Multipartite high-dimensional quantum state engineering via discrete-time quantum walk

Junhong Nie,* Meng Li,† and Xiaoming Sun‡

State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China and School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, China

(Received 17 August 2022; accepted 29 February 2024; published 29 March 2024)

Quantum state engineering, namely, the generation and control of arbitrary quantum states, is drawing more and more attention due to its wide applications in quantum information and computation. However, there is no general method in theory, and the existing schemes also depend heavily on the selected experimental platform. In this article, we give two schemes for the generating task of arbitrary quantum state in c-partite d-dimensional system, both of which are based on a discrete-time quantum walk with a 2^c-dimensional time- and position-dependent coin. The first procedure is a d-step quantum walk where all the d coins are nonidentity, while the second procedure is an $O(d)$-step quantum walk where only $O(\log d)$ coins are nonidentity. A concrete example of preparing generalized Bell states is given to demonstrate the first scheme we proposed. Furthermore, the first scheme can be applied to give an alternative approach to the quantum state preparation problem which is one of the fundamental tasks of quantum information processing. We design circuits for quantum state preparation with the help of our quantum state engineering scheme that match the best current result in both size and depth of the circuit asymptotically.

DOI: 10.1103/PhysRevA.109.032229

I. INTRODUCTION

High-dimensional states carry more information than qubits and thus can be exploited to reduce resource consumption of quantum information and computation tasks [1–3]. Multipartite high-dimensional states, especially high-dimensional entanglement [4], can be used as important sources in plenty of quantum information protocols, such as quantum teleportation [5], quantum key distribution [6–8], and quantum secret sharing [9]. Thus, quantum state engineering, i.e., the realization of coherent dynamics or multipartite high-dimensional quantum states by manipulating appropriate quantum mechanical systems, has become a hot topic in modern physics and computer sciences. In this work, we study the generation (or preparation) of multipartite high-dimensional states.

Discrete time quantum walk acts as a powerful tool in a lot of areas of quantum information and computation. From the computational theory perspective, it is universal for quantum computation [10–13], and serves as an important part in various quantum algorithms such as element distinctness [14], triangle finding [15], and so on. It is also proved that quantum algorithms for finding a marked vertex via quantum walk search is quadratic faster than classical random walk [16]. Apart from that, discrete-time quantum walk has been experimentally illustrated in some physical platforms, such as ion traps [17,18], optical [19,20], and superconducting processor [21,22].

Due to the flexibility and experimental feasibility of quantum walk, it has also been applied to the preparation of high-dimensional states [23,24]. In general, there are three kinds of quantum coins in quantum walks: time-dependent, position-dependent, time- and position-dependent. Innocenti *et al.* proposed a scheme for preparing most of the high-dimensional states on a single particle using a one-dimensional quantum walk with time-dependent coins [25], which is later experimentally realized in the linear-optics platform [26] and further serves as an experimental example in Ref. [27]. Moreover, Kadiri gives a sufficient condition of what states can be generated by quantum walk with time-dependent coin, of the entire system including both position and coin space [28]. However, Montero shows that quantum walks with time- and position-dependent coins can be used to retrieve any given distribution on a single particle [29,30], and this theoretical result has been experimentally realized recently [31].

However, most of the current schemes are only for single-qudit states, or there are limitations on the dimension of the bipartite quantum states discussed above. Motivated by this, we focus on preparing arbitrary states in multipartite systems. Due to the essence of preparing multipartite high-dimensional entanglements and the variety of physical platforms, we aim to seek a general framework to accomplish this task by taking advantage of quantum walk with time- and position-dependent coins. Here, we first put forward two quantum walk procedures generating arbitrary c-partite d-dimensional quantum state $|\phi\rangle$ with the help of a 2^c-dimensional time- and position-dependent coin. The first one has exactly d steps where all the d coins are nonidentitical. The second scheme is an $O(d)$-step quantum walk where only $O(\log d)$ coins are nonidentitical. Both of these procedures has nearly optimal

*niejunhong19z@ict.ac.cn

†Corresponding author: limeng2021@ict.ac.cn

‡sunxiaoming@ict.ac.cn

TABLE I. Relationship of our work with previous works. The multipartite engineering based on time-dependent coin mainly concerns a system consisting of one qudit and one qubit [28].

	One qudit	Multipartite
Time-dependent coin	[25]	qudit +qubit [28]
Experiments	[26,27]	–
Time- and position-dependent coin	[30]	**Our work**
Experiments	[31]	–

steps. A summary of the relationship of our work with other previous works is shown in Table I.

We also illustrate the first scheme with two examples, hoping for encouraging potential applications of engineering multipartite high-dimensional quantum states. The first example is about engineering generalized Bell states which firstserves as basis for the measurements Alice used in the quantum teleportation scheme [5]. Later in some quantum key distribution (QKD) protocols, it was used as the initial shared entanglements [32]. Here, we illustrate our algorithm with this particular state. For bipartite Bell states, our scheme has d steps. It is worth noting that one can indeed achieve $O(d)$ steps in this application using the scheme from Ref. [25]. However, as will be mentioned in Sec. V A, the realization of shift operator using Ref. [25]'s scheme is a lot more tricky compared with just using the native shift operator by leveraging our scheme.

Another example is the relationship between quantum engineering and quantum state preparation. Quantum circuit and quantum walk are two important models in quantum information and they are proved to be equivalent from the perspective of quantum computation. However, as mentioned earlier, there are several different approaches to physically realize quantum computation and quantum circuit model is suitable for describing quantum evolutions in some approaches, while the quantum walk model may be suitable for some others (e.g., the shift operator may be native in some physical realizations). Quantum state preparation is a fundamental problem asking for a quantum circuit preparing any given n-qubit quantum state of size and depth as small as possible [33–37]. When the number of ancillary qubits is $O(n)$, the asymptotically optimal circuit accomplishing this task has size $\Theta(2^n)$ and depth $\Theta(\frac{2^n}{n})$ [37]. Here we seek for a quantum walk protocol preparing a particular state on qudit systems. As explained above, these two problems are both for preparing states while they differ from each other on the quantum information models they are based on. Considering that quantum walk is universal for quantum computation, which means that any quantum circuit can be simulated by a quantum walk protocol, we explore the similarity of these two problems. We show that the same bounds can also be reached simultaneously by treating the problem in the quantum circuit model as engineering a bipartite $2^{\frac{n}{2}}$-dimensional quantum state and utilizing our scheme in the quantum walk model. To achieve this, we go the opposite way that quantum walk procedures can be implemented by efficient quantum circuits.

This paper is organized as follows. In Sec. II, we briefly introduce the model of quantum walks we will base on. In Sec. III, we propose our first scheme to engineer an arbitrary multipartite high-dimensional quantum state based on quantum walk with time- and position-dependent coins. The bipartite case where $c = 2$ is detailed in Sec. III A and the general multipartite case is sketched in Sec. III B. In Sec. IV, we show our second scheme which has $O(d)$ steps while only $O(\log d)$ coins are nonidentitical. We only detail the bipartite case with d a power of 2. After that in Sec. V, we give two examples where our scheme may be used. The first one is an illustration of our scheme by taking the generalized Bell state as an example, which is shown in Sec. V A. And the relationship between quantum state engineering and quantum state preparation is presented in Sec. V B. Finally, we end with a summary and outlook in Sec. VI.

II. PRELIMINARIES

Quantum walks are analogies of the well-known random walks in which at each step one tosses some coins (may be biased) to decide which direction to go. In this work, the system where quantum walks apply to consists of c particles and a 2^c-dimensional coin. That is, the system is the product of two Hilbert spaces: position space $\mathcal{H}_P$ in which the particles live and coin space $\mathcal{H}_C$. We use the computational basis, i.e., $\mathcal{H}_P$ spanned by $\{|\mathbf{x}\rangle : \mathbf{x} \in \mathbb{Z}^c, \mathbf{x} \geqslant 0\}$, and $\mathcal{H}_C$ spanned by $\{|\mathbf{c}\rangle : \mathbf{c} \in \{0, 1\}^c\}$. The initial state of the system is $|0\rangle^{\otimes c} \otimes |0\rangle^{\otimes c}$.

The kth step of a quantum walk on $\mathcal{H} = \mathcal{H}_P \otimes \mathcal{H}_C$ is described as a unitary $W^{(k)} = SC^{(k)}$, where $C^{(k)}$ is the coin-flipping operator and S is the conditional shift operator. Generally speaking, there are three kinds of coin operators which have also been illustrated experimentally in the literature, time-dependent [38,39], position-dependent [40,41], both time- and position-dependent [42,43]. As mentioned in Sec. I, in this work we use both the time- and position-dependent coin-flipping operator, that is,

$$C^{(k)} = \sum_{\mathbf{x}\geqslant 0} |\mathbf{x}\rangle \langle \mathbf{x}| \otimes C^{(k)}(\mathbf{x}). \tag{1}$$

The conditional shift operator S is defined as

$$S = \sum_{\substack{\mathbf{x}\geqslant 0\\ \mathbf{c}\in\{0,1\}^c}} |\mathbf{x}+\mathbf{c}\rangle \langle \mathbf{x}| \otimes |\mathbf{c}\rangle \langle \mathbf{c}| . \tag{2}$$

Let's illustrate this with the special case $c = 2$. The initial state is $|0, 0\rangle \otimes |0, 0\rangle$. For this four-dimensional coin, we may use alternative notations

$$|\circlearrowleft\rangle = |0, 0\rangle , |\rightarrow\rangle = |1, 0\rangle , |\uparrow\rangle = |0, 1\rangle , |\nearrow\rangle = |1, 1\rangle \tag{3}$$

to describe the computational basis on a 2D grid, so the initial state can also be written $|0, 0\rangle \otimes |\circlearrowleft\rangle$. Now we can define the coin operator $C^{(k)}$ and conditional shift operator S:

$$C^{(k)} = \sum_{x,y\geqslant 0} |x, y\rangle \langle x, y| \otimes C^{(k)}(x, y), \tag{4}$$

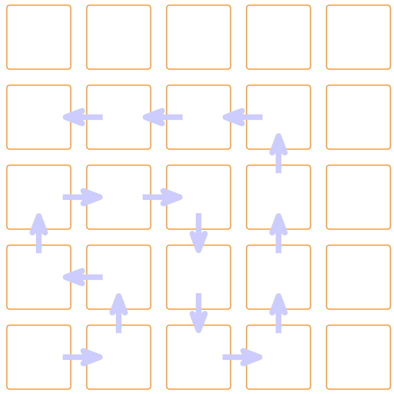


FIG. 1. Illustration of propagation of amplitudes using algorithm in Ref. [25].

where $C^{(k)}(x, y)'$s are all 4×4 unitaries, and

$$\begin{aligned} S(|x, y\rangle \otimes |\circlearrowleft\rangle) &= |x, y\rangle \otimes |\circlearrowleft\rangle , \\ S(|x, y\rangle \otimes |\rightarrow\rangle) &= |x+1, y\rangle \otimes |\rightarrow\rangle , \\ S(|x, y\rangle \otimes |\uparrow\rangle) &= |x, y+1\rangle \otimes |\uparrow\rangle , \\ S(|x, y\rangle \otimes |\nearrow\rangle) &= |x+1, y+1\rangle \otimes |\nearrow\rangle . \end{aligned} \tag{5}$$

In fact, this direction calibration is equivalent to the usual method (the particle moves left, right, up or down according to the coin state), and in this way we can have a much cleaner and more intuitive representation.

III. THEORETICAL SCHEME FOR ENGINEERING QUANTUM STATES

One can apparently make use of the scheme from Ref. [25] on c particles to produce arbitrary product states with only time-dependent coins used. However, engineering of multipartite entanglement states using only time-dependent coins is a bit more tricky. Indeed, one can encode the basis of the entire system in a zigzag fashion and utilize the scheme on the one-dimensional line (The propagation procedure of $d = 2$ is illustrated in Fig. 1), but this procedure takes d^c steps of quantum walk, and needs us to design coin operators delicately and manipulate shift operator in a complicated way. Also, this method reaches the lower bound of steps using time-dependent coins, which can be proved by a counting argument. Indeed, s steps of the time-dependent quantum walk has $O(s)$ different quantum coins and each of these coins provides constant number of degrees of freedom, so there are $O(s)$ flexible parameters in total. However, arbitrary states on c particles involve $O(d^c)$ parameters. Therefore, this scheme is not ideal for multipartite quantum state engineering from the perspective of the number of steps.

In this section, we will elaborate the theoretical protocol of realizing arbitrary multipartite high-dimensional quantum state engineering via a d-step quantum walk, which is nearly optimal. Here we fully explore the potential of the quantum walk with time- and position-dependent coins by focusing on the critical positions and splitting coin operators.

A. Bipartite case

Now we explain how our method works in the bipartite case. We make use of quantum walks with $c = 2$, which is detailed in Sec. II. Say one has an arbitrary state to be prepared $|\phi\rangle = \sum_{x,y=0}^{d-1} \alpha_{x,y} |x, y\rangle$, and the initial state is $|\Psi^{(0)}\rangle = |0, 0\rangle \otimes |\circlearrowleft\rangle$, then the procedure is described as the following d steps:

$$|\phi\rangle \otimes |\circlearrowleft\rangle = W^{(d-1)}W^{(d-2)} \cdots W^{(1)}W^{(0)} |\Psi^{(0)}\rangle . \tag{6}$$

The design of coin operators is the key to make the procedure suitable for arbitrary target quantum states.

Instead of directly design the coin operators in the quantum walk procedure, we accomplish the engineering task via an alternative procedure and turn it into the standard quantum walk procedure described in Eq. (6). The procedure we are looking at is denoted as

$$V^{(d-2)} \cdots V^{(1)}V^{(0)} |\Psi^{(0)}\rangle . \tag{7}$$

Later we will see the definition of the operators $V^{(k)}$. Also, we define some intermediate states

$$\begin{aligned} |\Psi^{(1)}\rangle &= V^{(0)} |\Psi^{(0)}\rangle , \\ |\Psi^{(2)}\rangle &= V^{(1)} |\Psi^{(1)}\rangle , \\ &\ldots , \\ |\Psi^{(d-1)}\rangle &= V^{(d-2)} |\Psi^{(d-2)}\rangle . \end{aligned} \tag{8}$$

For these $|\Psi^{(k)}\rangle'$s, we make a crucial convention that their coin spaces are all in the particular state $|\circlearrowleft\rangle$. That is, they are all of the form

$$|\Psi^{(k)}\rangle = |\phi^{(k)}\rangle \otimes |\circlearrowleft\rangle , \tag{9}$$

in which $|\phi^{(k)}\rangle = \sum_{x,y=0}^{k} \alpha_{x,y}^{(k)} |x, y\rangle$ is spanned by $\{|l, r\rangle : 0 \leqslant l, r \leqslant k\}$ for all $k = 0, 1, \ldots, d-1$ and $|\phi^{(d-1)}\rangle = |\phi\rangle$. That is, after $d-1$ steps, we reach the target state on position space. From now on, we use the notation $|\Psi^{(k)}\rangle$ and $|\phi^{(k)}\rangle$ interchangeably.

Now we define the operators $V^{(k)}$. Each $C^{(k)}$ in the walk operator $W^{(k)}$ in Eq. (6) is decomposed into two coin operators: $C^{(k)} = C_1^{(k)}C_2^{(k)}$ for $k = 0, \ldots, d-1$ with $C_2^{(0)} = C_1^{(d-1)} = I$. In addition we define $V^{(k)} = C_2^{(k+1)}SC_1^{(k)}$ for all $k = 0, 1, \ldots, d-2$. Intuitively, the coin operator $C_1^{(k)\prime}$s fork the directions according to the $\alpha_{x,y}^{(k+1)\prime}$s, and $C_2^{(k)\prime}$s are for the purpose of turning the states of coin space into $|\circlearrowleft\rangle$. Thus as

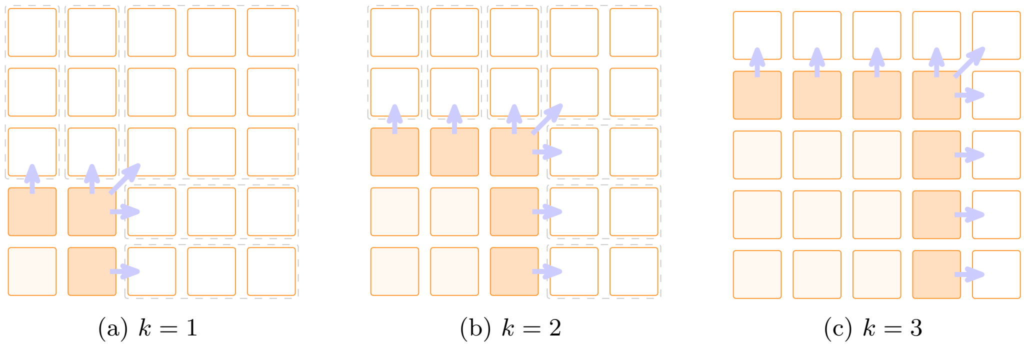


FIG. 2. Illustration of propagation of amplitudes along the algorithm process.

long as the state after shift operator ($SC_1^{(k)}|\Psi^{(k)}\rangle$) is given, the operator $C_2^{(k+1)}$ is settled automatically.

With these definitions in hand, we show how to engineer the state $|\phi\rangle$. This is described in an inductive fashion: suppose for any $|\Psi^{(k)}\rangle$, there exists operators $V^{(0)}, V^{(1)}, \ldots, V^{(k-1)}$ such that

$$|\Psi^{(k)}\rangle = V^{(k-1)}\cdots V^{(1)}V^{(0)}|\Psi^{(0)}\rangle, \tag{10}$$

we claim that for any given state $|\phi^{(k+1)}\rangle$, there exists operators $V^{(0)}, V^{(1)}, \ldots, V^{(k)}$ such that

$$|\Psi^{(k+1)}\rangle = V^{(k)}\cdots V^{(1)}V^{(0)}|\Psi^{(0)}\rangle. \tag{11}$$

Once this claim is proved, we can indeed reach our goal: simply let $|\Psi^{(d-1)}\rangle$ be equal to $|\phi\rangle\otimes|\circlearrowleft\rangle$, then there exists $V^{(0)}, V^{(1)}, \ldots, V^{(d-2)}$ such that

$$|\Psi^{(d-1)}\rangle = V^{(d-2)}\cdots V^{(1)}V^{(0)}|\Psi^{(0)}\rangle. \tag{12}$$

This indeed corresponds to a quantum walk procedure described in Eq. (6) because

$$S|\Psi^{(d-1)}\rangle = S(|\phi\rangle\otimes|\circlearrowleft\rangle) = |\phi\rangle\otimes|\circlearrowleft\rangle = |\Psi^{(d-1)}\rangle, \tag{13}$$

where the second equation is according to the definition of shift operator S in Eq. (5).

Given $|\phi^{(k+1)}\rangle$, to prove the claim, we delicately pick some $|\phi^{(k)}\rangle$ and computes two coins $C_1^{(k)}$ and $C_2^{(k+1)}$, such that $|\Psi^{(k+1)}\rangle = C_2^{(k+1)}SC_1^{(k)}|\Psi^{(k)}\rangle$ for all $k = 0, 1, \ldots, d-2$. Since the coin operator can take full control of the position space, let's denote

$$C_i^{(k)} = \sum_{x,y\geqslant 0} |x, y\rangle\langle x, y| \otimes C_i^{(k)}(x, y), \tag{14}$$

where $i = 1, 2$. Note that $|\phi^{(k)}\rangle$ is spanned by $\{|l, r\rangle : 0 \leqslant l, r \leqslant k\}$, which means $C_1^{(k)}(x, y)$ is meaningless for $x > k$ or $y > k$, and thus they can be set identity. To make the coin operator as simple as possible, $C_1^{(k)}(x, y)$ for $x, y < k$ are a set identity also. This, in fact, leads to the state $|\phi^{(k)}\rangle$ we choose.

The base case $|\Psi^{(0)}\rangle$ is fixed. For $k \geqslant 0$, suppose the given state is $|\phi^{(k+1)}\rangle = \sum_{x,y=0}^{k+1}\alpha_{x,y}^{(k+1)}|x, y\rangle$, then the algorithm sets

$$\alpha_{x,y}^{(k)} = \begin{cases} \alpha_{x,y}^{(k+1)}, & x, y < k, \\ \sqrt{\left|\alpha_{x,y}^{(k+1)}\right|^2 + \left|\alpha_{x+1,y}^{(k+1)}\right|^2}, & x = k, y < k, \\ \sqrt{\left|\alpha_{x,y}^{(k+1)}\right|^2 + \left|\alpha_{x,y+1}^{(k+1)}\right|^2}, & x < k, y = k, \\ \sqrt{\left|\alpha_{x,y}^{(k+1)}\right|^2 + \left|\alpha_{x+1,y}^{(k+1)}\right|^2 + \left|\alpha_{x,y+1}^{(k+1)}\right|^2 + \left|\alpha_{x+1,y+1}^{(k+1)}\right|^2}, & x = y = k. \end{cases} \tag{15}$$

The intuition behind setting these parameters is illustrated in Fig. 2. In the figure, each diagram represents an internal state of the algorithm. Each box corresponds to a position of the system's Hilbert space. Some boxes are colored by light (dark) orange to represent that the amplitude of the corresponding position is already (not yet) settled (that is, identical to target state $|\phi\rangle$), while the other boxes are not colored, which means that the amplitude of the corresponding position is zero for the current step. Figure 2 illustrates the propagation of amplitudes from $k = 1$ to $k = 3$.

For the system's state $|\Psi^{(k)}\rangle$ at step k, positions corresponding to the lower-left corner colored by light orange are settled. Each position on the dark boundary has a specific amplitude that is ready for amplitude propagation to positions in one dashed box in future steps. For example, the upper-right corner has amplitude that is the sum of all absolute values of amplitudes in the upper-right dashed box. In other words, our strategy is that the absolute values of amplitudes propagate along this process while the relative phases are adjusted locally.

Now suppose this particular $|\Psi^{(k)}\rangle$ has been prepared, the operators $C_1^{(k)}(x,y)$ are given by the following formula. Note that, for simplicity, we slightly abuse the notations such that the entire term is zero while dividing by zero

$$C_1^{(k)}(x,y)\,|\circlearrowleft\rangle = \begin{cases} |\circlearrowleft\rangle\,, & x,y<k,\\ \left(\alpha_{x,y}^{(k+1)}\,|\circlearrowleft\rangle + \alpha_{x+1,y}^{(k+1)}\,|\rightarrow\rangle\right)/\alpha_{x,y}^{(k)}, & x=k, y<k,\\ \left(\alpha_{x,y}^{(k+1)}\,|\circlearrowleft\rangle + \alpha_{x,y+1}^{(k+1)}\,|\uparrow\rangle\right)/\alpha_{x,y}^{(k)}, & x<k, y=k,\\ \left(\alpha_{x,y}^{(k+1)}\,|\circlearrowleft\rangle + \alpha_{x+1,y}^{(k+1)}\,|\rightarrow\rangle + \alpha_{x,y+1}^{(k+1)}\,|\uparrow\rangle + \alpha_{x+1,y+1}^{(k+1)}\,|\nearrow\rangle\right)/\alpha_{x,y}^{(k)}, & x=y=k, \end{cases} \tag{16}$$

In addition, $C_2^{(k)}(x,y)$ satisfies

$$\begin{aligned} C_2^{(k)}(x,y)\,|\circlearrowleft\rangle &= |\circlearrowleft\rangle\,, & x,y<k,\\ C_2^{(k)}(x,y)\,|\rightarrow\rangle &= |\circlearrowleft\rangle\,, & x=k, y<k,\\ C_2^{(k)}(x,y)\,|\uparrow\rangle &= |\circlearrowleft\rangle\,, & x<k, y=k,\\ C_2^{(k)}(x,y)\,|\nearrow\rangle &= |\circlearrowleft\rangle\,, & x=y=k. \end{aligned} \tag{17}$$

It's straightforward to see these $C_1^{(k)}(x,y)'$s and $C_2^{(k)}(x,y)'$s can be set to unitaries and they can indeed achieve our goal. Let's take the case $x=k, y<k$ as example, and the proof of the remaining cases are similar. The initial state at $x=k, y<k$ is $\alpha_{x,y}^{(k)}\,|x,y\rangle\otimes|\circlearrowleft\rangle$, and after $C_1^{(k)}(x,y)$, the shift operator S and $C_2^{(k+1)}(x,y)$, the state becomes

$$\begin{aligned} &\alpha_{x,y}^{(k)}\,|x,y\rangle\otimes|\circlearrowleft\rangle\\ &\xrightarrow{C_1^{(k)}} \alpha_{x,y}^{(k)}\,|x,y\rangle\otimes\left(\alpha_{x,y}^{(k+1)}\,|\circlearrowleft\rangle+\alpha_{x+1,y}^{(k+1)}\,|\rightarrow\rangle\right)/\alpha_{x,y}^{(k)}\\ &\xrightarrow{S} \alpha_{x,y}^{(k+1)}\,|x,y\rangle\otimes|\circlearrowleft\rangle+\alpha_{x+1,y}^{(k+1)}\,|x+1,y\rangle\otimes|\rightarrow\rangle\\ &\xrightarrow{C_2^{(k)}} \left(\alpha_{x,y}^{(k+1)}\,|x,y\rangle+\alpha_{x+1,y}^{(k+1)}\,|x+1,y\rangle\right)\otimes|\circlearrowleft\rangle\,. \end{aligned} \tag{18}$$

Also note that the $C_1^{(k)}(x,y)'$s and $C_2^{(k)}(x,y)'$s can be set identity for all $x,y<k$.

In fact, the explicit form of the $\alpha_{x,y}^{(k)\prime}$s in our algorithm for every step $k<d$ can be directly expressed by the amplitudes of the desired state $|\phi\rangle$, i.e., $\alpha_{x,y}{}'$s. For a fixed k, we have the following observations: $\alpha_{x,y}^{(k)\prime}$s are zero except for the case $x\leqslant k, y\leqslant k$. In addition, $\alpha_{x,y}^{(k)}=\alpha_{x,y}$ for $x<k, y<k$. So the only unsettled cases are $x=k, y<k$; $x<k, y=k$; and $x=k, y=k$. It's straightforward to see that

$$\alpha_{x,y}^{(k)} = \begin{cases} \alpha_{x,y}, & x,y<k,\\ \sqrt{\sum_{z=k}^{d-1}|\alpha_{z,y}|^2}, & x=k, y<k,\\ \sqrt{\sum_{w=k}^{d-1}|\alpha_{x,w}|^2}, & x<k, y=k,\\ \sqrt{\sum_{z,w=k}^{d-1}|\alpha_{z,w}|^2}, & x=y=k. \end{cases} \tag{19}$$

One can prove this by plugging it into Eq. (15) and making induction on k. With the help of Eq. (19), one can construct the coin operators $C_1^{(k)}(x,y)$ for every k explicitly.

The algorithm is shown in Algorithm 1. Below we show that the time complexity for computing the parameters on a classical machine is $O(d^2)$. To compute the elements of $C^{(k)}(x,y)'$s, one needs to compute the coefficients $\alpha_{x,y}^{(k)}$ for every k first, according to Eq. (15). Notice that at step k, there are only $2k+1$ of these coefficients that differs from $\alpha_{x,y}$, so the total effort of computing these coefficients for all k is $O(d^2)$. After that, the elements of the coin operators can be computed according to Eqs. (16) and (17). Again, recall that at step k, only $2k+1$ out of k^2 of the operators $C_{x,y}^{(k)}$ are nonidentity, and they are all 4×4 matrices whose elements can be computed from the $\alpha_{x,y}^{(k)\prime}$s. Thus, the total effort of computing these operators is $O(d^2)$ also. In summary, this algorithm takes $O(d^2)$ time to compute all the coin operators used in engineering $|\phi\rangle$.

B. Multipartite case

The multipartite case is a natural generalization of the bipartite case, so here we briefly sketch the scheme. Say again one has an arbitrary state to be prepared $|\phi\rangle=\sum_{\mathbf{x}<d}\alpha_{\mathbf{x}}\,|\mathbf{x}\rangle$, and the initial state is $|\Psi^{(0)}\rangle=|0^c\rangle\otimes|0^c\rangle$. Then the quantum walk procedure is

$$|\phi\rangle\otimes|0^c\rangle = SC^{(d-1)}SC^{(d-2)}\cdots SC^{(1)}SC^{(0)}\,|\Psi^{(0)}\rangle\,. \tag{20}$$

Again, we look at the alternative procedure

$$V^{(d-2)}\cdots V^{(1)}V^{(0)}\,|\Psi^{(0)}\rangle\,. \tag{21}$$

Denote $|\Psi^{(k)}\rangle=|\phi^{(k)}\rangle\otimes|0^c\rangle$ the intermediate state of step k in which

$$|\phi^{(k)}\rangle = \sum_{\mathbf{x}\leqslant k}\alpha_{\mathbf{x}}^{(k)}\,|\mathbf{x}\rangle\,, \tag{22}$$

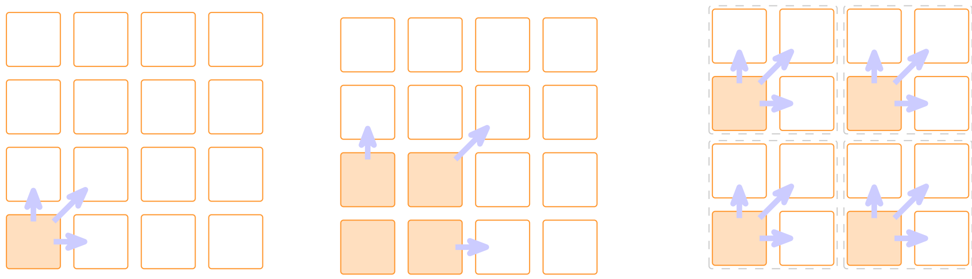
and

$$|\Psi^{(k+1)}\rangle = V^{(k)}\,|\Psi^{(k)}\rangle \tag{23}$$

for all $k=0,1,\ldots,d-2$. Recall that $V^{(k)}=C_2^{(k+1)}SC_1^{(k)}$, $C_i^{(k)}=\sum_{0\leqslant\mathbf{x}\leqslant k}|\mathbf{x}\rangle\langle\mathbf{x}|\otimes C_i^{(k)}(\mathbf{x})$, where $i=1,2$.

ALGORITHM 1. Quantum state engineering for bipartite systems

Input: a complex vector $\{\alpha_{x,y}\}$ describing the state $|\phi\rangle=\sum_{x,y=0}^{d-1}\alpha_{x,y}\,|x,y\rangle$ to be prepared
Output: $\{C^{(k)}\}_{k=0}^{d-1}$
1: **for** $k\leftarrow d-1, 0$ **do**
2: Compute all $\alpha_{x,y}^{(k)}$ according to Eq. (15)
3: Compute all $C_1^{(k)}(x,y)$ and $C_2^{(k)}(x,y)$ according to Eqs. (16) and (17), respectively
4: **end for**
5: Compute $C^{(k)}=C_1^{(k)}C_2^{(k)}$ for $k=0,1,\ldots,d-1$
6: **return** $\{C^{(k)}\}_{k=0}^{d-1}$



(a) $C_1^{(0)}$ (b) S (c) $C_2^{(1)}$ and future directions

FIG. 3. Propagation of one step of the alternative approach. This illustration corresponds to the operator $V^{(0)} = C_2^{(1)} S^{(2)} C_1^{(0)}$ for $d = 4$.

Now we describe how to design the operators $V^{(k)}$. We achieve this in an inductive fashion: suppose for any $|\Psi^{(k)}\rangle$, there exists an alternative procedure of k steps, we show that for any given state $|\phi^{(k+1)}\rangle$, there exists an alternative procedure of $k+1$ steps. That is, given $|\phi^{(k+1)}\rangle$, we delicately pick some $|\phi^{(k)}\rangle$ and compute two coins $C_1^{(k)}$ and $C_2^{(k+1)}$, such that $|\Psi^{(k+1)}\rangle = C_2^{(k+1)} S C_1^{(k)} |\Psi^{(k)}\rangle$.

The base case $k = 0$ is simple and similar to the bipartite case in Sec. III A. For $k > 0$, given arbitrary state $|\phi^{(k+1)}\rangle = \sum_{\mathbf{x} \leqslant k+1} \alpha_{\mathbf{x}}^{(k+1)} |\mathbf{x}\rangle$, the algorithm sets $\alpha_{\mathbf{x}}^{(k)}$'s in the following way. For $\mathbf{x} < k$, $\alpha_{\mathbf{x}}^{(k)}$ is simply set to be equal to $\alpha_{\mathbf{x}}^{(k+1)}$. For $\mathbf{x} \leqslant k$ with $x_i = k$ for some i, define a vector cylinder $V_{\mathbf{x}} = \{\mathbf{z} \in \{0, 1\}^c : z_j = 0 \text{ if } x_j < k\}$, then the algorithm sets

$$\alpha_{\mathbf{x}}^{(k)} = \sqrt{\sum_{\mathbf{z} \in V_{\mathbf{x}}} \left|\alpha_{\mathbf{x}+\mathbf{z}}^{(k+1)}\right|^2}. \tag{24}$$

Now suppose the state $|\phi^{(k)}\rangle$ defined above is prepared, $C_1^{(k)}(\mathbf{x})$ is given by

$$C_1^{(k)}(\mathbf{x}) |0^c\rangle = \sum_{\mathbf{z} \in V_{\mathbf{x}}} \alpha_{\mathbf{x}+\mathbf{z}}^{(k+1)} |\mathbf{z}\rangle / \alpha_{\mathbf{x}}^{(k)}. \tag{25}$$

In addition, $C_2^{(k)}(\mathbf{x})$ satisfies

$$C_2^{(k)}(\mathbf{x}) |\mathbf{z}\rangle = |0^c\rangle \tag{26}$$

for all $\mathbf{z} \in V_{\mathbf{x}}$. Note again that the $C_1^{(k)}(\mathbf{x})$'s and $C_2^{(k)}(\mathbf{x})$'s can be set identity for all $\mathbf{x} < k$.

IV. ALTERNATIVE SCHEME FOR ENGINEERING QUANTUM STATES

In this section, we elaborate another theoretical protocol of realizing arbitrary multipartite high-dimensional quantum state engineering via an $O(d)$-step quantum walk. While the scheme described in Sec. III consists of d steps of the quantum walk that each coin operator is nonidentitical, this alternative scheme has only $O(\log d)$ nonidentitical coin operators in total. This is achieved by taking advantage of time- and position-dependent coins further such that the engineering procedure becomes much more paralleled. We will illustrate this scheme by only the bipartite case when the dimension d is a power of 2, scheme for general d, and the multipartite case are both natural generalizations that are omitted.

Below we adopt the settings and notions used in Sec. III A. Suppose the state to be prepared is $|\phi\rangle = \sum_{x,y=0}^{d-1} \alpha_{x,y} |x, y\rangle$, and the initial state is $|\Psi^{(0)}\rangle = |0, 0\rangle \otimes |\circlearrowleft\rangle$. The whole procedure is described as Eq. (6). Again, instead of directly designing the coin operators, we decompose each $C^{(k)}$ into $C^{(k)} = C_1^{(k)} C_2^{(k)}$ and seek for another procedure that is equivalent to the original one. This time we look at a slightly different procedure, which is denoted as

$$V^{(\log d - 1)} \dots V^{(1)} V^{(0)} |\Psi^{(0)}\rangle, \tag{27}$$

where $V^{(i)} = C_2^{(i+1)} S^{\frac{d}{2^{i+1}}} C_1^{(i)}$ for $i = 0, 1, \dots, \log d - 1$. It can be verified that this procedure is indeed a quantum walk of $O(d)$ steps.

Intuitively, this scheme builds up the whole state by first engineering a state that has nonzero amplitudes only on some particular positions, say $|0, 0\rangle, |\frac{d}{2}, 0\rangle, |0, \frac{d}{2}\rangle, |\frac{d}{2}, \frac{d}{2}\rangle$, and then assuming one can engineer arbitrary bipartite states with dimension $\frac{d}{2}$, the whole state can be engineered by starting from these four positions and calling four state engineering procedures for dimension $\frac{d}{2}$ in parallel. An illustration for $d = 4$ is presented in Fig. 3. The figure shows the propagation of the first step $V^{(0)}$. The idea is dividing the whole position space into four regions dashed in Fig. 3(c). The scheme first distributes the amplitudes to these four regions according to their total amplitudes in target state $|\phi\rangle$ and call the procedure recursively to engineer the "substates" in these regions.

Like Sec. III A, we describe this scheme in an inductive fashion. However, instead of using an induction by the dimension d of the system in Sec. III A, here we use an induction by $\log d$. First we define notions $|\Psi^{(k)}\rangle$ for $k = 0, 1, \dots, \log d$:

$$|\Psi^{(k)}\rangle = |\phi^{(k)}\rangle \otimes |\circlearrowleft\rangle, \tag{28}$$

in which $|\phi^{(k)}\rangle = \sum_{x,y=0}^{2^k-1} \alpha_{x,y}^{(k)} |x, y\rangle$ and $|\phi^{(\log d)}\rangle = |\phi\rangle$. For these $|\Psi^{(k)}\rangle$'s, we remark that their coin spaces are all in the particular state $|\circlearrowleft\rangle$. Note that $|\phi^{(k)}\rangle$ is a quantum state of one-qudit system whose dimension is 2^k. We claim that suppose for any $|\Psi^{(k)}\rangle$, there exists operators $V^{(0)}, V^{(1)}, \dots, V^{(k-1)}$ of the form $V^{(j)} = C_2^{(j+1)} S^{2^{k-1-j}} C_1^{(j)}$ such that

$$|\Psi^{(k)}\rangle = V^{(k-1)} \dots V^{(1)} V^{(0)} |\Psi^{(0)}\rangle, \tag{29}$$

then for any given state $|\phi^{(k+1)}\rangle$, there exists operators $U^{(0)}, U^{(1)}, \dots, U^{(k)}$ of the form $U^{(j)} = C_2^{(j+1)} S^{2^{k-j}} C_1^{(j)}$ such that

$$|\Psi^{(k+1)}\rangle = U^{(k)} \cdots U^{(1)} U^{(0)} |\Psi^{(0)}\rangle . \tag{30}$$

Once this claim is proved, we can indeed reach our goal by simply letting $|\Psi^{(\log d)}\rangle$ be equal to $|\phi\rangle \otimes |\circlearrowleft\rangle$.

For $k \geqslant 0$, suppose the given state is $|\phi^{(k+1)}\rangle = \sum_{x,y=0}^{2^{k+1}-1} \alpha_{x,y}^{(k+1)} |x, y\rangle$, we construct four states starting at the four positions mentioned earlier. Denote

$$\begin{aligned}
|\phi_{0,0}^{(k)}\rangle &= \frac{1}{\sqrt{\gamma_{0,0}^{(k)}}} \sum_{x,y=0}^{2^k-1} \alpha_{x,y}^{(k+1)} |x, y\rangle , & \gamma_{0,0}^{(k)} &= \sum_{x,y=0}^{2^k-1} \left|\alpha_{x,y}^{(k+1)}\right|^2, \\
|\phi_{1,0}^{(k)}\rangle &= \frac{1}{\sqrt{\gamma_{1,0}^{(k)}}} \sum_{x,y=0}^{2^k-1} \alpha_{x+2^k,y}^{(k+1)} |x, y\rangle , & \gamma_{1,0}^{(k)} &= \sum_{x=2^k}^{2^{k+1}-1} \sum_{y=0}^{2^k-1} \left|\alpha_{x,y}^{(k+1)}\right|^2, \\
|\phi_{0,1}^{(k)}\rangle &= \frac{1}{\sqrt{\gamma_{0,1}^{(k)}}} \sum_{x,y=0}^{2^k-1} \alpha_{x,y+2^k}^{(k+1)} |x, y\rangle , & \gamma_{0,1}^{(k)} &= \sum_{x=0}^{2^k-1} \sum_{y=2^k}^{2^{k+1}-1} \left|\alpha_{x,y}^{(k+1)}\right|^2, \\
|\phi_{1,1}^{(k)}\rangle &= \frac{1}{\sqrt{\gamma_{1,1}^{(k)}}} \sum_{x,y=0}^{2^k-1} \alpha_{x+2^k,y+2^k}^{(k+1)} |x, y\rangle , & \gamma_{1,1}^{(k)} &= \sum_{x,y=2^k}^{2^{k+1}-1} \left|\alpha_{x,y}^{(k+1)}\right|^2.
\end{aligned} \tag{31}$$

According to the assumption, each of these four states can be engineered by a k-step procedure, which, for the procedure engineering $|\phi_{a,b}^{(k)}\rangle$ we denote $U_{a,b}^{(k)} \cdots U_{a,b}^{(2)} U_{a,b}^{(1)}$, for $a, b \in \{0, 1\}$, respectively, where $U_{a,b}^{(j)} = C_{2,a,b}^{(j+1)} S^{2^{k-j}} C_{1,a,b}^{(j)}$ by the definition of these operators. (Note that the indices of operators engineering $|\phi_{a,b}^{(k)}\rangle$ starts from 1 instead of 0 because they are used in step $1, \dots, k$ of the procedure engineering $|\phi^{(k+1)}\rangle$ and we want to keep these indices consistent.)

Now we show the design of the operators $U^{(j)}$ for $j = 0, 1, \dots, k$. The operator $U^{(0)} = C_2^{(1)} S^{2^k} C_1^{(0)}$ is described as follows:

$$\begin{aligned}
C_1^{(0)}(0,0)|\circlearrowleft\rangle = \sqrt{\gamma_{0,0}^{(k)}}\,|\circlearrowleft\rangle + \sqrt{\gamma_{1,0}^{(k)}}\,|{\rightarrow}\rangle \\
+ \sqrt{\gamma_{0,1}^{(k)}}\,|{\uparrow}\rangle + \sqrt{\gamma_{1,1}^{(k)}}\,|{\nearrow}\rangle
\end{aligned} \tag{32}$$

and $C_1^{(0)}(x, y) = I$ for $x > 0$ or $y > 0$. In addition $C_2^{(1)}(x, y)$ satisfies

$$\begin{aligned}
C_2^{(1)}(x, y)\,|\circlearrowleft\rangle &= |\circlearrowleft\rangle , x, y = 0, \\
C_2^{(1)}(x, y)\,|{\rightarrow}\rangle &= |\circlearrowleft\rangle , x = 2^k, y = 0, \\
C_2^{(1)}(x, y)\,|{\uparrow}\rangle &= |\circlearrowleft\rangle , x = 0, y = 2^k, \\
C_2^{(1)}(x, y)\,|{\nearrow}\rangle &= |\circlearrowleft\rangle , x = y = 2^k,
\end{aligned} \tag{33}$$

and $C_2^{(1)}(x, y) = I$ for other positions. It is easily checked that after applying operator $U^{(0)}$ on the initial state, we get the following state:

$$\begin{aligned}
\Big(\sqrt{\gamma_{0,0}^{(k)}}\,|0,0\rangle + \sqrt{\gamma_{1,0}^{(k)}}\,|2^k,0\rangle + \sqrt{\gamma_{0,1}^{(k)}}\,|0,2^k\rangle \\
+ \sqrt{\gamma_{1,1}^{(k)}}\,|2^k,2^k\rangle\Big) \otimes |\circlearrowleft\rangle .
\end{aligned} \tag{34}$$

The operator $U^{(j)} = C_2^{(j+1)} S^{2^{k-j}} C_1^{(j)}$ for $j = 1, 2, \dots, k$ satisfies

$$C_1^{(j)} = \sum_{x,y=0}^{2^{k+1}-1} |x, y\rangle\langle x, y| \otimes C_{1,\mathbf{1}_{x\geqslant 2^k},\mathbf{1}_{y\geqslant 2^k}}^{(j)} \tag{35}$$

and

$$C_2^{(j)} = \sum_{x,y=0}^{2^{k+1}-1} |x, y\rangle\langle x, y| \otimes C_{2,\mathbf{1}_{x\geqslant 2^k},\mathbf{1}_{y\geqslant 2^k}}^{(j)} , \tag{36}$$

where $\mathbf{1}_{x\geqslant 2^k}$ and $\mathbf{1}_{y\geqslant 2^k}$ are indicator functions that value 1 if and only if the conditions $x \geqslant 2^k$ and $y \geqslant 2^k$ are true, respectively. In other words, $C_1^{(j)}$ and $C_2^{(j)}$ uses different operators according to which region the position belongs to (see Fig. 3). The correctness comes naturally from assumption about the $U_{a,b}^{(j)}$'s for $a, b \in \{0, 1\}$, and the whole procedure has only $O(\log d)$ nonidentity coin operators as promised.

The algorithm is shown in Algorithm 2. Below we show that the time complexity for computing the parameters on a classical machine is $O(d^2)$. Indeed, the procedure for computing parameters according to $|\phi\rangle$ has four recursive calls each of which is of size $d/2$. It is $O(1)$ time to compute γ and $U^{(0)}$. In addition to that, it is also $O(1)$ time to compute other $U^{(j)}$ because these operators are merely a combination of the four groups of operators returned from the four recursive calls. So if the classical computing time for dimension d is $T(d)$, then the recursion satisfies $T(d) = 4T(d/2) + O(1)$ which results in $T(d) = O(d^2)$. In summary, this algorithm takes $O(d^2)$ time to compute all the coin operators used in engineering $|\phi\rangle$. In Table II, we compare these two schemes we proposed.

V. APPLICATIONS OF OUR SCHEME

In this section, we show two applications of our scheme for engineering quantum states in multipartite systems. In

ALGORITHM 2. Alternative scheme for bipartite systems

Input: a complex vector $\{\alpha_{x,y}\}$ describing the state $|\phi\rangle = \sum_{x,y=0}^{d-1} \alpha_{x,y} |x, y\rangle$ to be prepared
Output: $\{V^{(k)}\}_{k=0}^{\log d-1}$

procedure ComputeParameters $|\phi\rangle$
  $k \leftarrow$ dimension of $|\phi\rangle$
  Split $|\phi\rangle$ into four states $|\phi_{0,0}\rangle, |\phi_{0,1}\rangle, |\phi_{1,0}\rangle, |\phi_{1,1}\rangle$ according to Eq. (31)
  for $a, b \in \{0, 1\}$ **do**
    $\left[\gamma_{a,b}, \{U_{a,b}^{(j)}\}_{j=1}^{k}\right] \leftarrow$ ComputeParameters $(|\phi_{a,b}\rangle)$
  end for
  $\gamma \leftarrow \sum_{a,b\in\{0,1\}} \gamma_{a,b}$
  Compute $U^{(0)}$ according to Eqs. (32) and (33)
  Compute other $U^{(j)}$, $j = 1, 2, \ldots, k$ according to Eqs. (35) and (36)
  return $\gamma, \{U^{(j)}\}_{j=0}^{k}$
end procedure

$\left[\gamma, \{V^{(k)}\}_{k=0}^{\log d-1}\right] \leftarrow$ ComputeParameters $(|\phi\rangle)$
return $\{V^{(k)}\}_{k=0}^{\log d-1}$

Sec. V A, we illustrate our scheme by taking engineering generalized the Bell state as an example. Later in Sec. V B, we show how the quantum walk procedure obtained by our scheme can be used to generate a quantum circuit of size $O(2^n)$ and depth $O(\frac{2^n}{n})$ preparing any n-qubit state with $O(n)$ ancillary qubits, which also reaches the best quantum state preparation method accomplishing this task.

A. Taking generalized Bell state as example

The generalized Bell state, which is written as

$$|\phi_{n,m}\rangle = \frac{1}{\sqrt{d}} \sum_{j=0}^{d-1} e^{2\pi \mathrm{i} jn/d} |j\rangle \otimes |(j+m) \bmod d\rangle \tag{37}$$

for some n, m, first serves as basis for the measurements Alice used in the quantum teleportation scheme [5]. Later in some quantum key distribution (QKD) protocols, it was used as the initial shared entanglements [32]. An illustration of these states is shown in Fig. 4 where the white boxes represent for positions with amplitude zero. In this subsection, we illustrate our algorithm with this particular state. Our scheme has d steps. As mentioned in Sec. I, it is indeed possible to achieve $O(d)$ steps using the scheme in Ref. [25]. However, to achieve $O(d)$ steps, the nonzero positions must be encoded in a consecutive way. This leads to a challenge for the manipulation of the shift operator.

TABLE II. Comparison of two schemes we proposed.

	Sec. III	Sec. IV
Classical time	$O(d^2)$	$O(d^2)$
#steps	d	$O(d)$ [a]
$\#\{k, x, y : C^{(k)}(x, y) \neq I\}$ [b]	$O(d^2)$	$O(d^2)$
$\#\{k : C^{(k)} \neq I\}$ [c]	d	$\log d$
$\max_{0\leqslant k<d} \#\{x, y : C^{(k)}(x, y) \neq I\}$ [d]	$2d-1$	d^2

[a]The step is d when d is a power of 2, and will not exceed $2d$ in general.
[b]The third row counts for total nonidentity positions in all steps, and there are d^3 positions in total.
[c]The fourth row counts for nonidentity coin operators, and there are d coin operators since both of these two schemes have d steps.
[d]The fifth row counts for nonidentity positions of coin operator in a single step.

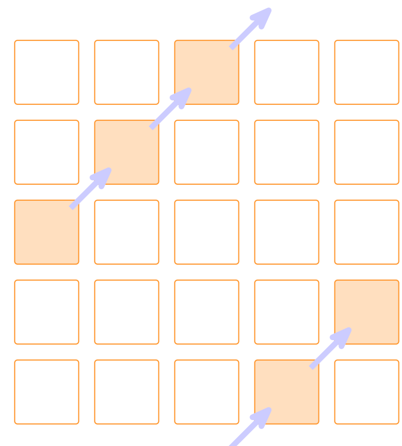


FIG. 4. Illustration of generalized Bell state with $n = 5$ and $m = 2$. White boxes represent for positions with amplitude zero. Arrows are one possible way to encode the nonzero positions onto a 1D line to utilize the scheme from Ref. [25].

We define the $|\Psi^{(k)}\rangle$'s and $|\phi^{(k)}\rangle$'s for $k = 0, 1, \ldots, d-1$ as described in Sec. III A, and will give the coin operators $C_1^{(k)}$'s and $C_2^{(k)}$'s explicitly.

Let $|\phi^{(d-1)}\rangle = |\phi_{n,m}\rangle$, the algorithm sets

$$\alpha_{x,y}^{(d-1)} = \begin{cases} e^{2\pi \mathrm{i} xn/d}/\sqrt{d}, & y \equiv x+m \pmod d, \\ 0, & \text{otherwise.} \end{cases} \tag{38}$$

For $k < d-1$, we suppose $0 \leqslant m < d$ without loss of generality. According to Eq. (19), there are three nontrivial cases we need to consider.

Case 1. $x = k, y < k$. Following Eq. (19), the only coordinates (k, y) such that $\alpha_{k,y}^{(k)}$ is not zero has its y coordinate satisfying the following constraints:

$$\begin{aligned} &y \equiv z + m \pmod d, \\ &k \leqslant z < d. \end{aligned} \tag{39}$$

That is, there exists $t \in \mathbb{Z}$ that $k \leqslant y - m - td < d$, which is

$$\frac{y-m}{d} - 1 < t \leqslant \frac{y-m}{d} - \frac{k}{d}. \tag{40}$$

Notice that $0 \leqslant y < k$ and $0 \leqslant m < d$, so $-1 < \frac{y-m}{d} < 1$, hence the only possible value of t is 0 and -1. If $t = 0$, then

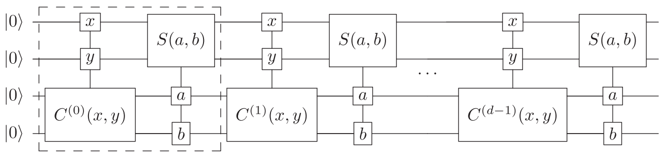


FIG. 5. Circuit of our algorithm. The controlled gate with label on its controlled space represents a "uniformly controlled gate." That is, when the state of the position space is $|x, y\rangle$, $C^{(k)}(x, y)$ is executed on the coin space.

$y - m \geqslant k$, which contradicts $y < k$; otherwise $t = -1$, and the constraint $k + m - d \leqslant y < m$ must be satisfied.

Case 2. $x < k, y = k$. Similar to case 1, the only coordinates (x, k) such that $\alpha^{(k)}_{x,k}$ is not zero has its x coordinate satisfying the following constraints:

$$\begin{aligned} w &\equiv x + m \pmod d, \\ k &\leqslant w < d. \end{aligned} \tag{41}$$

That is, there exists $t \in \mathbb{Z}$ that $k \leqslant x + m + td < d$, which is

$$\frac{-x-m}{d} + \frac{k}{d} \leqslant t < \frac{-x-m}{d} + 1. \tag{42}$$

Notice again that $0 \leqslant x < k$ and $0 \leqslant m < d$, so $-2 < \frac{-x-m}{d} \leqslant 0$, hence t can only be 0 or -1. If $t = -1$, then $x \geqslant k - m + d$, which leads to a contradiction; otherwise $t = 0$, and the constraint $k - m \leqslant x < d - m$ must be satisfied.

Case 3. $x = k, y = k$. According to Eq. (19), we need to count the number of integer solutions of the constraints below:

$$\begin{aligned} w &\equiv z + m \pmod d, \\ k &\leqslant z < d, \\ k &\leqslant w < d. \end{aligned} \tag{43}$$

Fix a particular z such that $k \leqslant z < d$. The same as the analysis in case 2, there is exactly one integer w such that (z, w) satisfies these constraints when $k \leqslant z < d - m$ or $k - m + d \leqslant z < d$. When $d - m \leqslant z < k - m + d$, there's no such w. Define a function $\delta : \mathbb{R} \to \mathbb{R}$ as

$$\delta(z) = \begin{cases} 1, & z \geqslant 0, \\ 0, & z < 0. \end{cases} \tag{44}$$

Denote the function $u : \mathbb{R} \to \mathbb{R}$ as $u(z) = z\delta(z)$. Then the total number of integer solutions is $u(d - m - k) + u(m - k)$, which is denoted $\sigma_m(k)$.

To summarize, for $k < d$ we have

$$\alpha^{(k)}_{x,y} = \begin{cases} \alpha_{x,y}, & x, y < k, \\ \frac{1}{\sqrt{d}}, & x = k, k + m - d \leqslant y < \min\{m, k\}, \\ \frac{1}{\sqrt{d}}, & k - m \leqslant x < \min\{d - m, k\}, y = k, \\ \sqrt{\frac{\sigma_m(k)}{d}}, & x = y = k, \\ 0, & \text{otherwise}. \end{cases} \tag{45}$$

One can obtain the corresponding coin operators via plugging the $\alpha^{(k)}_{x,y}$'s into Eq. (16). Notice that for each $C^{(k)}_1(x, y) \in \mathbb{C}^{4\times4}$, Eq. (16) has only four constraints, so he may pick the unitaries that are relatively simple. Here we give one possible version of the coin operators $C^{(k)}_1(x, y)$ for $k < d$ of preparing generalized Bell states directly:

$$C^{(k)}_1(x, y) = \begin{cases} e^{2\pi i kn/d} I \otimes I, & x = k, y = k + m - d, \\ X \otimes I, & x = k, k + m - d < y < \min\{m, k\}, \\ e^{2\pi i (k-m)n/d} I \otimes I, & x = k - m, y = k, \\ I \otimes X, & k - m < x < \min\{d - m, k\}, y = k, \\ D^{(k)}, & x = y = k, \\ I \otimes I, & \text{otherwise}, \end{cases} \tag{46}$$

in which I, X, Y are Pauli matrices and $D^{(k)}$ satisfies

$$D^{(k)} |\circlearrowleft\rangle = \begin{cases} \sqrt{\frac{1}{d-k}} e^{2\pi i kn/d} |\circlearrowleft\rangle + \sqrt{1 - \frac{1}{d-k}} |\nearrow\rangle, & m = 0, \\ \sqrt{\frac{\delta(m-k-1)}{\sigma_m(k)}} |\rightarrow\rangle + \sqrt{\frac{\delta(d-m-k-1)}{\sigma_m(k)}} |\uparrow\rangle + \sqrt{\frac{\sigma_m(k+1)}{\sigma_m(k)}} |\nearrow\rangle, & m \neq 0. \end{cases} \tag{47}$$

The operator $C^{(k)}_2(x, y)'$s can be determined by Eq. (17).

B. Relationship with quantum state preparation

For quantum computation, both quantum walk and quantum circuit are universal computing models. Quantum state preparation is a well-studied problem under circuit model, which aims to generate a given state $|\phi\rangle = \sum_{x=0}^{2^n-1} \alpha_x |x\rangle$ from $|0\rangle^{\otimes n}$ by a quantum circuit. In general, this problem is of great essence and a series of works have been devoted to reducing the resources consuming such as circuit size, circuit depth, and the number of ancillary qubits [33–37]. In particular, when one has $O(n)$ ancillary qubits, Sun *et al.* gave quantum circuits preparing any n-qubit states of $O(2^n)$ size and $O(\frac{2^n}{n})$ depth that

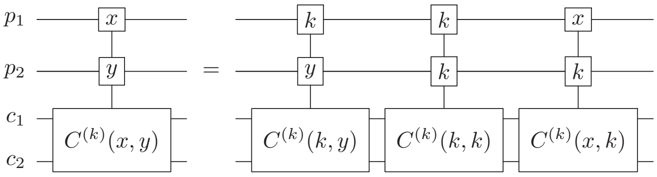


FIG. 6. $C^{(k)}$ can be treated as a UCG (left) and further decomposed into three UCG's (right).

reaches the lower bounds of size and depth simultaneously [37].

Now we show how our quantum engineering procedures via quantum walk can be used to design the quantum circuit for quantum state preparation with the same performance. Given a quantum state preparation instance of size n, we first divide the n qubits into two parts of $\lceil 0.5n \rceil$ each and treat the two parts as a bipartite system of dimension $2^{\lceil 0.5n \rceil}$ (by adding an ancillary qubit if n is odd). Then we utilize our scheme introduced in Sec. III on this system. This results in a quantum circuit with mainly two kinds of operators: the coin operators $C^{(k)}$ and the shift operators S. We need to decompose them into single-qubit gates and CNOT gates.

The decomposition of coin operators is mainly based on a method decomposing a uniformly controlled gate (UCG) with n-controlled qubits and constant number of targets into quantum circuit with size $O(2^n)$ and depth $O(\frac{2^n}{n})$ [37]. The coin operators $C^{(k)}$ can be seen as UCG's: as shown in Fig. 5, when the state of the position space is $|x, y\rangle$, $C^{(k)}(x, y)$ is executed on the coin space. Utilizing the techniques mentioned above directly, each coin operator can be implemented by a quantum circuit of size $O(2^n)$, but this makes the entire circuit too big. In fact, we can do much better by using the equivalence shown in Fig. 6, in which the controlled $C^{(k)}(k, y)$, $C^{(k)}(x, k)$ gates can be treated as one-qubit controlled UCG's. Using the technique from [37], these UCG's can be decomposed into a quantum circuit of $O(2^{0.5n})$ CNOT and single-qubit gates which has $O(\frac{2^{0.5n}}{n})$ depth. As shown in Fig. 7, this results in a quantum circuit of $O(2^{0.5n})$ Toffoli and two-qubit gates which has $O(\frac{2^{0.5n}}{n})$ depth. Each Toffoli and two-qubit gate can be decomposed into CNOT and single-qubit gates of constant size and depth. Thus, the overall size and depth of the decomposition of $C^{(k)}(x, y)$ is $O(2^{0.5n})$ and $O(\frac{2^{0.5n}}{n})$, respectively.

The shift operator S in this setting can be seen as controlled quantum adders. If one views the position space as two quantum registers, then different states of coin space control the "add one" operation on different registers. To be concrete, the first (second) register is added by one when the first (second) coin qubit is in state $|1\rangle$. The quantum adder is a well-studied object [44–46]. In our application, each shift operator S consists of two controlled "add one" operation on

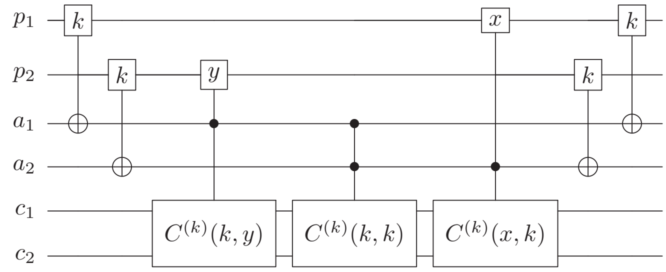


FIG. 7. Further transformation of the coin operator $C^{(k)}$. Two clean ancillary qubits a_1 (belonging to the first side of the system) and a_2 (belonging to the second side) are introduced. The initial states of the two ancillary qubits are both $|0\rangle$, and they end up with $|0\rangle$ as well.

$\lceil 0.5n \rceil$ qubits, and this can be realized by a quantum circuit of size $O(n)$ and depth $O(\log n)$ [46].

To sum all these costs, according to Fig. 5, there are $2^{\lceil 0.5n \rceil}$ coin operators each of size $O(2^{0.5n})$ and depth $O(\frac{2^{0.5n}}{n})$, and $2^{\lceil 0.5n \rceil}$ shift operators each of size $O(n)$ and depth $O(\log n)$. Thus, the overall size and depth of the circuit is $O(2^n)$ and $O(\frac{2^n}{n})$, which matches the result proposed in Ref. [37] as promised.

VI. DISCUSSION

In this work, we put forward two schemes for engineering arbitrary multipartite high-dimensional quantum states via quantum walks with time- and position-dependent coins, and discuss several potential areas that our scheme can have applications in. In fact, it can be applied in other quantum information protocols where a particular multipartite high-dimensional entangled state is needed. Also, it raises theoretical support for potential physical experiments of the quantum walk in multipartite systems since the both time- and position-dependent coin operation has been realized in real physical systems [31,43]. However, we believe that this engineering task can also be accomplished by quantum walk with position-dependent coins which provide enough degree of freedom, thus it deserves further exploration. In general, we hope that this is a possible direction toward better understanding of the quantum information involved in multipartite entanglements.

ACKNOWLEDGMENTS

This work was supported, in part, by the National Natural Science Foundation of China Grants No. 62325210, and No. 62301531, and the Strategic Priority Research Program of Chinese Academy of Sciences Grant No. XDB28000000, and China Postdoctoral Science Foundation Grant No. 2022M723209.

[1] Y. Wang, Z. Hu, B. C. Sanders, and S. Kais, Qudits and high-dimensional quantum computing, Front. Phys. **8**, 589504 (2010).

[2] H.-H. Lu, Z. Hu, M. S. Alshaykh, A. J. Moore, Y. Wang, P. Imany, A. M. Weiner, and S. Kais, Quantum phase estimation with time-frequency qudits in a single photon, Adv. Quantum Technol. **3**, 1900074 (2020).

[3] Y. Chi, Y. Yu, Q. Gong, and J. Wang, High-dimensional quantum information processing on programmable integrated photonic chips, Sci. China Inf. Sci. **66**, 180501 (2023).

[4] L. Amico, R. Fazio, A. Osterloh, and V. Vedral, Entanglement in many-body systems, Rev. Mod. Phys. **80**, 517 (2008).
[5] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels, Phys. Rev. Lett. **70**, 1895 (1993).
[6] T. Durt, D. Kaszlikowski, J.-L. Chen, and L. C. Kwek, Security of quantum key distributions with entangled qudits, Phys. Rev. A **69**, 032313 (2004).
[7] S. P. Walborn, D. S. Lemelle, M. P. Almeida, and P. H. S. Ribeiro, Quantum key distribution with higher-order alphabets using spatially encoded qudits, Phys. Rev. Lett. **96**, 090501 (2006).
[8] L. Sheridan and V. Scarani, Security proof for quantum key distribution using qudit systems, Phys. Rev. A **82**, 030301(R) (2010).
[9] A. Keet, B. Fortescue, D. Markham, and B. C. Sanders, Quantum secret sharing with qudit graph states, Phys. Rev. A **82**, 062315 (2010).
[10] A. M. Childs, Universal computation by quantum walk, Phys. Rev. Lett. **102**, 180501 (2009).
[11] A. M. Childs, D. Gosset, and Z. Webb, Universal computation by multiparticle quantum walk, Science **339**, 791 (2013).
[12] N. B. Lovett, S. Cooper, M. Everitt, M. Trevers, and V. Kendon, Universal quantum computation using the discrete-time quantum walk, Phys. Rev. A **81**, 042330 (2010).
[13] M. S. Underwood and D. L. Feder, Universal quantum computation by discontinuous quantum walk, Phys. Rev. A **82**, 042304 (2010).
[14] A. Ambainis, Quantum walk algorithm for element distinctness, SIAM J. Comput. **37**, 210 (2007).
[15] F. Magniez, M. Santha, and M. Szegedy, Quantum algorithms for the triangle problem, SIAM J. Comput. **37**, 413 (2007).
[16] A. Ambainis, A. Gilyén, S. Jeffery, and M. Kokainis, Quadratic speedup for finding marked vertices by quantum walks, in *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing* (ACM, New York, 2020), pp. 412–424.
[17] P. Xue, B. C. Sanders, and D. Leibfried, Quantum walk on a line for a trapped ion, Phys. Rev. Lett. **103**, 183602 (2009).
[18] H. Schmitz, R. Matjeschk, Ch. 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).
[19] C. Chen, X. Ding, J. Qin, Y. He, Y.-H. Luo, M.-C. Chen, C. Liu, X.-L. Wang, W.-J. Zhang, H. Li, L.-X. You, Z. Wang, D.-W. Wang, B. C. Sanders, C.-Y. Lu, and J.-W. Pan, Observation of topologically protected edge states in a photonic two-dimensional quantum walk, Phys. Rev. Lett. **121**, 100502 (2018).
[20] H. Tang, X.-F. Lin, Z. Feng, J.-Y. Chen, J. Gao, K. Sun, C.-Y. Wang, P.-C. Lai, X.-Y. Xu, Y. Wang *et al.*, Experimental two-dimensional quantum walk on a photonic chip, Sci. Adv. **4**, eaat3174 (2018).
[21] Z. Yan, Y.-R. Zhang, M. Gong, Y. Wu, Y. Zheng, S. Li, C. Wang, F. Liang, J. Lin, Y. Xu *et al.*, Strongly correlated quantum walks with a 12-qubit superconducting processor, Science **364**, 753 (2019).
[22] M. Gong, S. Wang, C. Zha, M.-C. Chen, H.-L. Huang, Y. Wu, Q. Zhu, Y. Zhao, S. Li, S. Guo *et al.*, Quantum walks on a programmable two-dimensional 62-qubit superconducting processor, Science **372**, 948 (2021).
[23] C. M. Chandrashekar, R. Srikanth, and R. Laflamme, Optimizing the discrete time quantum walk using a su (2) coin, Phys. Rev. A **77**, 032326 (2008).
[24] H. Majury, J. Boutari, E. O'Sullivan, A. Ferraro, and M. Paternostro, Robust quantum state engineering through coherent localization in biased-coin quantum walks, EPJ Quantum Technol. **5**, 1 (2018).
[25] L. Innocenti, H. Majury, T. Giordani, N. Spagnolo, F. Sciarrino, M. Paternostro, and A. Ferraro, Quantum state engineering using one-dimensional discrete-time quantum walks, Phys. Rev. A **96**, 062326 (2017).
[26] T. Giordani, E. Polino, S. Emiliani, A. Suprano, L. Innocenti, H. Majury, L. Marrucci, M. Paternostro, A. Ferraro, N. Spagnolo *et al.*, Experimental engineering of arbitrary qudit states with discrete-time quantum walks, Phys. Rev. Lett. **122**, 020503 (2019).
[27] A. Suprano, D. Zia, E. Polino, T. Giordani, L. Innocenti, A. Ferraro, M. Paternostro, N. Spagnolo, and F. Sciarrino, Dynamical learning of a photonics quantum-state engineering process, Adv. Photon. **3**, 066002 (2021).
[28] G. Kadiri, Steered discrete-time quantum walks for engineering of quantum states, Phys. Rev. A **108**, 012607 (2023).
[29] M. Montero, Classical-like behavior in quantum walks with inhomogeneous, time-dependent coin operators, Phys. Rev. A **93**, 062316 (2016).
[30] M. Montero, Quantum and random walks as universal generators of probability distributions, Phys. Rev. A **95**, 062326 (2017).
[31] R. Zhang, R. Yang, J. Guo, C.-W. Sun, Y.-C. Liu, H. Zhou, P. Xu, Z. Xie, Y.-X. Gong, and S.-N. Zhu, Arbitrary coherent distributions in a programmable quantum walk, Phys. Rev. Res. **4**, 023042 (2022).
[32] V. Karimipour, A. Bahraminasab, and S. Bagherinezhad, Quantum key distribution for d-level systems with generalized bell states, Phys. Rev. A **65**, 052331 (2002).
[33] L. Grover and T. Rudolph, Creating superpositions that correspond to efficiently integrable probability distributions, arXiv:quant-ph/0208112.
[34] 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).
[35] M. Plesch and Č. Brukner, Quantum-state preparation with universal gate decompositions, Phys. Rev. A **83**, 032302 (2011).
[36] X.-M. Zhang, M.-H. Yung, and X. Yuan, Low-depth quantum state preparation, Phys. Rev. Res. **3**, 043200 (2021).
[37] X. Sun, G. Tian, S. Yang, P. Yuan, and S. Zhang, Asymptotically optimal circuit depth for quantum state preparation and general unitary synthesis, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (IEEE, New York, 2023).
[38] M. C. Bañuls, C. Navarrete, A. Pérez, E. Roldán, and J. C. Soriano, Quantum walk with a time-dependent coin, Phys. Rev. A **73**, 062304 (2006).
[39] P. Xue, R. Zhang, H. Qin, X. Zhan, Z. H. Bian, J. Li, and B. C. Sanders, Experimental quantum-walk revival with a time-dependent coin, Phys. Rev. Lett. **114**, 140502 (2015).

[40] A. Suzuki, Asymptotic velocity of a position-dependent quantum walk, Quantum Info. Proc. **15**, 103 (2016).

[41] R. Ahmad, U. Sajjad, and M. Sajid, One-dimensional quantum walks with a position-dependent coin, Commun. Theor. Phys. **72**, 065101 (2020).

[42] P. Kurzyński and A. Wójcik, Quantum walk as a generalized measuring device, Phys. Rev. Lett. **110**, 200404 (2013).

[43] Z. Hou, J.-F. Tang, J. Shang, H. Zhu, J. Li, Y. Yuan, K.-D. Wu, G.-Y. Xiang, C.-F. Li, and G.-C. Guo, Deterministic realization of collective measurements via photonic quantum walks, Nat. Commun. **9**, 1414 (2018).

[44] T. G. Draper, Addition on a quantum computer, arXiv:quant-ph/0008033.

[45] S. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. P. Moulton, A new quantum ripple-carry addition circuit, arXiv:quant-ph/0410184.

[46] T. G. Draper, S. A. Kutin, E. M. Rains, and K. M. Svore, A logarithmic-depth quantum carry-lookahead adder, Quantum Inf. Comput. **6**, 351 (2006).