

Complex-phase extensions of the Szegedy quantum walk on graphsSergio A. Ortega ^{1,*} and Miguel A. Martin-Delgado ^{1,2,†}¹*Departamento de Física Teórica, Universidad Complutense de Madrid, 28040 Madrid, Spain*²*CCS-Center for Computational Simulation, Universidad Politécnica de Madrid, 28660 Boadilla del Monte, Madrid, Spain*

(Received 6 November 2024; accepted 18 February 2025; published 19 March 2025)

This work introduces a graph-phased Szegedy's quantum walk, which incorporates link phases and local arbitrary phase rotations (APR), unlocking new possibilities for quantum algorithm efficiency. We demonstrate how to adapt quantum circuits to these advancements, allowing phase patterns that ensure computational practicality. The graph-phased model broadens the known equivalence between coined quantum walks and Szegedy's model, accommodating a wider array of coin operators. Through illustrative examples, we reveal intriguing disparities between classical and quantum interpretations of walk dynamics. Remarkably, local APR phases emerge as powerful tools for marking graph nodes, optimizing quantum searches without altering graph structure. We further explore the surprising nuances between single and double operator approaches, highlighting a greater range of compatible coins with the latter. To facilitate these advancements, we present an improved classical simulation algorithm, which operates with superior efficiency. This study not only refines quantum walk methodologies but also paves the way for future explorations, including potential applications in quantum search and PageRank algorithms. Our findings illuminate the path towards more versatile and powerful quantum computing paradigms.

DOI: [10.1103/PhysRevA.111.032216](https://doi.org/10.1103/PhysRevA.111.032216)**I. INTRODUCTION**

Quantum walks are algorithms born from the quantization of classical Markov chains. They were first proposed in the discrete time version [1] and later in continuous time [2]. The same as classical walks, quantum walks have given rise to a wide variety of algorithms for problems such as triangle finding [3], element distinctness [4], and quantum search [5].

Quantum walks in discrete time usually required an inner degree of freedom, giving rise to what is called the coined quantum walk model [6]. These walks act on undirected graph and have difficulties quantizing arbitrary Markov chains. To solve this issue, Szegedy introduced a coinless quantum walk on bipartite graphs quantizing a general Markov chain on weighted graphs by a duplication process [7]. This quantum walk has been shown to be useful for problems of optimization [8–12], testing graph completeness [13], classification [14–17], quantum search [18–20], and machine learning [21].

Soon after Szegedy's work, an alternative version of Szegedy's quantum walk avoiding the need to duplicate the graph [22] was developed. This formulation produces more general results for classical weighted graphs [23] and, moreover, allows establishing an equivalence between Szegedy's quantum walk and the coined model [24–26].

Different modifications of Szegedy's quantum walk introducing complex phases have been proposed in the context of Grover walks [27], staggered quantum walks [28], and the quantum PageRank algorithm [16], where they have provided algorithms outperforming the standard walk. In this work we provide a further generalization of the phase extensions, obtaining what we call the graph-phased Szegedy's quantum walk. Moreover, we have studied how provided an efficient quantum circuit for the implementation of Szegedy's quantum walk [29,30], we can modify them to include the complex phase extensions.

An important result we find for our graph-phased Szegedy's model is that the set of coined quantum walks that can be cast into a Szegedy's walk is increased considerably. We review the conditions that must satisfy the coin operators [24] and provide new conditions considering also the phase extensions. Furthermore, we also show how the new extensions that we introduce in our work allow marking nodes in a graph using oracles. We also use this extension to review the proof that the method of absorbing vertices usually used for marking [7,18] is equivalent to the use of a different coin in the coined walk [5,25].

Finally, we provide an efficient classical simulation algorithm for the graph-phased Szegedy's quantum walk considering all possible phase extensions. For this, we have modified a recent algorithm that we proposed for the current Szegedy's walks, which scales as $O(N^2)$ in both time and memory requirements [31]. This algorithm outperforms previous simulation methods that require the explicit construction of the unitary evolution matrix, scaling as $O(N^3)$ for dense weighted graphs.

This paper is structured as follows. In Sec. II we review the formulation of Szegedy's quantum walk to later introduce

*Contact author: sergioan@ucm.es†Contact author: mardel@ucm.es

the graph-phased model. In Sec. III we show the efficient construction of quantum circuits. In Sec. IV we study the equivalence between the graph-phased Szegedy’s walk and the coined model. In Sec. V we show examples of coined quantum walks on the infinite line. In Sec. VI we show how some phase extensions can be used for marking nodes in a graph. In Sec. VII we present the classical simulation algorithm for the graph-phased walk considering all the phase extensions. Finally, we summarize and conclude in Sec. VIII.

II. GRAPH-PHASED SZEGEDY QUANTUM WALK

In this section we review the standard quantization of Szegedy’s quantum walk [7,22] and the current extensions found in the literature [16,27,28] to later generalize it introducing the graph-phased Szegedy’s model.

A. Szegedy’s quantum walk

A classical Markov chain is a stochastic process that can be represented as a random walk on the nodes of a graph. For a weighted graph with N nodes, this walk is described by a $N \times N$ transition matrix G . By convention in this work we choose that the transition matrix is column stochastic, so that all the columns add up to 1. Thus, the elements G_{ji} are the probabilities of the walker jumping from node i to node j . Let us define $p(t)$ as a N -dimensional column vector whose elements are the probabilities of the classical walker being at each node at time step t . Then the classical walk evolution is governed by the following equation:

$$p(t) = G^t p(0). \tag{1}$$

The classical Markov chain is quantized through Szegedy’s quantum walk [7,22]. In this quantum walk the Hilbert space is the span of all the vectors representing the $N \times N = N^2$ directed edges of the graph, i.e.,

$$\mathcal{H}_S = \text{span}\{|i\rangle_1 |j\rangle_2, i, j = 0, 1, \dots, N - 1\}, \tag{2}$$

where the states with indexes 1 and 2 refer to the nodes on two copies of the original graph. Thus, the states are defined over two quantum registers. In this paper we count the nodes of the network, and therefore the matrix indexes, from 0 to $N - 1$. We define the vectors

$$|\psi_i\rangle := |i\rangle_1 \otimes \sum_{k=0}^{N-1} \sqrt{G_{ki}} |k\rangle_2, \tag{3}$$

which are a superposition of the vectors representing the edges outgoing from the i th vertex, whose coefficients are given by the square root of the i th column of the matrix G . From these vectors we define the following projector operator:

$$\Pi := \sum_{i=0}^{N-1} |\psi_i\rangle \langle \psi_i|, \tag{4}$$

and use it to define a reflection operator on the subspace generated by the $|\psi_i\rangle$ states:

$$R := 2\Pi - \mathbb{1}. \tag{5}$$

The quantum walk evolution operator U_s is defined as

$$U_s := S_w R, \tag{6}$$

where S_w is a swap operator that exchanges the two quantum registers, i.e.,

$$S_w := \sum_{i,j=0}^{N-1} |i, j\rangle \langle j, i|. \tag{7}$$

The Szegedy’s quantum walk described in this manner is actually an alternative definition [22] that generalizes the original one [7] and is more suitable for establishing an equivalence with the coined quantum walk model [6], which we will explore further in Sec. IV. However, the original formulation made by Szegedy consisted on a coinless quantum walk based on two reflections, so that the quantum walk evolution operator was defined as $W_s = R_B R_A$. It turns out that $R_A = R$ and $R_B = S_w R S_w$, so that the original unitary operator W_s corresponds to two steps of the alternative version U_s , i.e., $W_s = U_s^2$. Thus, we will refer to it as the double Szegedy operator.

Finally, the initial state is usually constructed by a superposition of the $|\psi_i\rangle$ states, and the probability distribution of the walker after each time step of the quantum walk is usually obtained measuring the first register. However, there are also algorithms where the information of interest is obtained measuring the second register instead [14–17,19].

B. Complex phase extensions

An extension of Szegedy’s model, dubbed twisted Szegedy walk, introduced complex phases by means of a kind of complex-valued weights [27]. In our notation, this corresponds to an extended Szegedy walk where the $|\psi_i\rangle$ states are substituted by [28]

$$|\psi_i(\varphi)\rangle := \sum_{k=0}^{N-1} e^{i\varphi_{ik}} \sqrt{G_{ki}} |i\rangle_1 |k\rangle_2, \tag{8}$$

where φ is an $N \times N$ matrix, so that the element φ_{ij} is the phase associated to the edge state $|i\rangle_1 |j\rangle_2$. Thus, the unitary evolution operator is $U_s(\varphi) = S_w R(\varphi)$, where $R(\varphi)$ reflects around the subspace generated by the $|\psi_i(\varphi)\rangle$ states. When all the phases are 0 the standard Szegedy’s walk is recovered. Note that due to the convention of the transition matrix G being column-stochastic, the weight probability associated to edge state $|i\rangle_1 |j\rangle_2$ is G_{ji} . Thus, the real weights in the transition matrix G are associated with the elements of φ^T and vice versa.

Another modification of Szegedy’s quantum walk changing the reflection R by an arbitrary phase rotation (APR) was proposed in the context of quantum PageRank [16]. Although in the previous case N^2 phases were introduced, in this case a single phase denoted as θ is used to define the phase rotation operator,

$$R(\theta) := (1 - e^{i\theta})\Pi - \mathbb{1}, \tag{9}$$

so that the unitary evolution operator is $U_s(\theta) = S_w R(\theta)$. The standard walk is recovered for $\theta = \pi$.

In order to differentiate both phase extensions, we will call link phases to the phases φ_{ij} associated to the edges of the graph, and APR phase to the phase θ related to the arbitrary phase rotation. These two kind of extensions are compatible

and can be applied at the same time. Before constructing a general operator that combines all the phases, it is important to mention that the twisted Szegedy walk provided another extension with a complex phase for each edge acting inside the swap operator [27]. Nevertheless, this is beyond the scope of this work, as we are interested in the extensions of the reflection operator R , which acts as the coin of the coined quantum walk. Moreover, we can still extend further this operator.

C. Graph-phased Szegedy’s quantum walk

The APR phase θ acts in a global manner in the graph. However, the same as there are N^2 phases φ_{ij} associated to the edges of the graph, we could think of N APR phases θ_i associated to each of the nodes of the graph. Let us expand the projector Π in (9), and introduce the constant factor $(1 - e^{i\theta})$ in the sum:

$$R(\theta) = \sum_{i=0}^{N-1} (1 - e^{i\theta}) |\psi_i\rangle \langle \psi_i| - \mathbb{1}. \tag{10}$$

Now, we can give θ a different value θ_i for each node i . Let us define $\vec{\theta}$ as a vector with the N different APR phases. The phase rotation operator becomes

$$R(\vec{\theta}) = \sum_{i=0}^{N-1} (1 - e^{i\theta_i}) |\psi_i\rangle \langle \psi_i| - \mathbb{1}. \tag{11}$$

These APR phases act locally in the nodes of the graph, and the sum cannot be factorized with a projector operator.

Now we can define the graph-phased model joining all the phase extensions. Let us define the following operator:

$$\Sigma(\vec{\theta}, \varphi) := \frac{1}{2} \sum_{i=0}^{N-1} (1 - e^{i\theta_i}) |\psi_i(\varphi)\rangle \langle \psi_i(\varphi)|. \tag{12}$$

This operator generalizes the projector Π in (4), which is recovered for $\theta_i = \pi$ and $\varphi_{ij} = 0$. It is a kind of pseudoprojector, since acting on an arbitrary state provides a state in the subspace spanned by the $|\psi_i(\varphi)\rangle$ states. However, it does not corresponds to the actual orthogonal projection.

The phase rotation associated to this walk is defined as

$$R(\vec{\theta}, \varphi) := 2\Sigma(\vec{\theta}, \varphi) - \mathbb{1}. \tag{13}$$

The unitary evolution operator is defined as

$$U_s(\vec{\theta}, \varphi) := S_w R(\vec{\theta}, \varphi). \tag{14}$$

Again, we can think of the double Szegedy operator W_s using two times the operator $U_s(\vec{\theta}, \varphi)$. In general the phases of both operators can be different, so that the most general double Szegedy operator is as follows:

$$W_s(\vec{\theta}_1, \varphi_1, \vec{\theta}_2, \varphi_2) = U_s(\vec{\theta}_2, \varphi_2) U_s(\vec{\theta}_1, \varphi_1). \tag{15}$$

Before going on, let us fix some definitions about the different models we will deal with in this work.

Definition 1 (Standard Szegedy’s quantum walk). It is the Szegedy’s quantum walk without any phase extension.

Definition 2 (Link-phased Szegedy’s quantum walk). It is the Szegedy’s quantum walk with link phases φ_{ij} , but without APR phases.

Definition 3 (Szegedy’s quantum walk with global APR). It is the Szegedy’s quantum walk with the same APR phase θ for all nodes, but without link phases.

Definition 4 (Vertex-phased Szegedy’s quantum walk). It is the Szegedy’s quantum walk with multiple local APR phases θ_i , but without link phases.

Definition 5 (Graph-phased Szegedy’s quantum walk). It is the most general Szegedy’s quantum walk with all the phase extensions.

Unless otherwise stated, we will consider the single step operator U_s when we refer to each model.

III. QUANTUM CIRCUITS

Constructing a quantum circuit for Szegedy’s quantum walk is not a trivial task since it depends on the particular weighted graph where the walk takes place. However, there is a general structure for decomposing the circuit. Here we review the construction of quantum circuits for the standard model and later show how to modify them to include the phase extensions. Thus, provided a quantum walk for the standard Szegedy’s quantum walk, we can obtain the circuit of the graph-phased model.

A. Standard Szegedy’s circuit

Let us consider for simplicity that the number of nodes of the graph N can be expressed as a power of two, such that $N = 2^n$ for some integer n . Then the quantum circuit needs two quantum registers of n qubits each. For $N \neq 2^n$ we could augment the graph in a manner such that the original walk occurs in an invariant subspace of the augmented graph.

The evolution operator in (6) was $U_s = S_w R$. The swap S_w is trivially a bunch of swap gates between the qubits of the first register and the qubits of the second register. However, the reflection R is more complicated. This operator needs to be diagonalized [29,30]. With this purpose, we define the update operator V , which creates the $|\psi_i\rangle$ state if the first register is in state $|i\rangle_1$, provided that the second register is in $|0\rangle_2$. Thus, its action is defined as

$$V|i\rangle_1|0\rangle_2 = |\psi_i\rangle, \tag{16}$$

and the action on the rest of the computational basis is irrelevant as long as it is unitary. The update operator V diagonalizes the reflection operator R as $D = V^\dagger R V$, so that using (5) and (4) we obtain:

$$\begin{aligned} D &= 2 \sum_{i=0}^{N-1} V_i^\dagger |\psi_i\rangle \langle \psi_i| V_i - \mathbb{1} \\ &= 2 \sum_{i=0}^{N-1} |i\rangle_1 \langle i| \otimes |0\rangle_2 \langle 0| - \mathbb{1}. \end{aligned} \tag{17}$$

Factorizing the identity in both registers as $\mathbb{1} = \mathbb{1}_1 \otimes \mathbb{1}_2$, and using the completeness relation of the identity in the first register, we obtain

$$D = \mathbb{1}_1 \otimes (|0\rangle_2 \langle 0| - \mathbb{1}_2). \tag{18}$$

This operator corresponds to a reflection around the state $|0\rangle_2$ in the second register. This is a diagonal operator with eigen-

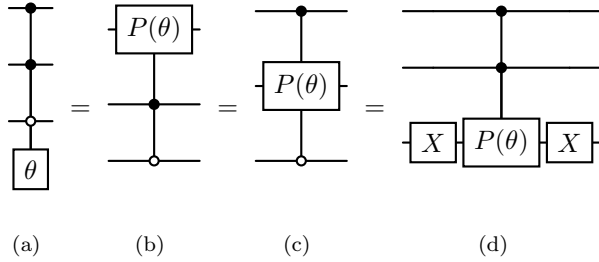


FIG. 1. (a) Quantum circuit representation of a multicontrolled- $P(\theta)$ gate on a ghost qubit, which in this example applies a relative phase $e^{i\theta}$ to the state $|110\rangle$. The real target can be whatever qubit as shown in (b)–(d), as long as the target qubit is surrounded by X gates if it must be in $|0\rangle$ for the application of the phase gate.

value $+1$ for $|0\rangle_2$ and -1 for the rest of the computational basis of the second register. In order to implement it, we actually use the operator $-D$, so that we end up applying the reflection R up to an unimportant global phase of -1 . Thus, we need to flip the phase of state $|0\rangle_2$. This is accomplished with a controlled- Z gate.

For the sake of generalizing later, let us instead use the phase gate $P(\theta)$:

$$P(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}, \quad (19)$$

which for the standard phase $\theta = \pi$ recovers the Z gate. When this gate is controlled, the target qubit can be whatever qubit since the phase $e^{i\theta}$ of the state cannot be attributed to a particular qubit. Thus, we use a representation in which all the qubits represent the state to whom we want to apply the phase $e^{i\theta}$, and they control the phase gate on a ghost target qubit [32], as shown in Fig. 1.

For the operator D , all the qubits in the second register control the gate $P(\pi)$ with the condition of all qubits being in state $|0\rangle$. Thus, the circuit for Szegedy's quantum walk operator U_s can be decomposed in a general manner as shown in Fig. 2. For the double operator W_s we would just apply twice this circuit.

The final problem is to provide a circuit for the update operator V . In general, we need to codify the N^2 transition probabilities of matrix G , so that the complexity of the circuit would scale at least as $O(N^2)$ for a general dense transition matrix. For sparse graphs, where there are few transition probabilities, this operator could be implemented efficiently [30]. Moreover, even if the matrix is dense but the graph has some symmetry properties, this operator could also be implemented efficiently. Different implementations of the update operators for graphs with symmetry can be found [29]. Supposing an efficient implementation of V is provided, in the following subsections we will show how to modify the standard circuit to include the phase extensions.

B. Adding APR phases

Let us consider the vertex-phased Szegedy's model, where the reflection R is substituted by the phase rotation operator $R(\vec{\theta})$ in (11) including local APR. We diagonalize it using the

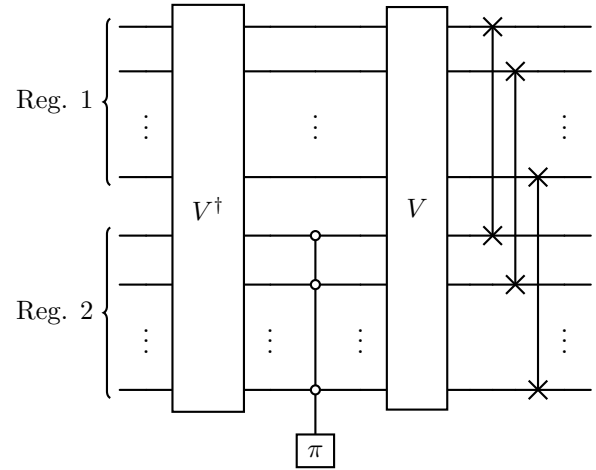


FIG. 2. Quantum circuit decomposition of the standard single Szegedy unitary evolution operator $U_s = S_w R$. Each register has n qubits for a graph with $N = 2^n$ nodes. The reflection $R = V D V^\dagger$, where D is a diagonal operator that implements a reflection around state $|0\rangle_2$ in the second register.

update operator V as before, so that $D(\vec{\theta}) = V^\dagger R(\vec{\theta}) V$. Then

$$\begin{aligned} D(\vec{\theta}) &= \sum_{i=0}^{N-1} (1 - e^{i\theta_i}) V_i^\dagger |\psi_i\rangle \langle \psi_i| V_i - \mathbb{1} \\ &= \sum_{i=0}^{N-1} (1 - e^{i\theta_i}) |i\rangle_1 \langle i| \otimes |0\rangle_2 \langle 0| - \mathbb{1} \\ &= \sum_{i=0}^{N-1} |i\rangle_1 \langle i| \otimes [(1 - e^{i\theta_i}) |0\rangle_2 \langle 0| - \mathbb{1}_2]. \end{aligned} \quad (20)$$

In the case of a global APR phase θ , we could again factorize it as a diagonal operator acting on the second register the same as in Eq. (18), which would correspond to a phase rotation around the state $|0\rangle_2$. This would have eigenvalue $-e^{i\theta}$ for $|0\rangle_2$ and -1 for the rest. The circuit would implement $-D(\theta)$ instead, so that applies a phase $e^{i\theta}$ to the state $|0\rangle_2$ letting the rest of the computational basis unchanged. Then, the circuit would be the same as the one shown in Fig. 2 but changing the phase π by the general phase θ .

However, if we have local APR phases, then the diagonal operator cannot be factorized. In this case the operator has the form of a uniformly controlled gate [33,34], which applies a different phase rotation around the state $|0\rangle_2$ depending on the state of the first register. Thus, in general it corresponds to N different phase rotations acting on the second register controlled by the N states of the computational basis of the first register. The quantum circuit for this case is shown in Fig. 3.

The construction of this circuit is in general inefficient, since we need N controlled-phase rotation gates. However, there can be cases where a more efficient circuit is possible. The local APR phases can be distributed in some manner that the same controlled phase rotation can be used to apply it to a bunch of states at the same time. For example, if we have the same phase θ_e for even nodes, and the same phase θ_o for odd nodes, then they can be implemented using only

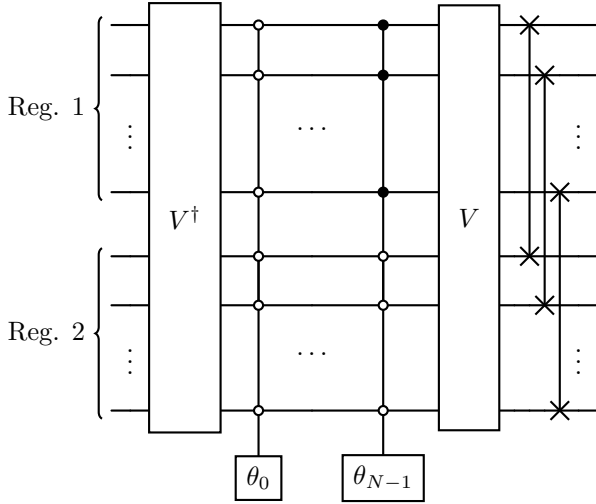


FIG. 3. Quantum circuit decomposition of the single unitary operator $U_s(\vec{\theta}) = S_w R(\vec{\theta})$ for the vertex-phased Szegedy's quantum walk. Each register has n qubits for a graph with $N = 2^n$ nodes. The phase rotation operator $R(\vec{\theta}) = VD(\vec{\theta})V^\dagger$, where $D(\vec{\theta})$ is a diagonal operator that implements N different phase rotations around state $|0\rangle_2$ in the second register. Thus, this circuit needs N phase rotations controlled by the first register, from state $|0\rangle_1$ with all qubits in $|0\rangle$ to state $|N - 1\rangle_1$ with all qubits in $|1\rangle$.

two controlled-phase rotations. In this case they would be controlled by the last qubit of the first register, which indicates the parity of the state.

Another important case where an efficient implementation is possible occurs when most of the nodes have the same phase θ , and there is a small set \mathcal{M} of special nodes with a different phase θ_k , where $k \in \mathcal{M}$. If we have a number of special nodes $M \ll N$, then there is a method for constructing the circuit only with $M + 1$ phase rotation gates. First, we apply a phase rotation of θ to the second register, with no control by the first register. Thus, a global APR phase θ is applied for all nodes. After that, we apply a phase rotation for each of the M special nodes, controlled by the first register being in the corresponding state. Let θ_k be the phase we want to apply to node k , which is a special node. Since we have already applied a phase θ to that node, the particular phase rotation for that node must now apply $\theta_k - \theta$, so that it deletes the previous rotation with θ . In Fig. 4 we show an example where node $N - 1$ is the only special node. Note that in general each of the special nodes can have a different local APR phase.

C. Adding link phases

We have seen that after diagonalizing the reflection operator R , the diagonal operator D , even with APR phases, does not depend on the vectors $|\psi_i\rangle$. Thus, it will not depend on the link phases φ_{ij} . In this case, the modification must be done to the update operator V . The diagonalization process is the same as before, but using the modified update operator $V(\varphi)$, such that $V(\varphi)|i\rangle_1|0\rangle_2 = |\psi_i(\varphi)\rangle$.

Provided a circuit implementation of the update operator V , a naive method to construct the operator $V(\varphi)$ would be to first apply V and then apply a controlled-phase gate (see

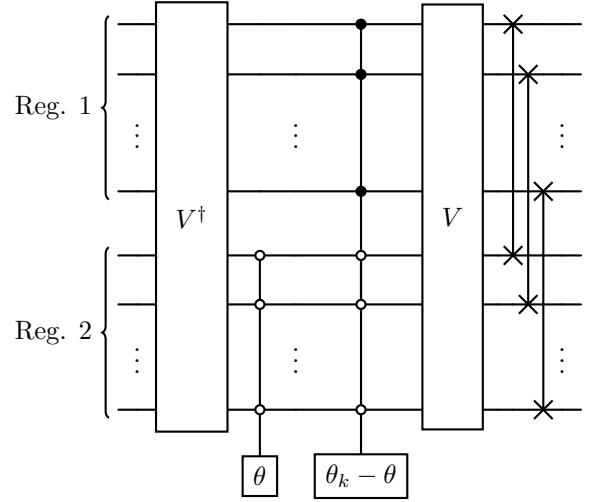


FIG. 4. Quantum circuit decomposition of the single unitary operator $U_s(\vec{\theta}) = S_w R(\vec{\theta})$ for the vertex-phased Szegedy's quantum walk on a graph where most nodes have the same local APR phase θ , and M special nodes have a different local APR phase. Each register has n qubits for a graph with $N = 2^n$ nodes. A phase rotation of θ is applied to all nodes, and then M different phase rotations controlled by the first register are applied for the special nodes. In this example, only node $N - 1$ has a different phase θ_k , so that the controlled phase rotation needs all qubits in the first register to be in state $|1\rangle$.

Fig. 1) for each computational basis state whose link phase φ_{ij} is different to 0. In general we would have N^2 link phases, so this method would be inefficient unless there are very few link phases different to 0, or they are distributed in a manner that several phases with the same value can be applied using a common controlled gate.

Another approach consists of constructing a new circuit for the modified update operator based on the standard one. The same as with the transition probabilities, if the link phases matrix φ is sparse or the distribution of phases in the graph follows some symmetry patterns, then we could construct an efficient operator $V(\varphi)$. An example of update operator with phases distributed with certain symmetry in cycles will be constructed in Sec. VC.

IV. EQUIVALENCE WITH THE COINED QUANTUM WALK

It is known that given a standard Szegedy's quantum walk, it can always be cast into the coined model, but only a restricted set of coins can be cast into Szegedy's model [24]. In this section we will review the coined quantum walk formulation and the equivalence with Szegedy's quantum walk in order to show how the phase extensions allows the graph-phased model to host a wider set of equivalent coins.

A. Coined quantum walk formulation

Although Szegedy's quantum walk takes place on weighted graphs, coined quantum walks are defined on undirected graphs. In Fig. 5(a) we show an example of a weighted

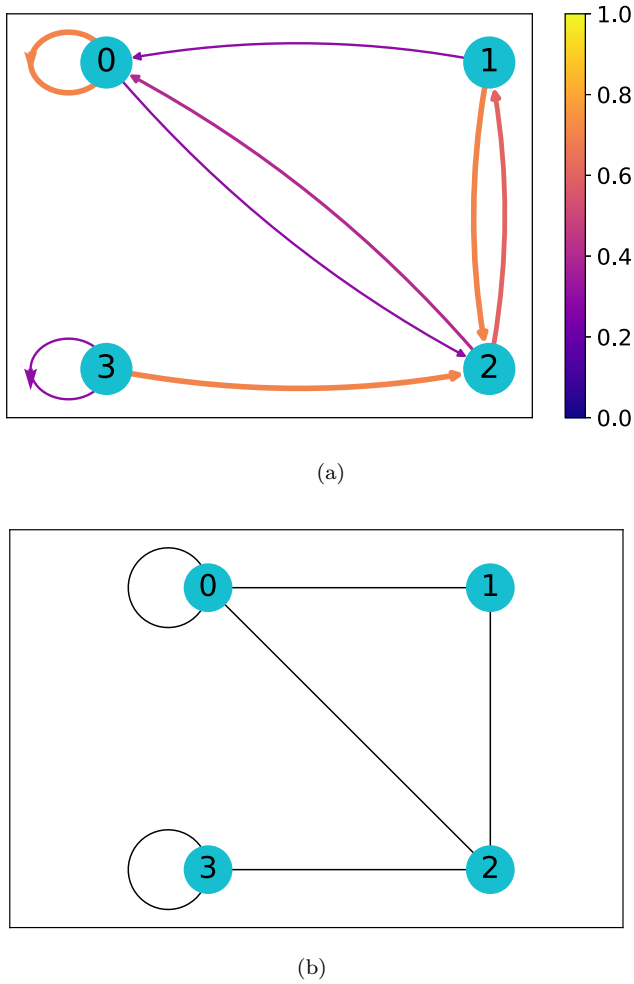


FIG. 5. (a) Weighted graph with $N = 4$ nodes whose transition matrix is given in (21). The weights of the directed edges are represented by the color map and are proportional to the width of the edges. Missing edges have a null transition probability. (b) Undirected graph with $N = 4$ nodes representing the backbone of the weighted graph in (a). The adjacency matrix is in (22). Note that although the weighted graph is asymmetric, the underlying undirected graph is symmetric. For example, in the weighted graph there is a directed edge from node 1 to node 0, but there is no directed edge from 0 to 1. This asymmetry is not reflected in the undirected graph, where a symmetric edge is present. The graphs have been plotted using the python library NetworkX [35].

graph with four nodes whose associated transition matrix is

$$G = \begin{pmatrix} 0.7 & 0.3 & 0.4 & 0 \\ 0 & 0 & 0.6 & 0 \\ 0.3 & 0.7 & 0 & 0.7 \\ 0 & 0 & 0 & 0.3 \end{pmatrix}. \quad (21)$$

The underlying undirected graph corresponds to the backbone of the graph, as it is shown in Fig. 5(b). Although the weighted graph is formed by directed edges with an associated transition probability, the undirected graph is formed by undirected edges, so they do not have an arrow indicating the direction. Moreover, note that the underlying undirected graph has an edge if there is a directed edge in the weighted

graph regardless of the direction. For example, between nodes 0 and 1 there is a directed edge from node 1 to node 0, but there is no directed edge from 0 to 1 since the corresponding transition probability is $G_{10} = 0$. However, this asymmetry is not reflected in the undirected graph, where an undirected edge connects nodes 0 and 1.

The adjacency matrix A of an undirected matrix is a symmetric matrix such that $A_{ij} = 1$ if there is an edge between nodes i and j . Otherwise $A_{ij} = 0$. Let us denote G^S as the result of symmetrizing the transition matrix G . Then A has non-null elements in the same positions as G^S . In this case the adjacency matrix is

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}. \quad (22)$$

Although self-loops are not usually considered, for the sake of establishing a relationship with Szegedy’s quantum walk we also need to consider self-loops. Thus, we provide a construction of coined quantum walks on arbitrary graphs [18] considering also self-loops. This construction uses the arc notation [36], so that the computational basis is formed by states representing directed edges of the graph. For undirected graphs we consider that each edge between two different nodes has associated two directed edges, whereas a self-loop only has a single directed edge. Thus, the Hilbert space where the coined quantum walk takes place is

$$\mathcal{H}_C = \text{span}\{|(i, j)\rangle : A_{ij} = 1\}, \quad (23)$$

where (i, j) represents the directed edge pointing from node i to node j . The dimension is $2|E_N| + |E_L|$, where E_N is the set of undirected edges between different nodes, and E_L the set of self-loops. For the example of Fig. 5(b) $E_N = 4$ and $E_L = 2$. Note that due to the symmetry of the undirected graph, if $|(i, j)\rangle$ is in the Hilbert space, then $|(j, i)\rangle$ is, too.

The unitary evolution operator U_c of the coined quantum walk is defined as follows:

$$U_c := S_c C, \quad (24)$$

where S_c is a shift operator and C the coin operator. For a general coined quantum walk on an undirected graph, the flip-flop shift operator is usually used. Its action is defined by [37]:

$$S_c |(i, j)\rangle = |(j, i)\rangle. \quad (25)$$

There are other possibilities for the shift operator, as the moving shift operator. However, in the end both quantizations are equivalent via a redefinition of the coin [18].

The coin operator C is expressed as:

$$C = \bigoplus_{i=0}^{N-1} C_i, \quad (26)$$

so that in general there is a different coin C_i associated to each of the N nodes of the graph. Thus, depending on the node where the walker is, which is represented by the tail of the directed edge, the coin acts differently. Moreover, in general each C_i matrix has a different dimension. Each C_i operator is a d_i -dimensional unitary operator, where d_i is the degree of

node i and acts in the subspace

$$\mathcal{H}_C^i = \text{span}\{|(i, k)\rangle : A_{ik} = 1\}, \quad (27)$$

which is formed by the d_i directed edges departing from node i . Usually the Grover diffusion operator [38] is used as coin [18], so that the matrix elements are

$$(C_i)_{ab} = \frac{2}{d_i} - \delta_{ab}. \quad (28)$$

B. Relation between coined and Szegedy’s quantum walks

A priori we cannot establish a relationship between coined quantum walks and Szegedy’s model because both Hilbert spaces have different dimensions. Moreover, the coined space \mathcal{H}_C cannot be expressed as a tensor product since each node has in general a different degree. We can augment the coined space into an N^2 -dimensional space considering also the edges (i, j) that are not present in the undirected graph. We call \mathcal{H}_C^A to this augmented space, and \mathcal{H}_C is a subspace where the walk takes place. The augmented space now is isomorphic to the Hilbert space of Szegedy’s quantum walk \mathcal{H}_S in (2). Let \mathcal{F} be the isomorphism between both Hilbert spaces:

$$\mathcal{F} : \mathcal{H}_C^A \rightarrow \mathcal{H}_S. \quad (29)$$

Using it on \mathcal{H}_C we can find a reduced subspace of \mathcal{H}_S where the equivalent Szegedy’s quantum walk should take place. Let us denote it as \mathcal{H}_S^R , so that $\mathcal{F} : \mathcal{H}_C \rightarrow \mathcal{H}_S^R$ and

$$\mathcal{H}_S^R = \text{span}\{|i\rangle_1 |j\rangle_2 : A_{ij} = 1\}. \quad (30)$$

Now we need to define an augmented N^2 -dimensional coined walk operator compatible with the augmented space. We define it directly acting on \mathcal{H}_S^A

$$U_c^A = S_c^A C^A. \quad (31)$$

For this operator to be equivalent to the coined walk in \mathcal{H}_C , it must have the same action as U_c in the reduced subspace \mathcal{H}_S^R after the application of the isomorphism and leaving it invariant. For the shift operator we can take $S_c^A = S_w$. From (25) it is trivial that the swap operator S_w acts equivalently to S_c in the reduced subspace. Moreover, \mathcal{H}_S^R is trivially invariant under S_w due to the symmetry of the undirected adjacency matrix A .

With regard to the coin, we can define a coin operator in \mathcal{H}_S^A as:

$$C^A = \sum_{i=0}^{N-1} |i\rangle_1 \langle i| \otimes C_i^A, \quad (32)$$

where now C_i^A is a N -dimensional unitary operator that acts nontrivially in

$$\mathcal{H}_S^i = \text{span}\{|k\rangle_2 : A_{ik} = 1\}, \quad (33)$$

conditioned by the first register being in state $|i\rangle_1$. In order to the coin C^A be equivalent to the coin C , the action in the reduced subspace must be provided by the isomorphism \mathcal{F} as

$$C^A |i\rangle_1 |j\rangle_2 := \begin{cases} \mathcal{F}(C|(i, j)) & \text{if } |i\rangle_1 |j\rangle_2 \in \mathcal{H}_S^R, \\ -|i\rangle_1 |j\rangle_2 & \text{if } |i\rangle_1 |j\rangle_2 \in (\mathcal{H}_S^R)^\perp. \end{cases} \quad (34)$$

We need the reduced subspace to be invariant under the action of C^A . The action on the states that are perpendicular

to \mathcal{H}_S^R is irrelevant as long as it does not mix the subspaces. Thus, we have freedom defining the action on the orthogonal complement. For the sake of establishing later an equivalence with Szegedy’s quantum walk, we define this action as the $-\mathbb{1}$ operator.

Since we have defined $S_c^A = S_w$, the equivalence with the graph-phased Szegedy’s quantum walk needs the operator $R(\vec{\theta}, \varphi)$ in (13) to be the coin operator C^A . For a state $|i\rangle_1 |j\rangle_2$ in the orthogonal complement of \mathcal{H}_S^R we have $A_{ij} = 0$, which implies that in the transition matrix $G_{ij} = G_{ji} = 0$. Thus, this state is perpendicular to all the $|\psi_i(\varphi)\rangle$ states in (8) and the action of $R(\vec{\theta}, \varphi)$ is just $-\mathbb{1}$. Thus, this subspace is invariant under $R(\vec{\theta}, \varphi)$, and due to unitarity \mathcal{H}_S^R also is. Therefore, we are closer to establish $C^A = R(\vec{\theta}, \varphi)$ and $U_c^A = U_s(\vec{\theta}, \varphi)$.

The last step is to find the expression of the individual coins C_i^A in (32). Let us rewrite the $|\psi_i(\varphi)\rangle$ states as:

$$|\psi_i(\varphi)\rangle = |i\rangle_1 \otimes |\omega_i(\varphi)\rangle_2, \quad (35)$$

where

$$|\omega_i(\varphi)\rangle_2 = \sum_{k=0}^{N-1} e^{i\varphi_{ik}} \sqrt{G_{ki}} |k\rangle_2. \quad (36)$$

Taking into account that $\mathbb{1} = \mathbb{1}_1 \otimes \mathbb{1}_2$, the completeness relation of the identity for the first register, and substituting the expression for the $|\psi_i(\varphi)\rangle$ states in (13), we have:

$$R(\vec{\theta}, \varphi) = \sum_{i=0}^{N-1} |i\rangle_1 \langle i| \otimes [(1 - e^{i\theta_i})|\omega_i(\varphi)\rangle_2 \langle \omega_i(\varphi)| - \mathbb{1}_2]. \quad (37)$$

Looking at the expression for the coin C^A in (32), we can identify the individual coins C_i^A with the phase rotations in the second register:

$$C_i^A = (1 - e^{i\theta_i})|\omega_i(\varphi)\rangle_2 \langle \omega_i(\varphi)| - \mathbb{1}_2. \quad (38)$$

When the augmented coins can be expressed in this form, then the coined quantum walk is equivalent to Szegedy’s quantum walk. The coins codify the transition probabilities G_{ji} and also the extended phases, and we have finally

$$\mathcal{F}(U_c|(i, j)) = U_s(\vec{\theta}, \varphi)|i\rangle_1 |j\rangle_2. \quad (39)$$

Note that \mathcal{H}_S^R is invariant under Szegedy’s quantum walk and it is actually where any Szegedy’s quantum walk takes place for a weighted graph, since it contains all the directed edges with non-null probability and their swapped versions, which appear due to the swap operator. Thus, Szegedy’s quantum walk is indeed quantized also on the undirected graph. In the case that for an undirected edge one of the two transition probabilities is null, this is taken into account by the coin. Nevertheless, this ghost directed edge plays also a role in the quantum state and cannot be removed.

The question that remains is what are the conditions that must be satisfied so that a coin can be expressed as in (38). In the case that we are provided with a Szegedy’s quantum walk, we can always define the coins that way, so that all Szegedy’s quantum walk that come from the quantization of a classical Markov chain with transition matrix G and extended phases $\vec{\theta}$ and φ can be cast into the coined model. In the case that the weighted graph is obtained normalizing the columns of the

adjacency matrix of an undirected graph, the equivalent coin is the Grover coin [22,25]. Thus, for arbitrary weighted graphs Szegedy's quantum walk is a generalization of Grover's quantum walk [39].

If on the contrary we are provided with a coined quantum walk, then only a restricted set of coins satisfy Eq. (38). From it we can formulate the following lemmas about the conditions that must be satisfied by a coined walk to be cast into Szegedy's quantum walk. Let us start with the standard model.

Lemma 1. Given a coined quantum walk U_c with a set of coin operators C_i of dimension d_i , there exists an equivalent standard Szegedy's quantum walk U_s if and only if for each coin C_i there are $d_i - 1$ eigenvalues -1 , and a single eigenvalue $+1$ whose eigenvector has real non-negative amplitudes.

In this case the coin must be a reflection around the state $|\omega_i\rangle$, so that $|\omega_i\rangle$ is an eigenstate with eigenvalue $+1$, and the rest of eigenvalues are -1 . Moreover, all the amplitudes of $|\omega_i\rangle$ are real positive or zero. This condition for the coins to be able to be cast into the standard Szegedy's model is in concordance with the conditions found in the literature [24]. Note that since eigenvectors that differ in a global phase are equivalent, the actual condition for the eigenvector is that there are no relative phases between the amplitudes.

Lemma 2. Given a coined quantum walk U_c with a set of coin operators C_i of dimension d_i , there exists an equivalent link-phased Szegedy's quantum walk $U_s(\varphi)$ if and only if for each coin C_i there are $d_i - 1$ eigenvalues -1 , and a single eigenvalue $+1$.

In this case the coin must be a reflection around the state $|\omega_i(\varphi)\rangle$. Again all the eigenvalues must be -1 except for the eigenstate $|\omega_i(\varphi)\rangle$, which is $+1$. However, thanks to the link phases this eigenstate can have any complex amplitudes. So that this model can host a bigger set of coins.

Lemma 3. Given a coined quantum walk U_c with a set of coin operators C_i of dimension d_i , there exists an equivalent vertex-phased Szegedy's quantum walk $U_s(\vec{\theta})$ if and only if for each coin C_i there are $d_i - 1$ eigenvalues -1 , and a single eigenvalue $-e^{i\theta_i}$ whose eigenvector has real non-negative amplitudes.

If we add APR phases, then the coin becomes a phase rotation operator. Thus, $|\omega_i\rangle$ is an eigenstate with eigenvalue $-e^{i\theta_i}$, and the rest of eigenvalues are -1 . Note that if we would only have a model with global APR, then all the coins should have the same eigenvalue different to -1 . However, thanks to the local APR phases introduced in this work, each node can have a coin with different eigenvalues.

Lemma 4. Given a coined quantum walk U_c with a set of coin operators C_i of dimension d_i , there exists an equivalent graph-phased Szegedy's quantum walk $U_s(\vec{\theta}, \varphi)$ if and only if for each coin C_i there are $d_i - 1$ eigenvalues -1 , and a single eigenvalue $-e^{i\theta_i}$.

Considering both link phases and local APR phases the eigenstate $|\omega_i(\varphi)\rangle$ can have any complex amplitude at the same time that the eigenvalue is arbitrary. Therefore the set of coins that can be cast is maximal, and encompasses the previous cases.

All these equivalences have been established between the coined walk and the single step Szegedy's operator U_s . If

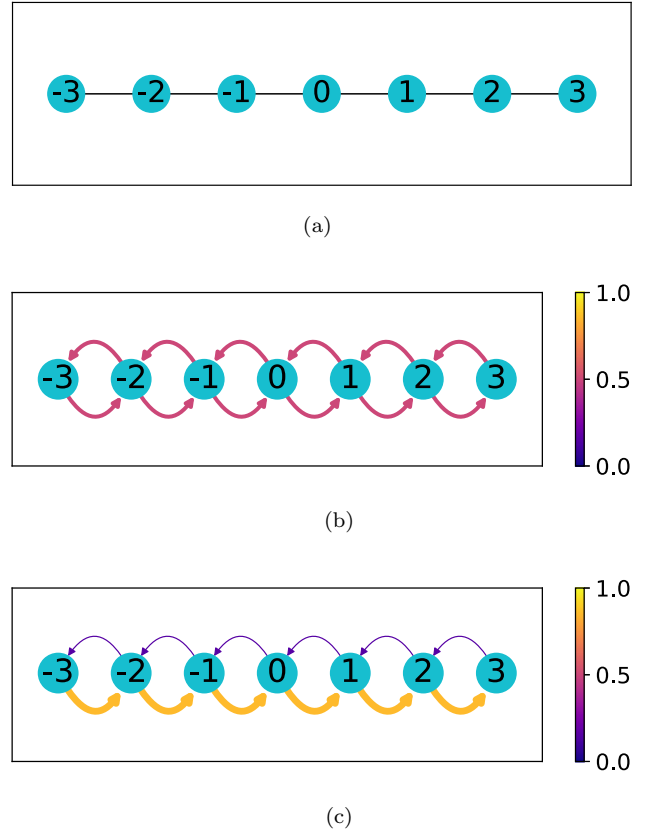


FIG. 6. (a) Undirected graph for an infinite one-dimensional line. For the sake of simplicity only seven nodes have been represented. (b) Weighted graph obtained by normalizing the adjacency matrix of the undirected graph in (a), so that all directed edges have the same transition probability. (c) Weighted graph associated to the coined walk with the Hadamard coin, where the probability to the right is bigger than to the left. The weights of the directed edges are represented by the colormap, and are proportional to the width of the edges. The graphs have been plotted using the python library NetworkX [35].

we can cast a coined walk into Szegedy's in this case, then trivially it can also be cast into Szegedy's model considering the original double operator W_s , so that one step of Szegedy's quantum walk would be equivalent to two steps of the coined walk, being the equivalent operator U_c^2 . Moreover, there can be cases where a coin cannot be cast into Szegedy's model considering the single step operator U_s , but it can be cast if we consider the double step operator W_s instead. We will show an example for the -1 coin in Sec. VIA.

V. EXAMPLE ON THE LINE

So far we have obtained the conditions for establishing an equivalence between Szegedy's model and coined quantum walks. In this section we will show how each model can be cast into the other using examples on the one-dimensional (1D) line. The undirected graph that shows the backbone of the line is shown in Fig. 6(a). The set of nodes ranges from $-\infty$ to ∞ , although we only show seven nodes for the sake of simplicity.

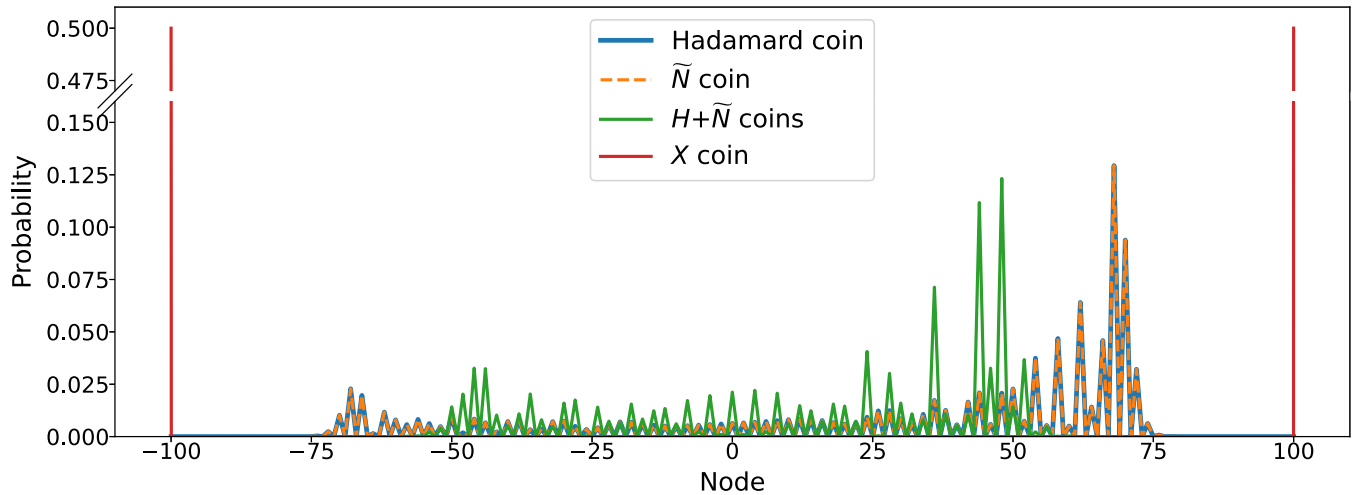


FIG. 7. Probability distribution results of the coined quantum walk simulations on a one-dimensional line using the X coin (red), the Hadamard coin (blue), the \tilde{N} coin (orange), and the coined walk with the Hadamard coin for even nodes and the \tilde{N} coin for odd nodes (green). Note that for the Hadamard and \tilde{N} coins the curves overlap. The initial state is $|\psi_0\rangle = (|0\rangle_1|1\rangle_2 + |0\rangle_1|-1\rangle_2)/\sqrt{2}$ and the unitary evolution has been applied 100 time steps.

A. Standard Szegedy's model and coined quantum walk

The naive transition matrix used for standard Szegedy's quantization is obtained by normalizing the adjacency matrix, so that a walker in node i has a probability of $1/d_i$ for jumping to each of the neighbor nodes. In this case the walker has a probability of one half for jumping either to the right or to the left. The associated weighted graph is shown in Fig. 6(b), and the transition matrix is

$$(G_u)_{ji} = \frac{1}{2}\delta_{j-1,i} + \frac{1}{2}\delta_{j+1,i}, \quad (40)$$

where the subindex u makes reference to the fact that it is obtained from the undirected graph. Since the degree of each node is $d_i = 2$, all coins C_i are going to be two-dimensional matrices. Moreover, all coins will be the same. By convention the 2D basis is ordered so that the first element corresponds to the directed edge pointing to the right, and the second one pointing to the left. Since there are no extended phases, the coins correspond to reflections around the state $|\omega_i\rangle = (1, 1)^T/\sqrt{2}$, which is obtained taking the square root of the columns of G_u and expressing it directly in the 2D basis of the coin. The rest of elements in the augmented space are always null. This is an eigenvector with eigenvalue $+1$, and the other eigenvalue must be -1 . Thus, it trivially corresponds to the Pauli X operator, which is the Grover coin in 2D.

To compare different quantum walks, we need an initial state for the simulation. We take as initial state $|\psi_0\rangle = (|0\rangle_1|1\rangle_2 + |0\rangle_1|-1\rangle_2)/\sqrt{2}$, which represents the walker at node 0. The probability distribution after 100 time steps is shown in Fig. 7 in red. The probability distribution moves ballistically, so that there are no probabilities at intermediate nodes, and the walker reach nodes 100 and -100 with the same probability.

A different quantization of the line walk is usually done by means of the Hadamard coin [6]. Since this coin puts in an equal superposition the computational basis of the coin register in 2D, it was thought to be a sensible quantization of

the classical walk on the undirected graph with G_u . However, when casting it into a Szegedy's quantum walk we obtain an unbiased transition matrix which does not corresponds to the classical walk on the undirected line [24]. The spectrum of H is $\sigma(H) = \{+1, -1\}$, and the eigenvector for eigenvalue $+1$ is $(h_R, h_L)^T$, where

$$h_R = \frac{1}{\sqrt{4 - 2\sqrt{2}}}, \quad (41)$$

$$h_L = \frac{\sqrt{2} - 1}{\sqrt{4 - 2\sqrt{2}}}. \quad (42)$$

Thus, by Lemma 1 we can cast it into the standard Szegedy's model. The transition probabilities are obtained by the square modulus of the amplitudes h_R and h_L , so that the walker has a probability of approximately 0.85 of jumping to the right and approximately 0.15 of jumping to the left. The transition matrix is therefore

$$(G_H)_{ji} = h_R^2\delta_{j-1,i} + h_L^2\delta_{j+1,i}. \quad (43)$$

The associated weighted graph is shown in Fig. 6(c), where we observe that the walk is biased to the right. The results of the simulation for 100 time steps using the same initial state as before are shown in Fig. 7 in blue. In this case the walk is slower, not reaching nodes ± 100 . Moreover, there are probabilities at intermediate nodes, and the distribution is biased to the right.

Note that despite the fact that the Hadamard coined walk and the unbiased Szegedy's walk are two quantizations based on the undirected graph, they are not equivalent. The equivalence between both models just means that given a transition matrix we can find a set of coins that reproduce that particular Szegedy's walk. However, different coins produce different quantizations that may be equivalent to quite different weighted graphs, despite being devised from the same classical walk as in the case of the Hadamard coin.

B. Graph-phased Szegedy’s model and coined quantum walk

Now let us show a different coin that cannot be cast into the standard Szegedy’s model. We have constructed the following coin:

$$\tilde{N} = -\frac{1+i}{2} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}. \tag{44}$$

The spectrum is $\sigma(\tilde{N}) = \{-i, -1\}$ and the eigenvector for eigenvalue $-i$ is $(1, i)^T/\sqrt{2}$. Thus, by Lemma 1 it cannot be cast into the standard model, but by Lemma 4 it can be cast into the graph-phased model. In this case the eigenvalue different to -1 was $-e^{i\theta_i}$, so that the APR phase for this coin is $\theta_i = \pi/2$. Again the transition probabilities are obtained taking the square modulus of the amplitudes of the eigenvector. In this case we obtain $1/2$ for both directions, so that the transition matrix $G_{\tilde{N}}$ associated to this coin is the same as for the undirected graph G_u in (40). Since the amplitudes are complex numbers, we also need link phases. For directed edges pointing to the right the amplitude is a real positive number and the link phase is $\varphi_{i,i+1} = 0$. However, for directed edges pointing to the left the link phase is $\varphi_{i,i-1} = \pi/2$. Thus, the link phases matrix for this coin is

$$(\varphi_{\tilde{N}})_{ij} = \frac{\pi}{2} \delta_{i,j+1}. \tag{45}$$

The results of the simulation using this coin for all nodes are shown in Fig. 7 in orange. The probability distribution is the same as for the Hadamard coin. This is surprising since for the \tilde{N} coin the associated weighted graph is unbiased, being the same as for the X coin. Thus, the extended phases play an important role in the walk, being able to modify somehow the transition probabilities.

Since both the Hadamard coin and the \tilde{N} coin produce the same results when they are used on all nodes, we want to examine what happens if we apply both at the same time on different nodes. We have quantized the walk on the line using the Hadamard coin for even nodes, and the \tilde{N} coin for odd nodes. Taking into account our convention for the transition matrix indexes, it is constructed taking the even columns of G_H and the odd columns of $G_{\tilde{N}}$. For the link phases matrix φ we take the odd rows of $\varphi_{\tilde{N}}$, and the even rows are null since the Hadamard coin has no link phases. The local APR phase for even nodes is the standard one $\theta_i = \pi$ and for odd nodes is $\theta_i = \pi/2$.

The results of the simulation are shown in Fig. 7 in green. We observe that despite the fact that H and \tilde{N} produce the same results when they are global coins, the results are different when they are mixed. This explicitly shows that both coins are not actually equivalent and that the graph-phased Szegedy’s model opens up a wide range of possibilities for quantizing classical Markov chains.

C. Quantum circuit for the $H + \tilde{N}$ coined quantum walk

Before finishing this section, we want to provide a quantum circuit construction for the graph-phased Szegedy’s quantum walk equivalent to the coined walk that uses both coins at the same time. A quantum walk in the infinite 1D line can be simulated in a finite 1D cycle as long as the time steps do not surpass a threshold from which the wavefront interferes

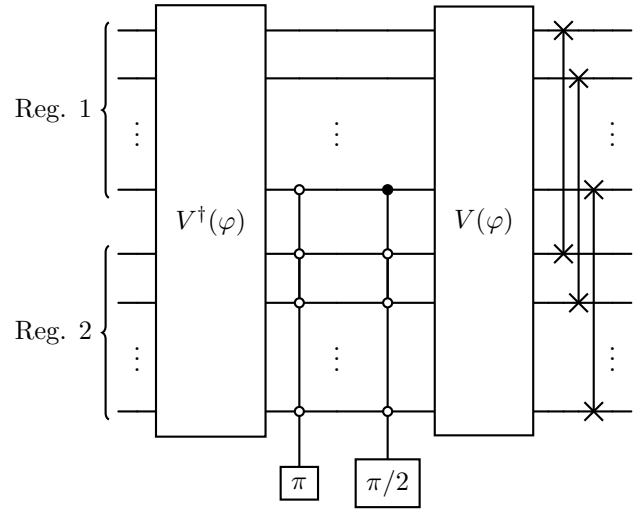


FIG. 8. Quantum circuit decomposition of the graph-phased single Szegedy unitary evolution operator $U_s = S_w R(\vec{\theta}, \varphi)$ for the coined walk with the coins H and \tilde{N} . Each register has n qubits for a cycle with $N = 2^n$ nodes. The last qubit of the first register, which determines the parity of the state, controls the application of a local APR phase π for even nodes and $\pi/2$ for odd nodes.

with itself when it completes a turn on the cycle. For a graph with $N = 2^n$ nodes, we need n qubits for each register.

In this case the local APR phases are distributed with a symmetric pattern, so that the diagonal operator $D(\vec{\theta})$ can be implemented efficiently. The last qubit of the first register indicates the parity of the node. If this control qubit is in state $|0\rangle$, then for even nodes we apply the phase rotation with $\theta = \pi$ on the second register. If on the contrary it is on state $|1\rangle$, then for odd nodes the phase rotation with $\theta = \pi/2$ is applied instead. Thus, the quantum circuit can be decomposed as shown in Fig. 8.

For the update operator $V(\varphi)$, in Fig. 9 we have constructed a circuit based on the update operator of the standard cyclic graph [29]. The update operator must act as $V(\varphi)|i\rangle_1|0\rangle_2 = |i\rangle_1|\omega_i(\varphi)\rangle_2$, where

$$|\omega_i(\varphi)\rangle_2 = \begin{cases} h_R|i+1\rangle_2 + h_L|i-1\rangle_2 & \text{if } i \text{ is even,} \\ \frac{1}{\sqrt{2}}|i+1\rangle_2 + \frac{i}{\sqrt{2}}|i-1\rangle_2 & \text{if } i \text{ is odd,} \end{cases} \tag{46}$$

where h_R and h_L are given in (41) and (42) respectively. Since we are considering a finite graph, the additions and subtractions are performed modulo N .

Let i be an even node. Then, the last qubit of the first register is in state $|0\rangle$, and the R_Y gate is applied to the first qubit of the second register, letting it in state $h_R|0\rangle_2 + h_L|1\rangle_2$. The following action of the CNOT gates and the last X gate let the second register in

$$h_R|1\rangle_2 + h_L|N-1\rangle_2. \tag{47}$$

The permutation operator \mathcal{P}^+ acting on a m -qubit system transforms the computational basis states $|x\rangle$ into $|x+1 \bmod 2^m\rangle$. This can be implemented efficiently with multi-controlled-NOT gates. The joint action of the \mathcal{P}^+ operators controlled by the first register this way corresponds to another permutation that transforms the computational basis states

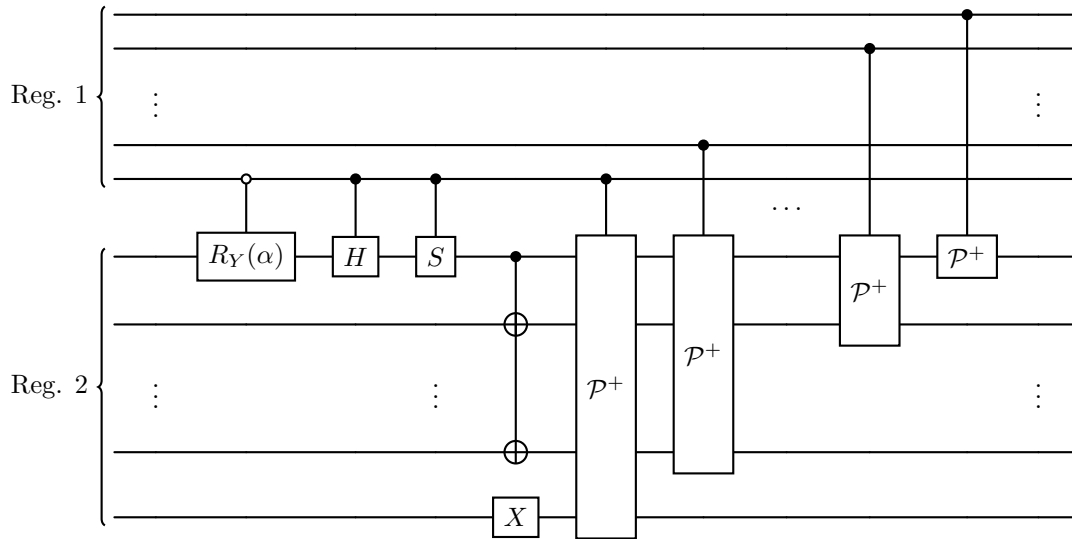


FIG. 9. Quantum circuit for the update operator $V(\varphi)$ of the graph-phased Szegedy’s walk equivalent to the coined walk with the H coin for even nodes and the \tilde{N} coin for odd nodes. The rotation matrix is $R_Y(\alpha) = \exp -i\alpha Y/2$, with Y being a Pauli matrix, and the value of the rotation angle is $\alpha = \pi/4$.

$|i\rangle_1|x\rangle_2$ into $|i\rangle_1|x + i \bmod N\rangle_2$ [29]. Therefore, when they are applied on the state in (47), the second register ends up in $|\omega_i(\varphi)\rangle_2$.

We can do an analogous reasoning for i being an odd node. In this case the last qubit of the first register is in state $|1\rangle$, so that the H and S gates are applied to the first qubit of the second register, letting it in state $(|0\rangle + i|1\rangle)/\sqrt{2}$. After the CNOT gates and the last X gate the second register ends up in

$$\frac{1}{\sqrt{2}}|1\rangle_2 + \frac{i}{\sqrt{2}}|N - 1\rangle_2, \quad (48)$$

and the controlled \mathcal{P}^+ operators permute the state into $|\omega_i(\varphi)\rangle_2$. Thus, we have proved that the circuit in Fig. 9 effectively performs the action of the update operator $V(\varphi)$.

VI. MARKING NODES WITH APR

Suppose there is a special set of nodes in a graph that we want to mark somehow so that the quantum evolution treats them in a different manner. We can mark them with a different local APR phase. A construction based on Fig. 4 would require the knowledge about who are these special nodes. However, we usually do not know them, and they are marked by a black-box function when they satisfy some conditions.

Let $f(x)$ be a classical function that decides if a particular node x satisfies some conditions, and let \mathcal{M} be the set of nodes that satisfy them. Then

$$f(x) := \begin{cases} 1 & \text{if } x \in \mathcal{M}, \\ 0 & \text{otherwise.} \end{cases} \quad (49)$$

Provided a classical circuit can be constructed for this function $f(x)$, it can be translated into quantum gates so that we can construct a quantum circuit for an oracle Q_f . This operator takes a quantum register with the information about the node x and adds the result of $f(x)$ in an ancilla qubit [40,41], such

that

$$Q_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle. \quad (50)$$

When the ancilla qubit starts in state $|0\rangle$ it just stores the result of $f(x)$.

For the sake of simplicity let us consider that we want to mark some nodes in a standard Szegedy’s walk so that they evolve with a different local APR phase when they satisfy the conditions of an oracle Q_f . We can construct a quantum circuit leveraging the update operator V of the standard walk, as shown in Fig. 10. In this case we make use of an ancilla qubit that takes the value of $f(x)$ to later control the phase rotation that deletes the global phase π and let the local APR phase θ_k for marked nodes. Thus, a marked node $k \in \mathcal{M}$ evolves with the local APR phase θ_k , instead of the standard phase π . The second oracle is used to uncompute the action of the first one, so that the ancilla qubit returns to its original state and can be traced out. Note that although we have used only an oracle for marking nodes, we can use multiple oracles to mark different sets of nodes with different phases, so that they evolve differently depending on the different conditions imposed by the oracles.

A. Searching marked nodes

The link phases have been previously used to search for marked arcs on graphs [42]. In this section we show that an important use case of the local APR phases is also the search problem. Nevertheless, in this case we want to use a quantum walk algorithm to search for marked nodes instead of marked arcs.

The first quantum walk search algorithm used the Grover coin for unmarked nodes and a different coin for marked nodes [5]. In particular, it was studied the case of the $-\mathbb{1}$ coin for marking nodes, finding an algorithm with quadratic speedup with respect to classical ones. This coin cannot be cast into the standard Szegedy’s walk since all the eigenvalues

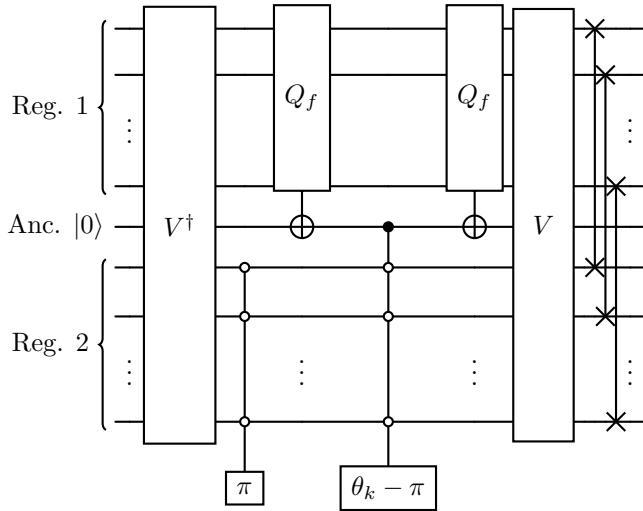


FIG. 10. Quantum circuit decomposition for marking nodes with local APR phases in a standard Szegedy's quantum walk. The first oracle turns the ancilla qubit into $|1\rangle$ if the first register represents a marked node. Then the ancilla applies a phase rotation on the second register deleting the global APR phase π and letting the local APR phase θ_k . Finally, the second oracle uncomputes the ancilla so that it returns to its original state and can be traced out.

are -1 . However, by Lemma 3 it can be cast into the vertex-phased Szegedy's model using $\theta_k = 0$ for marked nodes.

A different approach for quantum walk search algorithms is based on Szegedy's quantum walk with absorbing vertices [7,18]. Given a transition matrix G for a graph, nodes are marked turning them into sinks, so that they have a self-loop that does not allow the walker to escape once the walker steps on them. Let us denote the modified transition matrix as G' . This is obtained deleting the columns of the marked nodes and allocating a diagonal 1 element, so that

$$G'_{ji} = \begin{cases} G_{ji} & \text{if } i \notin \mathcal{M}, \\ \delta_{ji} & \text{if } i \in \mathcal{M}. \end{cases} \quad (51)$$

It was previously stated that marking with the -1 coin is equivalent to marking with absorbing vertices, although the proof was done assuming that the initial state has no amplitudes associated to self-loops [25]. In this work we will show that both walks are indeed not equivalent for arbitrary states, although they are equivalent when the double step Szegedy operator is considered. Moreover, we make the proof for general graphs, so that the global coin for unmarked nodes does not have to be the Grover coin.

On the one hand, the evolution operator for Szegedy's quantum walk using the -1 coin for marking nodes is $U_s(\vec{\theta}) = S_w R(\vec{\theta})$ and is obtained setting $\theta_i = \pi$ for unmarked nodes and $\theta_i = 0$ for marked nodes in (11). On the other hand, the evolution operator for Szegedy's quantum walk with absorbing vertices U'_s is obtained from the standard operator in (6) using the transition matrix G' . If we express the reflection operators in a coined form in a similar manner as in Eq. (37), and separate the sums for marked and unmarked nodes, then

the expressions of these operators are the following:

$$U_s(\vec{\theta}) = S \sum_{i \notin \mathcal{M}} |i\rangle_1 \langle i| \otimes (2|\omega_i\rangle_2 \langle \omega_i| - \mathbb{1}_2) + S \sum_{i \in \mathcal{M}} |i\rangle_1 \langle i| \otimes (-\mathbb{1}_2), \quad (52)$$

$$U'_s = S \sum_{i \notin \mathcal{M}} |i\rangle_1 \langle i| \otimes (2|\omega_i\rangle_2 \langle \omega_i| - \mathbb{1}_2) + S \sum_{i \in \mathcal{M}} |i\rangle_1 \langle i| \otimes (2|i\rangle_2 \langle i| - \mathbb{1}_2). \quad (53)$$

A priori both operators are not equivalent, since for marked nodes $U_s(\vec{\theta})$ applies the -1 coin, whereas U'_s applies a reflection around the self-loop state of the marked node.

We need to calculate how these operators act on the different states of the computational basis in order to examine properly the equivalence. For both operators the first term, corresponding to unmarked nodes, is the same. This is so because the APR phase for unmarked nodes is $\theta_i = \pi$, and the corresponding columns of G and G' are also the same. Thus, the action on a computational basis state $|i\rangle_1 |j\rangle_2$ with $i \notin \mathcal{M}$ is the same.

Now let us consider a state $|i\rangle_1 |j\rangle_2$ with $i \in \mathcal{M}$. In this case the action is given by the second terms of the operators. If $i \neq j$, then the action is $-S$ in both cases, so it is the same. However, if $i = j$, then the action of $U_s(\vec{\theta})$ is $-S$, but the action of U'_s is $+S$, so that $U_s(\vec{\theta})$ acts as $-U'_s$.

Let us define \mathcal{I}_{ML} as the subspace spanned by the marked self-loop states of the computational basis, i.e.,

$$\mathcal{I}_{\text{ML}} = \text{span}\{|i\rangle_1 |i\rangle_2 : i \in \mathcal{M}\}. \quad (54)$$

This space is an eigenspace for both operators, with eigenvalue $+1$ for U'_s and -1 for $U_s(\vec{\theta})$. Thus, it and its orthogonal complement are invariant subspaces. We can then factorize the action as follows:

$$U_s(\vec{\theta})|i\rangle_1 |j\rangle_2 = \begin{cases} -U'_s|i\rangle_1 |j\rangle_2 & \text{if } |i\rangle_1 |j\rangle_2 \in \mathcal{I}_{\text{ML}}, \\ U'_s|i\rangle_1 |j\rangle_2 & \text{if } |i\rangle_1 |j\rangle_2 \in \mathcal{I}_{\text{ML}}^\perp. \end{cases} \quad (55)$$

Although both operators are not the same, since both subspaces are invariant the action is equivalent when the initial state is in $\mathcal{I}_{\text{ML}}^\perp$. This corresponds to a state that has no amplitudes related to self-loops, so that the previously known equivalence in this case holds [25]. Nevertheless, note that for general states with self-loops, the only difference will be a relative phase -1 in the amplitudes of the self-loops. This relative phase plays no role in the probabilities of measuring the nodes in the computational basis, so that the results would be the same even in this case although the quantum states are not exactly equal. However, this equivalence would not be valid in case we measured in other different basis.

Usually, in quantum walk search algorithms based on absorbing vertices the double Szegedy operator W_s is used. Since the subspaces are invariant, we can take the square of Eq. (55), obtaining that

$$W_s(\vec{\theta}) = W'_s. \quad (56)$$

Thus, for this operator both walks are totally equivalent, taking into account that one Szegedy's step with absorbing

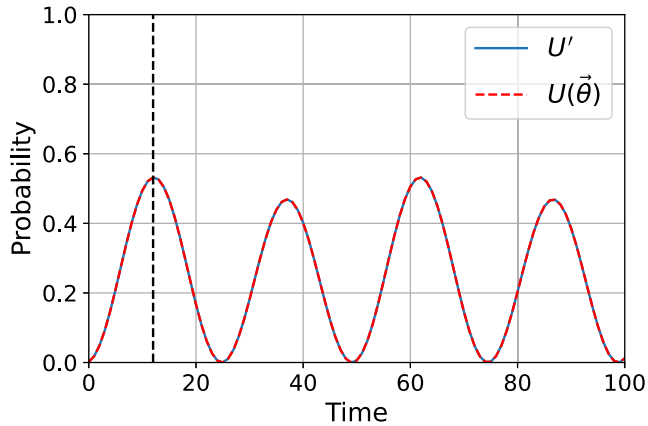


FIG. 11. Probability of measuring a marked node versus the time step of a quantum walk search algorithm on a complete graph with $N = 1000$ nodes and $M = 2$ marked nodes. The nodes have been marked with absorbing vertices in blue, and with a local APR phase $\theta_k = 0$ in red. Both algorithms are equivalent, so that the curves overlap. The vertical dashed line indicates the theoretical position of the first maximum, which occurs for $t_{\max} = 12$.

vertices corresponds actually to two steps of the coined walk with the -1 coin. Recall that by Lemma 1 the -1 coin cannot be cast into a single step of the standard Szegedy's model. However, this Lemma does not apply for the double Szegedy operator, since we have managed to cast this coin into an operator W'_s without phase extensions.

B. Example: Complete graph

A well-studied graph for quantum walk search algorithms based on absorbing vertices is the complete graph without loops [43]. The transition matrix is $G_{ji} = (1 - \delta_{ji})/(N - 1)$. The initial state of the system is constructed from the transition matrix of the original complete graph without marked nodes as

$$|\Psi^{(0)}\rangle := \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |\psi_i\rangle. \quad (57)$$

In this case the initial state has no self-loop amplitudes, so that the coined walk would be equivalent to the single step operator U'_s . Nevertheless, the quantum walk operator is usually W'_s . A maximum of probability for measuring one of the marked nodes occurs after a number of time steps given by [43]:

$$t_{\max} = \frac{\pi}{4} \sqrt{\frac{N}{2M}} - \frac{1}{4} + O\left(\sqrt{\frac{M}{N}}\right), \quad (58)$$

where M is the number of marked nodes. Thus, the algorithm has a quadratic speed-up with respect to a classical search, as the coined quantum walk.

In order to verify numerically that both models are equivalent, we have simulated the Szegedy's walk with $W_s(\vec{\theta})$ and W'_s for a graph with $N = 1000$ nodes and $M = 2$ marked nodes. The results are shown in Fig. 11, where we observe that both curves overlap. Moreover, note that the probability oscillates, despite the fact that in the underlying classical walk the walker cannot escape from absorbing vertices. This is due

to the unitary character of the evolution, and is related to the ghost links with null transition probability that play an actual role in Szegedy's quantum walk, as discussed in Sec. IV.

An efficient implementation of the update operator V for the complete graph can be found [29]. We can then use the construction in Fig. 10 to construct a quantum circuit for this quantum walk search algorithm based on an oracle, avoiding the explicit construction of the update operator V' for the graph with absorbing vertices. Although a form of constructing a circuit for V' using V and an oracle was already developed [24,44], this construction needed the application of the oracle two times, so that for the operator U'_s the oracle would be applied four times, instead of two as in our circuit. Furthermore, it needed a controlled version of the operator V , increasing the complexity of the circuit. Therefore, our implementation is simpler. Moreover, our circuit can mark with different phases so that it is not restricted only to absorbing vertices.

VII. CLASSICAL SIMULATION WITH SQUWALS

The Hilbert space of Szegedy's quantum walk \mathcal{H}_S is of dimension N^2 . Thus, in order to simulate the walk with a classical computer it is required to construct a $N^2 \times N^2$ matrix representing the unitary evolution operator. This operator has actually $O(N^3)$ non-null elements for arbitrary transition matrices, so that using a sparse representation of the matrix would provide a classical simulation algorithm scaling as $O(N^3)$ in both time and memory.

Recently we proposed a novel algorithm that avoids the explicit construction of this matrix, scaling as $O(N^2)$ for dense transition matrices. Moreover, our algorithm allows the extension with link phases and a global APR phase. Based on it, we developed a python library called SQUWALS [31]. To simulate the graph-phased Szegedy's model we need to modify our algorithm so that it also allows local APR phase extensions, which is not straightforward. In this section we show this modified algorithm, which we have used for all the simulations in this work.

The aim is to find algorithms for simulating the action of the general phase rotation $R(\vec{\theta}, \varphi)$ and the swap S_w , so that they can be used to simulate any walk operator $U_s(\vec{\theta}, \varphi) = S_w R(\vec{\theta}, \varphi)$. A generic vector in the Hilbert space \mathcal{H}_S can be written as

$$|\phi\rangle = \sum_{i,j=0}^{N-1} a_{ij} |i\rangle_1 |j\rangle_2. \quad (59)$$

We can represent this N^2 -dimensional vector as a $N \times N$ matrix Φ , whose elements are

$$\Phi_{ij} = a_{ji}. \quad (60)$$

Note that using this notation, the column index represents the first register, whereas the row index represents the second register. This is so because if we divide the vector state into blocks corresponding to each state of the computational basis of the first register, then each block corresponds to each column of the matrix state. Using this representation, the action of the swap operator S_w is simulated just transposing the

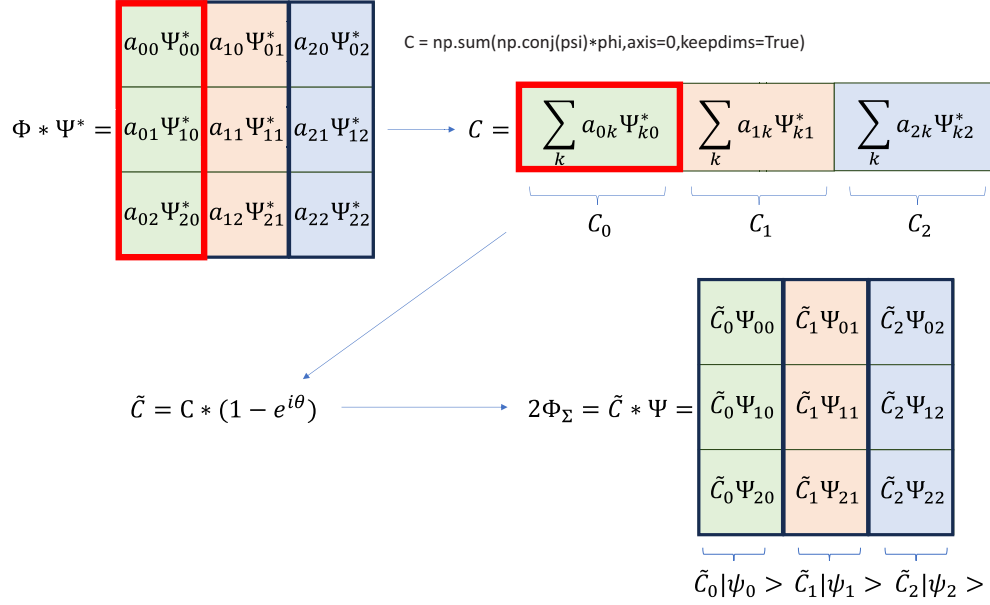


FIG. 12. How to obtain the matrix representing the state $2|\Phi\rangle_\Sigma$ in (61) for a network with $N = 3$ nodes. First, we obtain the vector C with the coefficients C_i in (64) by the element-wise multiplication between Φ and Ψ^* , and summing the elements of each column. Second, the vector C is multiplied element-wise with another vector with the factors $1 - e^{i\theta_i}$ to obtain the modified coefficients \tilde{C}_i in (65). Finally, the vector \tilde{C} is multiplied element-wise with the matrix Ψ . The result is a matrix where each column represents the non-null elements of $\tilde{C}_i|\psi_i(\varphi)\rangle$, which ends up representing the vector $2|\Phi\rangle_\Sigma$ as expressed in (66).

matrix. Nevertheless, the simulation of the phase rotation is quite more complicated.

The phase rotation operator was expressed as $R(\vec{\theta}, \varphi) = 2\Sigma(\vec{\theta}, \varphi) - \mathbb{1}$, where $\Sigma(\vec{\theta}, \varphi)$ was defined in (12). Let us denote $|\phi\rangle_\Sigma$ as the result of acting $\Sigma(\vec{\theta}, \varphi)$ on the state $|\phi\rangle$. We need to obtain this vector for the simulation. Indeed, since the factor 2 in $R(\vec{\theta}, \varphi)$ is going to cancel the factor 1/2 in $\Sigma(\vec{\theta}, \varphi)$, we calculate directly the state $2|\phi\rangle_\Sigma$:

$$2|\phi\rangle_\Sigma = 2\Sigma(\vec{\theta}, \varphi)|\phi\rangle = \sum_{i=0}^{N-1} (1 - e^{i\theta_i}) C_i |\psi_i(\varphi)\rangle, \quad (61)$$

where the coefficients C_i are obtained as $C_i = \langle \psi_i(\varphi) | \phi \rangle$. Let us rewrite the $|\psi_i(\varphi)\rangle$ states in (8) as:

$$|\psi_i(\varphi)\rangle = \sum_{k=0}^{N-1} \Psi_{ki} |i\rangle_1 |k\rangle_2, \quad (62)$$

with Ψ being a $N \times N$ matrix whose elements are

$$\Psi_{ij} = e^{i\varphi_{ji}} \sqrt{G_{ij}}. \quad (63)$$

This matrix can be obtained taking the element-wise square root of the transition matrix G , and multiplying it element-wise with the transpose of the matrix whose elements are the link phases $e^{i\varphi_{ij}}$. Using (59) and (62), the coefficients C_i are obtained as

$$C_i = \sum_{k=0}^{N-1} a_{ik} \Psi_{ki}^*, \quad (64)$$

where the asterisk denotes complex conjugation.

We can calculate the coefficients C_i in a vectorized way avoiding a *for* loop. We multiply element-wise the matrix

state Φ with the conjugate of the matrix Ψ and add across the rows of the resulting matrix, obtaining a vector C with the coefficients. In the upper panel of Fig. 12 we show an example for a network with $N = 3$ nodes.

We define the modified coefficients \tilde{C}_i as:

$$\tilde{C}_i = (1 - e^{i\theta_i}) C_i, \quad (65)$$

so that Eq. (61) can be rewritten as

$$2|\phi\rangle_\Sigma = \sum_{i=0}^{N-1} \tilde{C}_i |\psi_i(\varphi)\rangle. \quad (66)$$

The vector \tilde{C} representing the new coefficients is obtained creating a vector with the factors $1 - e^{i\theta_i}$ and multiplying it element-wise with the vector C . We can use the broadcasting feature of NumPy [45] to multiply element-wise the row vector \tilde{C} with the matrix Ψ . The resulting matrix has in the i th column the non-null elements of $|\psi_i(\varphi)\rangle$ multiplied by the coefficient \tilde{C}_i . Thus, it results in the matrix representing the vector $2|\phi\rangle_\Sigma$. The whole procedure is shown in Fig. 12 for the example with $N = 3$ nodes.

Now we have the action of the operator $2\Sigma(\vec{\theta}, \varphi)$, the action of $R(\vec{\theta}, \varphi)$ is trivially obtained by an element-wise matrix subtraction of the initial state, because using (13) we have

$$R(\vec{\theta}, \varphi)|\phi\rangle = 2|\phi\rangle_\Sigma - |\phi\rangle. \quad (67)$$

In all the procedures needed to simulate the Szegedy's evolution operator the bigger mathematical objects that intervene are $N \times N$ matrices. Thus, the memory complexity of the algorithm scales as $O(N^2)$. Moreover, since the multiplications are performed element-wise, they involve at most N^2 operations. Thus, the time complexity is also expected to scale as $O(N^2)$.

VIII. CONCLUSIONS

We have reviewed the formulation of Szegedy's quantum walk, and the current phase extensions in the literature based on link phases and a global APR, to generalize it including local APR phases. Therefore, we have defined different Szegedy's model depending on each kind of phase extensions and the graph-phased Szegedy's quantum walk including all the phase extensions.

Based on the quantum circuit construction of the standard Szegedy's walk, we have shown how a circuit can be modified in order to include phase extensions. In general cases where all the phase values would be different, the circuits would be inefficient. However, the same as happens with the transition matrix for the standard model, the phases can be distributed in some symmetric patterns so that an efficient implementation is possible, for example, if there is a small set of nodes with a different local APR phase or there are only two local APR phases values distributed between even and odd nodes.

It is known that under some circumstances the coined quantum walk model and the standard Szegedy's quantum walk are equivalent. We have reviewed this equivalence considering the different phase extensions and found that they can be used to cast a wider set of coins into Szegedy's model. Therefore, we have formulated some lemmas about the conditions that need to satisfy the coin operators in order to establish an equivalence with the different Szegedy's models. Of course, the graph-phased Szegedy's model can host the bigger set of coins.

We have shown how the two quantum walk models can be cast into each other using an example on the 1D line. On the one hand, in the case of the standard Szegedy's walk based on the undirected graph, the equivalent coin is the Pauli X matrix. On the other hand, a Hadamard coin walk is cast into a Szegedy's walk on an unbiased weighted graph, where the probability is bigger for jumping to the right. Simulations show that both coins are not equivalent, despite the fact that the Hadamard coin was thought to be a sensible quantization of the classical walk on the undirected graph. Furthermore, we have seen an example complex coin that cannot be cast into standard Szegedy's model but can be cast into the graph-phased model using the phase extensions. This coin produces the same results as the Hadamard coin when the coins are the same for all nodes. Nevertheless, when a coined walk is quantized using both coins simultaneously, distributing them between even and odd nodes, a new different walk is obtained. This result is quite surprising, and shows that coins that *a priori* seems to be equivalent, are not under different circumstances. Furthermore, based on the symmetric pattern of distribution of the extended phases, we have managed to construct a quantum circuit for this coined walk with two coins.

An important application of the local APR phases is that they can be used to mark nodes in a graph, so that the quantum evolution treats them in a different manner. We have seen how a quantum circuit based on oracles can be constructed, so that there is no need to modify the structure of the graph and it is more efficient than previously known constructions.

Quantum search algorithms are examples where nodes are marked. We have shown that marking with a null local APR phase is totally equivalent to marking with absorbing vertices in the case of using the double Szegedy operator W_s . When using the single operator U_s the same results are obtained after measuring in the computational basis, although both operators are not exactly equal.

Although the -1 coin cannot be cast into the standard Szegedy's walk with the single operator U_s , we have seen that we can cast it for the double operator W_s . This result is intriguing, as shows that when considering the double operator the set of coins that can host is wider, not needing any complex phase extension. More research about this fact is needed in the future.

Finally, we have provided a classical simulation algorithm for the graph-phased Szegedy's quantum walk, including local APR phases in our previous algorithm implemented in SQUWALS [31]. This algorithm scales as $O(N^2)$ in both time and memory requirements, so that is more efficient than previous algorithms scaling as $O(N^3)$.

A quantum PageRank algorithm using Szegedy's walk with global APR was the first algorithm showing the utility of the phase-rotation extension [16]. The link phases have also been considered for problems of state transfer in quantum walks [46]. In the future, it would be interesting to study further applications considering all the phase extensions at the same time. For example, taking into account that local APR phases act at the level of nodes, and link phases at the level of edges, there could be a relation with gauge symmetries. Moreover, we could include the phase extensions considering the twisted swap operator [27] and study how our graph-phased walk can be mapped to the general quantum singular value transform framework [47,48], which is an interesting but open problem.

ACKNOWLEDGMENTS

We acknowledge the support from the Spanish MINECO Grant No. MINECO/FEDER Projects, PID2021-122547NB-I00 FIS2021, the "MADQuantum-CM" project funded by Comunidad de Madrid, and by the Recovery, Transformation, and Resilience Plan—Funded by the European Union - NextGenerationEU and Ministry of Economic Affairs Quantum ENIA project. This work has also been financially supported by the Ministry for Digital Transformation and of Civil Service of the Spanish Government through the QUANTUM ENIA project call—Quantum Spain project and by the European Union through the Recovery, Transformation and Resilience Plan—NextGenerationEU within the framework of the Digital Spain 2026 Agenda. M.A.M.-D. has been partially supported by the U.S. Army Research Office through Grant No. W911NF-14-1-0103. S.A.O. acknowledges support from Universidad Complutense de Madrid—Banco Santander through Grant No. CT58/21-CT59/21.

DATA AVAILABILITY

The library SQUWALS with the new simulation algorithm and a tutorial is openly available [49].

- [1] Y. Aharonov, L. Davidovich, and N. Zagury, Quantum random walks, *Phys. Rev. A* **48**, 1687 (1993).
- [2] E. Farhi and S. Gutmann, Quantum computation and decision trees, *Phys. Rev. A* **58**, 915 (1998).
- [3] F. Magniez, M. Santha, and M. Szegedy, Quantum algorithms for the triangle problem, *SIAM J. Comput.* **37**, 413 (2007).
- [4] A. Ambainis, Quantum walk algorithm for element distinctness, *SIAM J. Comput.* **37**, 210 (2007).
- [5] N. Shenvi, J. Kempe, and K. B. Whaley, Quantum random-walk search algorithm, *Phys. Rev. A* **67**, 052307 (2003).
- [6] D. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani, Quantum walks on graphs, in *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing* (ACM, New York, 2001), pp. 50–59.
- [7] M. Szegedy, Quantum speed-up of Markov chain-based algorithms, in *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science* (IEEE, Piscataway, NJ, 2004), pp. 32–41.
- [8] J. Lemieux, B. Heim, D. Poulin, K. Svore, and M. Troyer, Efficient quantum walk circuits for metropolis-hastings algorithm, *Quantum* **4**, 287 (2020).
- [9] P. A. M. Casares, R. Campos, and M. A. Martin-Delgado, QFold: Quantum walks and deep learning to solve protein folding, *Quantum Sci. Technol.* **7**, 025013 (2022).
- [10] R. Campos, P. A. Casares, and M. A. Martin-Delgado, Quantum metropolis solver: QMS, *Quantum Mach. Intell.* **5**, 28 (2023).
- [11] G. Escrig, R. Campos, P. A. Moreno Casares, and M. A. Martin-Delgado, Parameter estimation of gravitational waves with a quantum metropolis algorithm, *Classical Quantum Gravity* **40**, 045001 (2023).
- [12] G. Escrig, R. Campos, H. Qi, and M. A. Martin-Delgado, Quantum Bayesian inference with renormalization for gravitational waves, *Astrophys. J. Lett.* **979**, L36 (2025).
- [13] S. Giordano and M. A. Martin-Delgado, Quantum algorithm for testing graph completeness, [arXiv:2407.20069](https://arxiv.org/abs/2407.20069).
- [14] G. D. Paparo and M. A. Martin-Delgado, Google in a quantum network, *Sci. Rep.* **2**, 444 (2012).
- [15] G. D. Paparo, M. Müller, F. Comellas, and M. A. Martin-Delgado, Quantum google in a complex network, *Sci. Rep.* **3**, 2773 (2013).
- [16] S. A. Ortega and M. A. Martin-Delgado, Generalized quantum PageRank algorithm with arbitrary phase rotations, *Phys. Rev. Res.* **5**, 013061 (2023).
- [17] S. A. Ortega and M. A. Martin-Delgado, Randomized SearchRank: A semiclassical approach to a quantum search engine, *Phys. Rev. Res.* **6**, 043014 (2024).
- [18] R. Portugal, *Quantum Walks and Search Algorithms* (Springer, New York, 2013).
- [19] H. Wang, J. Wu, X. Yang, P. Chen, and X. Yi, An enhanced quantum PageRank algorithm integrated with quantum search, in *Proceedings of the 8th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing* (IEEE, Piscataway, NJ, 2014), pp. 74–81.
- [20] R. A. Santos, Szegedy’s quantum walk with queries, *Quantum Inf. Process.* **15**, 4461 (2016).
- [21] G. D. Paparo, V. Dunjko, A. Makmal, M. A. Martin-Delgado, and H. J. Briegel, Quantum speedup for active learning agents, *Phys. Rev. X* **4**, 031002 (2014).
- [22] A. Childs, *Quantum Algorithms: Lecture 14: Discrete-time quantum walk* (University of Waterloo, 2008).
- [23] S. A. Ortega and M. A. Martin-Delgado, Discrete-time semiclassical szegedy quantum walks, *Physica A* **625**, 129021 (2023).
- [24] M. Sandbichler, The relationship between Coined and Szegedy-type quantum walks, Master’s thesis, University of Innsbruck, 2016.
- [25] T. G. Wong, Equivalence of Szegedy’s and coined quantum walks, *Quantum Inf. Process.* **16**, 1 (2017).
- [26] T. G. Wong, Coined quantum walks on weighted graphs, *J. Phys. A: Math. Theor.* **50**, 475301 (2017).
- [27] Y. Higuchi, N. Konno, I. Sato, and E. Segawa, Spectral and asymptotic properties of Grover walks on crystal lattices, *J. Funct. Anal.* **267**, 4197 (2014).
- [28] R. Portugal, R. A. Santos, T. D. Fernandes, and D. N. Gonçalves, The staggered quantum walk model, *Quantum Inf. Process.* **15**, 85 (2016).
- [29] T. Loke and J. B. Wang, Efficient quantum circuits for Szegedy quantum walks, *Ann. Phys.* **382**, 64 (2017).
- [30] C.-F. Chiang, D. Nagaj, and P. Wocjan, Efficient circuits for quantum walks, *Quantum Inf. Comput.* **10**, 0420 (2010).
- [31] S. A. Ortega and M. A. Martin-Delgado, Squwals: A Szegedy quantum walks simulator, *Adv. Quantum Technol.* **7**, 2400022 (2024).
- [32] T. Roy, L. Jiang, and D. I. Schuster, Deterministic Grover search with a restricted oracle, *Phys. Rev. Res.* **4**, L022013 (2022).
- [33] V. Bergholm, J. J. Vartiainen, M. Möttönen, and M. M. Salomaa, Quantum circuits with uniformly controlled one-qubit gates, *Phys. Rev. A* **71**, 052330 (2005).
- [34] M. Möttönen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa, Quantum circuits for general multiqubit gates, *Phys. Rev. Lett.* **93**, 130502 (2004).
- [35] A. Hagberg, D. Schult, and P. Swart, Exploring network structure, dynamics, and function using networkX, in *Proceedings of the 7th Python in Science Conference*, edited by G. Varoquaux, T. Vaught, and J. Millman (Scipy, Pasadena, CA, 2008), pp. 11–16.
- [36] P. Ren, T. Aleksić, D. Emms, R. C. Wilson, and E. R. Hancock, Quantum walks, Ihara zeta functions and cospectrality in regular graphs, *Quantum Inf. Process.* **10**, 405 (2011).
- [37] A. Ambainis, J. Kempe, and A. Rivosh, Coins make quantum walks faster, in *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms* (ACM, New York, 2005), pp. 1099–1108.
- [38] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing* (ACM, New York, 1996).
- [39] T. D. Mackay, S. D. Bartlett, L. T. Stephenson, and B. C. Sanders, Quantum walks in higher dimensions, *J. Phys. A: Math. Gen.* **35**, 2745 (2002).
- [40] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, UK, 2010).
- [41] A. Galindo and M. A. Martin-Delgado, Information and computation: Classical and quantum aspects, *Rev. Mod. Phys.* **74**, 347 (2002).

- [42] Y. Segawa, E. and Yoshie, Quantum search of matching on signed graphs, *Quantum Inf. Process.* **20**, 182 (2021).
- [43] R. A. M. Santos and R. Portugal, Quantum hitting time on the complete graph, *Int. J. Quantum Inf.* **08**, 881 (2010).
- [44] H. Krovi, F. Magniez, M. Ozols, and J. Roland, Quantum walks can find a marked element on any graph, *Algorithmica* **74**, 851 (2016).
- [45] C.R. Harris, K. J. Millman, and S. J. van der Walt *et al.*, Array programming with NumPy, *Nature (Lond.)* **585**, 357 (2020).
- [46] A. Chan and H. Zhan, Pretty good state transfer in discrete-time quantum walks, *J. Phys. A: Math. Theor.* **56**, 165305 (2023).
- [47] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (ACM, New York, 2019), pp. 193–204.
- [48] J. M. Martyn, Z. M. Rossi, A. K. Tan, and I. L. Chuang, Grand unification of quantum algorithms, *PRX Quantum* **2**, 040203 (2021).
- [49] S. A. Ortega, SQUWALS: A Szegedy QUantum WALks Simulator, Version 2, GitHub (2024), <https://github.com/OrtegaSA/SQUWALS-repo>.