

Comparison of quantum-walk implementations on noisy intermediate-scale quantum computers

Konstantinos Georgopoulos^{1,*}, Clive Emary², and Paolo Zuliani^{1,†}

¹*School of Computing, Newcastle University, Newcastle-upon-Tyne, NE4 5TG, United Kingdom*

²*Joint Quantum Centre Durham-Newcastle, School of Mathematics, Statistics and Physics, Newcastle University, Newcastle-upon-Tyne, NE1 7RU, United Kingdom*



(Received 15 October 2020; accepted 22 January 2021; published 8 February 2021)

This paper explores two circuit approaches for quantum walks: The first consists of generalized controlled inversions, whereas the second one effectively replaces them with rotation operations around the basis states. We show the theoretical foundation of the rotational implementation. The rotational approach nullifies the large amount of ancilla qubits required to carry out the computation when using the inverter implementation. Our results concentrate around the comparison of the two architectures in terms of structure, benefits, and detriments, as well as the computational resources needed for each approach. We show that the inverter approach requires exponentially fewer gates than the rotations but almost half the number of qubits in the system. Finally, we execute a number of experiments using an IBM quantum computer. The experiments show the effects of noise in our circuits. Small two-qubit quantum walks evolve closer to our expectations, whereas for a larger number of steps or state space the evolution is severely affected by noise.

DOI: [10.1103/PhysRevA.103.022408](https://doi.org/10.1103/PhysRevA.103.022408)

I. INTRODUCTION

This paper considers the quantum mechanical analog of a random walk on the line, or *quantum walk* [1]. Here, the evolution of the walker is guided by a balanced quantum coin in superposition. If we imagine a quantum particle that moves freely between adjacent discrete points on a line (see Fig. 1), then at each time step, the balanced coin is flipped and the quantum state undergoes a unitary transformation, otherwise called shift. Then the particle progresses according to the state of the quantum coin, thus evolving the walk.

By evolving the state of the system in a superposition, the walker or particle can seemingly follow all possible paths, propagating quadratically further as a function of the coin flips than in the classical case [2]. Significant here is also the effect of quantum interference in quantum walks, where two separate paths leading to the same point may be out of phase and cancel one another.

Quantum walks have the potential to speed up classical algorithms that are based on random walks [2–4]. There have been many systematic studies on this subject area and many of them can lead to further in-depth analysis of more advanced quantum algorithms, such as quantum Metropolis, quantum Markov chains, or quantum Monte Carlo methods [2,5–8]. An early work from Aharonov *et al.* [9] proves that, in the context of quantum walks on graphs, the walker’s propagation in the quantum case is quadratically faster than the classical random walk. The efficiency of quantum walks has been exploited in various cases in order to construct quantum algorithms [10,11] and speed up classical methods [12,13], sometimes

even exponentially [14,15]. Quantum walks have also been realized in a number of physical systems, including photons [16–21], cold atoms [22,23], and trapped ions [24].

The results of our work are concentrated around the comparison of two architectural approaches to quantum walks: (i) the generalized inverter approach [25] and (ii) the rotational approach [26,27], with no quantum-walk implementation of the latter, to the best of our knowledge. Through an analysis of the computational resources necessary we find that each architecture shows opposite advantages and disadvantages. The generalized inverter approach shows smaller execution times and requires exponentially less gates whereas the rotational approach lowers the number of qubits necessary for the implementation. This result is important within the noisy intermediate-scale quantum (NISQ) era [28], with the inability of existing quantum computers to reliably run even average-scale computations.

Finally, we examine the effects of noise for each implementation on a real quantum machine. Our experiments show that the results follow the theoretical expectations for small two-qubit systems, but quickly deviate for quantum walks with a three-qubit state space or larger. For the implementation we make use of IBMQ’s Qiskit development kit [29,30] in order to simulate and execute the quantum circuits. Currently, there is a lot of interest in running quantum applications on NISQ systems [31–35]. On the other hand, there are very few studies of quantum walks on hardware, with an early notable work by [36] representing the first implementation of a discrete coined quantum walk implemented on a quantum-information processor. Additionally, in a recent work [37] the authors examine the implementation of discrete-time quantum walks on cycles, two-dimensional lattices, and complete graphs, which are then executed on a different IBM quantum computer.

*k.georgopoulos2@newcastle.ac.uk

†paolo.zuliani@newcastle.ac.uk

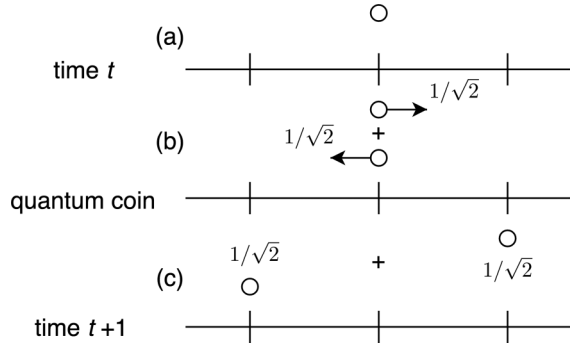


FIG. 1. Dynamics of the balanced quantum walk. (a) The walk begins at time t . (b) The flip of the quantum coin, where the particle is in equal superposition to go left or right. (c) The particle moves to generate the superposed state at time $t + 1$.

The paper is structured as follows. First, we give a theoretical overview of the quantum walk in Sec. II. Moving on, we discuss in Sec. III how a quantum walk can be implemented using generalized inverters and rotations around the basis states. Section IV offers an analysis of the implementation characteristics of each approach and a rigorous comparison of the two, before we outline the results from experimenting with both implementations in Sec. V. Finally, we discuss and conclude our research in Sec. VI.

II. QUANTUM WALKS

Throughout this paper we are occupied with a particle's one-dimensional discrete-time quantum walk on a finite cycle with N states and an arbitrary number of steps [38]. This quantum walk can be described as the repeated application of a unitary evolution operator U . This operator acts on a Hilbert space $\mathcal{H}^S \otimes \mathcal{H}^C$, where \mathcal{H}^C is the Hilbert space associated with a quantum coin and \mathcal{H}^S with the state space of the walk. In order to describe the quantum walk we define the operator U as

$$U = S(I \otimes C), \quad (1)$$

where S is the shift operator describing the walker's propagation and C is the quantum coin operator. We take the quantum coin to be a Hadamard operator with well-known matrix representation

$$C = H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

The walker, after each flip of the coin, can either *increase* or *decrease* its position by a step of 1. This is defined by the shift operator S and can be described via increment and decrement functions, as demonstrated by [25]. The mathematical description of the shift operator is

$$S = S^- \otimes |0\rangle\langle 0| + S^+ \otimes |1\rangle\langle 1|, \quad (2)$$

where $S^+ |x\rangle \rightarrow |x+1\rangle$ moves the walker one step to the right, increasing its position, and $S^- |x\rangle \rightarrow |x-1\rangle$ to the left, decreasing its position.

A very important aspect of a discrete-time quantum walk's evolution is what we call the *modularity property* [39]. This

property expresses the relationship between the parity of the number of coin flips, the initial position, and the resulting states of the walker. For example, if the particle starts on an even position (including $|0\rangle$) then, after an odd or even number of steps the position of the particle will be a superposition of $N/2$ odd or even states.

Another characteristic of this quantum walk is its *asymmetry*. After the evolution of the particle position, the probability of each state to be measured may not be the same. The reason for this phenomenon is quantum interference, which can be either constructive or destructive. This can affect the quantum walk for more than one iteration of the shift operator S . Precisely, the leftward path (S^-) interferes more destructively whereas the rightward path undergoes more constructive interference. In other words, the asymmetry is the result of the Hadamard coin introducing bias in the path selection [40].

III. INCREMENT AND DECREMENT FUNCTIONS

One way to implement the functions S^\pm , as expressed in Eq. (2), is to use generalized inverter gates, demonstrated by [25]. This implementation is discussed in Sec. III A below. In Sec. III B, we present the second approach that uses rotations around the basis states [26,27].

A. Using generalized inverter gates

We refer to generalized controlled operations as those controlled by more than one qubit. In this case, the operation in question will occur if and only if all the control qubits are in state $|1\rangle$. The most common example and one that is used extensively here is the three-qubit Toffoli gate, in which the target qubit will be inverted only if both the control qubits are in state $|1\rangle$.

The generalized controlled-NOT (CNOT) gates can be used to construct quantum circuits that implement the increment and decrement functions. As shown by [25], the increment and decrement functions can be realized with a single quantum circuit, but with opposite control logic. Figure 2(a) shows the higher level circuit for one iteration of the quantum walk for a state space of arbitrary size. The general implementation of the increment and decrement circuits is shown in Fig. 2(b).

The realization of the individual circuits in Fig. 2(b) for a small quantum walk on a $N = 8$ cycle using elementary and Toffoli gates is shown in Figs. 3(a) and 3(b).

Both the increment and decrement circuits of these two figures look more complicated than their respective schematic in Fig. 2(b). The reason is the lack of direct implementation of any generalized inverters other than the Toffoli gate in IBMQ's Qiskit. Any inverter gate with more than two control qubits requires intermediate computations stored in ancilla qubits. This leads to a significant increase of the workspace (i.e., the number of qubits needed for the computation) that grows with the size of the state space. Precisely, a generalized CNOT gate with n_c control qubits requires additional $n_c - 1$ ancilla qubits for the implementation (refer to Appendix A). For example, considering IBMQ's 15-qubit Melbourne machine, we can implement a quantum walk on a cycle with at most $2^8 = 256$ states.

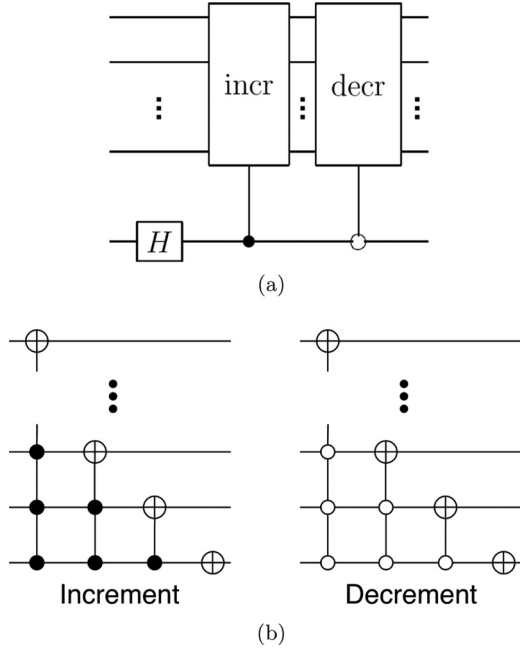


FIG. 2. (a) Implementation of one step for the quantum walk of a particle. (b) Quantum circuits for increment and decrement operations. A filled control circle means that the control qubits have to be in state $|1\rangle$ in order for the operation to occur. An empty control circle means they have to be in state $|0\rangle$.

B. Using rotations

In this section we present a solution that uses *rotations* around the basis states [26,27] to implement the quantum walk without an ancilla register, thus lowering the vast increase in computational resources resulting from adding qubits to the system. Referring back to the previous section's example, we can use the 15-qubit Melbourne machine to run a quantum

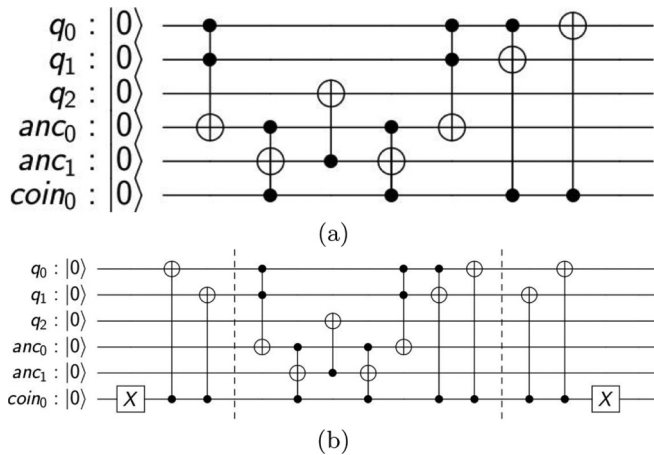


FIG. 3. (a) Increment circuit. The three-qubit q_n register is the state space of the quantum walk, the coin register $coin_0$ represents the Hadamard coin. The ancilla qubits used for the computation are anc_0 and anc_1 . (b) Decrement circuit. Important here is the need to invert all the control qubits (including the coin) at the start of the computation and uncompute them at the end.

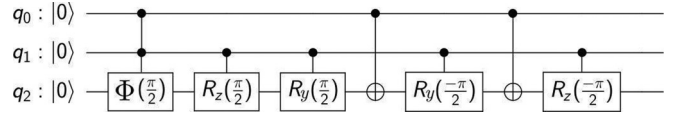


FIG. 4. Quantum circuit implementing a Toffoli gate (X_{cc}) using conditioned rotations.

walk with rotations on a cycle with $2^{14} = 16384$ positions, a significant rise compared to the generalized CNOT approach.

Another benefit of the rotational approach holds when simulating quantum walks on classical machines. As the size of the state space increases exponentially with the number of qubits, classical computers very quickly start struggling to cope with the size of the workspace. The rotational implementation offers a way around this problem.

Before we show the quantum circuit for the rotational approach, we need to start by expressing two very important lemmas, first introduced and proven by Barenco *et al.* [26].

Lemma 1. [26] *For any unitary operator W there exist operators Φ, A, B , and C such that $ABC = I$ and $\Phi AXBXC = W$, where Φ is a phase operator of the form $\Phi = e^{i\delta} \times I$ with $\delta \in \mathbb{R}$, X is the Pauli- X , and I the identity matrix.*

What we learn from Lemma 1 is that we can express any unitary operator and, for our case, a NOT gate, as a sequence of operators $\Phi AXBXC$. The existence of the operator Φ compensates for the fact that the NOT gate is not an $SU(2)$. Thus, we can narrow our efforts down to finding the appropriate Φ, A, B, C operators that suit our specific needs.

We start by presenting a quantum circuit that implements a Toffoli gate, denoted X_{cc} , via a number of controlled rotations, as shown in Fig. 4. The specific rotation gates needed can be defined through the unitary matrix $R_y(\theta)$ defined as

$$R_y(\theta) = \begin{pmatrix} \cos \theta/2 & -\sin \theta/2 \\ \sin \theta/2 & \cos \theta/2 \end{pmatrix}, \quad (3)$$

where, in our case, we require $\theta = \pi/2$. The rotation operator can be implemented using IBM's U_3 gate, as shown in Appendix B. The next operation is expressed by the unitary operator $R_z(\phi)$, given as

$$R_z(\phi) = \begin{pmatrix} e^{i\phi/2} & 0 \\ 0 & e^{-i\phi/2} \end{pmatrix}. \quad (4)$$

Similarly to the rotation operators, we assign $\phi = \pi/2$.

Finally, since the inverter gate is not a special unitary (i.e., it does not have determinant 1), there is the need for the additional phase gate $\Phi(\delta)$ defined as

$$\Phi(\delta) = \begin{pmatrix} e^{i\delta} & 0 \\ 0 & e^{i\delta} \end{pmatrix},$$

where, for our case, we identify $\delta = -\pi/2$.

Thus, we can now rewrite the rotation operation for an inverter gate X as

$$X \equiv \Phi(\pi/2)R_z(\pi/2)R_y(\pi/2)XR_y(-\pi/2)XR_z(-\pi/2),$$

where $A = R_z(\pi/2)R_y(\pi/2)$, $B = R_y(-\pi/2)$, and $C = R_z(-\pi/2)$. By modifying the operators to accommodate for the right matrix dimensions, we can implement the Toffoli gate.

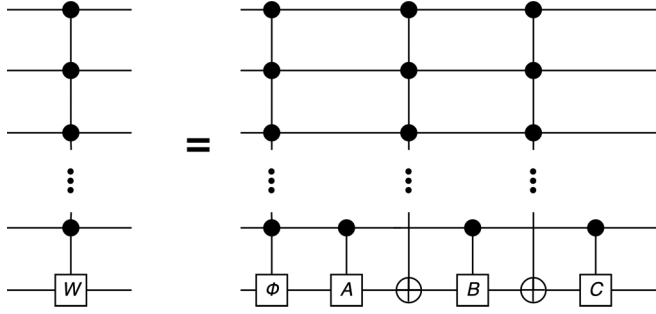


FIG. 5. Generalized rotational network that implements a unitary controlled by an arbitrary number of control qubits.

The next step is to generalize this quantum circuit so it can accommodate more than two control qubits, i.e., create a generalized CNOT using rotations. In order to do this, we can use another lemma from Barenco *et al.* [25]. In this context, we need to introduce the notation $\wedge_{n-1}(U)$ as used by [26,27].

For any unitary matrix $U = \begin{pmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{pmatrix}$ and $m \in \{0, 1, 2, \dots\}$ we define the $(m+1)$ -bit (2^{m+1} -dimensional) operator $\wedge_m(U)$ as

$$\begin{aligned} \wedge_m(U) &(|x_1, \dots, x_m, y\rangle) \\ &= \begin{cases} u_{y0}|x_1, \dots, x_m, 0\rangle + u_{y1}|x_1, \dots, x_m, 1\rangle & \text{if } \wedge_{k=1}^m x_k = 1, \\ |x_1, \dots, x_m, y\rangle & \text{if } \wedge_{k=1}^m x_k = 0, \end{cases} \end{aligned}$$

where \wedge_k denotes the AND operation of the relevant k values.

Lemma 2. [26] For any unitary W , a $\wedge_{n-1}(W)$ gate can be simulated by a network of rotation and phase operators, as shown in Fig. 5, with Φ , A , B , and C as in Lemma 1.

Lemma 2 describes a way to expand any generalized unitary with an arbitrary number m of control qubits to a network of controlled rotations and generalized CNOTs of the form $\Phi A X B X C$. It is easy to see that, if $W = X$, we can iteratively expand each one of the generalized CNOT gates to such a network. The expansion will stop when the generalized inverter gates end up being regular Toffoli gates. After the transformation of the initial approach to rotation operations, the 2×2 operator W applied to the target qubit is the regular inverter

$$W = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad (5)$$

with the dimensions of the matrix representation defined according to the dimensionality of the workspace.

Thus, we can produce a quantum circuit that implements a generalized CNOT gate with an arbitrary number of control qubits without depending on the use of any ancilla qubits. This logic can be applied to any unitary operator [26,27].

We can now integrate this implementation to the increment and decrement circuits. We substitute the generalized CNOT gates with the network described in Lemma 2. Any Toffoli, CNOT, or inverter X gate remains the same. A visualization of an increment quantum circuit on four qubits is shown in Fig. 6. The decrement circuit will follow similar logic with the difference that all the control qubits have to be inverted at the start of the computation.

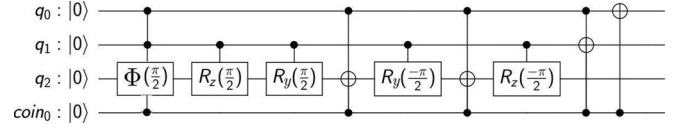


FIG. 6. Rotational implementation of an increment circuit for a three-qubit state space and one-qubit coin.

IV. IMPLEMENTATION CHARACTERISTICS

For our experiments we implement a quantum walk on an N cycle. This means that the walker can be in N different states and needs $n = \log N$ qubits to represent it. One additional single-qubit quantum register is needed for the quantum coin and, in the case of the generalized inverter approach, an ancilla register. We find that the number of qubits needed for the inverter approach increases as $2 \log N$, whereas for the rotational approach as $1 + \log N$.

The difference in the size of the workspace between the two approaches increases linearly with the state space of the walk, i.e., as we move onto larger cycles. On the other hand, the rotational circuit needs an increasingly larger number of gates than the generalized CNOT approach. This is shown with the following propositions.

Proposition 1. The number of gates that participate in the generalized inverter implementation of the quantum-walk circuit increases polylogarithmically with the size of the state space N as $O(\log^2 N)$.

Proof. For a state space $N \geq 8$, any gate that needs more than two control qubits is expanded to a network with ancilla qubits. The number of gates necessary for this expansion can be expressed as $2 \sum_{n_c=3}^{\log N} (2n_c - 1)$, where n_c is the number of control qubits necessary for each operation. The additional gates needed will be the inverters with two or less control qubits and the Hadamard gate. For a state space of $N < 8$ there will be no operations with more than two control qubits and the number of gates will be simply calculated by the inverters and the Hadamard gate.

Thus, the number of gates for the generalized inverter implementation can be expressed as

$$v_c = \begin{cases} 2 \sum_{n_c=3}^{\log N} (2n_c - 1) + 2 \log N + 5, & \text{if } N \geq 8 \\ 2 \log N + 5, & \text{if } 2 \leq N < 8. \end{cases} \quad (6)$$

From the above equation we can see that the sum $\sum_{n_c=3}^{\log N} (2n_c - 1)$ provides the dominant growth rate. We find that

$$\begin{aligned} \sum_{n_c=3}^{\log N} (2n_c - 1) &= \sum_{n_c=1}^{\log N} (2n_c - 1) - \sum_{n_c=1}^2 (2n_c - 1) \\ &= \log^2 N - 4, \end{aligned}$$

as the sum $\sum_{n_c=1}^{\log N} (2n_c - 1) = \log^2 N$ is the known sum of the first $\log N$ odd natural numbers and $\sum_{n_c=1}^2 (2n_c - 1) = 4$. Thus, the number of gates increases with the size of the state space N as $O(\log^2 N)$. ■

Proposition 2. The number of gates that participate in the rotational implementation of the quantum-walk circuit increases linearly with the size of the state space N as $O(N)$.

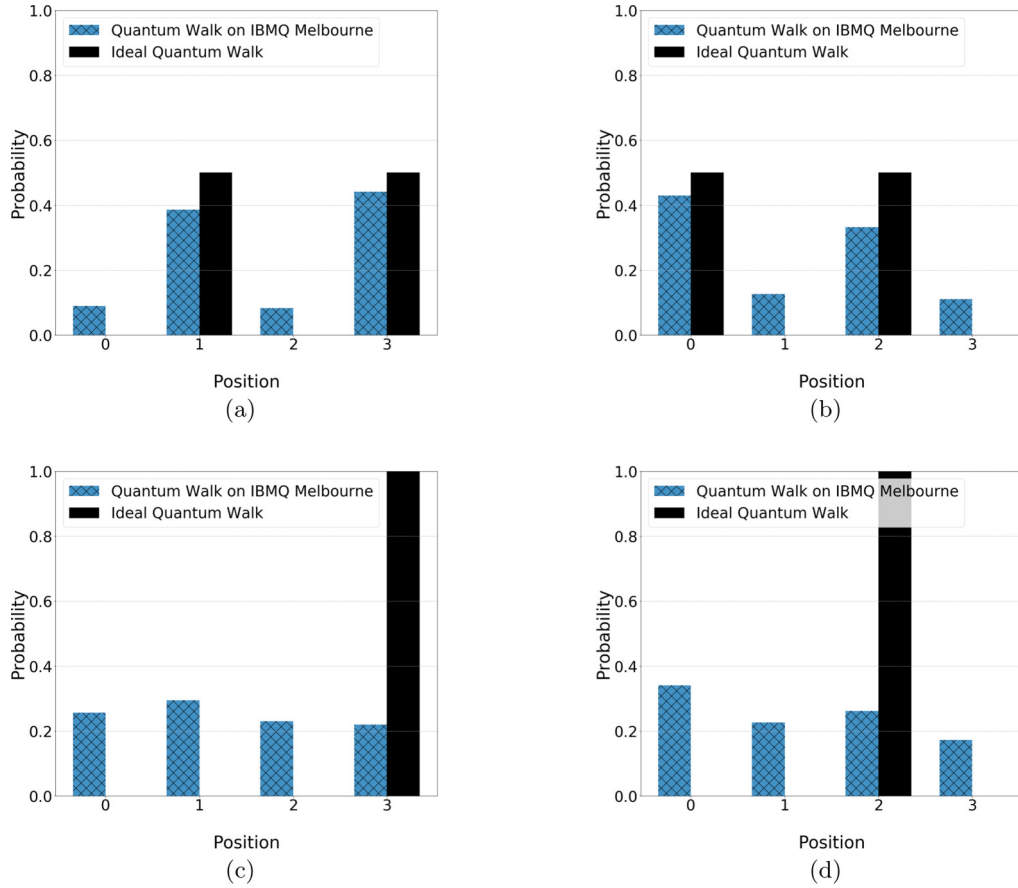


FIG. 7. Probability distributions of two-qubit quantum walks on the IBMQ Melbourne computer (crossed bar) and on an ideal simulator (solid bar) for (a) one step, (b) two steps, (c) three steps, and (d) four steps, with generalized inverter approach (rotations are not needed); 95% confidence intervals are smaller than 10^{-3} , hence are not displayed.

Proof. For $N \geq 8$ any operation with more than three control qubits will be expanded according to the network of Lemma 2. It is evident that every inverter as in Fig. 5 will need to be expanded to a rotational network until all gates need only two control qubits. This leads to a number of $\sum_{j=3}^{\log N} [10 \sum_{n_c=j}^{\log N} (2^{n_c-j})]$ gates before the last step as well as $\sum_{j=3}^{\log N} (2^{2-j+\log N})$ gates on the last step of the expansion. For operations with two or less control qubits, there is no need for rotations or expansions.

For $N < 8$ there will be no operations with more than two control qubits and the circuit will not need a rotational approach. The number of gates for the rotational implementation ν_r can be expressed as

$$\nu_r = \begin{cases} \sum_{j=3}^{\log N} [(2^{2-j+\log N}) + 10 \sum_{n_c=j}^{\log N} (2^{n_c-j})] \\ \quad + 2 \log N + 5, & \text{if } N \geq 8 \\ 2 \log N + 5, & \text{if } 2 \leq N < 8. \end{cases} \quad (7)$$

From Eq. (7) we find that again the sum provides the dominant growth. In this case it represents the well-known sum of a geometric progression, where the largest growth would be given by $2^{\log N} = N$. Thus, the number of gates increases linearly with the size of the state space N as $O(N)$. ■

Liu *et al.* [27] present an analysis on how n -qubit controlled unitaries can be realized by one-qubit and CNOT gates using exponential and polynomial complexity, respectively.

Our propositions, however, analyze the complexity of architectures implementing a quantum walk, not simply multiqubit controlled gates.

As evident from Propositions 1 and 2, the number of gates needed by the rotational implementation increases exponentially faster than for the generalized inverters. Of course, since the exact same circuit is repeated after every coin flip, for arbitrary coin flips t and the same state space N the number of gates is t times the number of gates needed for each implementation.

By studying the propositions we can draw some conclusions regarding the complexity of each approach. For a small state space, the generalized inverter implementation is more efficient in terms of gates, with the efficiency increasing with N . On the other hand, the inverter approach is less efficient in terms of workspace size, quickly surpassing the number of qubits that classical computers can simulate, or the capacity of near-term quantum machines.

V. EXPERIMENTAL RESULTS

In this section we present the results from running quantum-walk experiments on 4- and 8-cycles as a simulation on a classical machine and on the quantum computer. The state space is of size two and three qubits, respectively. We initialize the walker on position $|0\rangle$.

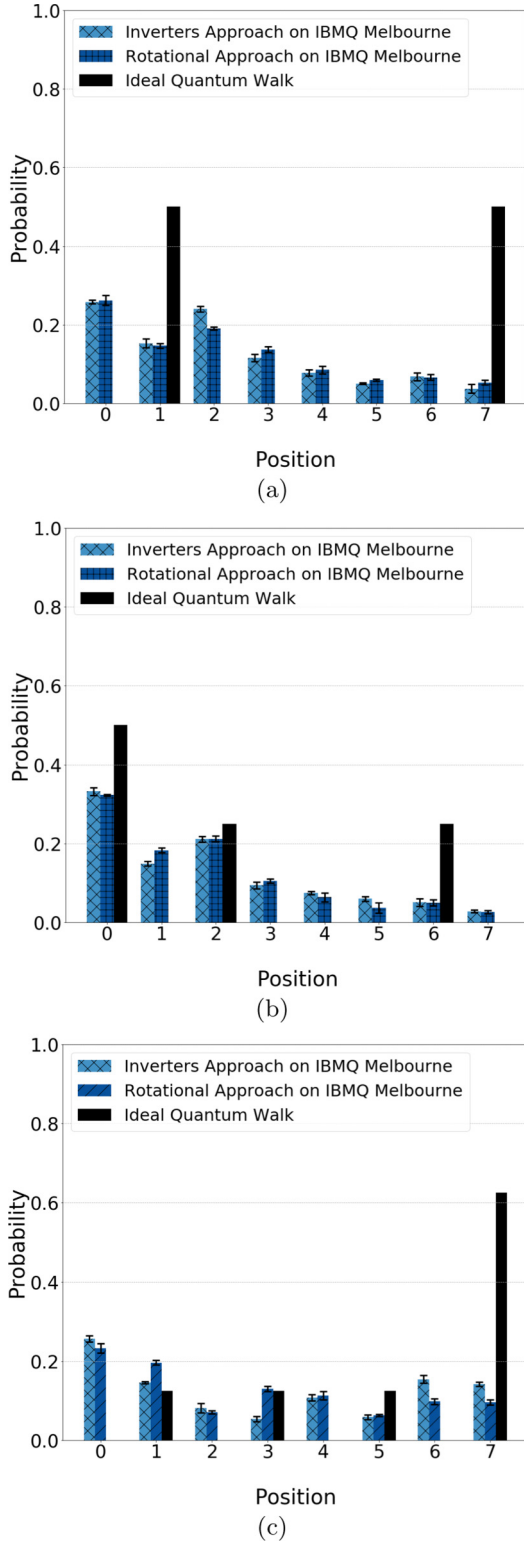


FIG. 8. Probability distributions of quantum walks on three qubits for (a) one step, (b) two steps, and (c) three steps using IBMQ’s Melbourne machine for the generalized inverter (crossed bar) and rotational (tilled bar) approach and an ideal simulator (solid bar), on which both approaches are identical. Error bars are calculated with 95% confidence intervals.

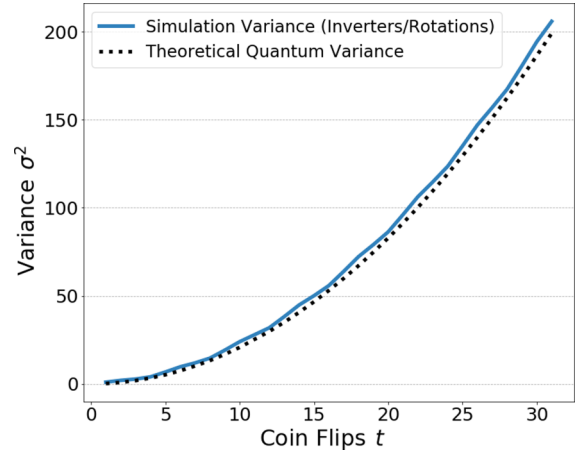


FIG. 9. Variance of the quantum walk as a function of the coin flips. Here, the simulation variance (solid line) is the one computed on a $N = 256$ -cycle via the simulations of either the inverter or the rotational circuit. Theoretical variance (dotted line) is calculated from Eq. (8).

The results of the simulations and experiments for both approaches on two- and three-qubit states are given in Figs. 7 and 8, respectively. We first discuss the results of the classical simulations before moving to the quantum computer. The code is available in [41].

A. Experiments on noise-free simulator

The simulations were run on a MacBook Pro 2017 computer with a 2.3-GHz Intel Core i5 processor and 16 GB of memory. There are three important points to observe here: (i) the asymmetry of the probability distributions, (ii) the modular behavior, and (iii) the variance of the quantum walk. The effects of the asymmetry can be seen as the imbalance in the ideal probability distribution, as for example in Fig. 8(c) with state $|7\rangle$ appearing with higher probability than the rest.

The second point involves the modularity of the quantum walk. Having initialized the walker on state $|0\rangle$ (considered an even state) and evolving for an odd or even number of steps, we predict that the measurement outcome will be an odd or even one, respectively. Indeed that is the case, with the outcomes measured also satisfying that only $N/2$ states are observed. This stands true for both approaches.

The final point refers to the variance of the quantum walk. It has been proven that Markov chains show near-quadratic increase in the variance with respect to time as opposed to their classical analogs [2]. Theoretically, the variance σ^2 as a function of the coin flips t can be calculated as [42]

$$\sigma^2 = \frac{\sqrt{2}-1}{2} t^2 \approx 0.2t^2. \quad (8)$$

By computing the simulated quantum-walk variance we can verify this quadratic tendency for both implementations, as depicted in Fig. 9, concluding that both our quantum circuits are likely to be correct implementations of a quantum walk.

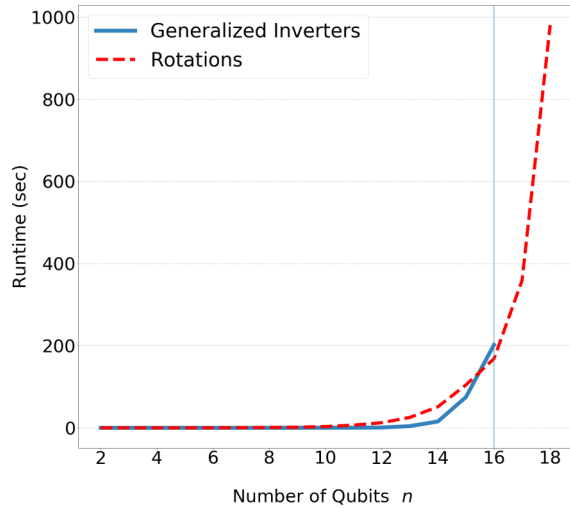


FIG. 10. The simulation runtimes for the generalized inverter (solid line) and rotational (dashed line) implementations. For $n > 16$ our classical machine is unable to simulate the generalized inverter implementation. Simulations are run for one step. The number of qubits n only refers to the state space of the walk.

Finally, the simulation runtimes for different number of qubits on an 8-cycle for the two approaches are presented in Fig. 10. It is visible that the runtime of the quantum-walk circuit increases exponentially with the number of qubits. For the generalized inverter approach, our classical machine is unable to simulate the circuit for $n > 16$ due to lack of memory. On the other hand, the rotational approach is able to simulate the quantum walk for state spaces larger than 16 qubits.

B. Experiments on quantum computer

The results we obtain by executing the experiments on the quantum processor are quite different. Both the quantum-walk experiments are executed on IBMQ's 15-qubit Melbourne machine. Due to limitations on the number of iterations we can submit to the machine, we repeat the quantum walk 1000 times in what constitutes a batch of trials. We repeat such batches of trials 100 times and muster the probability distributions from the average results of the 100 000 repetitions of the experiment. The resulting probability distribution for three steps of the quantum walk on a 4- or 8-cycle can be seen in Fig. 7 and 8, respectively. It is noteworthy here that since for a quantum walk on a 4-cycle there are no inverters with more than two control qubits, a rotational implementation is not needed.

We can see from Figs. 7 and 8 that the empirical distributions differ greatly from the simulations. None of the properties that we expect to see from a quantum walk are present. More specifically, the process no longer exhibits modular behavior as we see states that should not occur, especially on the walks with larger state space or number of steps. Unfortunately, the resulting probability distributions are completely different to the theoretical ones, making it difficult to correctly calculate the variance. Thus, no remarks regarding the variance can be made with certainty.

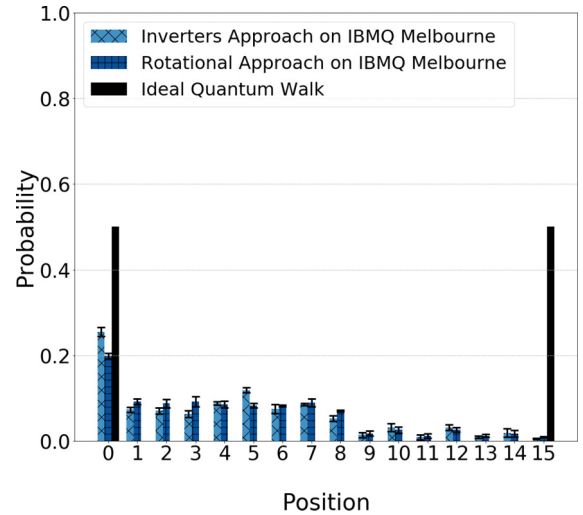


FIG. 11. Probability distributions of generalized inverter (crossed bar), rotational (tiled bar), and ideal (solid bar) quantum walks on four qubits for one step. Error bars are calculated with 95% confidence intervals.

In another paper, Balu *et al.* [43] realize a topological quantum walk on a 4-cycle for one step of the walker. This implementation differs from both approaches we are concerned with in our work. The results capture the topological effects of the quantum walk at a single step, but fail to do so for more steps due to noise. Qualitatively, the results of [43] match the theoretical expectations for one step of the walk in a way similar to ours.

Taking into account the nominal execution time of the different gates that participate in the circuits, as given by IBMQ, we can calculate the overall runtime of the quantum computations. For one step of the walk with three-qubit state space, the generalized CNOT circuit's execution time is approximately 42 ms and for the rotational circuit 76 ms. Similarly, for one and two steps of the walk on two qubits, the execution times are 10.5 and 21 ms, respectively. These execution times are purely for the operations themselves, meaning we do not take into account the state preparation (if it occurs), the measurement, or the buffer time between operations on the same qubit.

In general, we find that shorter quantum-walk circuits, for example, one or two steps of a two-qubit quantum walk, can generally provide results closer to our expectations, as shown in Figs. 7(a) and 7(b). This is due to the low execution time of the quantum circuit, as calculated above, compared to the average coherence time of the three qubits participating in the circuit (i.e., 64.25 ms) and the smallest decoherence time among these qubits (i.e., 56 ms). For more than two steps we find that the distribution starts to deviate from the expected [Figs. 7(c) and 7(d)].

As shown in Figs. 8(a)–8(c), the effects of the noise remain intense for both approaches on an 8-cycle. The average coherence of the qubits participating in the circuit is found to be 58.6 ms for the six qubits in the inverter approach and 63.6 ms for the four qubits of the rotational approach.

Finally, similar results are obtained on larger state spaces. One example is one step of the walk on a four-qubit state space, shown in Fig. 11. Thus, we can conclude that, for one

or two steps on a two-qubit state space, the quantum walk behaves relatively close to the expectation, whereas for more than three steps or for a three-qubit state space, where the runtime of the circuit is longer and approaches the coherence time, the effects of the noise are overwhelming.

VI. DISCUSSION

Throughout this paper, two approaches are used to implement a quantum walk: (i) the generalized inverter [25] and (ii) the rotational approach. Our experiments show that using generalized inverters keeps the implementation simple and the circuit shallow but requires an ancilla register. The number of ancilla qubits increases linearly with the number of control qubits, quickly leading to a large workspace, limiting the capabilities of our experiment. The rotational approach deals with this limitation by rendering the ancilla register obsolete, allowing us to experiment with a much larger state space for our quantum walks. The disadvantage of the rotational approach is the larger complexity of the resulting circuit.

It is evident from our experiments that the two implementations of quantum walks offer opposite advantages and disadvantages. Implementing a quantum walk with generalized inverters shows smaller execution time and requires exponentially less gates as a function of the size of the state space or, in other words, smaller circuit depth. On the other hand, the rotational circuit requires less qubits to be used in the workspace, as there is no need for an ancilla register, but the circuit is much deeper than the first approach.

The dependency of the operations between the qubits within the architectures does not allow for any gates to be run in parallel, restricting the width of the circuit at each time step. Thus, the runtime of the quantum circuit quickly surpasses the coherence time of the qubits, leading to immensely noisy distributions. For very small state spaces (i.e., two qubits) we see distributions closer to the theory with lower level of noise. This does not hold for walks with a three-qubit state space or larger. The execution time of the circuit with relation to the coherence time of the qubits greatly affects the resulting distribution of the quantum walk.

Due to the stochastic nature of the noise, it is very difficult to draw safe conclusions from the comparison of the two approaches on the quantum computer. We can, however, point out that since the rotational circuit is deeper, the cumulative error due to hardware infidelities will be more extensive. Additionally, since the computation also takes longer in the case of the rotational circuit, the active qubits have higher chance to decohere. Thus, we believe it is safe to claim that the rotational circuit will be the noisier of the two.

As for the comparison between the two approaches, we can streamline the discussion using the quantum volume [44], which is an architecture-neutral figure of merit that showcases the performance of a quantum computer when running quantum circuits. It depends on the number of physical qubits in the machine, the number of qubits utilized by the circuit, the depth and width of the circuit, as well as the average effective error rates of the two-qubit gates implemented in the quantum computer.

Following the above discussion, we can calculate the quantum volume that each circuit would require to be run on

the same quantum architecture. More specifically, considering as an example one step of a quantum walk with a three-qubit state space on IBMQ's Melbourne machine, the generalized inverter approach requires a quantum volume of approximately 20.812, whereas the rotational approach requires 28.905 (for more information on the calculation of the quantum volume for the three-qubit example, see Appendix C). In a similar way we can calculate the volume for the larger quantum-walk circuits. Thus, we can safely conclude that the generalized inverter approach would require the smaller quantum volume of the two implementations due to the much smaller circuit depth and subsequently reduced execution time compared to the rotational approach, as well as lower the cumulative effective error due to hardware infidelities.

It is noteworthy that, to the best of our knowledge, no optimal implementation of quantum walks currently exists. The generalized inverter approach proposed by [25] constitutes an efficient circuit. Other implementations might exist that could perhaps prove to be more efficient and could be the subject of further comparison.

Finally, due to large presence of noise in the quantum machine, we cannot draw firm conclusions regarding today's quantum hardware. In this paper we have worked out a resource theory for and compared two different implementations of quantum walks. Therefore, considering the limitations and hardware constraints of NISQ machines, our analysis can assist on the realization of quantum walks on near-term quantum computers in an efficient way.

ACKNOWLEDGMENTS

This work was supported by the Engineering and Physical Sciences Research Council, Centre for Doctoral Training in Cloud Computing for Big Data (Grant No. EP/L015358/1).

APPENDIX A: GENERALIZED INVERTER GATE FOR ARBITRARY CONTROL QUBITS

In many circumstances, we need to control an inversion with an arbitrary number of $n_c > 2$ control qubits. A solution can be given by introducing intermediate computations, with their results stored in an ancilla register of size $n_c - 1$ [45]. A visualization of this solution for a generalized CNOT gate with n_c control qubits is shown in Fig. 12. This decomposition of the generalized CNOT gate can be further simplified to use just regular CNOT operations.

APPENDIX B: REALIZATION OF ROTATION OPERATIONS WITH IBM's U_3 GATE

Here we show how IBM Qiskit's [29] U_3 gate can be used to create a rotation operator of the form $R_y(\theta)$. The $U_3(\theta, \phi, \lambda)$ gate is a single-qubit gate with three Euler angles. The gate implements the following operator:

$$U_3(\theta, \phi, \lambda) = \begin{pmatrix} \cos(\theta/2) & -e^{i\lambda} \sin(\theta/2) \\ e^{i\phi} \sin(\theta/2) & e^{i\lambda+i\phi} \cos(\theta/2) \end{pmatrix}. \quad (\text{B1})$$

For our case we can assign $\theta = \pi/2$, $\phi = \lambda = 0$, thus getting

$$R_y(\pi/2) = U_3(\pi/2, 0, 0),$$

with matrix representation easily deduced from Eq. (B1).

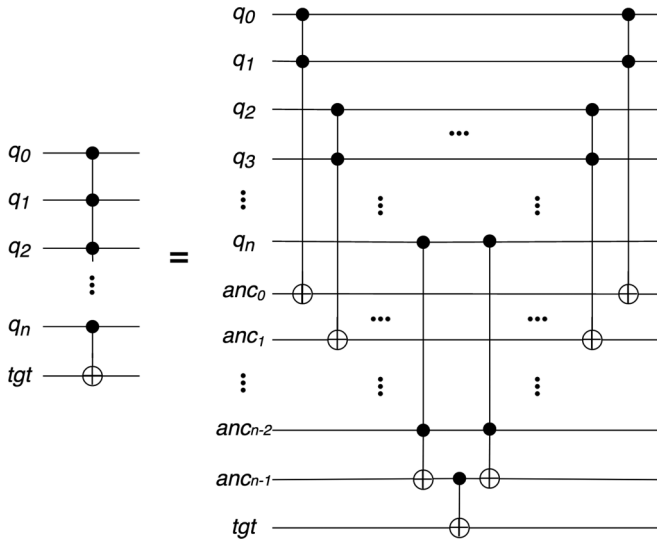


FIG. 12. Generalized Toffoli gate with n control qubits (q_0 to q_n), $n - 1$ ancilla qubits (anc_0 to anc_{n-1}), and one target qubit (tgt).

APPENDIX C: CALCULATING THE QUANTUM VOLUME

The quantum volume, as introduced by Moll *et al.* [44], is a figure of merit that characterizes the performance of a quantum computer. In the context of this paper, we use the quantum volume in order to attach an additional metric that

shows the required resources of each quantum-walk approach on IBMQ's 15-qubit Melbourne machine.

The quantum volume can be defined as [44]

$$V_Q = \max_{n < N} \left(\min \left[n, \frac{1}{n\epsilon_{\text{eff}}(n)} \right]^2 \right), \quad (\text{C1})$$

where n is the size of the workspace necessary for the computation, N is the number of qubits within the quantum computer, and $\epsilon_{\text{eff}}(n)$ is the average effective two-qubit gate error rates of the qubits that participate in the circuit, following the connectivity of the architecture.

Thus, we can now easily calculate the quantum volume of the three-qubit experiment we use as an example. It is important here that, due to automatic optimization of the connectivity before execution in the computer, additional qubits that store intermediate quantum states have to be taken into account when calculating the quantum volume. Subsequently, the size of the workspace for the generalized inverter approach is $n_c = 8$ and for the rotational is $n_r = 6$. For the qubits that participate in the workspace of each approach, we can compute the average error rates as $\epsilon_{\text{eff}}^c = 2.74 \times 10^{-2}$ and $\epsilon_{\text{eff}}^r = 3.10 \times 10^{-2}$ for the inverter and rotational approach, respectively, as of the day of the experiment.

Substituting the above values in Eq. (C1), we compute the quantum volume for the generalized inverter approach as $V_Q^c = 20.812$ and for the rotational approach as $V_Q^r = 28.905$. In a similar fashion we can calculate the quantum volume for a quantum walk of arbitrary size.

- [1] J. Kempe, Quantum random walks: An introductory overview, *Contemp. Phys.* **44**, 307 (2003).
- [2] M. Szegedy, Quantum speed-up of Markov chain based algorithms, in *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, FOCS '04* (IEEE Computer Society, Washington, DC, 2004), pp. 32–41.
- [3] P. C. Richter, Quantum speedup of classical mixing processes, *Phys. Rev. A* **76**, 042306 (2007).
- [4] N. Shenvi, J. Kempe, and K. Birgitta Whaley, Quantum random-walk search algorithm, *Phys. Rev. A* **67**, 052307 (2003).
- [5] A. N. Chowdhury and R. D. Somma, Quantum algorithms for gibbs sampling and hitting-time estimation, *Quantum Inf. Comput.* **17**, 41 (2017).
- [6] A. Montanaro, Quantum speedup of Monte Carlo methods, *Proc. R. Soc. A: Math., Phys. Eng. Sci.* **471**, (2015).
- [7] K. Temme, T. J. Osborne, K. G. Vollbrecht, D. Poulin, and F. Verstraete, Quantum metropolis sampling, *Nature (London)* **471**, 87 (2011).
- [8] A. Dhahri, C. K. Ko, and H. J. Yoo, Quantum Markov chains associated with open quantum random walks, *J. Stat. Phys.* **176**, 1272 (2019).
- [9] D. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani, Quantum walks on graphs, in *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing, STOC '01* (ACM, New York, 2001), pp. 50–59.
- [10] A. Ambainis, Quantum walk algorithm for element distinctness, *SIAM J. Comput.* **37**, 210 (2007).
- [11] F. Magniez, A. Nayak, J. Roland, and M. Santha, Search via quantum walk, in *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing, STOC '07* (ACM, New York, 2007), pp. 575–584.
- [12] A. M. Childs and J. Goldstone, Spatial search by quantum walk, *Phys. Rev. A* **70**, 022314 (2004).
- [13] A. Montanaro, Quantum-walk speedup of backtracking algorithms, *Theory Comput.* **14**, 1 (2018).
- [14] 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 Thirty-fifth ACM Symposium on Theory of Computing, STOC '03* (ACM, New York, 2003), pp. 59–68.
- [15] J. Kempe, Discrete quantum walks hit exponentially faster, *Probab. Theory Relat. Fields* **133**, 215 (2005).
- [16] L. Sansoni, F. Sciarrino, G. Vallone, P. Mataloni, A. Crespi, R. Ramponi, and R. Osellame, Two-Particle Bosonic-Fermionic Quantum Walk Via Integrated Photonics, *Phys. Rev. Lett.* **108**, 010502 (2012).
- [17] T. Rakovszky and J. K. Asboth, Localization, delocalization, and topological phase transitions in the one-dimensional split-step quantum walk, *Phys. Rev. A* **92**, 052311 (2015).
- [18] S. Muel, A. Celi, P. Massignan, J. K. Asboth, M. Lewenstein, and C. Lobo, Topological bound states of a quantum walk with cold atoms, *Phys. Rev. A* **94**, 023631 (2016).
- [19] M. Grafe and A. Szameit, Integrated photonic quantum walks, *J. Phys. B: At., Mol. Opt. Phys.* **53**, 073001 (2020).

- [20] M. A. Broome, A. Fedrizzi, B. P. Lanyon, I. Kassal, A. Aspuru-Guzik, and A. G. White, Discrete Single-Photon Quantum Walks With Tunable Decoherence, *Phys. Rev. Lett.* **104**, 153602 (2010).
- [21] A. Schreiber, K. N. Cassemiro, V. Potoček, A. Gábris, P. J. Mosley, E. Andersson, I. Jex, and C. Silberhorn, Photons Walking The Line: A Quantum Walk With Adjustable Coin Operations, *Phys. Rev. Lett.* **104**, 050502 (2010).
- [22] M. Karski, L. Förster, J. Choi, A. Steffen, W. Alt, D. Meschede, and A. Widera, Quantum walk in position space with single optically trapped atoms, *Science* **325**, 174 (2009).
- [23] C. Robens, W. Alt, D. Meschede, C. Emary, and A. Alberti, Ideal Negative Measurements in Quantum Walks Disprove Theories Based on Classical Trajectories, *Phys. Rev. X* **5**, 011003 (2015).
- [24] H. Schmitz, R. Matjeschk, C. Schneider, J. Glueckert, M. Enderlein, T. Huber, and T. Schaetz, Quantum Walk Of A Trapped Ion In Phase Space, *Phys. Rev. Lett.* **103**, 090504 (2009).
- [25] B. L. Douglas and J. B. Wang, Efficient quantum circuit implementation of quantum walks, *Phys. Rev. A* **79**, 052335 (2009).
- [26] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, Elementary gates for quantum computation, *Phys. Rev. A* **52**, 3457 (1995).
- [27] Y. Liu, G. L. Long, and Y. Sun, Analytic one-bit and CNOT gate constructions of general n-qubit controlled gates, *Int. J. Quantum Inf.* **06**, 447 (2008).
- [28] J. Preskill, Quantum computing in the NISQ era and beyond, *Quantum* **2**, 79 (2018).
- [29] IBM Qiskit, <https://qiskit.org/>.
- [30] IBM Quantum Experience, <https://www.ibm.com/quantum-computing/technology/experience>
- [31] C. Blank, D. K. Park, J.-K. K. Rhee, and F. Petruccione, Quantum classifier with tailored quantum kernel, *npj Quantum Inf.* **6**, 41 (2020).
- [32] N.-N. Huang, W.-H. Huang, and C.-M. Li, Identification of networking quantum teleportation on 14-qubit ibm universal quantum computer, *Sci. Rep.* **10**, 3093 (2020).
- [33] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta, Validating quantum computers using randomized model circuits, *Phys. Rev. A* **100**, 032328 (2019).
- [34] A. Martín, B. Candelas, A. Rodríguez-Rozas, J. Martín-Guerrero, X. Chen, L. Lamata, R. Orús, E. Solano, and M. Sanz, Towards pricing financial derivatives with an IBM quantum computer, [arXiv:1904.05803v1](https://arxiv.org/abs/1904.05803).
- [35] G.-R. Feng, Y. Lu, L. Hao, F.-H. Zhang, and G.-L. Long, Experimental simulation of quantum tunneling in small systems, *Sci. Rep.* **3**, 2232 (2013).
- [36] C. A. Ryan, M. Laforest, J. C. Boileau, and R. Laflamme, Experimental implementation of a discrete-time quantum random walk on an nmr quantum-information processor, *Phys. Rev. A* **72**, 062317 (2005).
- [37] F. Acasiete, F. P. Agostini, J. K. Moqadam, and R. Portugal, Implementation of quantum walks on IBM quantum computers, *Quantum Inf. Proc.* **19**, (2020).
- [38] A. Ambainis, E. Bach, A. Nayak, A. Vishwanath, and J. Watrous, One-dimensional quantum walks, in *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing, STOC '01* (ACM, New York, 2001), pp. 37–49.
- [39] D. Reitzner, D. Nagaj, and V. Buzek, Quantum walks, *Acta Phys. Slovaca, Rev. Tutoriales* **61**, 603 (2011).
- [40] H. Krovi, Symmetry in quantum walks, [arXiv:0711.1694v1](https://arxiv.org/abs/0711.1694).
- [41] <https://github.com/pzuliani/qwalk>.
- [42] A. C. Orthey, Jr. and E. P. M. Amorim, On the spreading of quantum walks starting from local and delocalized states, [arXiv:1706.06257](https://arxiv.org/abs/1706.06257).
- [43] R. Balu, D. Castillo, and G. Siopsis, Physical realization of topological quantum walks on IBM-q and beyond, *Quantum Sci. Technol.* **3**, 035001 (2018).
- [44] N. Moll, P. Barkoutsos, L. S. Bishop, J. M. Chow, A. Cross, D. J. Egger, S. Filipp, A. Fuhrer, J. M. Gambetta, M. Ganzhorn *et al.*, Quantum optimization using variational algorithms on near-term quantum devices, *Quantum Sci. Technol.* **3**, 030503 (2018).
- [45] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th ed. (Cambridge University Press, New York, 2011).