Resource-efficient universal photonic processors based on time-multiplexed hybrid architectures

Jonas Lammers,[1,*] Laura Ares,[2] Federico Pegoraro,[1] Philip Held,[1] Benjamin Brecht,[1] Jan Sperling,[2] and Christine Silberhorn[1]

[1] *Paderborn University, Integrated Quantum Optics, Institute for Photonic Quantum Systems (PhoQS), Warburger Str. 100, 33098 Paderborn, Germany*

[2] *Paderborn University, Theoretical Quantum Science, Institute for Photonic Quantum Systems (PhoQS), Warburger Str. 100, 33098 Paderborn, Germany*

(Received 26 September 2025; revised 29 January 2026; accepted 2 April 2026; published 5 May 2026)

For the ever-growing field of quantum information processing, large-scale, efficient multiport interferometers serving as photonic processors are required. In this context, the suitability of quantum walks as the interferometric base for universal computation has been theoretically proven. In this work, we bridge the gap between theoretical proposals and state-of-the-art experimental capabilities by providing the recipe for the implementation of a universal photonic processor in discrete-time quantum walks. Specifically, we present the protocol for translating arbitrary linear transformations into the coin and step operator of a quantum walk and map these to the experimental parameters of the established time-multiplexed platform [A. Schreiber *et al.*, Phys. Rev. Lett. **104**, 050502 (2010)]. We show that our interface is highly scalable and resource efficient due to the hybrid encoding consisting of multiple degrees of freedom. Finally, we prove that our system is highly resilient against experimental imperfections and show that it compares favorably against existing architectures.

DOI: 10.1103/x99y-2sms

I. INTRODUCTION

In the field of optical quantum information processing, most protocols rely on the interference between photons in a multiport interferometer. Universal interferometers, capable of implementing any linear transformation between a large number of optical modes, also referred to as photonic processors, are at the heart of photonic quantum computation schemes [1–3]. They are also powerful tools for optical networking and communication [4–6], forming the bases for distributed quantum computation based on hybrid systems [7]. With the recent development around boson sampling and Gaussian boson sampling [8–12], the scaling of photonic processors towards a larger number of modes has become an important issue. To be useful for most applications, photonic processors need to fulfill three criteria: exhibit low loss, involve a high number of optical modes, and implement arbitrary unitary transformations with high fidelity.

In recent years, multiple platforms for realizing photonic processors have emerged. The most prevalent platform utilizes spatial encoding, where photonic processors are based on a network of beam splitters and phase shifters following the configuration by either Reck *et al.* [13] or Clements *et al.* [14]. While these schemes have shown high fidelity and low loss in integrated photonic chips [15–17], the number of components required scales quadratically with the number of modes. In order to overcome this scaling problem, research has shifted towards using different encoding schemes, resulting in two separate approaches. The first approach is based on time-frequency-bin encoding [18–20], allowing for monolithic implementations that reduce the number of required components to one. Monolithic photonic frequency processors are generally based on nonlinear conversion processes, which suffer in current state-of-the-art demonstration from low efficiencies and requiring high spectral resolution [21]. The second approach to reducing the number of components is to reuse them in a loop architecture, where the underlying mode structure is encoded in time bins [22–24]. Using protocols based purely on time bins allows for nonuniversal photonic processors with extremely high number of modes and low losses [10], as well as for universal systems with high losses [11]. A promising way to solve this tradeoff between universality and loss is time-multiplexed

*Contact author: jonas.lammers@uni-paderborn.de

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

hybrid systems combining multiple degrees of freedom in a time-multiplexed architecture. Its feasibility has been theoretically and experimentally pointed out, for instance, in Refs. [22,25,26].

In the adjacent field of quantum simulation, generalized hybrid systems, namely quantum walks, have been a long-established platform based on large-scale interferometry. Quantum walks, as originally developed by Y. Aharonov [27], analyze the spread of one or more photons—typically referred to as walkers—over a large graph structure. They are hybrid systems where one subsystem—typically referred to as position—represents the graph's vertices while the other—typically referred to as coin—controls the transitions between vertices. Using the underlying graph representation, it has been proven that quantum walks can serve as a platform for gate-based universal computation [28,29]. Furthermore, combined with measurement-induced nonlinearities and single photons, quantum walks are capable of implementing both single-qubit and two-qubit gates in the form of C-NOT gates [30].

As the aforementioned protocols for universal quantum computation strive to implement quantum gates between qubits encoded using the graph's vertices, they neglect the coin subsystem for mode encoding, which worsens scalability. Furthermore, for a universal photonic processor, we require a protocol (typically referred to as the compiler protocol) capable of translating arbitrary linear transformations $\hat{U}_{\text{Target}}$ into the individual photonic processor parameters that should be programmed. To our knowledge, there is currently no compiler protocol that allows the use of discrete-time quantum walks as a universal processor in such a way that both the coin and position degrees of freedom are used to encode the underlying mode structure.

In this work, we show how a discrete-time quantum walk can be directly used as a universal photonic processor. We provide the compiler protocol, translating arbitrary linear transformations into their smallest footprint in a quantum walk. Furthermore, we introduce a setup scheme for quantum walks in the time-multiplexed domain that can perform as universal processors by applying our findings. We prove that, in our scheme, transition probabilities are completely immune to most expected experimental causes of losses and phase noise. Finally, we carry out a systematic study on the influence of experimental imperfections, comparing the performance of the different state-of-the-art platforms for universal processors. We conclude that the proposed setup compares favorably with existing schemes in terms of loss and phase noise resilience.

II. QUANTUM WALK

Discrete-time quantum walks (DTQWs) describe the evolution of one or more quantum particles on a graph structure. Each walker starts localized at some graph vertex and its probability amplitude moves to one of the adjacent vertices according to its coin state, allowing for quantum interference. The state of the walker is therefore composed of its position and coin state, belonging in the composite Hilbert space $\mathscr{H} = \mathscr{H}_{\text{pos}} \otimes \mathscr{H}_{\text{coin}}$. Here, we consider a two-dimensional coin space, $\mathscr{H}_{\text{coin}} = \text{span}\{|0\rangle, |1\rangle\}$, and an infinite-dimensional position Hilbert space $\mathscr{H}_{\text{pos}} = \text{span}\{|x\rangle : x \in \mathbb{Z}\}$.

The evolution of a DTQW occurs by iteratively manipulating the coin state of the walker, followed by the transition towards the next position according to its coin state. Mathematically, we describe each iteration n as performing a coin operation followed by a shift operation. The coin operation manipulates the coin state of the walker at each position and each iteration individually,

$$\hat{C}(n) = \sum_{x\in\mathbb{Z}} |x\rangle\langle x| \otimes \sum_{p,q=0}^{1} c_{p,q}^{(x)}(n)\, |p\rangle\langle q|\,, \tag{1}$$

where $c_{p,q}^{(x)}$ represents the transition probability amplitude from the coin state $|q\rangle$ to $|p\rangle$, at position x. The shift operator transforms the position of the walker as

$$\hat{S} = \sum_{x\in\mathbb{Z}} \left(|x-1\rangle\langle x| \otimes |0\rangle\langle 0| + |x+1\rangle\langle x| \otimes |1\rangle\langle 1|\right). \tag{2}$$

Here we construct the unitary at each step $\hat{U}(n)$ as firstly applying the shift and then the coin operation, as this matches the proposed experimental setup more closely. Note that the following treatment can be performed equivalently for a unitary operator that firstly applies the coin and then the shift operation. Thus, the unitary evolution at each step is $\hat{U}(n) = \hat{C}(n)\hat{S}$, which, after N steps, generates the full evolution of the quantum walk as

$$\hat{U}_N = \prod_{n=1}^{N} \hat{U}(n). \tag{3}$$

The aim of this work is to implement any target unitary by a DTQW $\hat{U}_{\text{Target}} = \hat{U}_N$, where every step $\hat{U}(n)$ in Eq. (3) corresponds to one loop of our proposed experimental setup presented in Sec. V.

III. BUILDING-BLOCK DECOMPOSITION

In this section, we present the mathematical decomposition of the unitary evolution $\hat{U}_N$ that allows us to find the configuration for the experimental setup (cf. Sec. V) that will implement the target unitary U_{Target}. First, we identify the smallest unit cell of a coined quantum walk evolution such that the complete process can be represented as an iteration of this element. Next, we map this unit cell to a local network representation where it can be understood

as applying a specific sequence of nearest-neighbor operations. Using the derived local network representation, we utilize the decomposition proposed in Ref. [31] to show how to determine the individual transition amplitudes of the coin operator to realize arbitrary target unitaries. In total, this section lays the theoretical fundamentals for the compiler algorithm presented in Sec. IV.

A. Unit cell

We first decompose the quantum walk position space into even and odd components, $\mathscr{H}_{\text{pos},e} = \text{span}\{|x\rangle : x \in 2\mathbb{Z}\}$ and $\mathscr{H}_{\text{pos},o} = \text{span}\{|x\rangle : x \in 2\mathbb{Z}+1\}$, respectively. Thus, the complete Hilbert space can be written as $\mathscr{H} = (\mathscr{H}_{\text{pos},e} \oplus \mathscr{H}_{\text{pos},o}) \otimes \mathscr{H}_{\text{coin}} = \mathscr{H}_e \oplus \mathscr{H}_o$, where $\mathscr{H}_{e/o} = \mathscr{H}_{\text{pos},e/o} \otimes \mathscr{H}_{\text{coin}}$ and $\oplus$ denotes the direct sum.

Following this decomposition, we can group even and odd positions, $\{-N,\ldots,2,0,2,\ldots,N,-N+1,\ldots,-1,1,\ldots,N-1\}$, and rewrite the coin operator in a matrix form,

$$\hat{C}(n) = \begin{pmatrix} \hat{C}_e(n) & 0 \\ 0 & \hat{C}_o(n) \end{pmatrix}, \tag{4}$$

where $\hat{C}_{e/o}(n)$ acts on $\mathscr{H}_{e/o}$. The shift operator switches even to odd positions and vice versa, which generates the antidiagonal block structure

$$\hat{S} = \begin{pmatrix} 0 & \hat{S}_o \\ \hat{S}_e & 0 \end{pmatrix}. \tag{5}$$

In this matrix form, the evolution operator for each step is given by

$$\hat{U}(n) = \begin{pmatrix} 0 & \hat{C}_e(n)\hat{S}_o \\ \hat{C}_o(n)\hat{S}_e & 0 \end{pmatrix}, \tag{6}$$

which clearly shows that even and odd positions evolve independently. That is, each step switches the space of positions we are working with. To avoid this, we directly consider a two-step evolution,

$$\hat{U}(n+1)\hat{U}(n) = \begin{pmatrix} \hat{C}_e(n+1)\hat{S}_o\hat{C}_o(n)\hat{S}_e & 0 \\ 0 & \hat{C}_o(n+1)\hat{S}_e\hat{C}_e(n)\hat{S}_o \end{pmatrix}, \tag{7}$$

where even positions stay even and odd positions stay odd, completely dividing the operation space. It is therefore sufficient to study the structure of the linear transformation of a double step in one position subspace. Note that, for an odd number of steps, we can still define the unit cell until the final step, followed by a switch between even and odd positions that can be effectively implemented by a permutation operation. The resulting unit cell for the evolution of even positions in any N-step quantum walk is therefore given by $\hat{U}_{\text{cell}}(n) = \hat{C}_e(n+1)\hat{S}_o\hat{C}_o(n)\hat{S}_e$.

Furthermore, note that as even and odd positions evolve independently in the quantum walk network, they can be used to implement two parallel unitary transformations. This could be useful as this allows us, for example, to implement indefinite casual order experiments [32], through a straightforward population of both even and odd positions and interfering them after the quantum walk network.

While the unit-cell operation only transforms even-numbered positions into even-numbered positions, as seen in Eq. (7), $\hat{S}_e$ and $\hat{S}_o$ still switch even and odd positions. The next step is to transform these operators such that each operator of the individual blocks exclusively acts on one subspace. To this end, we introduce the so-called transition operator

$$\hat{M} = \sum_{x\in\mathbb{Z}} |x+1\rangle\langle x| \otimes \hat{\mathbb{1}}_{\text{coin}}, \tag{8}$$

where $\hat{\mathbb{1}}_{\text{coin}}$ is the identity matrix acting on the coin subsystem. This hermitian operator allows us to define operators which all act on the same subspace, $\mathscr{H}_e$,

$$\hat{U}_{\text{cell}}(n) = \underbrace{\hat{C}_e(n+1)}_{\hat{C}_e(n+1)} \underbrace{\hat{S}_o\hat{M}}_{\hat{S}'_o} \underbrace{\hat{M}^\dagger\hat{C}_o(n)\hat{M}}_{\hat{C}'_o(n)} \underbrace{\hat{M}^\dagger\hat{S}_e}_{\hat{S}'_e}. \tag{9}$$

In summary, we have constructed a two-step unit cell of an N-step evolution, which acts and is composed of operators that only act on the even-numbered position space $\mathscr{H}_e$. The next step is to find the underlying structure of this unit cell that enables us to map an N-step evolution to a local network.

B. Local network representation

As we have seen above, the space of states in a coined quantum walk has a bipartite structure where we can identify coin and position degrees of freedom as the subsystems. Once we restrict to the subspace of even positions, we introduce the bijective function $f : 2\mathbb{Z}\otimes\{0,1\} \to \mathbb{Z}, (x,p) \overset{f}{\mapsto} x+p$, serving as an isomorphism between spaces.

Working in the transformed Hilbert space $f(\mathscr{H}_e)$, we now introduce a transition operation $\hat{M}' = \sum_{z\in\mathbb{Z}} |z\rangle\langle z+1|$ and insert it into our unit cell as follows:

$$\hat{U}_{\text{cell}}(n) = \hat{C}_e(n+1) \underbrace{\hat{S}'_o\hat{M}'}_{\hat{S}''_o} \underbrace{(\hat{M}')^\dagger\hat{C}'_o(n)\hat{M}'}_{\hat{C}''_o(n)} \underbrace{(\hat{M}')^\dagger\hat{S}'_e}_{\hat{S}''_e}. \tag{10}$$

The resulting operators, representing our unit cell, can now be written as

$$\begin{aligned}
\hat{C}_e(n) &= \sum_{x\in 2\mathbb{Z}} \sum_{p,q=0}^{1} c_{p,q}^{(x)}(n)\,|x+p\rangle\,\langle x+q|\,,\\
\hat{C}_o''(n) &= \sum_{x\in 2\mathbb{Z}} \sum_{p,q=0}^{1} c_{p,q}^{(x+1)}(n)\,|x+1+p\rangle\,\langle x+1+q|\,,\\
\hat{S}_o'' &= \sum_{x\in 2\mathbb{Z}} \sum_{p=0}^{1} |x+2-p\rangle\,\langle x+1+p|\,,\\
\hat{S}_e'' &= \sum_{x\in 2\mathbb{Z}} \sum_{p=0}^{1} |x+1-p\rangle\,\langle x+p|\,,
\end{aligned} \tag{11}$$

where all operations act only on neighboring modes, resulting in a local network. In order to better understand the structure of this local network, we decompose the coin operators into a sequence of beam-splitter operators acting on modes a and b,

$$\begin{aligned}
\hat{T}_{a,b}(n) &= \sum_{z\in\mathbb{Z}\setminus\{a,b\}} |z\rangle\,\langle z| + \sum_{i,j\in\{a,b\}} T_{a,b}^{(i,j)}(n)\,|i\rangle\,\langle j|\\
&= \begin{pmatrix}
\ddots & \vdots & \vdots & \vdots & \vdots & \\
\cdots & 1 & 0 & 0 & 0 & \cdots\\
\cdots & 0 & T_{a,b}^{(a,a)}(n) & T_{a,b}^{(a,b)}(n) & 0 & \cdots\\
\cdots & 0 & T_{a,b}^{(b,a)}(n) & T_{a,b}^{(b,b)}(n) & 0 & \cdots\\
\cdots & 0 & 0 & 0 & 1 & \cdots\\
 & \vdots & \vdots & \vdots & \vdots & \ddots
\end{pmatrix},
\end{aligned} \tag{12}$$

with $b = a+1$ and $a \in \mathbb{Z}$. This allows us to rewrite the coin operators as

$$\begin{aligned}
\hat{C}_e''(n) &= \prod_{x\in 2\mathbb{Z}} \hat{T}_{x,x+1}(n),\\
\hat{C}_o''(n) &= \prod_{x\in 2\mathbb{Z}} \hat{T}_{x+1,x+2}(n),
\end{aligned} \tag{13}$$

with $T_{a,a+1}^{(i,j)}(n) = c_{i-a,j-a}^{(a)}(n)$.

Further, the shift operators are permutation operations where, for all $x \in 2\mathbb{Z}$, $\hat{S}_o''$ routes $x+1 \leftrightarrow x+2$, while $\hat{S}_e''$ routes $x \leftrightarrow x+1$. We can also represent these shift operations as next-neighbor beam splitters

$$\begin{aligned}
\hat{S}_e'' &= \prod_{x\in 2\mathbb{Z}} \hat{T}_{x,x+1}(n),\\
\hat{S}_o'' &= \prod_{x\in 2\mathbb{Z}} \hat{T}_{x+1,x+2}(n),
\end{aligned} \tag{14}$$

where each beam splitter implements a Pauli X gate. This further implies that they can be merged with the beam-splitter operations of the coin operators, changing the individual weights without changing the structure of the interferometer.

In total, we decompose the unit-cell operation as

$$\hat{U}_{\text{cell}}(n) = \prod_{x\in 2\mathbb{Z}} \hat{T}_{x,x+1}(n+1) \prod_{y\in 2\mathbb{Z}} \hat{T}_{y+1,y+2}(n), \tag{15}$$

implementing a local network with an infinite dimensionality.

Finally, we introduce reflective boundary conditions to map the infinite-dimensional quantum walk evolution $\hat{U}_N$, acting on $f(\mathscr{H}_e)$, to the target unitary $\hat{U}_{\text{Target}}$, acting on $\mathbb{C}^K$. For this purpose, we identify K modes by choosing $\lfloor K/2 \rfloor$ positions $\{0, 2, \ldots, 2\lfloor K/2 \rfloor - 2\}$ that are populated by both coin modes, resulting in the mode set $Z = f(\{|0\rangle, |2\rangle, \ldots, |2\lfloor K/2 \rfloor - 2\rangle\} \otimes \{|0\rangle, |1\rangle\}) = \{|0\rangle, |1\rangle, \ldots, |2\lfloor K/2 \rfloor - 1\rangle\}$ with $|Z| = 2\lfloor K/2 \rfloor$. This choice results in boundary cases when one of the selected modes $z \in Z$ overlaps with a nonselected mode $m \in f(\mathscr{H}_e)\backslash Z$ in time $\lfloor z/2 \rfloor = \lfloor m/2 \rfloor$. In order to keep the selected subset of modes separated, we impose reflective boundary conditions by implementing the Pauli X operation using the coin operation at these time bins. The resulting beam-splitter operation $\hat{T}_{z,m}$ implements the identity and will therefore be ignored in what follows for the programming of the network.

An example of the resulting network structure for a 6×6 target unitary is displayed in Fig. 1. The sequence of beam-splitter operations that defines our quantum walk evolution unitary is

$$\begin{aligned}
\hat{U}_N = {} & \hat{T}_{1,2}(0)\hat{T}_{3,4}(0)\hat{T}_{0,1}(1)\hat{T}_{2,3}(1)\hat{T}_{4,5}(1)\hat{T}_{1,2}(2)\hat{T}_{3,4}(2)\\
& \times \hat{T}_{0,1}(3)\hat{T}_{2,3}(3)\hat{T}_{4,5}(3)\hat{T}_{1,2}(4)\hat{T}_{3,4}(4) = \hat{U}_{\text{Target}},
\end{aligned}$$

where we already filtered out the coin operations like $\hat{T}_{-1,0}(0)$ that implement Pauli X in order to realize the boundary condition and therefore cannot be used to program the unitary.

C. Programming beam splitters

So far, we have decomposed a quantum walk into a specific sequence of local beam-splitter operations $\hat{T}_{a,b}$. The next step is to determine the values of reflectivities and

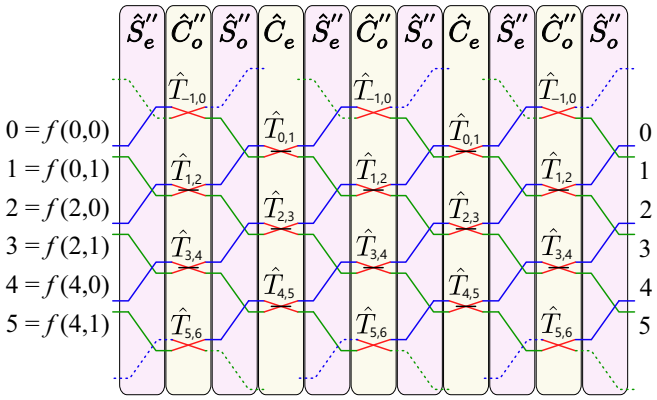

FIG. 1. Schematics of an equivalent local network interferometer for a six-step quantum walk implementing an arbitrary 6×6 unitary. Note that the resulting beam-splitter operations $\hat{T}_{-1,0}$ and $\hat{T}_{5,6}$ implement a fixed Pauli X gate realizing the required boundary conditions.

phases of each beam splitter for the specific target unitary. As a quantum walk maps to a local network of beam splitters, we can follow the proposed protocol in Ref. [31]. They show that programming a local network to implement an arbitrary unitary is equivalent to sorting a list. Specifically, they show that one can map any target unitary $\hat{U}_K \in \mathbb{C}^{K\otimes K}$ of arbitrary size K to its list of mode indices $l = \sigma(1,2,\ldots,K)$ and a bijective permutation σ. From there, programming the local network is equivalent to sorting this list, where each beam splitter allows one to swap two elements in the list. Once the list is sorted from lowest to highest mode index $(1,2,\ldots,K)$, the unitary is programmed and the compiler is completed. For more details, see the Appendix.

As derived above, our local network is composed of the beam-splitter sequence $M_{2n+1} = \left(\hat{T}_{-1,0},\ldots,\hat{T}_{2\left\lfloor\frac{K}{2}\right\rfloor-1,2\left\lfloor\frac{K}{2}\right\rfloor}\right)$ in odd steps and $M_{2n} = \left(\hat{T}_{0,1},\ldots,\hat{T}_{2\left\lfloor\frac{K}{2}\right\rfloor,2\left\lfloor\frac{K}{2}\right\rfloor+1}\right)$ in even steps. This effectively means that at every even step we can swap the modes $(j-1,j)$, while at every odd step we can swap the modes $(j,j+1)$ for $j \in \{0,2,4,\ldots,2\lfloor K/2\rfloor\}$. This matches exactly the sequence of operations required for the sorting algorithm known as parallel bubble sort [33].

Therefore, programming our system to implement an arbitrary unitary operation is equivalent to performing parallel bubble sort on a list. As parallel bubble sort is capable of sorting any permuted list, we have thus proven that our system is capable of implementing any unitary. Secondly, as an inverted list $l = (K, K-1,\ldots,1)$ is the worst-case scenario for the algorithm in which it requires at most $N(N-1)/2$ operations and we can map any unitary to an unsorted list, we have shown that at maximum we require $N(N-1)/2$ coin operations or N quantum walk steps. Furthermore, this means that for any other permutation, only a subset of the full N-step quantum walk is required. Note that there exists a mathematical decomposition of the target matrix, called the Bruhat decomposition, which decomposes any given matrix into its list of mode indices $l = \sigma(1,2,\ldots,K)$ and all possible permutations σ. Therefore, this decomposition can potentially find a shorter compile time and shallower quantum walk circuit for any target unitary, as demonstrated in [31].

IV. COMPILER PROTOCOL

As we have seen above, a quantum walk with a two-dimensional coin can be represented by repeatedly applying a sequence of beam splitters and routing operations. In the following, we present an algorithm to determine the values of the coin operations that need to be implemented so that the quantum walk compiles to the target $K \times K$ matrix, $\hat{U}_{\text{Target}}$.

Step 0. We select $\lfloor K/2+2\rfloor$ even-numbered positions $\{-2,0,2,4,\ldots,2\lfloor K/2\rfloor\}$. As each position has two coin modes, this results in $2\lfloor K/2+2\rfloor$ modes involved in the quantum walk evolution. We label each mode given by the position x and the coin state p as $z = x+p$. This results in the set of modes $Z = \{-2,-1,0,\ldots,2\lfloor K/2\rfloor+1\}$. Note that, in general, the compiler requires K quantum walk steps in order to implement the desired target unitary.

Step 1. We collect the beam splitter sequence M_n for each quantum walk step $n \in \{1,\ldots,K\}$.

(1) If n is odd, we retrieve the beam-splitter sequence $M_n = (\hat{T}_{-1,0},\hat{T}_{1,2},\ldots,\hat{T}_{2\lfloor K/2\rfloor-1,2\lfloor K/2\rfloor})$.
(2) If n is even, we retrieve the beam-splitter sequence $M_n = (\hat{T}_{0,1},\hat{T}_{2,3},\ldots,\hat{T}_{2\lfloor K/2\rfloor,2\lfloor K/2\rfloor+1})$.

Step 2. We decompose our target unitary $\hat{U}_{\text{Target}}$ into the two upper-diagonal matrices $\hat{A}_1$ and $\hat{A}_2$ as well as the permutation operator $\hat{F}$ such that $\hat{U}^{\dagger}_{\text{Target}} = \hat{A}_1\hat{F}\hat{A}_2$. Next, we retrieve the unsorted sequence $l = (l_0,\ldots,l_{K-1})$ from the permutation operator $\hat{F}$ such that each entry is $(F)_{j,l_j} = 1$.

Step 3. We perform the parallel bubble sort algorithm on the unsorted list l using our quantum walk beam-splitter sequence M_n. For each step $n \in \{1,\ldots,K\}$ and each beam splitter $\hat{T}_{m,m+1} \in M_n$, we then proceed as follows.

(1) If $m < 0$ or $m+1 \geq K$, then we implement boundary conditions by setting the beam-splitter operation to the identity.
(2) Else if $l_m \leq l_{m+1}$, then set the beam-splitter operation to the identity.
(3) Else if $l_m > l_{m+1}$, then swap l_m and l_{m+1}. For this, we set the beam-splitter parameters such that

$$\frac{(A_1)_{m,m+1}}{(A_1)_{m+1,m+1}} = -\frac{T^{(1,1)}_{m,m+1}}{T^{(1,0)}_{m,m+1}} = (A_1)_{m,m+1}. \tag{16}$$

We further set $(\hat{A}_1')_{m,m}, (\hat{A}_1')_{m+1,m+1} = 1$ by propagating a diagonal matrix through to $\hat{A}_2$ [31].

With the sorted list, the permutation operator $\hat{F}$ implements the identity, and we can retrieve the diagonal operation $\hat{D} = \hat{A}_1'\hat{A}_2'$.

Step 4. We reduce the footprint of the evolution by dropping all steps that implement only the identity.

Step 5. We adjust all beam-splitter operations by applying a Pauli X gate at the beginning, compensating for the permutation of the shift operator. Then we read out each beam splitter $\hat{T}_{m,m+1} \in M_n$ at each step n as the coin operations that need to programmed with $(\forall i,j \in \{0,1\}) c_{i,j}^{(m)}(n) = T_{m,m+1}^{(m+i,m+j)}(n)$.

Step 6. We perform the quantum walk evolution $\hat{U}_N$ as determined in steps 0 to 5, followed by phase shifts according to the diagonal operation $\hat{D}^\dagger$ on each of the resulting modes. The resulting unitary operation is $\hat{D}^\dagger\hat{U}_N = \hat{U}_{\text{Target}}$.

V. TIME-MULTIPLEXED ARCHITECTURE

In this section, we present an experimental scheme based on the time-multiplexed architecture [22] capable of implementing our proposed quantum-walk-based universal processor. The complete setup is displayed in Fig. 2. Note that the structure of the experimental system has already been successfully used for many applications involving DTQWs [22,34–36]. Because of the recursive structure of the quantum walk, it becomes intuitive to implement the processor in a loop architecture. We encode the two-dimensional coin degree of freedom in polarization, and the position degree of freedom is realized using time-bin encoding. The coin operation must be programmed individually for each position x at each step n, thus a high-speed programmable polarization device is needed. A suitable option would be an electro-optic modulator (EOM) operating in the megahertz regime using bulk crystals [37] and gigahertz on integrated platforms [38]. Typically, EOMs induce a controllable phase shift in the circular polarization basis, resulting in the transformation matrix

$$\hat{E}(\alpha) = \begin{pmatrix} \cos\alpha & i\sin\alpha \\ i\sin\alpha & \cos\alpha \end{pmatrix}, \tag{17}$$

where the angle $\alpha \propto V$ has a linear dependence on the applied voltage V. Using quarter-wave plates $\hat{Q}(\beta)$, we can transform this operation such that it rotates the polarization along the equator of the Bloch sphere,

$$\hat{Q}(0^\circ)\hat{E}\left(\alpha(x,n)\right)\hat{Q}(90^\circ) = \begin{pmatrix} \cos\alpha(x,n) & \sin\alpha(x,n) \\ -\sin\alpha(x,n) & \cos\alpha(x,n) \end{pmatrix}, \tag{18}$$

while applying half-wave plates $\hat{H}(\beta)$ allows for an arbitrary phase shift,

$$\hat{H}(22.5^\circ)\hat{E}(\phi(x,n))\hat{H}(22.5^\circ) = -e^{i\phi(x,n)}\begin{pmatrix} 1 & 0 \\ 0 & e^{-2i\phi(x,n)} \end{pmatrix}, \tag{19}$$

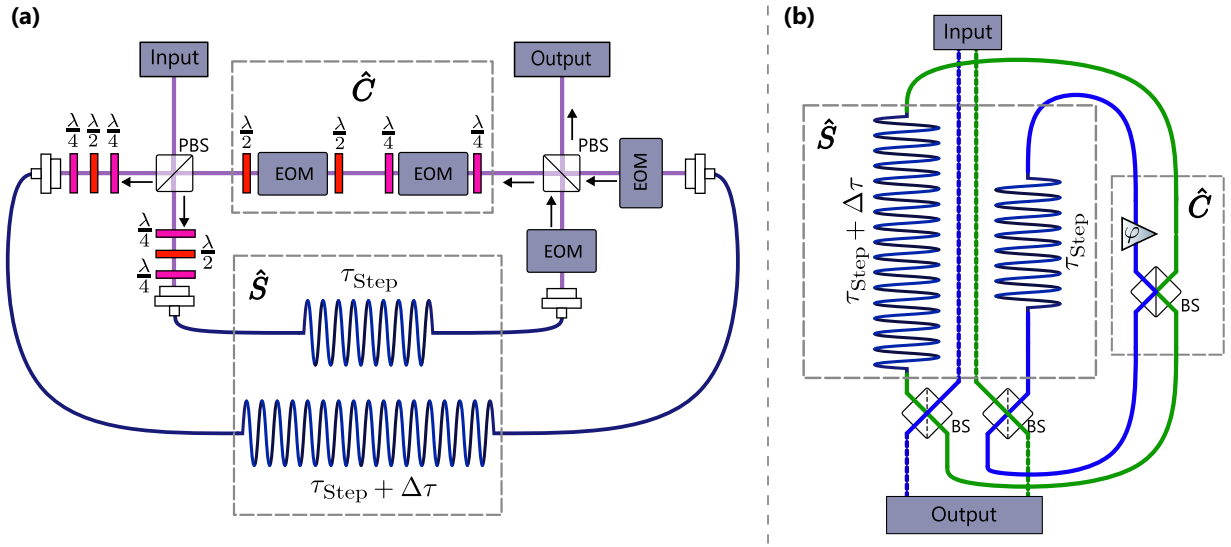

FIG. 2. Experimental implementations of a time-multiplexed quantum walk, where (a) shows a coin operation using polarization encoding, and (b) shows a coin operation implemented using dual rail encoding. Specifically, from (a) to (b), the electro-optic modulators (EOMs) for the coin translate to an arbitrary beam splitter (BS) followed by an arbitrary phase shifter (φ), and the EOMs in the unbalanced Mach-Zehnder interferometer translate to two beam splitters that can be set to complete transmission or complete reflectance for the incoupling/outcoupling.

at every temporally encoded step n and position x. Then, with two EOMs, one can construct an arbitrary beam-splitter transformation of the form

$$\hat{T}_{a,b}(x,n) = \begin{pmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \\ \cdots & 1 & 0 & 0 & 0 & \cdots \\ \cdots & 0 & \cos\alpha e^{-2i\phi} & \sin\alpha & 0 & \cdots \\ \cdots & 0 & -\sin\alpha e^{-2i\phi} & \cos\alpha & 0 & \cdots \\ \cdots & 0 & 0 & 0 & 1 & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \tag{20}$$

where $\alpha = \alpha(x,n)$ and $\phi = \phi(x,n)$. The voltage to apply to the EOMs at every step n and every position x is directly determined by the compiler algorithm presented in Sec. IV.

The shift operator is performed by an unbalanced Mach-Zehnder interferometer, implemented by two polarizing beam splitters (PBSs); see Fig. 2(a). The two polarization modes at each position enter one port of a PBS, which separates them into two optical paths with different lengths. This effectively means that the time bin for one polarization is shifted by τ, while the other is shifted by $\tau + \Delta\tau$. Finally, a second PBS is oriented so that both optical paths are recombined and forwarded towards the coin operation if the polarization is unchanged in the arms of the unbalanced Mach-Zehnder interferometer. Summing up, we can write the resulting operation explicitly as

$$\hat{S}_\tau = \sum_{t\in\Gamma} |t+\tau+\Delta\tau\rangle\langle t| \otimes |H\rangle\langle H| + |t+\tau\rangle\langle t| \otimes |V\rangle\langle V|, \tag{21}$$

where Γ is the set of all time bins. If we now shift our reference frame after each shift operator by $\tau + \Delta\tau/2$, we retrieve the desired shift operator

$$\hat{S}_T = \sum_{t\in\Gamma} \left|t+\frac{\Delta\tau}{2}\right\rangle\langle t| \otimes |H\rangle\langle H| + \left|t-\frac{\Delta\tau}{2}\right\rangle\langle t| \otimes |V\rangle\langle V|, \tag{22}$$

up to the isomorphism $g : \Gamma \to \mathbb{Z}, (t) \overset{g}{\mapsto} t/(\Delta\tau/2)$ for $\Gamma = \{x(\Delta\tau/2) : x \in \mathbb{Z}\}$. If we now pump this system with a pulse train with a pulse separation of $\Delta\tau$, we can populate up to $\tau/\Delta\tau - 1$ position modes. Note that this architecture allows us to directly utilize squeezed light from a parametric down-conversion type II process, where the signal and idler fields are encoded in polarization.

Lastly, to insert light in our quantum walk and measure its outcome, we perform the incoupling into and outcoupling from the loop by switching the polarization inside the unbalanced Mach-Zehnder interferometer. This allows us to choose the second port of the outcoupling PBS as the output, while the second port of the incoupling PBS serves as the input. Additional waveplates in each arm of the unbalanced Mach-Zehnder interferometer allow us to compensate for polarization rotations due to fiber-optical perturbations.

VI. PERFORMANCE ANALYSIS

In this section, we investigate the effect of imperfections (in particular, phase noise and losses) on the performance of the proposed system. Then we compare the resulting behavior with the performance of universal processors implemented using other prominent architectures based on spatial and temporal encoding.

A. Phase and loss resilience

We model experimental imperfections such as phase noise and photon loss using modified versions of the shift and coin operators

$$\hat{\mathscr{S}}(n) = \sum_{x\in\mathbb{Z}}\sum_{p=0}^{1} h(p,x,n)\,|x-1+2p\rangle\langle x| \otimes |p\rangle\langle p|,$$
$$\hat{\mathscr{C}}(n) = \sum_{x\in\mathbb{Z}}\sum_{p,q=0}^{1} g(p,q,x,n)c_{p,q}^{(x)}(n)\,|x\rangle\langle x| \otimes |p\rangle\langle q|, \tag{23}$$

which are, in general, no longer unitary. Here, $h(p,x,n) \in \mathbb{C}$ and $g(p,q,x,n) \in \mathbb{C}$ account for the experimental imperfections depending on the coin degree of freedom p,q, position x, and step n, and are henceforth referred to as imperfection terms. Note that losses imply that these terms have an absolute value below 1, and phase noise results in a nonzero complex phase.

To formally understand this model, we can consider the general Lindblad master equation governing the realistic process [39,40]

$$\partial_t\hat{\rho} = \frac{1}{i\hbar}\left[\hat{H},\hat{\rho}\right] - \sum_{x,p,n}\left(\frac{1}{2}\left\{\hat{L}_{x,p,n}^\dagger\hat{L}_{x,p,n},\hat{\rho}\right\} - \hat{L}_{x,p,n}^\dagger\hat{\rho}\hat{L}_{x,p,n}\right), \tag{24}$$

where $[\cdot,\cdot]$ and $\{\cdot,\cdot\}$ represent the commutator and anticommutator, respectively. The unitary evolution of the system is governed by the Hamiltonian, $\hat{H}$, while dissipation at each step n, position x, and for each coin degree of freedom p is captured through the Lindblad operators $\hat{L}_{x,p,n}$. Effectively, our loss model considers the damping term $\frac{1}{2}\left\{\hat{L}_{x,p,n}^\dagger\hat{L}_{x,p,n},\hat{\rho}\right\}$, while neglecting the jumping $\hat{L}_{x,p,n}^\dagger\hat{\rho}\hat{L}_{x,p,n}$. This is a reasonable approximation if the probability for a quantum jump is efficiently small [41]. Note that the solution where only the damping term

is considered resembles a postselection procedure, fixing the input and output photon numbers, and results in $|g(x,p,n)|, |h(x,p,n)| < 1$ in Eq. (23). Consequently the following derivation cannot be directly applied to experiments such as Gaussian boson sampling [42] if losses are present, but unleashes its potential in quantum information-processing protocols with single photons.

As we investigate a time-multiplexed system that completes N steps in the realm of microseconds [22], we can assume that temporal and acoustic fluctuations are sufficiently small not to influence our evolution. This means that our imperfections are time-insensitive for each run, thus independent of the position x and constant for all steps n. We can therefore reduce the dependence of our imperfection terms to just the coin degree of freedom, $h(p)$ and $g(p,q)$.

In addition, we assume a separable imperfection term $g(p,q) = g_o(p)g_i(q)$, where $g_{o/i}$ represents arbitrary phase shifts and losses before (i) and after (o) the ideal operation. This allows us to construct the diagonal operators $\hat{G}_{i/o} = \hat{\mathbb{1}}_{\text{pos}} \otimes \sum_{p=0}^{1} g_{i/o}(p)\,|p\rangle\langle p|$, only acting on the coin degree of freedom and thus commuting with the shift operation. In this manner, the nonideal coin operator $\hat{\mathscr{C}}(n)$ can be related to the ideal operator $\hat{C}(n)$ by means of diagonal operations,

$$\hat{\mathscr{C}}(n) = \hat{G}_i \hat{C}(n) \hat{G}_o. \tag{25}$$

As the next step, we look for a similar relation between the ideal and nonideal shift operators in terms of diagonal matrices. To this end, we reformulate the nonideal shift operator as $\hat{\mathscr{S}} = \sum_{x,y\in\mathbb{Z}} \sum_{p=0}^{1} h(p)\delta_{y,x-1+2p}\,|y\rangle\langle x| \otimes |p\rangle\langle p|$, where we introduce the Kronecker delta specifying that we only evaluate our function where $y = x - 1 + 2p$. Using this Kronecker delta, we can reformulate our imperfection term in terms of the positions x and y as

$$h(p)\delta_{p,(y+1-x)/2} = \underbrace{\sqrt{h(0)h(1)}}_{g} \underbrace{\left(\sqrt{\frac{h(0)}{h(1)}}\right)^{y-x}}_{k}, \tag{26}$$

where we can identify a global term $g \in \mathbb{C}$ and a coefficient $k \in \mathbb{C}$ as our imperfection term. Using this definition, we can write the nonideal shift operator as

$$\hat{\mathscr{S}} = g\hat{D}\hat{S}\hat{D}^{-1}, \tag{27}$$

where $\hat{D} = \sum_{z\in\mathbb{Z}} k^z\,|z\rangle\langle z| \otimes \hat{\mathbb{1}}_{\text{coin}}$ is a diagonal matrix. This shows that the nonideal shift operator can be written as the ideal shift operator plus diagonal operations and a global term. Note that the diagonal operator acts only on the position subsystem and therefore commutes with the coin operation.

Putting both the nonideal shift and coin operator together, we can construct the corresponding nonideal, nonunitary evolution operator at every step n, $\hat{\mathscr{U}}(n) = \hat{\mathscr{C}}(n)\hat{\mathscr{S}}$. After N steps, the nonideal evolution operation $\hat{\mathscr{U}}_N$ is given by

$$\begin{aligned}\hat{\mathscr{U}}_N &= \prod_{n=0}^{N}\left[\hat{\mathscr{U}}(n)\right] = \hat{G}_i \prod_{n=0}^{N}\left[\hat{C}(n)\underbrace{\hat{G}_o\hat{\mathscr{S}}\hat{G}_i}_{\hat{\mathscr{S}}'}\right]\hat{G}_i^{-1} \\ &= \hat{G}_i \prod_{n=0}^{N}\left[\hat{C}(n) g\hat{D}\hat{S}\hat{D}^{-1}\right]\hat{G}_i^{-1} \\ &= g^N \underbrace{\hat{G}_i\hat{D}}_{\hat{A}} \prod_{n=0}^{N}\left[\hat{C}(n)\hat{S}\right]\underbrace{\hat{D}^{-1}\hat{G}_i^{-1}}_{\hat{A}^{-1}} = g^N\hat{A}\hat{U}_N\hat{A}^{-1},\end{aligned} \tag{28}$$

consisting of the ideal operation $\hat{U}_N$ plus an imperfection-dependent diagonal matrix $\hat{A} \in \mathbb{C}^{N\times N}$ and global term $g^N \in \mathbb{C}$. From the explicit form of the imperfection matrix

$$\hat{A} = \sum_{x\in\mathbb{Z}} \left(\frac{h'(0)}{h'(1)}\right)^{z/2} |z\rangle\langle z| \otimes \sum_{p=0}^{1} g_i(p)\,|p\rangle\langle p| \tag{29}$$

with $h'(p) = h(p)g_i(p)g_o(p)$, we see that the experimental noise impacts the time bins z independently of the coin modes p. Under typical experimental conditions, we expect that the coin-noise process effects both polarizations nearly identically ($g_i(0) \approx g_i(1)$) as both polarizations share a common spatial path. With these considerations, the strength of our hybrid encoding consisting of multiple degrees of freedom becomes apparent as the expected impact of experimental noise is near zero on the coin subsystem. This allows us, for example, to easily perform partial state tomography independently of the experimental noise, measuring the coin state at each position.

We further analyze the implications of the diagonal noise matrices $\hat{A}$ on the implemented unitary evolution depending on the individual contributions.

If the imperfections considered only amount to a phase difference ($h'(0) = e^{i\alpha}, h'(1) = e^{i\beta}$), this results in the coefficient term $e^{i\frac{\alpha-\beta}{2z}}$. The resulting diagonal matrices apply a phase shift before and after the desired interferometer. In the framework of boson sampling, this means that for scattershot boson sampling [43] these diagonal matrices are of no consequence, while they change the considered unitary transformation for Gaussian boson sampling [42].

If the imperfections considered only amount to a loss term ($h'(0) = a, h'(1) = b$), this results in the coefficient term $k = \sqrt{a/b}$. Note that, for balanced losses $a = b$, $k = 1$ and the diagonal matrix $\hat{D}$ becomes the identity. Unbalanced losses, however, manifest themselves as a shift of

the probability at the input and output of the unitary, impacting both scattershot and Gaussian boson sampling.

B. Comparison between architectures

To conclude, we compare the performance of our proposed scheme with other known architectures in terms of fault tolerance. We chose the architectures proposed by Reck *et al.* [13] and Clements *et al.* [14] as the main candidates for spatial implementations, and the Motes-Gilchrist-Dowling-Rohde (MGDR) architecture [23] as an example of a different time-multiplexed setup.

All of the above schemes decompose the target unitary into a sequence of individual beam-splitter operations between neighboring optical modes. Therefore, imperfections mainly arise in terms of losses and phase shifts from the connection between individual beam splitters. Accordingly, we use the imperfection model where the individual beam splitters are considered ideal, while the connections introduce the experimental imperfection. This results in the beam-splitter transformation

$$\hat{T}_{a,b} \mapsto \begin{pmatrix} \alpha_o^{(a,b)} & 0 \\ 0 & \beta_o^{(a,b)} \end{pmatrix} \hat{T}_{a,b} \begin{pmatrix} \alpha_i^{(a,b)} & 0 \\ 0 & \beta_i^{(a,b)} \end{pmatrix}, \tag{30}$$

where $\alpha_{o/i}^{(a,b)}, \beta_{o/i}^{(a,b)} \in \mathbb{C}_{\leq 1}$ are arbitrary parameters representing loss and phase shifts at the input (i) and output (o) of a beam splitter for each mode a, b. Note that, for time-multiplexed systems, these imperfection parameters are assumed to be constant over all modes as the same physical components are reused in a loop during the evolution for all modes.

We characterize the effect of these imperfections by averring the fidelity $\mathscr{F}$ and similarity $\mathscr{O}$ over five hundred 20×20 unitaries, each with five hundred random patterns of imperfections chosen within a normal distribution with a standard deviation of σ. We define the fidelity as

$$\mathscr{F} = \left| \frac{\mathrm{tr}\left(\hat{\mathscr{U}}_N \hat{U}_{\mathrm{Target}}^{\dagger}\right)}{\sqrt{N}\sqrt{\mathrm{tr}\left(\hat{\mathscr{U}}_N \hat{\mathscr{U}}_N^{\dagger}\right)}} \right|^2, \tag{31}$$

where $\hat{\mathscr{U}}_N$ is the resulting transformation matrix including experimental imperfections. This standard measure of fidelity for unitaries disregards global terms, while still being sensitive to the relative phase of the individual transition amplitudes. On the other hand, the similarity, defined as

$$\mathscr{O} = \frac{\sum_{x,y=0}^{N} \left| (\mathscr{U}_N)_{x,y} \left(U_{\mathrm{Target}}\right)_{x,y} \right|}{\sqrt{\sum_{x,y=0}^{N} \left| (\mathscr{U}_N)_{x,y} \right|^2} \sqrt{\sum_{x,y=0}^{N} \left| \left(U_{\mathrm{Target}}\right)_{x,y} \right|^2}}, \tag{32}$$

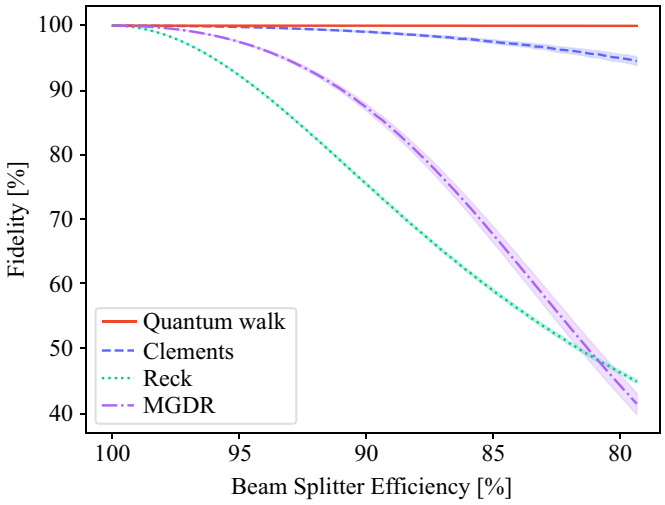

FIG. 3. Effect of randomly varying beam-splitter efficiencies with a standard deviation of $\sigma = 5\%$ on the fidelity with the target matrix. The shaded area represents one standard deviation over all averaged unitaries and loss patterns.

where we treat the individual operators as high-dimensional vectors and then calculate their overlap, is insensitive to all phase terms.

We first study the effect of losses per beam splitter on the resulting fidelity for all four architectures by sampling the imperfection terms from a Gaussian distribution. The results are displayed in Fig. 3 for varying losses with $\sigma = 5\%$ per beam splitter.

The first observation we can make is that the fidelity of the implemented unitary in the quantum walk architecture is completely independent of the average loss. In contrast, the Clements architecture becomes marginally affected, and, for the Reck and MGDR architectures the fidelity drops significantly. This difference in the effect of losses on the resulting fidelity comes from the symmetrization of the arrangement of the constituting operations. As discussed in Ref. [14], unbalanced loss resulting from unsymmetrical arrangements has a nonrecoverable impact on the resulting interference. Therefore, architectures such as the Reck and MGDR are strongly affected, while more symmetrical arrangements such as the Clements architecture are only marginally affected. Note that the Clements architecture is affected due to its border conditions, while our system implements border conditions indistinguishably from the coin operations, resulting in complete symmetry of the underlying local network.

Next, we investigate the effect of random phase shifts between individual beam splitters, with imperfection terms $|\alpha_{o/i}^{(a,b)}|^2 = |\beta_{o/i}^{(a,b)}|^2 = 1$. For this purpose, we sample each imperfection term individually by randomly selecting a phase $e^{i\varphi}$ from a Gaussian distribution with varying standard deviations. The results are displayed in Fig. 4 for all four architectures considered.

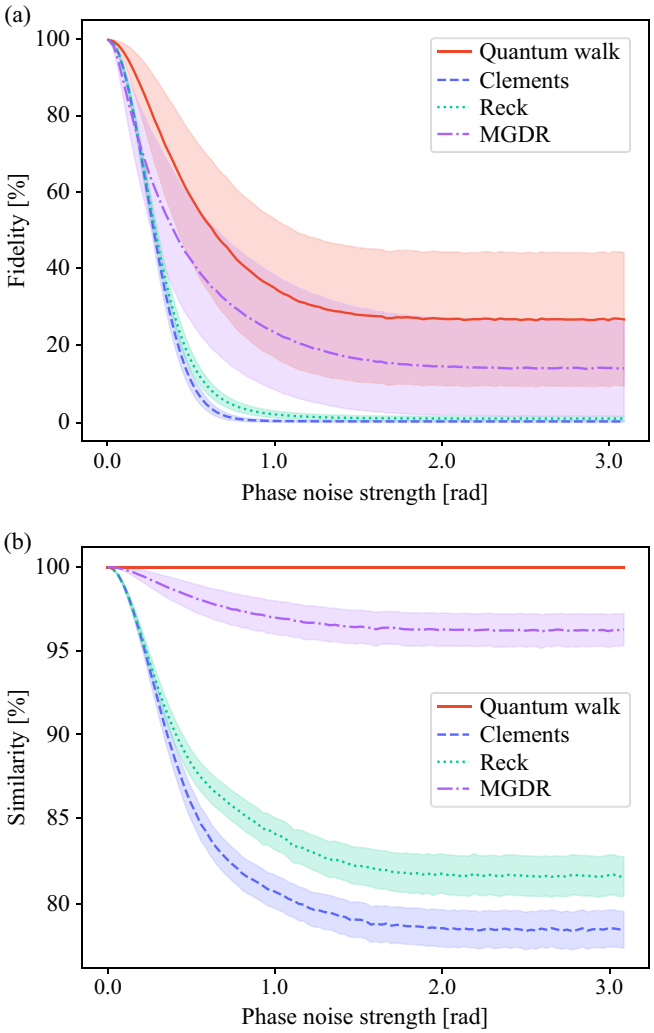

FIG. 4. Effect of random phase noise on (a) the resulting fidelity and (b) similarity, with the target matrix. The shaded area represents one standard deviation over all averaged unitaries and phase noise patterns.

Here, we can see that the similarity remains at 100%, meaning that the underlying transition amplitudes of the desired unitary can be implemented in our quantum walk architecture completely independent of phase noise. We further see that this not true for all other architectures considered, where such phase noise would need to be minimized. Finally, we see that for the phase noise considered the resulting fidelity of the implemented unitary compares favorably with all other architectures.

VII. CONCLUSIONS

In this paper, we systematically investigated the performance of a coined discrete-time quantum walk as a photonic quantum processor. Specifically, we show that for a two-dimensional coin space and reprogrammable coin operation, any $N \times N$ target unitary evolution can be realized in an N-step quantum walk. This closes the gap between theory and experiments by providing a recipe to translate arbitrary unitaries into the experimental parameters required to be implemented in a quantum walk. We proposed a time-multiplexed hybrid encoding where both the coin and position degree of freedom are exploited, requiring $\lfloor (N(N-1))/2 \rfloor$ time bins and operations, as well as a fixed number of optical components for reproducing any $N \times N$ unitary evolution. This makes the proposed system strongly scalable and more resource efficient in comparison to existing spatial [13,14] and time-multiplexed [11,23] architectures.

Furthermore, we introduced an experimental setup capable of implementing our findings and provided the compiler algorithm that translates the target evolution into the individual coin operations to be programmed. Our proposed architecture is capable of implementing two independent unitary evolutions in parallel, which could be beneficial for applications such as indefinite causal order experiments [32] or error correction protocols such as unitary averaging [44]. We further analyzed the impact of the most common experimental imperfections on the performance of the photonic quantum processor, proving that the resulting evolution of our quantum walk is completely insensitive to photon losses and highly resilient against phase noise. Moreover, we showed that it compares favorably across multiple configurations of loss and phase noise with some of the most prominent existing architectures.

ACKNOWLEDGMENTS

We acknowledge financial support by the European Commission through the Horizon Europe project EPIQUE (Grant No. 101135288). We acknowledge the financial support of the Deutsche Forschungsgemeinschaft (DFG) via the TRR 142/3 (Project No. 231447078, Subproject No. C10).

J.L., P.H., and F.P. developed the experimental platform. J.L. with the assistance of L.A. derived the protocol and theoretical proofs. J.L. and L.A. wrote the manuscript. P.H. contributed with ideas on system stability. C.S., J.S., and B.B. supervised the work. C.S. initiated the work. All authors discussed the results and commented on the manuscript.

DATA AVAILABILITY

The data that support the findings of this article are not publicly available. The data are available from the authors upon reasonable request.

APPENDIX: LOCAL NETWORK PROGRAMMING

For completeness, we provide a short summary of the result of T. G. de Brugière *et al.* [31] on the programming

of a local network to compile as an arbitrary unitary. Further, we provide a method to represent any target unitary as an inverted list in their formalism as a worst-case compiler for our photonic processor.

First, we decompose the hermitian conjugate of our target matrix into two upper-triangular matrices $\hat{A}_1, \hat{A}_2$ and two permutation matrices $\hat{P}, \hat{F}$, such that

$$\hat{U}^{\dagger}_{\text{Target}} = \hat{P}\hat{A}_1\hat{F}\hat{A}_2. \tag{A1}$$

This decomposition can be achieved by, for example, using the LU decomposition procedure decomposing $\hat{U}^{\dagger}_{\text{Target}}$ into a lower-triangular $\hat{L}_1$, upper-triangular $\hat{A}_2$, and permutation $\hat{P}$ matrix, such that $\hat{U}^{\dagger}_{\text{Target}} = \hat{P}\hat{L}_1\hat{A}_2$. From there we can transform the lower-triangular matrix $\hat{L}_1 = \hat{F}^{-1}\hat{F}\hat{L}_1\hat{F}^{-1}\hat{F}$ using the antidiagonal operator $\hat{F}$, such that $\hat{U}^{\dagger}_{\text{Target}} = \hat{P}'\hat{A}_1\hat{F}\hat{A}_2$, where $\hat{A}_1 = \hat{F}\hat{L}_1\hat{F}^{-1}$ and $\hat{P}' = \hat{P}\hat{F}^{-1}$.

Given this representation, the next step is to investigate the impact of a local beam-splitter operation $\hat{T}_{a,a+1}$ on our target matrix. Specifically, we consider

$$\hat{T}_{a,a+1}\hat{P}^{\dagger}\hat{U}^{\dagger}_{\text{Target}} = \underbrace{\hat{T}_{a,a+1}\hat{A}_1\hat{B}}_{\hat{A}'_1}\underbrace{\hat{B}^{-1}\hat{F}}_{\hat{F}'}\hat{A}_2, \tag{A2}$$

where $\hat{B}$ is an invertible matrix chosen such that $\hat{A}'_1$ is again an upper-triangular matrix and $\hat{F}'$ a permutation matrix. First of all, if the beam splitter $\hat{T}_{a,a+1}$ implements the identity, nothing changes, and the resulting permutation operator $\hat{F}' = \hat{F}$ remains unchanged. It has been proven [31] that the reflectivity and phase of any beam splitter $\hat{T}_{a,a+1}$ can be chosen such that $\hat{B}$ implements a flip between mode a and $a+1$, and the identity everywhere else. This causes a flip of the rows a and $a+1$ in the resulting permutation operator $\hat{F}' = \hat{B}_1^{-1}\hat{F}$. For any sequence of beam splitters $(T_{m,m+1}, m \in M)$ representing a local network M, which reduces the permutation operator $\hat{F}' \mapsto \mathbb{1}$ to the identity, we see that

$$\left(\prod_{m\in M}\hat{T}_{m,m+1}\right)\hat{P}^{\dagger}\hat{U}^{\dagger}_{\text{Target}} = \hat{A}'_1\hat{\mathbb{1}}\hat{A}'_2 = \hat{D}, \tag{A3}$$

where $\hat{D} = \hat{A}'_1\hat{A}'_2$ is a unitary upper-triangular matrix and therefore a diagonal matrix. From there, we can directly write our target matrix

$$\hat{D}^{\dagger}\left(\prod_{m\in M}\hat{T}_{m,m+1}\right)\hat{P}^{\dagger} = \hat{U}_{\text{Target}}, \tag{A4}$$

as this sequence of beam splitters up to a diagonal matrix $\hat{D}$ and permutation matrix $\hat{P}^{\dagger}$.

The next step is to consider the local network required in order fully transition the permutation operator $\hat{F}$ to the identity operation via local swaps. Here, we map the permutation operator $\hat{F}$ resulting from the decomposition of an arbitrary target matrix $\hat{U}_{\text{Target}}$ into the list of modes $l = \sigma(1, 2, \ldots, N)$ permuted by the bijective permutation σ. Specifically, the permutation σ maps each mode index j to the index $\sigma(j) = l_j$ where the permutation operator $\hat{F}$ is nonzero $\left((F)_{j,l_j} = 1\right)$. For example, the antidiagonal permutation operator results in the sequence $l = (N-1, N-2, \ldots, 1, 0)$, while the identity operator results in $l = (0, 1, \ldots, N-2, N-1)$. We now take the beam-splitter sequence M and, for each beam splitter $\hat{T}_{m,m+1}$ with $m \in M$, either flip the entries l_m and l_{m+1} of the sequence l or keep it as it is. See Fig. 5 for a scheme of this process. Once the resulting sequence l is equal to the sequence for the identity, the permutation operator $\hat{F}$ is equal to the identity, which means that our chosen sequence of beam splitters M can implement the target unitary $\hat{U}_{\text{Target}}$. Therefore, verifying that a sequence of beam splitters is capable of implementing the target matrix is equivalent to proving that it is capable of sorting the sequence l.

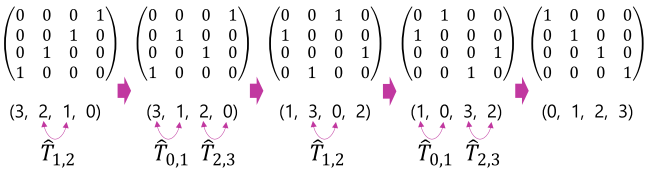

FIG. 5. Example process of programming an arbitrary 4×4 unitary using the resulting quantum walk beam-splitter sequence.

[1] H. J. Briegel, D. E. Browne, W. Dür, R. Raussendorf, and M. Van den Nest, Measurement-based quantum computation, Nat. Phys. **5**, 19 (2009).

[2] M. V. Larsen, C. Chamberland, K. Noh, J. S. Neergaard-Nielsen, and U. L. Andersen, Fault-tolerant continuous-variable measurement-based quantum computation architecture, PRX Quantum **2**, 030325 (2021).

[3] J. Carolan, C. Harrold, C. Sparrow, E. Martín-López, N. J. Russell, J. W. Silverstone, P. J. Shadbolt, N. Matsuda, M. Oguma, M. Itoh, G. D. Marshall, M. G. Thompson, J. C. F. Matthews, T. Hashimoto, J. L. O'Brien, and A. Laing, Universal linear optics, Science **349**, 711 (2015).

[4] L. Chen, E. Hall, L. Theogarajan, and J. Bowers, Photonic switching for data center applications, IEEE Photonics J. **3**, 834 (2011).

[5] K. Jacobs, R. Wu, X. Wang, S. Ashhab, Q.-M. Chen, and H. Rabitz, Fast quantum communication in linear networks, Europhys. Lett. **114**, 40007 (2016).

[6] F. Ewert, M. Bergmann, and P. van Loock, Ultrafast long-distance quantum communication with static linear optics, Phys. Rev. Lett. **117**, 210501 (2016).

[7] X.-B. Xu, W.-T. Wang, L.-Y. Sun, and C.-L. Zou, Hybrid superconducting photonic-phononic chip for quantum information processing, Chip **1**, 100016 (2022).

[8] H.-S. Zhong *et al.*, Quantum computational advantage using photons, Science **370**, 1460 (2020).
[9] H.-S. Zhong *et al.*, Quantum computational advantage using photons, Science **370**, 1460 (2020).
[10] L. S. Madsen, F. Laudenbach, M. F. Askarani, F. Rortais, T. Vincent, J. F. F. Bulmer, F. M. Miatto, L. Neuhaus, L. G. Helt, M. J. Collins, A. E. Lita, T. Gerrits, S. W. Nam, V. D. Vaidya, M. Menotti, I. Dhand, Z. Vernon, N. Quesada, and J. Lavoie, Quantum computational advantage with a programmable photonic processor, Nature **606**, 75 (2022).
[11] S. Yu *et al.*, A universal programmable Gaussian boson sampler for drug discovery, Nat. Comput. Sci. **3**, 839 (2023).
[12] J. Gao, X.-W. Wang, W.-H. Zhou, Z.-Q. Jiao, R.-J. Ren, Y.-X. Fu, L.-F. Qiao, X.-Y. Xu, C.-N. Zhang, X.-L. Pang, H. Li, Y. Wang, X.-M. Jin *et al.*, Quantum advantage with membosonsampling, Chip **1**, 100007 (2022).
[13] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, Experimental realization of any discrete unitary operator, Phys. Rev. Lett. **73**, 58 (1994).
[14] W. R. Clements, P. C. Humphreys, B. J. Metcalf, W. S. Kolthammer, and I. A. Walmsley, Optimal design for universal multiport interferometers, Optica **3**, 1460 (2016).
[15] F. Hoch, T. Giordani, L. Castello, G. Carvacho, N. Spagnolo, F. Ceccarelli, C. Pentangelo, S. Piacentini, A. Crespi, R. Osellame, E. F. Galvão, and F. Sciarrino, Modular quantum-to-quantum Bernoulli factory in an integrated photonic processor, Nat. Photonics **19**, 12 (2025).
[16] A. Barzaghi, M. Bénéfice, F. Ceccarelli, G. Corrielli, V. Galli, M. Gardina, V. Grimaldi, J. Kaczorowski, F. Malaspina, R. Osellame, C. Pentangelo, A. Rocchetto, and A. Rudi, in *Proc. SPIE 13391, Quantum Computing, Communication, and Simulation V*, 1339112 (2025).
[17] J. M. Arrazola, N. Quesada, D. J. Broome, A. A. Gaggero, M. J. Pugh, S. Slussarenko, M. E. Simmons, M. J. B. Prystawko, A. Acín, J. L. O'Brien *et al.*, Quantum circuits with many photons on a programmable nanophotonic chip, Nature **591**, 54 (2021).
[18] S. De, V. Ansari, J. Sperling, S. Barkhofen, B. Brecht, and C. Silberhorn, Realization of high-fidelity unitary operations on up to 64 frequency bins, Phys. Rev. Res. **6**, L022040 (2024).
[19] L. Serino, C. Eigner, B. Brecht, and C. Silberhorn, Programmable time-frequency mode-sorting of single photons with a multi-output quantum pulse gate, Opt. Express **33**, 5577 (2025).
[20] P. Folge, M. Stefszky, B. Brecht, and C. Silberhorn, A framework for fully programmable frequency-encoded quantum networks harnessing multioutput quantum pulse gates, PRX Quantum **5**, 040329 (2024).
[21] L. Serino, J. Gil-Lopez, M. Stefszky, R. Ricken, C. Eigner, B. Brecht, and C. Silberhorn, Realization of a multi-output quantum pulse gate for decoding high-dimensional temporal modes of single-photon states, PRX Quantum **4**, 020306 (2023).
[22] A. Schreiber, K. N. Cassemiro, V. Potoček, A. Gábris, P. J. Mosley, E. Andersson, I. Jex, and Ch. Silberhorn, Photons walking the line: A quantum walk with adjustable coin operations, Phys. Rev. Lett. **104**, 050502 (2010).
[23] K. R. Motes, A. Gilchrist, J. P. Dowling, and P. P. Rohde, Scalable boson sampling with time-bin encoding using a loop-based architecture, Phys. Rev. Lett. **113**, 120501 (2014).
[24] P. P. Rohde, Simple scheme for universal linear-optics quantum computing with constant experimental complexity using fiber loops, Phys. Rev. A **91**, 012306 (2015).
[25] D. Su, I. Dhand, L. G. Helt, Z. Vernon, and K. Brádler, Hybrid spatiotemporal architectures for universal linear optics, Phys. Rev. A **99**, 062301 (2019).
[26] M. Borghi, E. Brusaschi, M. Liscidini, M. Galli, and D. Bajoni, Bipartite Gaussian boson sampling in the time-frequency-bin domain with squeezed light generated by a silicon nitride microresonator, Quantum Inf. **11**, 135 (2025).
[27] Y. Aharonov, L. Davidovich, and N. Zagury, Quantum random walks, Phys. Rev. A **48**, 1687 (1993).
[28] A. M. Childs, Universal computation by quantum walk, Phys. Rev. Lett. **102**, 180501 (2009).
[29] 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).
[30] F. Pegoraro, P. Held, J. Lammers, B. Brecht, and C. Silberhorn, Demonstration of a photonic time-multiplexed C-NOT gate, arXiv:2412.02478 [quant-ph].
[31] T. G. de Brugière, R. Mezher, S. Currie, and S. Mansfield, Designs of linear optical interferometers with minimal depth and component count, Phys. Rev. A **113**, 033512 (2026).
[32] M. Araújo, F. Costa, and C. Brukner, Computational advantage from quantum-controlled ordering of gates, Phys. Rev. Lett. **113**, 250402 (2014).
[33] N. Habermann, Parallel neighbor-sort (or the glory of the induction principle), CMU Computer Science Report, Tech. Rep. AD-759 248 (1972).
[34] C. S. Hamilton, S. Barkhofen, L. Sansoni, I. Jex, and C. Silberhorn, Driven discrete time quantum walks, New J. Phys. **18**, 073008 (2016).
[35] T. Nitsche, S. Barkhofen, R. Kruse, L. Sansoni, M. Štefaňák, A. Gábris, V. Potoček, T. Kiss, I. Jex, and C. Silberhorn, Probing measurement-induced effects in quantum walks via recurrence, Sci. Adv. **4**, eaar6444 (2018).
[36] F. Pegoraro, P. Held, S. Barkhofen, B. Brecht, and C. Silberhorn, Dynamic conditioning of two particle discrete-time quantum walks, Phys. Scr. **98**, 034005 (2023).
[37] L. Carosini, V. Oddi, F. Giorgino, L. M. Hansen, B. Seron, S. Piacentini, T. Guggemos, I. Agresti, J. C. Loredo, and P. Walther, Programmable multiphoton quantum interference in a single spatial mode, Sci. Adv. **10**, eadj0993 (2024).
[38] B.-C. Pan, H.-X. Liu, H.-C. Xu, Y.-S. Huang, H. Li, Z.-J. Yu, L. Liu, Y.-C. Shi, and D.-X. Dai, Ultra-compact lithium niobate microcavity electro-optic modulator beyond 110 GHz, Chip **1**, 100029 (2022).
[39] G. Lindblad, On the generators of quantum dynamical semigroups, Commun. Math. Phys. **48**, 119 (1976).
[40] V. Gorini, A. Kossakowski, and E. C. G. Sudarshan, Completely positive dynamical semigroups of N-level systems, J. Math. Phys. **17**, 821 (1976).

[41] C.-Y. Ju, A. Miranowicz, G.-Y. Chen, and F. Nori, Non-Hermitian Hamiltonians and no-go theorems in quantum information, Phys. Rev. A **100**, 062118 (2019).

[42] C. S. Hamilton, R. Kruse, L. Sansoni, S. Barkhofen, C. Silberhorn, and I. Jex, Gaussian boson sampling, Phys. Rev. Lett. **119**, 170501 (2017).

[43] A. P. Lund, A. Laing, S. Rahimi-Keshari, T. Rudolph, J. L. O'Brien, and T. C. Ralph, Boson sampling from a Gaussian state, Phys. Rev. Lett. **113**, 100502 (2014).

[44] R. J. Marshman, D. Singh, T. C. Ralph, and A. P. Lund, Unitary averaging with fault and loss tolerance, Phys. Rev. A **109**, 062436 (2024).