results used for comparison, and thus further testing on advanced emulators or real quantum hardware is necessary to fully assess the impact of noise over our QCA implementations.

4 × 4 *torus.* We simulate QW on a 4 × 4 torus. The state count histograms (Fig. 7) show behavior dependent on the initial condition, with good alignment between noisy and noiseless outputs in early steps. To quantify noise effects, we analyze the Hellinger fidelity and the ℓ^1 distance over time (Figs. 5 and 6). Compared to 1D cycles, 2D metrics degrade faster due to a doubled number of two-qubit gates per step, as shown in Eq. (3). Compared to Ref. [9], which reports a fidelity of ≈0.51 and $\ell^1 \approx 0.485$ after one step, our implementation achieves 0.92 and 0.035, respectively, reaching their values only after ≈10× more steps. Once again, we stress that these results are from an emulator, and further testing on real hardware or advanced emulators is needed to fully gauge noise impact.

2. Quantum walk search

4-, 8-, and 16-cycles. We simulate QW search on 4-, 8-, and 16-cycles. State count histograms (Fig. 8) show correct behavior, with higher probability at the marked vertex. To further assess the performance of our QCA implementation, we use the metrics in Sec. IV B (see Fig. 9):

(1) *Marked vertex probability*: It is consistently higher than the nonmarked vertex probabilities, modulo some os-

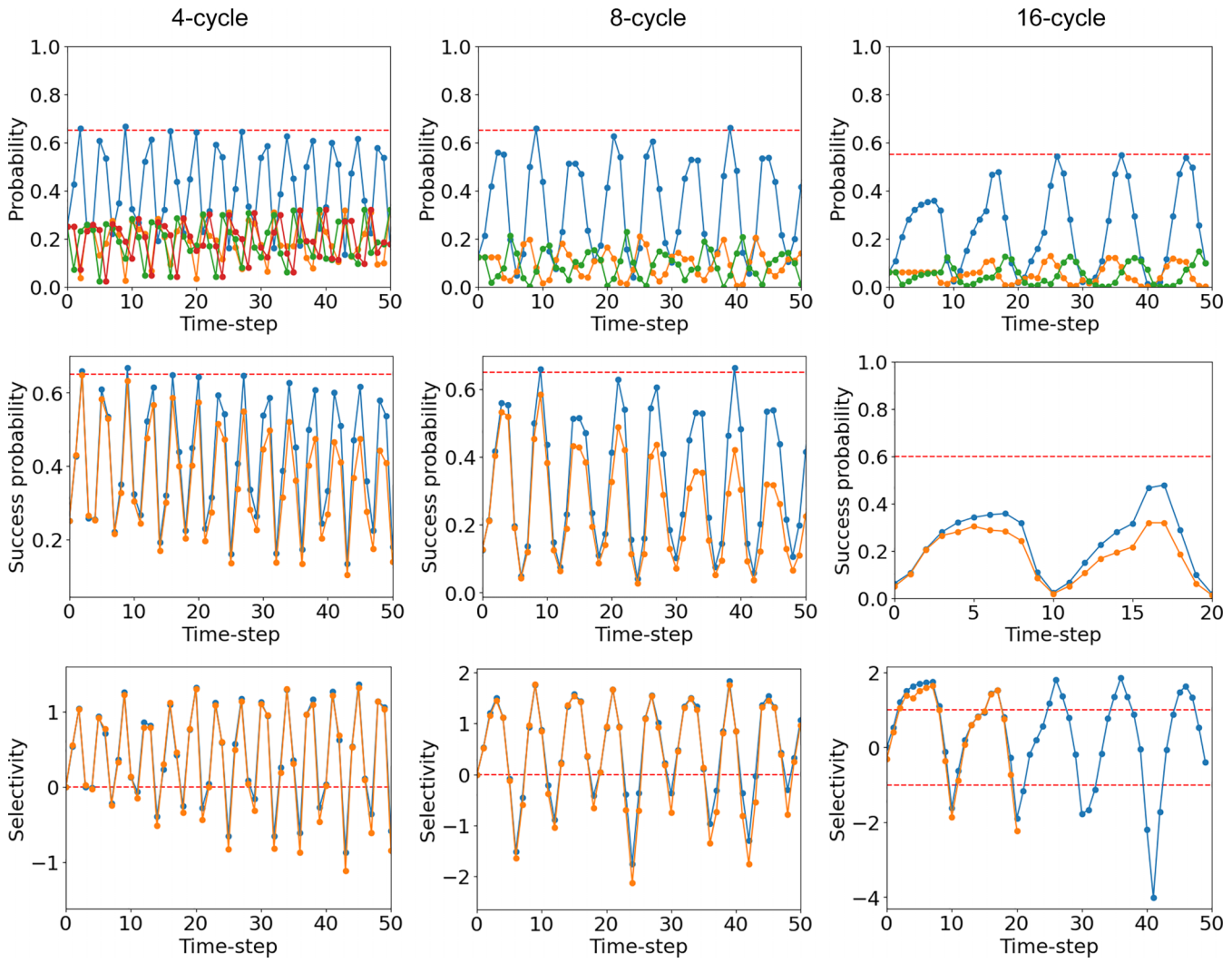


FIG. 9. Noiseless (Aer) and noisy (*Callisto*) simulations of QW search via QCA. First row: Noiseless vertex probabilities vs time step (for 8- and 16-cycles, some vertex curves are not shown for visual clarity). Second row: Noiseless (orange) and noisy (blue) success probabilities vs time step. Third row: Noiseless (orange) and noisy (blue) selectivity vs time step.

cillations due to the lattice finite-size effects (see first and second rows of Fig. 10). Selectivity plots (third row of Fig. 10) confirm that larger lattices yield more stable amplification. The similarity between the noisy (orange) and noiseless (blue) curves confirms that the oscillations stem from finite lattice size effects. Indeed, larger lattices result in a more stable marked vertex probability, as shown in the torus selectivity study (Fig. 11).

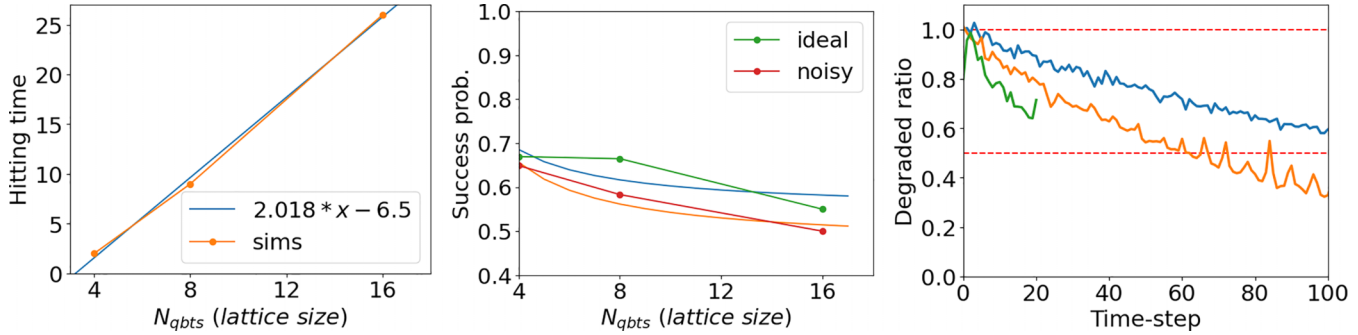


FIG. 10. Noiseless (Aer) and noisy (*Callisto*) simulations of QW search via QCA over 4-, 8-, and 16-cycles. Left: Hitting time vs number of qubits (lattice size) along with a linear fit. Middle: Noiseless and noisy success probabilities vs number of qubits (lattice size) along with an inverse linear fit. Right: Degraded ratio vs time step.

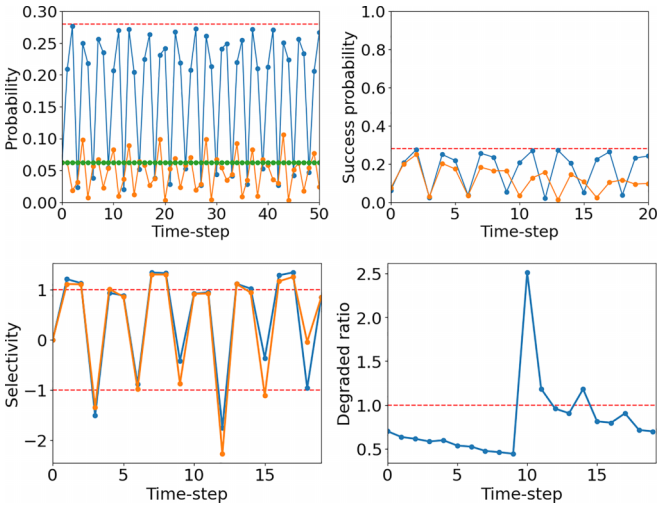


FIG. 11. Noiseless (Aer) and noisy (*Callisto*) simulations of QW search via QCA over 4×4 torus. Top left: Noiseless vertex probabilities vs time step (overlapping vertex curves are not shown for visual clarity). Top right: Noiseless (orange) and noisy (blue) success probabilities vs time step. Bottom left: Noiseless (orange) and noisy (blue) selectivity vs time step. Bottom right: Degraded ratio vs time step.

(2) *Hitting time*: It display a trend with acceptable agreement with a linear fit (first plot, last row of Fig. 10), consistent with theoretical predictions [62], although additional data are required to confirm its validity.

(3) *Success probability*: It decreases with the lattice size (second plot of last row of Fig. 10), following a trend that could be interpreted as linearly inverse with the lattice size (number of qubits). Our *scipy* fit supports this, but more data are needed to accurately characterize the trend.

(4) *Degraded ratio*: $R_s = \frac{P_{s,\text{noisy}}}{P_{s,\text{ideal}}}$ (Fig. 10, last plot of last row) shows that noise reduces probabilities uniformly across marked and nonmarked vertices, preserving search functionality.

(5) *Hellinger fidelity and ℓ^1 distance*: Both metrics (Figs. 5 and 6) degrade faster for larger lattices and for QW search (versus standard QW) due to more expensive initialization and use of lower-fidelity gates (see Table I and Algorithms 1 and 2). As time steps increase, the metrics approach values expected from random output.

Once again, we emphasize that our simulations are run on an emulator, not a real quantum computer, so further testing on advanced emulators or actual hardware is needed to accurately assess noise impact.

4×4 *torus.* We simulate QW search on a 4×4 torus. The state count histograms (Figs. 12 and 7) show correct QW search behavior, with minor discrepancies between noiseless and noisy counts in early steps. Benchmarking metrics and quantities (Fig. 11) reveal a success probability of $\approx$0.28—comprehensibly lower than the one-dimensional success probabilities—being hit after two time steps and noise-induced oscillations in the degraded ratio. The Hellinger fidelity and ℓ^1 distance (Figs. 5 and 6) show faster degradation in 2D compared to 1D, due to the increased number of two-qubit gates per time step on the torus [see Eq. (3)]. To our knowledge, this is the first evaluation of QW search on a torus using QCA on an NISQ processor. Further simulations on larger 2D or higher-dimensional lattices are needed to better assess the advantages and limitations of QW search via QCA.

V. CONCLUSION

In this work, we mapped QCA implementations of QW and QW search algorithms over N-cycles and $N \times N$ torus graphs to C12's Quantum Electronics cQED hardware. Using Qiskit SDK, we ran noiseless and noisy simulations, respectively, with Qiskit Aer simulator and C12s in-house emulator *Callisto*. In particular, we ran simulations over different cycle sizes and on a 4×4 torus. We analyzed the simulation outputs with Qiskit's state count histogram plots, which, serving as a proof of concept, show our implementation to correctly reproduce the expected QW and QW search behavior.

For both the QW and QW search, we assessed the noise impact on our simulation results using the Hellinger fidelity and ℓ^1 distance, benchmarking our implementation. Our study shows (see Figs. 5 and 6) that the two metrics gradually deteriorate with the number of algorithm steps (or, equivalently, circuit depth), due to gate errors stacking up at each time step, thus causing leakage to incorrect states, whether on or off the lattice sites. For the QW, a comparison of our QCA implementation over the emulator *Callisto* with similar studies [27,33] shows promising results. However, our simulations being run on an emulator, further investigations with a more advanced emulator or a real quantum processor are needed to better understand noise effects. For the QW search instead, we ran simulations with the emulator Callisto on 4-, 8-, and 16-cycle and 4×4 torus graphs, and analyzed the results using suitable metrics and quantities, such as hitting time, success probability, selectivity, and degraded ratio [69] (see Fig. 10). In 1D, we compared our results with theoretical predictions for hitting time and success probability. Selectivity showed oscillations in the marked vertex probability due to finite-size effects in both 1D and 2D. Noise analysis, using the degraded ratio, Hellinger fidelity, and ℓ^1 distance, shows stability within the first hundred time steps, with larger lattices experiencing more noise. Further simulations on larger and higher-dimensional lattices are needed for a deeper understanding of QW search via QCA.

To the authors' knowledge, this work provides the first investigation of QW and QW search implementations via QCA on an NISQ processor emulator. Our analysis benchmarks C12's quantum computer via its emulator, *Callisto*, and the QCA algorithmic framework, demonstrating their effectiveness in early NISQ implementations of QW and QW search algorithms. The noninteracting one-particle sector QCA model, equivalent to the QW, can be easily implemented on C12's hardware, requiring for its time-evolution operator the XY gate present in C12's basis gate set. More broadly, our study shows the QCA fruitfulness as an algorithmic framework for NISQ implementations, due to its *close-to-physics* spatiotemporal definition. Indeed, despite an unfavorable linear scaling of the lattice size with the number of qubits, the cycle and torus QW and QW search implementations via QCA require a time-evolution operator

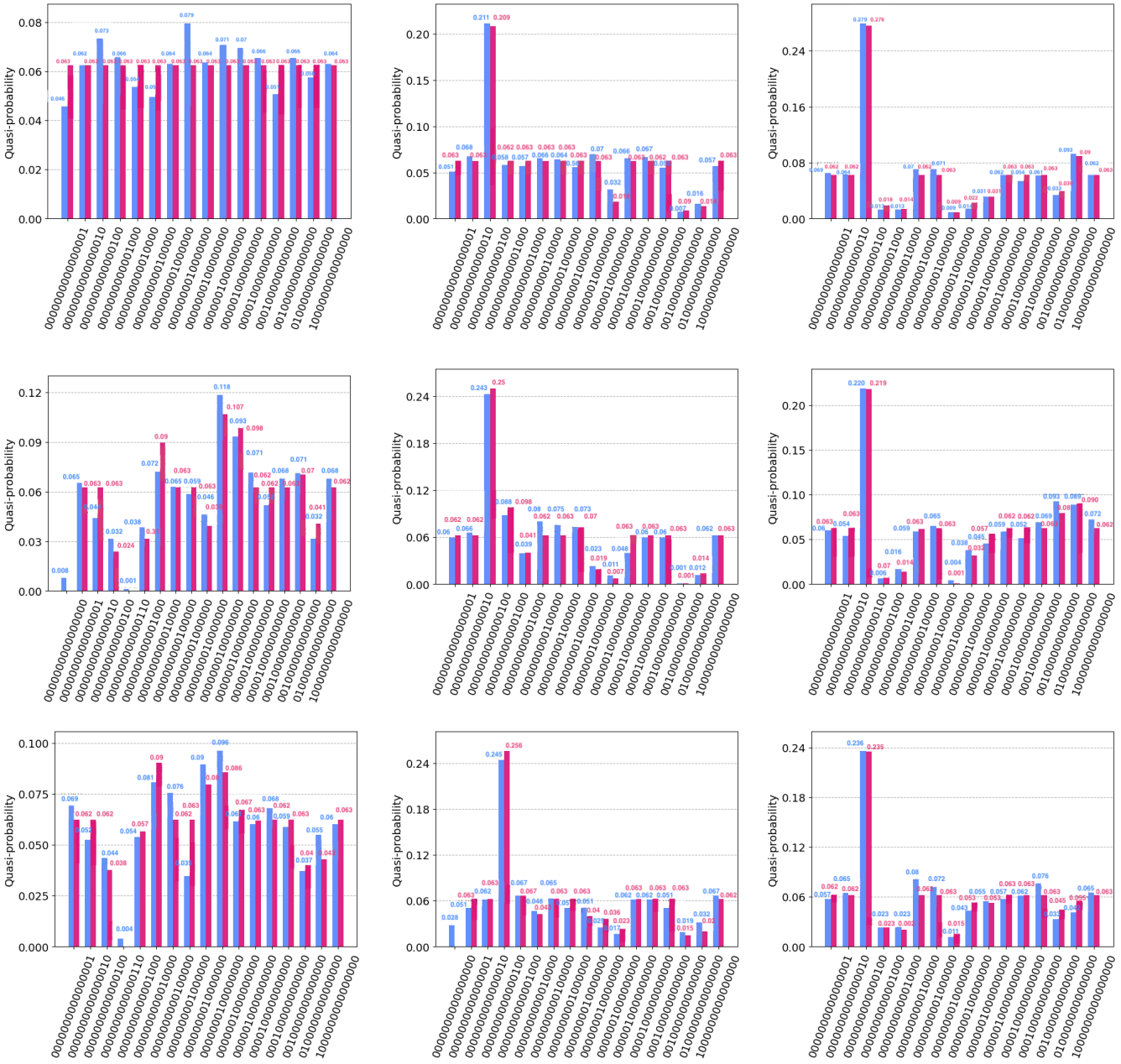


FIG. 12. Eight steps of QW search via QCA over a 4 × 4 torus with marked vertex (3, 0). Noiseless Qiskit Aer (blue) and noisy (blue bars) and noiseless (purple bars) simulations with 10 000 shots.

composed of common two-qubit gates only. This translates into a favorable linear scaling of the two-qubit gate count and circuit's depth with the algorithm steps, scarce of big overhead due to complex gate decomposition proper of other circuit implementations.

While the emulator we employed accounts for the main sources of error in the processor, primarily those arising from incoherent processes, it could be further refined by incorporating coherent errors, such as quantum crosstalk, which have already been investigated in these platforms [43]. Although cQED spin-qubit platforms are not yet readily available for experimental use, following the first demonstration of a two-qubit gate in 2024 [40], their availability for experiments can reasonably be expected within the next 2–3 years.

In conclusion, our results are promising, yet further investigation is required for testing our implementations: (1) over different lattices or higher-dimensional structures [3,28]; (2) over some use cases, like asset-price evolution modeling in finance [70] or for network analyses [71]; and (3) for state preparation, e.g., in Ref. [30]. Additionally, exploring more complex simulations, such as QFT theories or topological phases of matter [15,72], with initial tests on the Schwinger model (1 + 1) QED [14], would be valuable.

ACKNOWLEDGMENTS

This work was supported by the PEPR integrated project EPiQ ANR-22-PETQ-0007; by the ANR JCJC DisQC ANR-

22-CE47-0002-01 funded by the French National Research Agency; and the French government under the France 2030 investment plan, as part of the Initiative d'Excellence d'Aix-Marseille Université—A*MIDEX AMX-21-RID-011. We thank Viktor Radovic for his great help with simulations and insightful discussions and Mathieu Roget for reviewing the manuscript.

DATA AVAILABILITY

The data that support the findings of this article are openly available [73]; embargo periods may apply.

[1] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).
[2] Y. Aharonov, L. Davidovich, and N. Zagury, Quantum random walks, Phys. Rev. A **48**, 1687 (1993).
[3] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, Exponential algorithmic speedup by a quantum walk, in *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, *STOC'03* (Association for Computing Machinery, New York, 2003), pp. 59–68.
[4] K. Georgopoulos, C. Emary, and P. Zuliani, Comparison of quantum-walk implementations on noisy intermediate-scale quantum computers, Phys. Rev. A **103**, 022408 (2021).
[5] X. Qiang, S. Ma, and H. Song, Quantum walk computing: Theory, implementation, and application, Intell Comput. **3**, 0097 (2024).
[6] N. Shenvi, J. Kempe, and K. B. Whaley, Quantum random-walk search algorithm, Phys. Rev. A **67**, 052307 (2003).
[7] V. Wadhia, N. Chancellor, and V. Kendon, Cycle discrete-time quantum walks on a noisy quantum computer, Eur. Phys. J. D **78**, 29 (2024).
[8] L. Razzoli, G. Cenedese, M. Bondani, and G. Benenti, Efficient implementation of discrete-time quantum walks on quantum computers, Entropy **26**, 313 (2024).
[9] R. Portugal, *Quantum Walks and Search Algorithms* (Springer, Cham, 2013).
[10] J. Watrous, On one-dimensional quantum cellular automata, in *Proceedings of IEEE 36th Annual Foundations of Computer Science* (IEEE, Piscataway, NJ, 1995), pp. 528–537.
[11] P. Arrighi, An overview of quantum cellular automata, Nat. Comput. **18**, 885 (2019).
[12] L. E. Hillberry, M. T. Jones, D. L. Vargas, P. Rall, N. Y. Halpern, N. Bao, S. Notarnicola, S. Montangero, and L. D. Carr, Entangled quantum cellular automata, physical complexity, and Goldilocks rules, Quantum Sci. Technol. **6**, 045017 (2021).
[13] L. Piroli and J. I. Cirac, Quantum cellular automata, tensor networks, and area laws, Phys. Rev. Lett. **125**, 190402 (2020).
[14] K. Sellapillay, P. Arrighi, and G. Di Molfetta, A discrete relativistic spacetime formalism for 1 + 1-QED with continuum limits, Sci. Rep. **12**, 2198 (2022).
[15] K. Sellapillay, A. D. Verga, and G. Di Molfetta, Entanglement dynamics and ergodicity breaking in a quantum cellular automaton, Phys. Rev. B **106**, 104309 (2022).
[16] T. L. M. Guedes, D. Winter, and M. Müller, Quantum cellular automata for quantum error correction and density classification Phys. Rev. Lett. **133**, 150601 (2024).
[17] P. C. S. Costa, R. Portugal, and F. Melo, Quantum walks via quantum cellular automata, Quantum Inf. Proc. **17**, 226 (2018).
[18] N. Schuch and J. Siewert, Natural two-qubit gate for quantum computation using the XY interaction, Phys. Rev. A **67**, 032301 (2003).
[19] J. Siewert, R. Fazio, G. M. Palma, and E. Sciacca, Aspects of qubit dynamics in the presence of leakage, J. Low Temp. Phys. **118**, 795 (2000).
[20] D. Mozyrsky, V. Privman, and M. L. Glasser, Indirect interaction of solid-state qubits via two-dimensional electron gas, Phys. Rev. Lett. **86**, 5112 (2001).
[21] A. Imamog¯lu, D. D. Awschalom, G. Burkard, D. P. DiVincenzo, D. Loss, M. Sherwin, and A. Small, Quantum information processing using quantum dot spins and cavity QED, Phys. Rev. Lett. **83**, 4204 (1999).
[22] C12 Quantum Electronics, https://www.c12qe.com.
[23] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, and J. M. Gambetta, Quantum computing with Qiskit, arXiv:2405.08810.
[24] C12 Quantum electronics, https://www.c12qe.com/callisto.
[25] Z.-X. Jin and S.-M. Fei, Quantifying quantum coherence and nonclassical correlation based on Hellinger distance, Phys. Rev. A **97**, 062342 (2018).
[26] K. Zhang, P. Rao, K. Yu, H. Lim, and V. Korepin, Implementation of efficient quantum search algorithms on NISQ computers, Quantum Inf. Proc. **20**, 233 (2021).
[27] F. Acasiete, F. P. Agostini, J. K. Moqadam, and R. Portugal, Implementation of quantum walks on IBM quantum computers, Quantum Inf. Proc. **19**, 426 (2020).
[28] A. M. Childs, M. Coudron, and A. S. Gilani, Quantum algorithms and the power of forgetting, in *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, *Leibniz International Proceedings in Informatics (LIPIcs)*, edited by Y. Tauman Kalai (Schloss Dagstuhl—Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2023), Vol. 251, pp. 37:1–37:22.
[29] S. Marsh and J. B. Wang, A quantum walk-assisted approximate algorithm for bounded NP optimisation problems, Quantum Inf. Proc. **18**, 61 (2019).
[30] Y.-J. Chang, W.-T. Wang, H.-Y. Chen, S.-W. Liao, and C.-R. Chang, Preparing random state for quantum financing with quantum walks, arXiv:2302.12500.
[31] D. M. Abrams, N. Didier, B. R. Johnson, M. P. da Silva, and C. A. Ryan, Implementation of XY entangling gates with a single calibrated pulse, Nat. Electron. **3**, 744 (2020).
[32] C. Huang, T. Wang, F. Wu, D. Ding, Q. Ye, L. Kong, F. Zhang, X. Ni, Z. Song, Y. Shi, H.-H. Zhao, C. Deng, and J. Chen, Quantum instruction set design for performance, Phys. Rev. Lett. **130**, 070601 (2023).
[33] A. Shakeel, Efficient and scalable quantum walk algorithms via the quantum Fourier transform, Quantum Inf. Proc. **19**, 323 (2020).
[34] C. Gidney, https://algassert.com/circuits/2015/06/12/Constructing-Large-Increment-Gates.html.

[35] J. Nie, W. Zi, and X. Sun, Quantum circuit for multi-qubit Toffoli gate with optimal resource, arXiv:2402.05053.

[36] T. Khattar and C. Gidney, Rise of conditionally clean ancillae for optimizing quantum circuits Quantum **9**, 1752 (2024).

[37] A. Cottet, T. Kontos, and B. Douçot, Electron-photon coupling in mesoscopic quantum electrodynamics, Phys. Rev. B **91**, 205417 (2015).

[38] A. Blais, A. L. Grimsmo, S. M. Girvin, and A. Wallraff, Circuit quantum electrodynamics, Rev. Mod. Phys. **93**, 025005 (2021).

[39] S. Haroche and D. Kleppner, Cavity quantum electrodynamics, Phys. Today **42**(1), 24 (1989).

[40] J. Dijkema, X. Xue, P. Harvey-Collard, M. Rimbach-Russ, S. Snoo, G. Zheng, A. Sammak, G. Scappucci, and L. Vandersypen, Cavity-mediated iSWAP oscillations between distant spins, Nat. Phys. **21**, 168 (2025).

[41] G. Burkard, T. D. Ladd, A. Pan, J. M. Nichol, and J. R. Petta, Semiconductor spin qubits, Rev. Mod. Phys. **95**, 025003 (2023).

[42] J. Yoneda, K. Takeda, T. Otsuka, T. Nakajima, M. R. Delbecq, G. Allison, T. Honda, T. Kodera, S. Oda, Y. Hoshi, *et al.*, A quantum-dot spin qubit with coherence limited by charge noise and fidelity higher than 99.9%, Nat. Nanotechnol. **13**, 102 (2018).

[43] A. Mammola, M. M. Desjardins, and Q. Schaeverbeke, Optimal connectivity from idle-qubit residual-coupling crosstalk in a cavity-mediated entangling gate, Phys. Rev. A **111**, 042621 (2025).

[44] B. Neukelmance, B. Hue, Q. Schaeverbeke, L. Jarjat, A. Théry, J. Craquelin, W. Legrand, T. Cubaynes, G. Abulizi, J. Becdelievre, *et al.*, Microsecond-lived quantum states in a carbon-based circuit driven by cavity photons, Nat. Commun. **16**, 5636 (2025).

[45] D. A. Steck, Quantum and atom optics, http://steck.us/teaching (2006).

[46] M. Göppl, A. Fragner, M. Baur, R. Bianchetti, S. Filipp, J. M. Fink, P. J. Leek, G. Puebla, L. Steffen, and A. Wallraff, Coplanar waveguide resonators for circuit quantum electrodynamics, J. Appl. Phys. **104**, 113904 (2008).

[47] V. Sazonova, Y. Yaish, H. Ustünel, D. Roundy, T. A. Arias, and P. L. McEuen, A tunable carbon nanotube electromechanical oscillator, Nature (London) **431**, 284 (2004).

[48] R. Leturcq, C. Stampfer, K. Inderbitzin, L. Durrer, C. Hierold, E. Mariani, M. G. Schultz, F. von Oppen, and K. Ensslin, Franck-Condon blockade in suspended carbon nanotube quantum dots, Nat. Phys. **5**, 327 (2009).

[49] A. Benyamini, A. Hamo, S. Kusminskiy, F. von Oppen, and S. Ilani, Real-space tailoring of the electron-phonon coupling in ultraclean nanotube mechanical resonators, Nat. Phys. **10**, 151 (2014).

[50] J. Moser, A. Eichler, J. Güttinger, M. I. Dykman, and A. Bachtold, Nanotube mechanical resonators with quality factors of up to 5 million, Nat. Nanotechnol. **9**, 1007 (2014).

[51] W. Yang, C. Urgell, S. L. De Bonis, M. Margańska, M. Grifoni, and A. Bachtold, Fabry-Pérot oscillations in correlated carbon nanotubes, Phys. Rev. Lett. **125**, 187701 (2020).

[52] A. V. Kuhlmann, J. Houel, A. Ludwig, L. Greuter, D. Reuter, A. D. Wieck, M. Poggio, and R. J. Warburton, Charge noise and spin noise in a semiconductor quantum device, Nat. Phys. **9**, 570 (2013).

[53] C. Müller, J. H. Cole, and J. Lisenfeld, Towards understanding two-level-systems in amorphous solids: insights from quantum circuits, Rep. Prog. Phys. **82**, 124501 (2019).

[54] E. Paladino, L. Faoro, G. Falci, and R. Fazio, Decoherence and $1/f$ noise in Josephson qubits, Phys. Rev. Lett. **88**, 228304 (2002).

[55] E. Paladino, Y. M. Galperin, G. Falci, and B. L. Altshuler, $1/f$ noise: Implications for solid-state quantum information, Rev. Mod. Phys. **86**, 361 (2014).

[56] M. Sarovar, T. Proctor, K. Rudinger, K. Young, E. Nielsen, and R. Blume-Kohout, Detecting crosstalk errors in quantum information processors, Quantum **4**, 321 (2020).

[57] H.-P. Breuer and F. Petruccione, *The Theory of Open Quantum Systems* (Oxford University Press, Oxford, 2007).

[58] G. Lindblad, On the generators of quantum dynamical semigroups, Commun. Math. Phys. **48**, 119 (1976).

[59] V. Gorini, A. Kossakowski, and E. C. G. Sudarshan, Completely positive dynamical semigroups of *N*-level systems, J. Math. Phys. **17**, 821 (1976).

[60] V. H. E. Hdlinger, Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen, J. Reine Angew. Math. **1909**, 210 (1909).

[61] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th anniversary ed. (Cambridge University Press, Cambridge, 2010).

[62] N. Lovett, M. Everitt, M. Trevers, D. Mosby, D. Stockton, and V. Kendon, Spatial search using the discrete time quantum walk, Nat. Comput. **11**, 23 (2010).

[63] A. Tulsi, Faster quantum-walk algorithm for the two-dimensional spatial search, Phys. Rev. A **78**, 012310 (2008).

[64] A. Ambainis, J. Kempe, and A. Rivosh, Coins make quantum walks faster, in *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, *SODA'05* (Society for Industrial and Applied Mathematics, USA, 2005), pp. 1099–1108.

[65] S. S. Tannu and M. Qureshi, Ensemble of diverse mappings: Improving reliability of quantum computers by orchestrating dissimilar mistakes, in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, *MICRO'52* (Association for Computing Machinery, New York, 2019), pp. 253–265.

[66] Y. Wang and P. S. Krstic, Prospect of using Grover's search in the noisy-intermediate-scale quantum-computer era, Phys. Rev. A **102**, 042609 (2020).

[67] T. Kitagawa, M. S. Rudner, E. Berg, and E. Demler, Exploring topological phases with quantum walks, Phys. Rev. A **82**, 033429 (2010).

[68] IBM Quantum, https://quantum.cloud.ibm.com.

[69] K. Zhang, K. Yu, and V. Korepin, Quantum search on noisy intermediate-scale quantum devices, Europhys. Lett. **140**, 18002 (2022).

[70] S. De Backer, L. E. C. Rocha, J. Ryckebusch, and K. Schoors, On the potential of quantum walks for modeling financial return distributions, Physica A **657**, 130215 (2025).

[71] P. Chawla, R. Mangal, and C. M. Chandrashekar, Discrete-time quantum walk algorithm for ranking nodes on a network, Quantum Inf. Proc. **19**, 158 (2019).

[72] T. Farrelly, A review of quantum cellular automata, Quantum **4**, 368 (2020).

[73] https://github.com/c12qe/QW_and_QW_search.git.