IOP Publishing New J. Phys. 27 (2025) 114508 https://doi.org/10.1088/1367-2630/ae1866

New Journal of Physics
The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft DPG
IOP Institute of Physics

Published in partnership with: Deutsche Physikalische Gesellschaft and the Institute of Physics

PAPER

A generalized reconstruction model in circuit cutting and nonlocal-gate-based distributed quantum computation

Yi Sun[1], **Changhua Zhu**[1,2,3,*], **Yuan Zhao**[1] and **Guangwu Hou**[1]

[1] School of Telecommunications Engineering, Xidian University, Xi'an, Shaanxi 710071, People's Republic of China
[2] Collaborative Innovation Center of Quantum Information of Shaanxi Province, Xidian University, Xi'an, Shaanxi 710071, People's Republic of China
[3] Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts and Telecommunications, Xi'an, Shaanxi 710121, People's Republic of China
[*] Author to whom any correspondence should be addressed.

E-mail: chhzhu@xidian.edu.cn

Keywords: distributed quantum computation, circuit cutting, results reconstruction, multi-objective simulated annealing

OPEN ACCESS

RECEIVED
19 August 2025

REVISED
14 October 2025

ACCEPTED FOR PUBLICATION
28 October 2025

PUBLISHED
7 November 2025

Abstract

In the current noisy intermediate-scale quantum era, the limited number of high-fidelity qubits and restricted circuit depth pose significant challenges for large-scale quantum computation. Fortunately, distributed quantum computing (DQC) provides a feasible solution by dividing large quantum circuits into smaller subcircuits that can be executed on existing quantum processors. In this work, we propose a generalized model of circuit reconstruction (GMCR), which is capable of handling complex cutting patterns such as U-type structures to recover the output of the original circuit from the subcircuit results. In addition to the number of nonlocal gates and execution rounds, we introduce a new objective function in multi-objective simulated annealing (MOSA)-based cutting algorithm, the number of required SWAP operations in the subsequent mapping from logical qubits to physical qubits, which is used to satisfy the hardware connectivity constraints and to further decrease the complexity of quantum circuit compiling. We verified the GMCR model by cutting five circuits: encoding circuit for the Steane 7-qubit code, circuit of Shor's algorithm, quantum supremacy circuit, quantum circuit of Bernstein–Vazirani algorithm, and circuit of approximate quantum Fourier transform. In the case of the Steane 7-qubit code, the number of reconstruction rounds was reduced from 337 to 156 under a fixed nonlocal gate count of two, while the number of SWAP operations was also reduced from 10 to 7 compared with the earlier MOSA-based algorithm. For the U-type subcircuits, using the GMCR model, the original results can be obtained, but cannot be obtained by the dynamic definition, approximate reconstruction algorithm, and fast reconstruction algorithm. This work plays an important role in implementing large-scale DQC, a typical application of future quantum Internet.

1. Introduction

Quantum computing leverages the fundamental quantum mechanical properties, quantum state superposition [1], quantum entanglement [2], and quantum interference [3] to enable exponential parallelism and high-efficiency information processing. Consequently, it may offer significant computational advantages over classical computing in specific domains, including machine learning [4], chemistry [5], cryptanalysis [6], and finance [7]. However, to fully realize the computational speedup offered by quantum computing, a large number of high-quality qubits and large-depth quantum circuits are required. For example, Shor's well-known algorithm for integer factorization requires millions of physical qubits to encode sufficient logical qubits to solve problems on a practical scale [8]. In the current noisy intermediate-scale quantum (NISQ) era [9], a practical quantum computer with a relatively small number of qubits remains insufficient for large-scale, fault-tolerant quantum computing. Fortunately, distributed quantum computing (DQC) is a feasible method to address this problem [10].

Using a DQC, a large quantum circuit can be divided into several small-scale subcircuits with a small number of qubits and a shallow circuit depth. Such subcircuits can be implemented by current quantum processors with a limited qubit coherence time. DQC can be classified into two categories: the first category involves connecting and coordinating multiple quantum processors which contain nonlocal quantum gates [11]. With the development of quantum Internet, the entangled quantum states can be obtained on-demand at any two quantum processors in the quantum network, so that nonlocal quantum gates can be implemented on-demand with additional measurements, classical communications and local single-qubit gates. The first scheme is scalable. The second category aims to integrate multiple small-scale quantum processors each of which runs the subcircuit deriving from circuit cutting. Since the subcircuits run independently, this kind of DQC scheme is also scalable with the cost of quantum processors and classical reconstruction processing. In addition, the parallel execution of subcircuits allows efficient processing in distributed quantum systems. While, the classical reconstruction complexity grows exponentially with the number of cut qubits, the combination of two schemes is a better choice. Quantum circuit cutting decomposes a large-scale quantum circuit into several smaller subcircuits. A reduction in the number of quantum gates in the subcircuits lowers the demand for high-quality qubits, thereby significantly improving the accuracy of the computation results. Quantum circuit reconstruction (QCR) refers to recovering the output of the original circuit based on the results of the subcircuits. In quantum circuit cutting, a well-chosen partition point can significantly reduce the dimensionality of the quantum circuit and computational complexity of the reconstruction process. In addition, the choice of the reconstruction method is another key challenge.

Grover began the first DQC work on distributed data processing using entanglement (named by 'telecomputation') [12]. Various circuit partitioning methods have been proposed [13–15]. In the original work [16] and improved works [17–21], the proposed reconstruction schemes were applied to special quantum circuits. For more general and complex cutting scenarios, we propose a generalized reconstruction scheme. Several studies have focused on the use of nonlocal gates and circuit cutting. In previous studies [22], both circuit cutting and nonlocal gate-based DQC were investigated. In addition, coupling between qubits in NISQ hardware is limited, and two-qubit gates can only be implemented between physically adjacent qubits [12]. Currently, the mainstream chip structure is a two-dimensional nearest-neighbor layout that limits the operational flexibility. Additional swapping operations are required to implement multiple-qubit gate operation among multiple non-adjacent physical qubits in practical chip. These increase the additional overhead. In this case, swap operation becomes important step of mapping from logical qubits in quantum circuits to physical qubits in practical quantum processors, which is one of the tasks in quantum circuit compiling [23–25]. While, during the procedure of cutting the multiple-qubit gates with shorter distance between the input qubits can be located in one subcircuit for reducing the swap operations. Hence, we add a new objective function, the number of swap operations in the subcircuits.

The structure of this paper is arranged as follows. In section 2, we provide a comprehensive review and comparison of existing quantum circuit cutting and reconstruction methods. In section 3, an improved circuit cutting scheme is proposed by adding a new objective function to minimize the number of swap operations in the multi-objective simulated annealing (MOSA) algorithm. Section 4 proposes a universal QCR model and validates the algorithm using simulation software developed using the Python programming language in conjunction with IBM's Qiskit library. Finally, we discuss the potential advantages and limitations of the proposed reconstruction model, summarize the main contributions of this study, and outline future research directions.

2. Related works

Reconstruction of the results of the original circuit is an important step in the DQC. The classical resources consumed grow exponentially with the number of cut qubits. The prerequisite is a reconstruction model, in which the correct results can be calculated efficiently. Early representative works, such as CutQC [18], established a complete engineering framework using exact reconstruction and dynamic definition (DD) queries, but its computational overhead scales exponentially with the number of qubits and cuts. To address this bottleneck, Chen *et al* [26]. pioneered a new approach by reformulating reconstruction as a probabilistic sampling task. They employed Markov Chain Monte Carlo (MCMC) methods, specifically the Metropolis-Hastings algorithm, to significantly reduce the dependency on the qubit count. The core idea is to construct a Markov chain whose stationary distribution matches the target probability distribution of the reconstructed quantum circuit. To guide the sampling process, a temperature parameter (T), inspired by SA in statistical physics, is introduced. At high temperature, the algorithm promotes broad exploration of the state space, while gradually lowering the temperature

focuses sampling on high-probability states for exploitation, enabling efficient approximation of the circuit output without reconstructing the full probability distribution. Although this scheme reduces qubit-count dependence, it still proves inefficient for handling multiple cuts. Building upon this, Lian *et al* [27]. further optimized the sampling process by introducing the Hamiltonian Monte Carlo (HMC) algorithm, which substantially improved both sampling efficiency and reconstruction speed, although their current implementation is limited to single-cut scenarios the existed works, in which it is difficult to obtain results for complex cutting method, for example a U-type cutting method (see section 4). Accordingly, a generalized reconstruction model is required.

Many studies have been published on the QCP technology. In 2019, Perlin *et al* [28] developed a circuit cutting scheme that leverages Karger's MIN-CUT algorithm to identify severe connections within quantum circuits. This method enables the decomposition of large circuits into smaller independently simulated modules. Circuit cutting, or fragmentation, is a technique that enables the execution of large quantum circuits by breaking them into smaller manageable subcircuits. Ayral *et al* [29] provided the first experimental demonstration of this method using a superconducting quantum processor. Their work showed that by executing smaller circuit fragments and classically recombining their results, it is possible to simulate quantum circuits whose width or depth exceeds the native capacity of one quantum hardware. Tang *et al* proposed a circuit cutting approach, CutQC, using the mixed-integer programming (MIP) technique [18] Lowe *et al* [30] proposed a circuit partitioning method based on random measurement values. Saleem *et al* [31] reduced the overhead during reconstruction by minimizing the number of partitions. Hou *et al* [22] proposed a MOSA algorithm to choose the cutting positions with minimum reconstruction runs and minimum nonlocal CNOT gates. In these existing studies, multiple-qubit gates on adjacent qubits in subcircuits have not been considered. We add a new objective function to address this issue to decrease the number of swapping circuits in quantum compiling.

In summary, regarding reconstruction, our work is similar to previous studies in that it also aims to reduce the computational cost of reconstruction, but differs in that we propose a more general reconstruction model that can handle complex cutting methods, such as the U-type cutting scheme. Regarding circuit cutting, our work is similar to earlier approaches in targeting the reduction of nonlocal quantum gates and compilation overhead, but differs in that we additionally consider multi-qubit gates on adjacent qubits within subcircuits, and introduce a new objective function to decrease the number of swapping circuits during quantum compiling, thereby further improving overall performance.

3. Distributed quantum computation based on circuit cutting and nonlocal gates

Circuit cutting can also be named as circuit partitioning, which can be divided into two categories: qubit partitioning and gate partitioning. As shown in figure 1, qubit partitioning splits quantum circuit into two subcircuits by cutting the wire of qubit q_1. Qubits q_0 and q_1 are in one subcircuit and q_1 and q_2 are in the other subcircuit. Gate partitioning is another circuit cutting scheme by which a multiple-qubit gate operating in one quantum circuit can be converted into one which is implemented by multiple quantum subcircuits, with one qubit in each subcircuit and classical communications between them. This kind of multiple-qubit gate by multiple subcircuits (or multiple quantum computers) is called by nonlocal quantum gate. As shown in figure 2, the second CNOT gate is converted into nonlocal CNOT gate in which control qubit q_1 is in first subcircuit and q_2 is in the second subcircuit (typical schemes can refer to [32, 33]. In our scheme we adopt the existed nonlocal CNOT gate design based on entangled state, single-qubit measurements and classical data-controlled single-qubit gate. We here mainly introduce the principle of circuit cutting-based DQC for our generalized reconstruction algorithm in section 4.

3.1. Principle of DQC based on circuit cutting

In the theoretical framework of quantum circuits, any unitary operator can be decomposed elegantly into a linear combination of orthogonal matrix bases. Taking Pauli matrices $\boldsymbol{I}$, $\boldsymbol{Z}$, $\boldsymbol{X}$, and $\boldsymbol{Y}$ as an example, after normalization, they form a standard orthogonal basis. Any 2×2 matrix $\boldsymbol{A}$ can be precisely decomposed into a linear combination of these bases, as expressed below,

$$\boldsymbol{A} = \frac{Tr(\boldsymbol{A}\boldsymbol{I})\,I + Tr(\boldsymbol{A}\boldsymbol{X})\,X + Tr(\boldsymbol{A}\boldsymbol{Y})\,Y + Tr(\boldsymbol{A}\boldsymbol{Z})\,Z}{2} \tag{1}$$

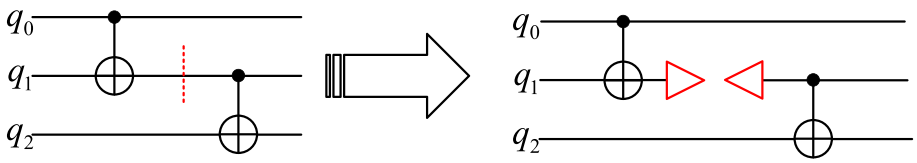

Figure 1. Schematic illustration of single-qubit partitioning. The figure shows a three-qubit circuit that is divided into two two-qubit subcircuits by cutting one qubit wire.

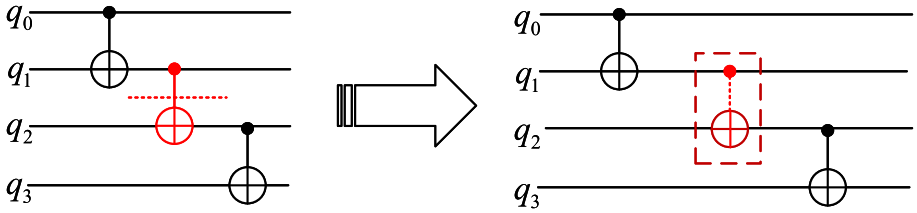

Figure 2. Schematic illustration of gate-based partitioning. The figure shows a three-qubit circuit divided into two two-qubit subcircuits through gate partitioning.

The Pauli matrices in the formula can be further expanded as a combination of their eigenbases:

$$\begin{aligned} \boldsymbol{I} &= |0\rangle\langle 0| + |1\rangle\langle 1| \\ \boldsymbol{X} &= |+\rangle\langle +| - |-\rangle\langle -| \\ \boldsymbol{Y} &= |+i\rangle\langle +i| - |-i\rangle\langle -i| \\ \boldsymbol{Z} &= |0\rangle\langle 0| - |1\rangle\langle 1| \end{aligned} \tag{2}$$

where, $|+\rangle$ and $|-\rangle$, $|i\rangle$ and $|-i\rangle$,$|0\rangle$ and $|1\rangle$ are the eigenstates of Pauli operators $\boldsymbol{X}, \boldsymbol{Y}$ and $\boldsymbol{Z}$, respectively. By using linear combination of $|0\rangle$ and $|1\rangle$, $|\pm\rangle = \frac{|0\rangle \pm |1\rangle}{\sqrt{2}}$ and $|\pm i\rangle = \frac{|0\rangle \pm i|1\rangle}{\sqrt{2}}$.

For a measurement M on a quantum state with density operator (or density matrix) ρ, the mean value of measurement results can be given by $Tr(\rho M)$. The density operator of a qubit (or qubits) in a quantum circuit can be expressed with an equivalent form. As an example, the state with density operator $\boldsymbol{A}$ in the cutting location shown in figure 1 can be expressed in the form of equation (3) equivalently, where c_i is the coefficient, $\psi_i(\boldsymbol{A}) = \mathrm{Tr}(\boldsymbol{A}\boldsymbol{O_i})$, $\boldsymbol{O_i}$ is Pauli measurement operator on $\boldsymbol{A}$ and ρ_i is the density operator which acts as the input to the second subcircuit [16],

$$\boldsymbol{A} = \sum_{i=1}^{8} c_i \psi_i(\boldsymbol{A}) \tag{3}$$

$$\begin{aligned} \boldsymbol{O_1} &= \boldsymbol{I}, \rho_1 = |0\rangle\langle 0|, c_1 = +1/2 \\ \boldsymbol{O_2} &= \boldsymbol{I}, \rho_2 = |1\rangle\langle 1|, c_2 = +1/2 \\ \boldsymbol{O_3} &= \boldsymbol{X}, \rho_3 = |+\rangle\langle +|, c_3 = +1/2 \\ \boldsymbol{O_4} &= \boldsymbol{X}, \rho_1 = |-\rangle\langle -|, c_4 = -1/2 \\ \boldsymbol{O_5} &= \boldsymbol{Y}, \rho_1 = |+i\rangle\langle +i|, c_5 = +1/2 \\ \boldsymbol{O_6} &= \boldsymbol{Y}, \rho_1 = |-i\rangle\langle -i|, c_6 = -1/2 \\ \boldsymbol{O_7} &= \boldsymbol{Z}, \rho_1 = |0\rangle\langle 0|, c_7 = +1/2 \\ \boldsymbol{O_8} &= \boldsymbol{Z}, \rho_1 = |1\rangle\langle 1|, c_8 = -1/2. \end{aligned} \tag{4}$$

In equations (3) and (4), ρ_i is the corresponding density operator of eigenbase of Pauli operator $\boldsymbol{O_i}$, and the corresponding eigenvalues by $2c_i$. Thus, the measurement and state preparation can be performed independently in parallel in the two subcircuits. We name the first subcircuit the prefix subcircuit and the second one the suffix subcircuit. Since the eigenvalue of the identity matrix $\boldsymbol{I}$ is $+1$, and the corresponding eigenstates can be any two orthogonal states, we can choose the eigenstates of the $\boldsymbol{Z}$ operator. Thus, measuring a qubit in either the I or Z basis corresponds to the same quantum circuit.

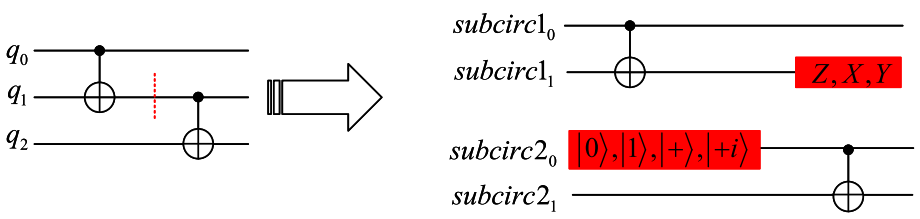

Figure 3. Example of cutting a three-qubit circuit into two smaller subcircuits, each containing two qubits. The subcircuits are generated by cutting the q_1 wire between the two CNOT gates. The three variations of $subcircuit_1$,$\boldsymbol{Z}$,$\boldsymbol{X}$,$\boldsymbol{Y}$ denote Pauli measurements, respectively.

Therefore, in practice, only three measurement bases (Z, X, and Y) are required for the prefix subcircuit. Additionally, states $|-\rangle$ and $|-i\rangle$ can be obtained by applying a Z gate (phase flip) to states $|+\rangle$ and $|+i\rangle$, respectively, reducing the required quantum states from six to four basic ones.

3.2. Results reconstruction

An illustration of single-qubit circuit cutting and reconstruction is shown in figure 3. Let the input to an n-qubit quantum circuit be initialized in the product state $|q_0,\ldots,q_{n-1}\rangle$, where each $q_i \in \{|0\rangle,|1\rangle,|+\rangle,|+i\rangle\}$ denotes a standard single-qubit basis state. The output qubits are measured in the basis $M_0,\ldots,M_{n-1}$, with each $M_i \in \{I,X,Y,Z\}$ representing a Pauli observable or the identity. We denote this circuit configuration as $C(|q_0,\ldots,q_{n-1}\rangle;M_0,\ldots,M_{n-1})$, where C specifies a quantum circuit operating on the given input state and subject to measurements in the corresponding bases. It is important to note that $subcirc1_1$ does not directly contribute to the final output of the original quantum circuit. As a result, the measurement outcomes obtained from executing subcircuit 1 must be adjusted by multiplying a factor of ±1, which is determined by the specific measurement results of qubits within $subcirc1_1$. Specifically, each measurement outcome of subcircuit 1 should be attributed to the final output as:

$$\begin{cases} \overline{x0},\ \overline{x1} \to +\bar{x} & M = I \\ \overline{x0} \to +\bar{x} & \\ \overline{x1} \to -\bar{x} & M = [Z,\ X,\ Y] \end{cases} \tag{5}$$

where $\bar{x}$ is the measurement outcome of the qubits in the $subcirc1_0$.

By substituting equation (4) into equation (1) and simplifying the expression, we obtain the following result [18]:

$$\begin{aligned} A_1 &= [Tr(AI) + Tr(AZ)]\,|0\rangle\langle 0| \\ A_2 &= [Tr(AI) - Tr(AZ)]\,|1\rangle\langle 1| \\ A_3 &= Tr(AX)\,[2\,|+\rangle\langle +| - |0\rangle\langle 0| - |1\rangle\langle 1|] \\ A_4 &= Tr(AY)\,[2\,|+i\rangle\langle +i| - |0\rangle\langle 0| - |1\rangle\langle 1|]\,. \end{aligned} \tag{6}$$

As an illustrative example, we show how to compute the probability of the uncut circuit outputting the state $|000\rangle$. In this case, the corresponding portion of subcircuit 1 is the state $|0\rangle$. Based on equation (5), the reconstruction process requires four terms from subcircuit 1, which are associated with this outcome:

$$\begin{aligned} p_{1,1} &= p(|00\rangle\,|I) + p(|01\rangle\,|I) + p(|00\rangle\,|Z) - p(|01\rangle\,|Z) \\ p_{1,2} &= p(|00\rangle\,|I) + p(|01\rangle\,|I) - p(|00\rangle\,|Z) + p(|01\rangle\,|Z) \\ p_{1,3} &= p(|00\rangle\,|X) - p(|01\rangle\,|X) \\ p_{1,4} &= p(|00\rangle\,|Y) - p(|01\rangle\,|Y)\,. \end{aligned} \tag{7}$$

The relevant state of subcircuit 2 is $|00\rangle$. Hence, its four terms are:

$$\begin{aligned} p_{2,1} &= p(|00\rangle\,||0\rangle) \\ p_{2,2} &= p(|00\rangle\,||1\rangle) \\ p_{2,3} &= 2p(|00\rangle\,||+\rangle) - p(|00\rangle\,||0\rangle) - p(|00\rangle\,||1\rangle) \\ p_{2,4} &= 2p(|00\rangle\,||+i\rangle) - p(|00\rangle\,||0\rangle) - p(|00\rangle\,||1\rangle)\,. \end{aligned} \tag{8}$$

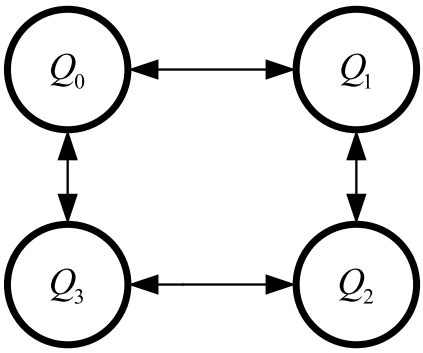

Figure 4. Four-qubit quantum processor model. Q_0 is connected to Q_1 and Q_3 via couplers, which allows a CNOT gate to be applied on the qubit pairs $\{Q_0, Q_1\}$ and $\{Q_0, Q_3\}$ in either direction. However, Q_0 is not directly connected to Q_2, so a CNOT gate cannot be applied on these two qubits directly. Used with permission of Association for Computing Machinery, from [25]; permission conveyed through Copyright Clearance Center, Inc.

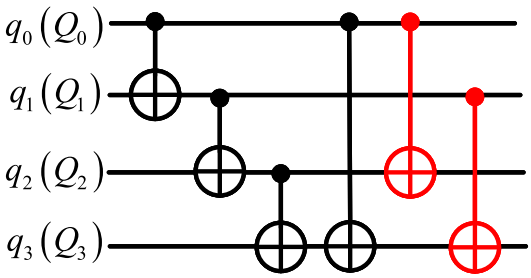

Figure 5. Four-qubit circuit. This quantum circuit consists of six CNOT gates. The initial logical-to-physical qubit mapping is given by $\{q_0 \mapsto Q_0, q_1 \mapsto Q_1, q_2 \mapsto Q_2, q_3 \mapsto Q_3\}$.

During the classical post-processing stage, the complete probability distribution of the original uncut circuit can be reconstructed by utilizing the relevant outputs from the two smaller subcircuits. This involves calculating and summing four specific Kronecker product pairs. Specifically,the final reconstructed probability of the uncut state $|000\rangle$ is

$$p(|000\rangle) = \frac{1}{2}\sum_{i=1}^{4} p_{1,i} \otimes p_{2,i}. \tag{9}$$

The mathematical theory of circuit cutting [16] proves that CutQC [18] output strictly equals the output of the uncut circuit.

3.3. The objective functions of optimizing circuit cutting positions

The number of nonlocal gates and the number of execution rounds required for subcircuit clusters were used as two objective functions in previous work [22]. However, the logical quantum circuits should run in practical quantum processors, and hardware-specific constraints, e.g. limited qubit connectivity, should be taken into account. Therefore, these two objective functions (metrics) are insufficient. Next, the detailed will be expressed.

Quantum circuits are composed of multiple quantum gates that are functionally analogous to the logic gates in classical computing. However, owning to the physical implementation challenges of the current quantum hardware, the physical qubit topology is relatively limited. Therefore, one task of a quantum compiler is to map logical qubits into specific physical qubits in practical quantum hardware.

A 4-qubit quantum processor is shown in figure 4. It only allows two-qubit gate operations between the following pairs of physical qubits: $\{Q_0, Q_1\}$, $\{Q_1, Q_2\}$, $\{Q_2, Q_3\}$ and $\{Q_3, Q_0\}$; However, there is no physical connection between qubits $\{Q_0, Q_2\}$ and $\{Q_1, Q_3\}$, so a two-qubit gate cannot be applied directly to them.

Suppose that we want to execute a small quantum circuit containing six CNOT gates on this 4-qubit device, as shown in figure 5. The initial mapping from logical to physical qubits is defined as follows: $\{q_0 \to Q_0, q_1 \to Q_1, q_2 \to Q_2, q_3 \to Q_3\}$. Under this mapping, four CNOT gates can be executed directly on a physical device. However, the 5th and 6th CNOT gates (highlighted in red in figure 5 cannot be

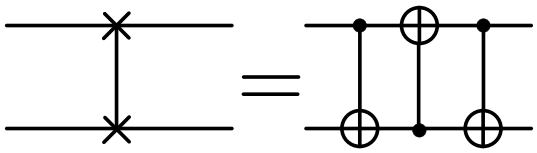

Figure 6. Decomposition of SWAP Operation. We employ SWAP operations to change the qubit mapping by exchanging the states between two qubits. It consists of three CNOT gates. Used with permission of Association for Computing Machinery, from [25]; permission conveyed through Copyright Clearance Center, Inc.

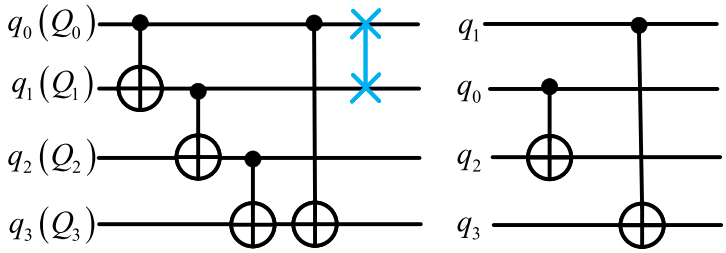

Figure 7. The updated quantum circuit is now executable after inserting a SWAP operation between q_0 and q_1 following the fourth CNOT gate. The first four CNOT gates can be executed under the initial mapping. After the inserted SWAP, the mapping is updated to $\{q_0 \mapsto Q_1, q_1 \mapsto Q_0, q_2 \mapsto Q_2, q_3 \mapsto Q_3\}$. The remaining two CNOT gates can now be executed under this updated mapping.

executed directly because the corresponding pairs of physical qubits are not connected. Therefore, it is necessary to dynamically adjust the qubit mapping during circuit execution to ensure that all CNOT gates can be executed successfully on the hardware.

We update the qubit mapping by introducing SWAP operations, which change the physical positions of two qubits by swapping their states. A SWAP operation consists of three CNOT gates (as shown in figure 6). By consecutively applying multiple SWAP operations, a logical qubit can be moved to any desired physical location. Even if the target qubits are not adjacent in the initial connectivity graph, they can be moved to an interactive position with the help of intermediate swaps, thereby enabling the desired two-qubit gate operations.

Figure 7 shows the updated circuit after inserting the SWAP operation following the fourth CNOT gate. The SWAP operation acts on qubits q_0 and q_1, and the corresponding mapping is updated as $\{q_0 \to Q_1, q_1 \to Q_0, q_2 \to Q_2, q_3 \to Q_3\}$. The updated mapping allows the remaining two CNOT gates to be executed successfully in the physical architecture.

By inserting an appropriate number of SWAP operations into the quantum circuit, we can satisfy all physical execution constraints of the two-qubit gates, thereby generating a circuit compatible with the hardware while maintaining the functionality of the original quantum circuit. However, due to the limitations of current NISQ devices, introducing additional SWAP operations also brings the following issues.

First, the number of operations increased. SWAP operations consist of multiple gates, and are not ideal operations themselves, which may introduce more noise and increase the overall error rate.

Second, the circuit depth was increased. Additional gate operations prolong the circuit execution time, allowing more decoherence effects to accumulate and further reducing the fidelity of the circuit.

Additional SWAP operations introduce significant overhead in terms of fidelity and execution time. Therefore, we aimed to minimize the number of SWAP operations to reduce the overall error rate and execution time. Thus, we used the number of swap operations in the subcircuits as the third objective function f_3. Actually, the physical mapping stage remains one of the primary bottlenecks in quantum compilation, as it often requires inserting numerous SWAP gates to comply with hardware connectivity restrictions. The objective directs the MOSA algorithm toward coupling-aware partitioning schemes, where logically correlated qubits are assigned to physically adjacent hardware qubits. This anticipatory optimization minimizes inter-qubit distances within each subcircuit, thereby reducing the number of SWAP operations required during subsequent compilation and mapping. Consequently, the compiled circuits exhibit shallower depth, shorter execution time, and lower accumulated noise.

Algorithm 1. Multi-Objective Simulated Annealing-based circuit cutting(MOSA).

Input: QC, S_{init},T_{init},T_{min},T,I,α,POP,IND_{init},DS
Output: *Pareto frontier*

while $T > T_{min}$ **do**
  for $i \in I$ **do**
    while S_{new} is not valid
      $S_{new} = RO(S_{\text{old}})$
    end while
    $(f_1,f_2,f_3) = (Obj_1(S_{new}), Obj_2(S_{new}), Obj_3(S_{new}))$
    $IND_{new} = (f_1,f_2,f_3)$
    if IND_{new} is superior to $POP[i]$ **then**
      $POP[i] = IND_{new}$
      $S_{opt} = S_{new}$
      Add IND_{new}, S_{opt} into DS
    else
      $\Delta f = \left|f_1^{new} - f_1^{old}\right| + \left|f_2^{new} - f_2^{old}\right| + \left|f_3^{new} - f_3^{old}\right|$
      $P_{accept} = e^{(-\Delta f/T)}$
      if random(0,1) P_{accept} **then**
        $POP[i] = IND_{new}$
      end if
    end if
    $S_{\text{old}} = S_{new}$
  end for
  $T = T \cdot \alpha$
end while
Choose the Pareto frontier from DS

3.4. MOSA-based circuit cutting

SA is a heuristic algorithm inspired by the metallurgy annealing process, where a metal material is heated and then slowly cooled to achieve a stable state. It can escape the local minima by probably accepting suboptimal solutions. SA algorithm starts with an initial solution and a high temperature. Then new solution is generated by a small change to the current solution and the temperature is decreased slowly at each iteration. The new solution will be accepted if it is better than the current one, or be accepted probably if it is worse than the current one. SA stops when a low temperature or the number of iterations is reached. SA is widely applied in combinatorial optimization problems. MOSA is an extension of SA in which there are multiple objective functions. In MOSA, a new solution is feasible when it dominates the current one, i.e. at least one objective function value is without any deterioration. Detailed will be given later in this section.

The process of the improved MOSA algorithm is presented in algorithm 1. The inputs of the algorithm includes: the original quantum circuit QC, the initial cutting scheme S_{init}, the initial temperature T_{init}, the minimum temperature T_{min}, the current temperature T, the number of iterations at each temperature I, the cooling rate α, the new cutting scheme S_{new}, the randomly chosen operator RO, the previous valid cutting scheme S_{old}, the objective function Obj_1 to calculate f_1, the objective function Obj_2 to calculate f_2, the objective function Obj_3 to calculate f_3, POP represents a population, which denotes a set of I individuals (f_1,f_2,f_3), each individual is also called IND, the original individual is called IND_{init}, new individual generated during iteration is written as IND_{new}, and the set of dominant individuals generated in the iterative process of population individuals is called dominant species, simplified as DS. At the end of the algorithm, the Pareto frontier is filtered out from DS, and we can choose the most suitable cutting scheme from the pareto frontier points according to the actual situation. The initial Settings are as follows: $IND_{\text{new}} = (Obj_1(S_{\text{init}}), Obj_2(S_{\text{init}}), Obj_3(S_{\text{init}}))$, $POP = \left\{IND_{\text{init}}^0, IND_{\text{init}}^1, \ldots, IND_{\text{init}}^{I-1}\right\}$, $T = T_{\text{init}}$, $S_{\text{old}} = S_{\text{init}}$.

The MOSA algorithm begins by initializing the essential parameters, including the individual, population, temperature, and previous cutting configuration. The core execution of the algorithm spans from Lines 1 to 22, iterating until system temperature falls below the predefined minimum threshold T_{min}. Within each temperature level T the procedure from Lines 2 to 20 performs the SA process. For every instance in population set I, a new candidate solution S_{new} is randomly produced via the RO operator and assessed against the current solution using objective functions. If the resulting indicator IND_{new} demonstrates superiority over the existing population entry $POP[i]$, it replaces it, and the corresponding

optimal state S_{opt} is recorded in the solution archive DS. Otherwise, $POP[i]$ may still be updated with IND_{new} based on the probability dictated by the Metropolis acceptance criterion.

The applied stochastic modification technique involves altering the position of the coordinates within the circuit partitioning configuration. This is achieved by first selecting a subcircuit fragment at random, then randomly selecting one coordinate from it, and finally, relocating this coordinate to a different subcircuit. These randomized operations can be categorized into three distinct types :(1) randomly modifying the width of the selected subcircuit fragment, (2) randomly adjusting the depth of the fragment, and (3) simultaneously altering the width and depth of the fragment in a stochastic manner.

Two distinct criteria are employed to determine whether a newly generated solution should replace an existing one. The first is based on Pareto dominance, defined as follows. Let A, B, C represent the objective values of the current solution, with the optimization goal being the minimization of both. For a new candidate solution with objectives (A', B', C'), the following scenarios are considered:(1) If $A' \leqslant A, B' \leqslant B, C' \leqslant C$ and at least one of these inequalities is strict, then (A', B', C') is said to dominate (A, B, C);(2) if any value in (A', B', C') is greater than its corresponding value in (A, B, C), then the new solution does not dominate the old one. In our method, a new individual is deemed better than the previous one only when it satisfies the dominance condition described in Case (1). The second replacement rule is based on the Metropolis criterion, which allows acceptance of a suboptimal solution with a certain probability. This strategy, commonly employed in SA, helps the algorithm escape from local minima. In the MOSA framework, the Metropolis acceptance probability is defined as: $P_{accept} = e^{(-\Delta f/T)}$, where the objective difference Δf is calculated as the sum of the absolute changes: $\Delta f = \left|f_1^{new} - f_1^{old}\right| + \left|f_2^{new} - f_2^{old}\right| + \left|f_3^{new} - f_3^{old}\right|$.

In algorithm 1, by incorporating f_3 into the MOSA framework, the search process is guided to explore a broader and more physically realistic solution space, enabling the algorithm to achieve a balanced trade-off between cutting cost and mapping cost. As a result, the optimization process converges toward a Pareto front that more accurately reflects overall hardware feasibility. Rather than merely accelerating convergence in terms of iteration count, the proposed method enhances convergence quality, allowing the algorithm to more effectively approach globally meaningful Pareto-optimal solutions in which circuit partitions are near-optimal from both logical and physical perspectives. While, it is worth noting that by MOSA algorithm and the new added objective function might not help to converge faster to optimized circuits than existing works. The algorithm aims at reducing the running time of DQC based on optimized cutting scheme, not the cutting scheme itself. In the quantum circuits examples in section 5, we choose the number of swapping operations as one of the performance metrics.

4. Generalized model of circuit reconstruction (GMCR) in circuit-cutting-based DQC

To overcome the limited applicability, low precision, and weak scalability of current reconstruction algorithms, as discussed in section 2, we propose a generalized QCR model that supports any number of qubit cuts and is capable of performing complete reconstruction on multi-qubit cutting circuits. Unlike existing methods, our approach does not rely on the presence or strength of entanglement between qubits, making it applicable to a wide range of circuit structures, including weakly entangled, strongly entangled, and even fully non-entangled circuits. While the classical reconstruction complexity grows exponentially with the number of cut qubits, the approach remains practically scalable for moderate numbers of cuts, and the parallel execution of subcircuits allows efficient processing in distributed quantum systems. In this section, we first present the reconstruction model in the L-type cutting case and then the model in the U-type case. Third, we propose a method for resolving the problem of discontinuous cutting and reconstruction order. Finally, a complete circuit reconstruction process is proposed.

4.1. The reconcontruction of L-type cutting case

Given an arbitrary N-qubit quantum state $|\psi\rangle$, a direct decomposition of the identity operator $I = \sum_{b\in\{0,1\}} |b\rangle\langle b|$ on the nth qubit of $|\psi\rangle$ can be represented as follows:

$$|\psi\rangle = I_n|\psi\rangle \simeq \sum_{b\in\{0,1\}} |b\rangle_n \otimes ({}_n\langle b|\,\psi\rangle) \tag{10}$$

where I_n performs the identity operation on the nth qubit, and the symbol $\simeq$ indicates equality in cases where the order of the quantum bits (i.e. the arrangement of tensor factors) may be different. Where ${}_n\langle b|\,\psi\rangle$ represents the quantum state of a subsystem composed of N-1 quantum bits obtained by projecting the nth qubit of $|\psi\rangle$ into $|b\rangle$

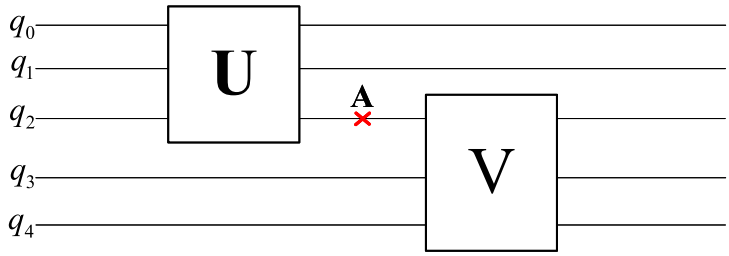

Figure 8. Example of cutting a five-qubit circuit into two smaller subcircuits of three qubits each by single-qubit cutting.

Figure 8 show an example circuit in which 5-qubit circuit is divided into two subcircuits by cutting the wire at position A. The output quantum state can be represented by

$$\begin{aligned}|\psi\rangle_{0-4}^{\text{out}} &= |\psi\rangle_{0,1}^{\text{out}} \otimes |\psi\rangle_{2,3,4}^{\text{out}} \\ &= \sum_{b_1\in\{0,1\}} {}_A\langle b_1|\, \boldsymbol{U}|\psi\rangle_{0,1,2}^{\text{in}} \otimes \boldsymbol{V}\left(|b_1\rangle_A \otimes |\psi\rangle_{3,4}^{\text{in}}\right).\end{aligned} \tag{11}$$

The quantum state can also be represented by the density operator as follows:

$$\begin{aligned}\rho_{0-4}^{\text{out}} &= |\psi\rangle_{0-4}^{out}\langle\psi| \\ &= \left[\sum_{b_1\in\{0,1\}} {}_A\langle b_1|\, \boldsymbol{U}|\psi\rangle_{0,1,2}^{\text{in}} \otimes \boldsymbol{V}\left(|b_1\rangle_A \otimes |\psi\rangle_{3,4}^{\text{in}}\right)\right] \\ &\quad \times \left[\sum_{b_1'\in\{0,1\}} {}_{0,1,2}^{\text{in}}\langle\psi|\, \boldsymbol{U}^+ |b_1'\rangle_A \otimes \left({}_A\langle b_1'| \otimes {}_{3,4}^{\text{in}}\langle\psi|\right) \boldsymbol{V}^+\right] \\ &= \sum_{b_1,b_1'\in\{0,1\}} Tr_A\left[|b_1'\rangle_A\langle b_1|\, \boldsymbol{U}|\psi\rangle_{0,1,2}\langle\psi|\, \boldsymbol{U}^+\right] \otimes \left[\boldsymbol{V}\left((|b_1\rangle_A\langle b_1'|) \otimes \left(|\psi\rangle_{3,4}^{\text{in}}\langle\psi|\right)\right)\boldsymbol{V}^+\right] \\ &= Tr_A\left[|0\rangle_A\langle 0|\, \boldsymbol{U}|\psi\rangle_{0,1,2}\langle\psi|\, \boldsymbol{U}^+\right] \otimes \left[\boldsymbol{V}\left((|0\rangle_A\langle 0|) \otimes \left(|\psi\rangle_{3,4}^{\text{in}}\langle\psi|\right)\right)\boldsymbol{V}^+\right] \\ &\quad + Tr_A\left[|0\rangle_A\langle 1|\, \boldsymbol{U}|\psi\rangle_{0,1,2}\langle\psi|\, \boldsymbol{U}^+\right] \otimes \left[\boldsymbol{V}\left((|1\rangle_A\langle 0|) \otimes \left(|\psi\rangle_{3,4}^{\text{in}}\langle\psi|\right)\right)\boldsymbol{V}^+\right] \\ &\quad + Tr_A\left[|1\rangle_A\langle 0|\, \boldsymbol{U}|\psi\rangle_{0,1,2}\langle\psi|\, \boldsymbol{U}^+\right] \otimes \left[\boldsymbol{V}\left((|0\rangle_A\langle 1|) \otimes \left(|\psi\rangle_{3,4}^{\text{in}}\langle\psi|\right)\right)\boldsymbol{V}^+\right] \\ &\quad + Tr_A\left[|1\rangle_A\langle 1|\, \boldsymbol{U}|\psi\rangle_{0,1,2}\langle\psi|\, \boldsymbol{U}^+\right] \otimes \left[\boldsymbol{V}\left((|1\rangle_A\langle 1|) \otimes \left(|\psi\rangle_{3,4}^{\text{in}}\langle\psi|\right)\right)\boldsymbol{V}^+\right].\end{aligned} \tag{12}$$

According to equation (12), each possible output state of qubits 0 and 1 is determined by the measurement of qubit 2 at point A using the operator $|b_1'\rangle_A\langle b_1|$. The output states of qubits 2, 3, and 4 are obtained from operation V with the input states $|b_1\rangle_A\langle b_1'|$, as shown in line 4 of equation (12). Obviously, $|b_1'\rangle_A\langle b_1|$ is the conjugate transpose of $|b_1\rangle_A\langle b_1'|$.

The matrix $|b_1'\rangle_A\langle b_1|$ can have 4 values, which are $|0\rangle_A\langle 0|$, $|0\rangle_A\langle 1|$, $|1\rangle_A\langle 0|$, $|1\rangle_A\langle 1|$. We use Pauli matrices $\boldsymbol{I}$, $\boldsymbol{Z}$, $\boldsymbol{X}$ and $\boldsymbol{Y}$ to denote them, as follows:

$$\begin{aligned}|0\rangle\langle 0| &= \frac{1}{2}(\boldsymbol{I}+\boldsymbol{Z}) \\ |0\rangle\langle 1| &= \frac{1}{2}(\boldsymbol{X}+j\boldsymbol{Y}) \\ |1\rangle\langle 0| &= \frac{1}{2}(\boldsymbol{X}-j\boldsymbol{Y}) \\ |1\rangle\langle 1| &= \frac{1}{2}(\boldsymbol{I}-\boldsymbol{Z}).\end{aligned} \tag{13}$$

For each summation term in the second part of equation (12), the density operator $|b_1\rangle_A\langle b_1'|$ can be prepared using the density operators $|0\rangle\langle 0|, |1\rangle\langle 1|, |+\rangle\langle +|, |+i\rangle\langle +i|$ of the four basic input quantum states. The states $|0\rangle\langle 1|, |1\rangle\langle 0|$ can be prepared as follows:

$$|0\rangle\langle 1| = -\frac{1+j}{2}|0\rangle\langle 0| - \frac{1+j}{2}|1\rangle\langle 1| + |+\rangle\langle +| + j|+i\rangle\langle +|$$
$$|1\rangle\langle 0| = -\frac{1-j}{2}|0\rangle\langle 0| - \frac{1-j}{2}|1\rangle\langle 1| + |+\rangle\langle +| - j|+i\rangle\langle +|. \quad (14)$$

Substituting equations (13) and (14) into equation (12), we obtain:

$$\begin{aligned}
\rho_{0-4}^{\text{out}} = \frac{1}{2}\Big[& Tr_A\left[(I+Z)\,\boldsymbol{U}|\psi\rangle_{0,1,2}^{\text{in}}\langle\psi|\,\boldsymbol{U}^\dagger\right] \otimes \boldsymbol{V}\left(|0\rangle_A\langle 0| \otimes |\psi\rangle_{3,4}^{\text{in}}\langle\psi|\right)\boldsymbol{V}^\dagger \\
& + Tr_A\left[(X+jY)\,\boldsymbol{U}|\psi\rangle_{0,1,2}^{\text{in}}\langle\psi|\,\boldsymbol{U}^\dagger\right] \\
& \otimes \boldsymbol{V}\left(\left(-\frac{1-j}{2}|0\rangle_A\langle 0| - \frac{1-j}{2}|1\rangle_A\langle 1| + |+\rangle_A\langle +| - j|+i\rangle_A\langle +i|\right) \otimes |\psi\rangle_{3,4}^{\text{in}}\langle\psi|\right)\boldsymbol{V}^\dagger \\
& + Tr_A\left[(X-jY)\,\boldsymbol{U}|\psi\rangle_{0,1,2}^{\text{in}}\langle\psi|\,\boldsymbol{U}^\dagger\right] \\
& \otimes \boldsymbol{V}\left(\left(-\frac{1+j}{2}|0\rangle_A\langle 0| - \frac{1+j}{2}|1\rangle_A\langle 1| + |+\rangle_A\langle +| + j|+i\rangle_A\langle +i|\right) \otimes |\psi\rangle_{3,4}^{\text{in}}\langle\psi|\right)\boldsymbol{V}^\dagger \\
& + Tr_A\left[(I-Z)\,\boldsymbol{U}|\psi\rangle_{0,1,2}^{\text{in}}\langle\psi|\,\boldsymbol{U}^\dagger\right] \otimes \boldsymbol{V}\left(|1\rangle_A\langle 1| \otimes |\psi\rangle_{3,4}^{\text{in}}\langle\psi|\right)\boldsymbol{V}^\dagger\Big] \\
= \frac{1}{2}\Big[& Tr_A\left[\left(\boldsymbol{U}|\psi\rangle_{0,1,2}^{\text{in}}\langle\psi|\,\boldsymbol{U}^\dagger\right)(I+Z)\right] \otimes \boldsymbol{V}\left(|0\rangle_A\langle 0| \otimes |\psi\rangle_{3,4}^{\text{in}}\langle\psi|\right)\boldsymbol{V}^\dagger \\
& + Tr_A\left[\left(\boldsymbol{U}|\psi\rangle_{0,1,2}^{\text{in}}\langle\psi|\,\boldsymbol{U}^\dagger\right)(X+jY)\right] \\
& \otimes \boldsymbol{V}\left(\left(-\frac{1-j}{2}|0\rangle_A\langle 0| - \frac{1-j}{2}|1\rangle_A\langle 1| + |+\rangle_A\langle +| - j|+i\rangle_A\langle +i|\right) \otimes |\psi\rangle_{3,4}^{\text{in}}\langle\psi|\right)\boldsymbol{V}^\dagger \\
& + Tr_A\left[\left(\boldsymbol{U}|\psi\rangle_{0,1,2}^{\text{in}}\langle\psi|\,\boldsymbol{U}^\dagger\right)(X-jY)\right] \otimes \\
& \boldsymbol{V}\left(\left(-\frac{1+j}{2}|0\rangle_A\langle 0| - \frac{1+j}{2}|1\rangle_A\langle 1| + |+\rangle_A\langle +| + j|+i\rangle_A\langle +i|\right) \otimes |\psi\rangle_{3,4}^{\text{in}}\langle\psi|\right)\boldsymbol{V}^\dagger \\
& + Tr_A\left[\left(\boldsymbol{U}|\psi\rangle_{0,1,2}^{\text{in}}\langle\psi|\,\boldsymbol{U}^\dagger\right)(I-Z)\right] \otimes \boldsymbol{V}\left(|1\rangle_A\langle 1| \otimes |\psi\rangle_{3,4}^{\text{in}}\langle\psi|\right)\boldsymbol{V}^\dagger\Big]. \quad (15)
\end{aligned}$$

In equation (15) $Tr_A\left[\left(\boldsymbol{U}|\psi\rangle_{0,1,2}^{\text{in}}\langle\psi|\,\boldsymbol{U}^+\right)|I\right]$ means the density operator of the residual qubits, q_0 and q_1, when the cut qubit q_2 at A is measured by I after unitary operating $\boldsymbol{U}$ with input state $|\psi\rangle_{0,1,2}^{\text{in}}$. Similarly, $Tr_A\left[\left(\boldsymbol{U}|\psi\rangle_{0,1,2}^{\text{in}}\langle\psi|\,\boldsymbol{U}^+\right)|X\right]$, $Tr_A\left[\left(\boldsymbol{U}|\psi\rangle_{0,1,2}^{\text{in}}\langle\psi|\,\boldsymbol{U}^+\right)|Y\right]$, $Tr_A\left[\left(\boldsymbol{U}|\psi\rangle_{0,1,2}^{\text{in}}\langle\psi|\,\boldsymbol{U}^+\right)|Z\right]$ are the ones in X, Y, and Z measurement, respectively. If the output quantum state of U subcircuit is denoted by $|q_2q_1q_0\rangle$, then there are eight possible base states. For I measurement with two eigenvalues $+1$ over q_2 and calculating the mean value, that is partial tracing over q_2, the probabilities of state $|q_1q_0\rangle$ are as follows:

$$\begin{aligned}
p(00) &= p(000|I) + p(100|I) \\
p(01) &= p(001|I) + p(101|I) \\
p(10) &= p(010|I) + p(110|I) \\
p(11) &= p(011|I) + p(111|I) \quad (16)
\end{aligned}$$

$\boldsymbol{V}\left(\left(|0\rangle_A\langle 0|\right) \otimes \left(|\psi\rangle_{3,4}^{\text{in}}\langle\psi|\right)\right)\boldsymbol{V}^+$ represents $\boldsymbol{V}$operation with the state of q_2 $|0\rangle$ and the state of q_3 and q_4 $|\psi\rangle_{3,4}$. This output state of subcircuit V can be denoted as $|q_4q_3q_2\rangle$. The related output probabilities were obtained via simulation on a Python-based quantum simulator. While, the same procedure can also be executed on an actual practical quantum processor that can be accessed (this has not been implemented in this paper). Furthermore, the output probabilities of the original circuit can be obtained, for example, the probability of state $|00000\rangle$ can be calculated by

$$\begin{aligned}
p(|00000\rangle) = \frac{1}{2}\Big(& [p(|000\rangle|I) + p(|100\rangle|I) + p(|000\rangle|Z) - p(|100\rangle|Z)]\,p(|000\rangle|0\rangle) \\
& + [p(|000\rangle|X) - p(|100\rangle|X) + j(p(|000\rangle|Y) - p(|100\rangle|Y))] \\
& \cdot \left[-\frac{1-j}{2}p(|000\rangle|0\rangle) - \frac{1-j}{2}p(|000\rangle|1\rangle) + p(|000\rangle|+\rangle) - jp(|000\rangle|+i\rangle)\right] \\
& + [p(|000\rangle|X) - p(|100\rangle|X) - j(p(|000\rangle|Y) - p(|100\rangle|Y))]
\end{aligned}$$

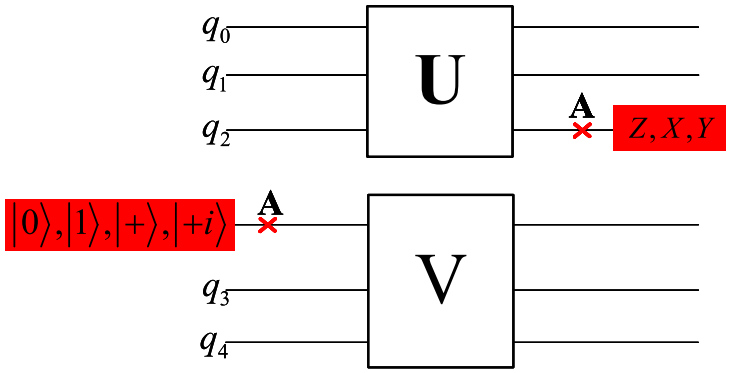

Figure 9. Illustration of single-qubit circuit cutting and decomposition. The three variations of the U subcircuit and four variations of the V subcircuit can then be evaluated on a three-qubit quantum device.

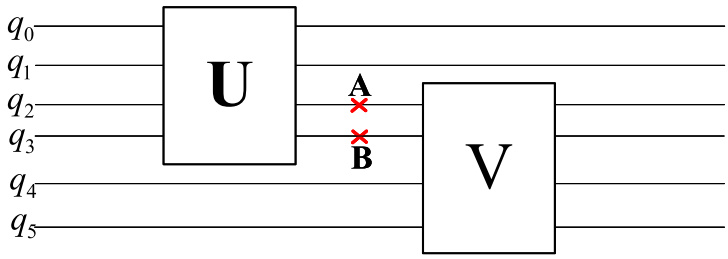

Figure 10. Example of cutting a six-qubit circuit into two smaller subcircuits of four qubits each by inveresed L-type two-qubit circuit cutting.

$$\cdot\left[-\frac{1+j}{2}p(|000\rangle|0\rangle)-\frac{1+j}{2}p(|000\rangle|1\rangle)+p(|000\rangle|+\rangle)+jp(|000\rangle|+i\rangle)\right]$$
$$+\left[p(|000\rangle|I)+p(|100\rangle|I)-p(|000\rangle|Z)+p(|100\rangle|Z)\right]p(|000\rangle|1\rangle)\Big). \tag{17}$$

As previously described, in equation (12), $|b_1'\rangle_A\langle b_1|$ in $\mathrm{Tr}_A\left[|b_1'\rangle_A\langle b_1|\boldsymbol{U}|\psi\rangle_{0,1,2}\langle\psi|\boldsymbol{U}^+\right]$ represents a measurement operator that is treated as an equivalent Pauli operator measurement and calculate the expectation value of its measurement result. For the second part of each sum term of equation (12), where $\left[\boldsymbol{V}\left(|b_1\rangle_A\langle b_1'|\otimes|\psi\rangle_{3,4}\langle\psi|\right)\boldsymbol{V}^+\right]$ uses $|b_1\rangle_A\langle b_1'|$ as input, we replace $|b_1\rangle_A\langle b_1'|$ with the corresponding four standard quantum states. Thus, the five-qubit circuit becomes equivalent to two sub-circuits, as illustrated in figure 9.

The first subcircuit requires measurement n of the output of qubit 2. Because measurements in the I and Z bases yielded the same results, only three sets of measurements were required. For the second subcircuit, the input at position A consisted of four quantum states, requiring four different sets of measurement data corresponding to the four inputs. By substituting the measurement results and output probabilities of the two subcircuits into equation (17), we obtain the reconstructed results of the original circuit.

One-qubit circuit cutting can be easily extended to two or more qubit cutting cases, an example of two-qubit L-type cutting case is shown in figure 10.

Its output quantum state can be expressed as

$$|\psi\rangle_{0-5}^{\mathrm{out}}=|\psi\rangle_{0,1}^{\mathrm{out}}\otimes|\psi\rangle_{2\sim5}^{\mathrm{out}}=\sum_{b_1,b_2\in\{0,1\}}{}_{A,B}\langle b_1b_2|\boldsymbol{U}|\psi\rangle_{0\sim3}^{in}\otimes\boldsymbol{V}\left(|b_1b_2\rangle_{A,B}\otimes|\psi\rangle_{4,5}^{in}\right). \tag{18}$$

The density operator of this state is

$$\rho_{0\sim5}^{\mathrm{out}}=|\psi\rangle_{0\sim5}^{\mathrm{out}}\langle\psi|$$
$$=\left[\sum_{b_1,b_2\in\{0,1\}}{}_{A,B}\langle b_1b_2|\boldsymbol{U}|\psi\rangle_{0\sim3}^{\mathrm{in}}\otimes\boldsymbol{V}\left(|b_1b_2\rangle_{A,B}\otimes|\psi\rangle_{4,5}^{\mathrm{in}}\right)\right]$$

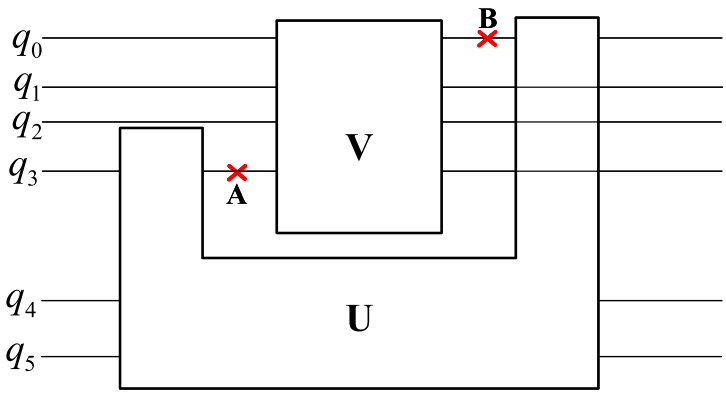

Figure 11. Example of cutting a six-qubit circuit into two smaller subcircuits of four qubits each using U-type two-qubit circuit cutting.

$$
\begin{aligned}
&\cdot\left[\sum_{b_1',b_2'\in\{0,1\}} {}^{\text{in}}_{0\sim3}\langle\psi|\, \boldsymbol{U}^{+}\,|b_1'b_2'\rangle_{A,B}\otimes\left({}_{A,B}\langle b_1'b_2'|\otimes{}^{\text{in}}_{4,5}\langle\psi|\right)\boldsymbol{V}^{+}\right]\\
&=\sum_{b_1,b_2,b_1',b_2'\in\{0,1\}}\left[{}_{A,B}\langle b_1b_2|\,\boldsymbol{U}|\psi\rangle^{\text{in}}_{0\sim3}\langle\psi|\,\boldsymbol{U}^{+}\,|b_1'b_2'\rangle_{A,B}\right]\\
&\otimes\left[\boldsymbol{V}\left(|b_1b_2\rangle_{A,B}\otimes|\psi\rangle^{\text{in}}_{4,5}\right)\left({}_{A,B}\langle b_1'b_2'|\otimes{}^{\text{in}}_{4,5}\langle\psi|\right)\boldsymbol{V}^{+}\right]\\
&=\sum_{b_1,b_2,b_1',b_2'\in\{0,1\}} Tr_{A,B}\left[|b_1'b_2'\rangle_{A,B}\langle b_1b_2|\,\boldsymbol{U}|\psi\rangle^{\text{in}}_{0\sim3}\langle\psi|\,\boldsymbol{U}^{+}\right]\\
&\otimes\left[\boldsymbol{V}\left(|b_1b_2\rangle_{A,B}\langle b_1'b_2'|\right)\otimes\left(|\psi\rangle^{\text{in}}_{4,5}\langle\psi|\right)\boldsymbol{V}^{+}\right].
\end{aligned}
\tag{19}
$$

The meaning and function of each term are the same as those in the one-qubit-cutting case.

4.2. The reconstruction of U-type cutting case

The classical post-processing required for circuit cutting presents a fundamental trade-off between reconstruction fidelity and computational cost. Existing methodologies can be broadly categorized into two distinct paradigms: exact and approximate reconstruction.The exact reconstruction paradigm, rooted in the foundational theory of Peng *et al* [28]. and advanced by frameworks such as CutQC [18], aims to deterministically calculate the full probability distribution of the original circuit. CutQC's DD query is a sophisticated implementation of this, which excels at efficiently identifying all solution states for circuits with sparse outputs, such as those found in many structured quantum algorithms (e.g. Bernstein–Vazirani (BV)). However, it still cannot support the reconstruction for all types of circuit cutting scenarios, despite being a general and representative case. In contrast, the approximate reconstruction paradigm, pioneered by Chen *et al* [26] using MCMC and further refined by Lian *et al* [27] with HMC sampling, forgoes exactness for efficiency. These methods treat reconstruction as a sampling problem, focusing only on the identification of high-probability bitstrings. This approach is highly effective for optimization tasks such as QAOA, where the primary goal is to find one or a few optimal solutions, and the precise probabilities of sub-optimal states are irrelevant. By doing so, they achieve a runtime that scales much more favorably, particularly for problems where the solution space is concentrated. However, the inherent limitation is that they provide a probabilistic and incomplete picture of the output. In particular, Lian's method is only applicable to circuits with a single cut, and has not yet been extended to scenarios involving multiple cuts. Therefore, extending the existing algorithms to multi-bit cutting scenarios and expanding them to all quantum circuits is a promising direction for future research. To address this limitation, we propose a U-type cutting scheme that serves as a general and representative case for all bitwise cuts in circuit-cutting-based DQC.

An example of a two-qubit U-type circuit cutting is shown in figure 11. Its output state can be expressed as:

$$
\begin{aligned}
|\psi\rangle^{\text{out}}_{0\sim5}&=|\psi\rangle^{\text{out}}_{0,4,5}\otimes|\psi\rangle^{\text{out}}_{1,2,3}\\
&=\sum_{b_1,b_2\in\{0,1\}}\left[{}_{A}\langle b_1|\,\boldsymbol{U}\left(|b_2\rangle_{B}\otimes|\psi\rangle^{in}_{3,4,5}\right)\right]\otimes\left[{}_{B}\langle b_2|\,\boldsymbol{V}\left(|\psi\rangle^{in}_{0,1,2}\otimes|b_1\rangle_{A}\right)\right].
\end{aligned}
\tag{20}
$$

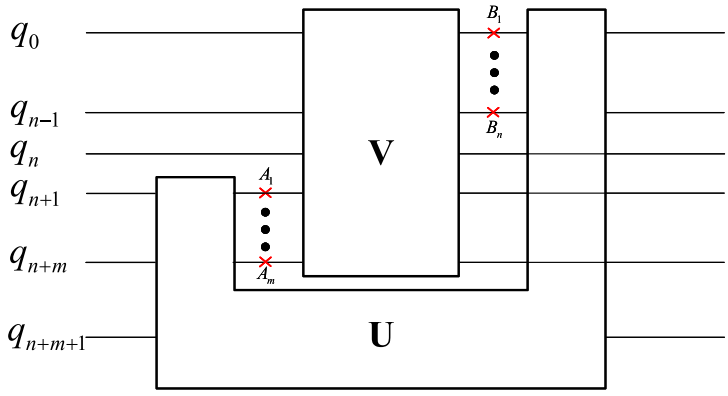

Figure 12. Example of cutting a multi-qubit circuit into two smaller subcircuits by U-type N-qubit circuit cutting,demonstrating the scalability of the proposed reconstruction model.

The corresponding density operator is:

$$\begin{aligned}
\rho_{0\sim5}^{\text{out}} &= |\psi\rangle_{0\sim5}^{\text{out}}\langle\psi| \\
&= \left[\sum_{b_1,b_2\in\{0,1\}}\left({}_A\langle b_1|\boldsymbol{U}\left(|b_2\rangle_B\otimes|\psi\rangle_{3,4,5}^{\text{in}}\right)\right)\otimes\left({}_B\langle b_2|\boldsymbol{V}\left(|\psi\rangle_{0,1,2}^{\text{in}}\otimes|b_1\rangle_A\right)\right)\right] \\
&\quad\times\left[\sum_{b_1',b_2'\in\{0,1\}}\left(\left({}_B\langle b_2'|\otimes{}_{3,4,5}^{\text{in}}\langle\psi|\right)\boldsymbol{U}^\dagger|b_1'\rangle_A\right)\otimes\left(\left({}_A\langle b_1'|\otimes{}_{0,1,2}^{\text{in}}\langle\psi|\right)\boldsymbol{V}^\dagger|b_2'\rangle_B\right)\right] \\
&= \sum_{b_1,b_2,b_1',b_2'\in\{0,1\}}\left[{}_A\langle b_1|\boldsymbol{U}\left(|b_2\rangle_B\otimes|\psi\rangle_{3,4,5}^{\text{in}}\right)\left({}_B\langle b_2'|\otimes{}_{3,4,5}^{\text{in}}\langle\psi|\right)\boldsymbol{U}^\dagger|b_1'\rangle_A\right] \\
&\quad\otimes\left[{}_B\langle b_2|\boldsymbol{V}\left(|\psi\rangle_{0,1,2}^{\text{in}}\otimes|b_1\rangle_A\right)\left({}_A\langle b_1'|\otimes{}_{0,1,2}^{\text{in}}\langle\psi|\right)\boldsymbol{V}^\dagger|b_2'\rangle_B\right] \\
&= \sum_{b_1,b_2,b_1',b_2'\in\{0,1\}}Tr_A\left[|b_1'\rangle_A\langle b_1|\boldsymbol{U}\left(|b_2\rangle_B\otimes|\psi\rangle_{3,4,5}^{\text{in}}\right)\left({}_B\langle b_2'|\otimes{}_{3,4,5}^{\text{in}}\langle\psi|\right)\boldsymbol{U}^\dagger\right] \\
&\quad\otimes Tr_B\left[|b_2'\rangle_B\langle b_2|\boldsymbol{V}\left(|\psi\rangle_{0,1,2}^{\text{in}}\otimes|b_1\rangle_A\right)\left({}_A\langle b_1'|\otimes{}_{0,1,2}^{\text{in}}\langle\psi|\right)\boldsymbol{V}^\dagger\right].
\end{aligned}\tag{21}$$

From equation (21), we can observe that U-type cutting is a general type of cutting. L-type cutting is a special type of U-type cutting. We can also extend this to the N-qubit circuit cutting case, as shown in figure 12.

There are m cutting qubits in the input of subcircuit V, positioned at $A_1, A_2, \ldots, A_m$, and n cutting qubits in the output of V located at $B_1, B_2, \ldots, B_n$. The output density operator, or reconstructed output state, can be given by

$$\begin{aligned}
\rho_{0\sim n+m+1}^{\text{out}} &= |\psi\rangle_{0\sim n+m+1}^{\text{out}}\langle\psi| \\
&= \left[\sum_{\zeta}\left({}_A\langle a|\boldsymbol{U}\left(|b\rangle_B\otimes|\psi\rangle_{n+1\sim n+m+1}^{\text{in}}\right)\right)\otimes\left({}_B\langle b|\boldsymbol{V}\left(|\psi\rangle_{0\sim n}^{\text{in}}\otimes|a\rangle_A\right)\right)\right] \\
&\quad\cdot\left[\sum_{\zeta'}\left(\left({}_B\langle b'|\otimes\langle\psi|_{n+1\sim n+m+1}^{\text{in}}\right)\boldsymbol{U}^+|a'\rangle_A\right)\otimes\left(\left({}_A\langle a'|\otimes\langle\psi|_{0\sim n}^{\text{in}}\right)\boldsymbol{V}^+|b'\rangle_B\right)\right] \\
&= \sum_{\zeta,\zeta'}\left[{}_A\langle a|\boldsymbol{U}\left(|b\rangle_B\otimes|\psi\rangle_{n+1\sim n+m+1}^{\text{in}}\right)\left({}_B\langle b'|\otimes\langle\psi|_{n+1\sim n+m+1}^{\text{in}}\right)\boldsymbol{U}^+|a'\rangle_A\right] \\
&\quad\otimes\left[{}_B\langle b|\boldsymbol{V}\left(|\psi\rangle_{0\sim n}^{\text{in}}\otimes|a\rangle_A\right)\left({}_A\langle a'|\otimes\langle\psi|_{0\sim n}^{\text{in}}\right)\boldsymbol{V}^+|b'\rangle_B\right] \\
&= \sum_{\zeta,\zeta'}Tr_A\left[|a'\rangle_A\langle a|\boldsymbol{U}\left(|b\rangle_B\langle b'|\otimes|\psi\rangle_{n+1\sim n+m+1}^{\text{in}}\langle\psi|\right)\boldsymbol{U}^+\right] \\
&\quad\otimes Tr_B\left[|b'\rangle_B\langle b|\boldsymbol{V}\left(|\psi\rangle_{0\sim n}^{\text{in}}\langle\psi|\otimes|a\rangle_A\langle a'|\right)\boldsymbol{V}^+\right].
\end{aligned}\tag{22}$$

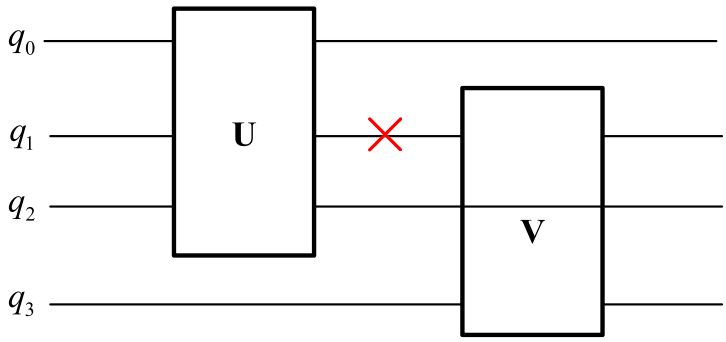

Figure 13. Example of cutting a four-qubit circuit into two smaller subcircuits of two qubits each by single-qubit cutting. Here, q_0 and q_2 are the outputs of U, and q_1 and q_3 are the outputs of V. The combination of qubits in each subcircuit is not consecutive, illustrating a discontinuous cutting scheme.

In equation (22), $\zeta = a_1, \cdots, a_m, b_1, \cdots, b_n \in \{0,1\}$, $\zeta' = a_1', \ldots, a_m', b_1', \ldots, b_n' \in \{0,1\}$, $a = a_1 \cdots a_m$, $b = b_1 \cdots b_n$, $a' = a_1' \cdots a_m'$, $b = b_1' \cdots b_n'$, $A = A_1, ..., A_m$, $B = B_1, ..., B_n$.

Considering that projection measurements are usually applied to obtain the bases state probability distribution, we only need to replace $|b\rangle_B \langle b'|$ in $Tr_A \left[|a'\rangle_A \langle a| \boldsymbol{U} \left(|b\rangle_B \langle b'| \otimes |\psi\rangle^{\text{in}}_{n+1\sim n+m+1} \langle \psi| \right) \boldsymbol{U}^+ \right]$ with the preparation of the quantum state and replace $|a'\rangle_A \langle a|$ with measurement bases. The conversion process is the same as that presented in section 4.1. Complex equations are no longer given here.

4.3. Discontinuous cutting and reconstruction order

Discontinuous cutting is a typical case in the process of QCR. Discontinuous cutting means that non-adjacent qubits, referring to the sequence of input qubits of the original circuit, are partitioned into the same subcircuits, and the qubits at the output of each subcircuit are nonadjcent. An example is shown in figure 13, where q_0, q_2 are the outputs of U, and q_1, q_3 are the outputs of V. In the process of reconstructing the results, the choice of traversal order for the Cartesian product has a decisive impact on the final output result obtained from discontinuous cutting. If the original results generated by the computation are used directly as the reconstruction output, the sequence will not align with the expected order. Therefore, the computed results must be reordered.

Specifically, for the circuit shown in figure 13, different traversal orders lead to two distinct output arrangements.

V-priority traversal mode (V is the outer loop, U is the inner loop): The output sequence produced by this mode exhibits a clear V-dominated characteristic, and the corresponding base states of the reconstructed results are [0000, 0001, 0100, 0101, 0010, 0011, 0110, 0111, 1000, 1001, 1100, 1101, 1010, 1011, 1110, 1111].

U-priority traversal mode (U is the outer loop, V is the inner loop): This mode generates a U-dominated sequence characteristic, and the corresponding base states of the reconstructed results are: [0000, 0010, 1000, 1010, 0001, 0011, 1001, 1011, 0100, 0110, 1100, 1110, 0101, 0111, 1101, 1111].

Hence, different traversal orders lead to a systematic shift in the distribution pattern of the output states in the Hilbert space. Therefore, when performing reconstruction calculations, it is important to detemine whether the traversal mode is U-priority or V-priority.

In the reconstruction process, we need five groups of parameters for each sub-circuit: the operating results(output base state probability distribution), the qubit index positions of all qubits of in the original quantum circuit, the front cutting positions of the sub-circuit, the rear cutting positions of the sub-circuit, and the priorities of the sub-circuits in this reconstruction process.

After separating the original quantum circuit, we obtained the coordinate sets of each subcircuit. To restore the subcircuits from these unordered coordinate sets, we must use the coordinate form described in appendix. The complete coordinate information can be used to obtain the corresponding quantum gate attributes. The index positions of the quantum bits can be directly extracted from the y-values of the coordinate set. The front and rear cutting positions can be determined by checking whether the quantum bit boundaries of each subcircuit match the qubit boundaries of the original quantum circuit. If they do not, the quantum bits of that sub-circuit must lie at the cutting points. Thus, four of the five groups of parameters for the quantum circuit were determined. The last parameter, the priority of the subcircuit, is complex to determine, and we discuss it in detail below.

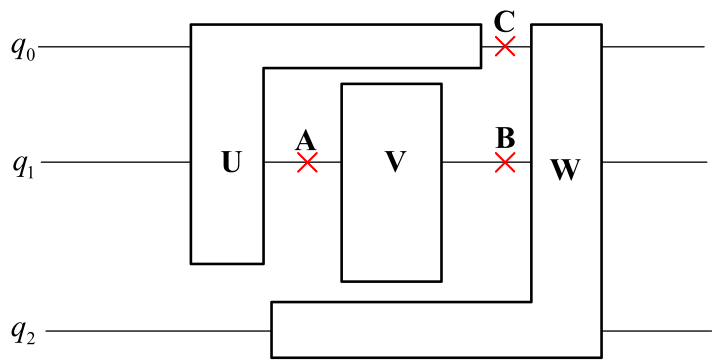

Figure 14. This is a special case of quantum circuit reconstruction. When reconstructing in the order $U \to V \to W$, the original circuit output is correctly restored. However, reconstructing in the order $U \to W \to V$ causes the intermediate subcircuit V to be discarded incorrectly.

Algorithm 2. Quantum circuit reconstruction automatic sorting (QCR-AS).

Input: *SCS*
Output: *SSCS*
for $SC \in SCS$ **do**
  if $PC_{SC} = 0$ **then**
    $FC_{SC}^{new} = FC_{SC}$, $RC_{SC}^{new} = RC_{SC}$
    remove *SC* from *SCS* and add *SC* to *SSCS*
  end if
end for
$P = 1$
while len(*SCS*) > 1 **do**
  if RC_{SC}^{old} is not empty **then**
    for $(x_i, q_i) \in RC_{SC}^{old}$ **do**
      Find all $(x_j, q_j) \in FC_{SCRS}$ such that $x_j > x_i$ and $q_j = q_i$
      Add them to *AC*
    end for
    $SC_{\text{closest}} = SCS(AC_x^{min})$
  else
    for $(x_i, q_i) \in FC_{SC}^{old}$ **do**
      Find all $(x_j, q_j) \in RC_{SCRS}$ such that $x_j < x_i$ and $q_j = q_i$
      Add them to *AC*
    end for
    $SC_{\text{closest}} = SCS(AC_x^{max})$
  end if
  $PS_{SC}^{closest} = P$, $P = P + 1$
  $FC_{SC}^{new} = UP(FC_{SC}^{old}, FC_{SC})$, $RC_{SC}^{new} = UP(RC_{SC}^{old}, RC_{SC})$
  remove $SC_{closest}$ from *SCS* and add $SC_{closest}$ to *SSCS*
end while
$PC_{SCC}^{last} = P$ and add SCC_{last} to *SSCS*

As shown in the figure 14, this is a special case of QCR. We found that when reconstructing in the order of $U \to V \to W$, the original circuit output is correctly restored. However, if the order is $U \to W \to V$, the intermediate subcircuit V is incorrectly discarded. This situation arises because rear cutting point A of U and front cutting point B of W are located at the same quantum bit index position, causing the algorithm to skip over the intermediate V subcircuit during the stitching calculation.

To address this issue, we have developed a reconstruction automatic sorting (AS) algorithm that can intelligently recognize and avoid errors such as this cross-sub-circuit stitching.

This algorithm 2, QCR-AS, is used to automatically reorder and reconstruct the sub-circuit set of a quantum circuit. The inputs of the algorithm include the sub-circuit set of a quantum circuit SCS. Where the sub-circuit SC, the front cutting coordinate set of the sub-circuit $[(x_1, q_1), (x_2, q_2), (x_3, q_3), \ldots]$ FC_{SC}, the rear cutting coordinate set of the sub-circuit $\left[\left(x_1', q_1'\right), \left(x_2', q_2'\right), \left(x_3', q_3'\right), \ldots\right]$ RC_{SC}, the priority of sub-circuit of the sub-circuit PS_{SC}, the sub-circuit cluster composed of the current sub-circuit and its closest adjacent sub-circuit, along with the newly generated front cutting coordinate set

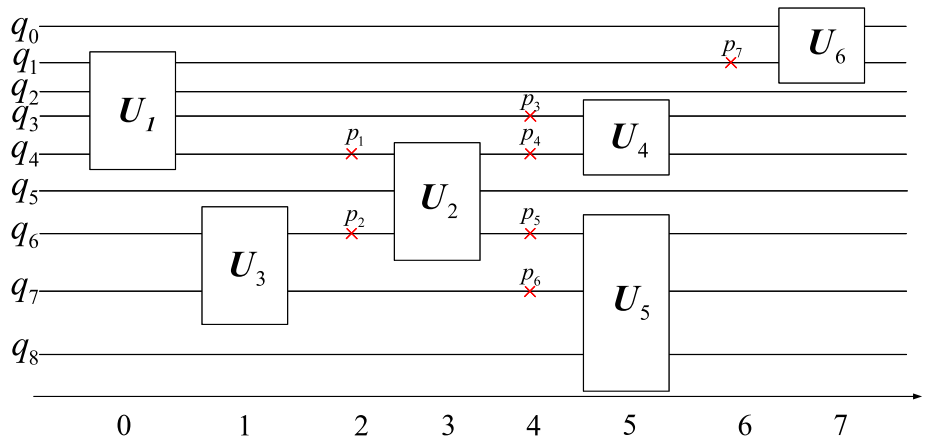

Figure 15. Example of N-qubit cutting and N-qubit circuit reconstruction. Six subcircuits are reconstructed sequentially in a predefined order to restore the output probability distribution of the original circuit.

$\left[\left(x_1^{\text{new}}, q_1^{\text{new}}\right), \left(x_2^{\text{new}}, q_2^{\text{new}}\right), \left(x_3^{\text{new}}, q_3^{\text{new}}\right), \ldots\right]$ $\text{FC}_{\text{SCC}}^{\text{new}}$, and the one from the previous iteration is denoted as $\text{FC}_{\text{SCC}}^{\text{old}}$, the sub-circuit cluster composed of the current sub-circuit and its closest adjacent sub-circuit, along with the newly generated rear cutting coordinate set $[x_1^{\text{new}\prime}, q_1^{\text{new}\prime}), (x_2^{\text{new}\prime}, q_2^{\text{new}\prime}), (x_3^{\text{new}\prime}, q_3^{\text{new}\prime}), \ldots]$ $\text{RC}_{\text{SCC}}^{\text{new}}$, and the one from the previous iteration is denoted as $\text{RC}_{\text{SCC}}^{\text{old}}$, the front cutting coordinate sets of all remaining sub-circuits $[(x_1, q_1), (x_2, q_2), (x_3, q_3), \ldots]$ FC_{SCRS}, the rear cutting coordinate sets of all remaining sub-circuits $\left[\left(x_1', q_1{}'\right), \left(x_2', q_2{}'\right), \left(x_3', q_3{}'\right), \ldots\right]$ RC_{SCRS}, the set of the closest adjacent cutting coordinates $\left[\left(x_1^{\text{adjacent}}, q_1\right), \left(x_2^{\text{adjacent}}, q_2\right), \ldots\right]$ AC, the closest adjacent sub-circuit to the current sub-circuit $\text{SC}_{\text{closest}}$. The outputs of the algorithm include the sorted sub-circuit set SSCS. Only one subcircuit has a priority of 0 (as the starting point), and the rest have a priority of -1.

In the initialization phase(Lines 1–7), Traversing SCS, the sub-circuit with a priority of 0 is taken as the starting point, and its FC, RC as $\text{FC}_{\text{SC}}^{\text{new}}$, $\text{RC}_{\text{SC}}^{\text{new}}$, and removed from the set, and then added to SSCS. If the current number of SCS is greater than one, then:

If $\text{RC}_{\text{SC}}^{\text{old}}$ is not empty, find the nearest front cutting coordinate with the minimum depth. For $\text{RC}_{\text{SC}}^{\text{old}}$, we must select all coordinate pairs from FC_{SCRS} that satisfy the following conditions: the minimum front cutting coordinate x_j is found from the set of front cutting coordinates that satisfy $x_j > x_i$ and $q_j = q_i$. During this process, we will obtain AC. Next,we find the coordinates in this set with the smallest x-coordinate $(x_j^{\min}, q_j)$ $AC_x^{\min}$. The sub-circuit with this front cutting coordinate is the nearest adjacent sub-circuit $\text{SCS}(\text{AC}_x^{\min})$; if no rear cutting coordinates exist, then the front cutting coordinates are used to find the nearest rear cutting coordinate with the maximum depth. For $\text{FC}_{\text{SC}}^{\text{old}}$, we need to select all coordinate pairs from RC_{SCRS} that satisfy the given conditions:the maximum front cutting coordinate x_j is found from the set of front cutting coordinates that satisfy $x_j < x_i$ and $q_j = q_i$. During this process, we obtain AC. Next,we find the coordinate in this set with the largest x-coordinate $(x_j^{\max}, q_j)$ $AC_x^{\max}$. The sub-circuit with this front cutting coordinate is the nearest adjacent sub-circuit $\text{SCS}(\text{AC}_x^{\max})$.

In algorithm 2, we define a global priority variable P, which is initially set to 1. Each time SC_{closest} is found, we assign P to $\text{PS}_{\text{SC}}^{\text{closest}}$, and then increment P. Subsequently, $\text{FC}_{\text{SC}}^{\text{new}}$, $\text{RC}_{\text{SC}}^{\text{new}}$ are computed based on ($\text{FC}_{\text{SC}}^{\text{old}}$, FC_{SC}) and ($\text{RC}_{\text{SC}}^{\text{old}}$, RC_{SC}), where this computation is performed by the function UP. $\text{SC}_{\text{closest}}$ was then removed from SCS and added to SSCS. This process is repeated until the number of SCS is less than one, at which point the loop terminates. When only one subcircuit remains in SCS, its priority is simply set to P and added to SSCS. The entire sub-circuit priority update, and sorting of the sub-circuits are completed.

4.4. Complete process of QCR

We will present the complete reconstruction process of the quantum circuit, with the circuit in figure 15 as an example. Based on the AS algorithm described in section 4.3, one of the reconstruction orders was $U_1 \to U_2 \to U_4 \to U_5 \to U_6 \to U_3$.

The client is required to provide the following data for each sub-circuit: the ground state probability distribution of various derived sub-circuits, the bit indices of each quantum bit in the original quantum circuit, the front cutting coordinates, the rear cutting coordinates, and the reconstruction priority of the sub-circuit in the original quantum circuit.

For a server, upon receiving the client's subcircuit data set, it first sorts the sub-circuits according to their priorities. Reconstruction calculations were then performed sequentially. The basic process is as follows. The output probabilities of the combined subcircuit of the first and second subcircuits are calculated according to the GMCR algorithm. The position of this new subcircuit was also updated. The output probabilities of further combined subcircuits of the third subcircuit and recent combined subcircuits from the first and the second subcircuits are calculated. This operation continues until all the subcircuits are processed. Reconstruction work was completed.

Here, we elaborate on the reconstruction step of combining subcircuits U_1 and U_2. We define each entry in the data format as a triple: [an exhaustive combination and permutation of input quantum states ('0', '1', '+', 'i' denote $|0\rangle$, $|1\rangle$, $|+\rangle$, $|i\rangle$, respectively; use '' if no front cutting coordinates), an exhaustive combination and permutation of measurement bases (Z, X, Y; use '' if no rear cutting coordinates), and the corresponding probability distribution (CPD)].

For U_1: in the original quantum circuit, the qubit indices are [1, 2, 3, 4], the front cutting coordinates are [] (if none, use an empty set to represent it), and the rear cutting coordinates are [(6, 1), (4, 3), (2, 4)](the depth of the reconstruction circuit is shown at the bottom of figure 15). The desired formats for U_1 are: ['', *ZZZ*, CPD_1], ['', *ZZX*, CPD_2], ..., ['', *YYY*, CPD_{27}]. Clearly, U_1's derived subcircuits have $3^3 = 27$ sets of basis state probability distributions. For U_2: in the original quantum circuit, the qubit indices are [4, 5, 6], the front cutting coordinates are [(2, 4), (2, 6)], and the rear cutting coordinates are [(4, 4), (4, 6)]. The desired format for U_2 is: [*00*, *ZZ*, CPD_1], [*00*, *ZX*, CPD_2], ..., [*ii*, *YY*, CPD_{144}]. Clearly, U_2's derived sub-circuits have $4^2 \times 3^2 = 144$ sets of basis state probability distributions.

The server processes these data according to the following steps:

(1) We observe that U_1 has no front cutting coordinates, where U_2 has rear cutting coordinates [(4,4), (4,6)]. Because none of U_1's qubit indices in the front cutting coordinates match any qubit indices in U_2's rear cutting coordinates, we determine that U_1's internal front cutting coordinates are [], and likewise, its external front cutting coordinates are also []. Next, we analyze U_1's internal rear cutting coordinates. U_1's rear cutting coordinates are [(6,1), (4,3), (2,4)], and U_2's front cutting coordinates are [(2,4), (2,6)]. Among these, U_1's rear cutting coordinate (2,4) matches U_2's front cutting coordinate (2,4) in terms of the same qubit index 4. The depth of the rear cutting coordinate of subcircuit U_1 is 2, which is less than or equal to the depth of the front cutting coordinate of subcircuit U_2, whose value is 2. Thus, we identify (2,4) as the internal rear cutting coordinates of U_1. The remaining coordinates (6,1) and (4,3) were considered as the external rear cutting coordinates of U_1. Internal cutting coordinates were used for the reconstruction computation, whereas external cutting coordinates were used for grouping. Based on the external front cutting coordinates [] and external rear cutting coordinates [(6,1), (4,3)], we regroup the 27 basis state probability distributions of U_1 provided by the client into 9 groups. Here, we explain what is meant by a grouped basis state probability distribution. Taking ['', *ZZZ*, CPD_1], ['', *ZZX*, CPD_1], and ['', *ZZY*, CPD_1], in this case, we fix Z-basis measurements on qubit indices [1,3], while qubit index 4 is measured under Z,X, and Y measurements, resulting in three groups of distinct outcomes. Three outcomes were prepared for use in the reconstruction formula. The remaining eight groups followed the same logic; hence, we reorganized the 27 original datasets from the client into a new set of nine grouped datasets.

(2) We computed the corresponding data for U_2. First, we determined the internal front cutting coordinates for U_2. U_2's front cutting coordinates are [(2,4), (2,6)], and U_1's rear cutting coordinates are [(6,1), (4,3), (2,4)]. Among these, U_2's front cutting coordinate (2,4) matches U_1's front cutting coordinate (2,4) in terms of the same qubit index 4. The depth of the front cutting coordinate of subcircuit U_2 is 2, which is greater than or equal to the depth of the rear cutting coordinate of subcircuit U_1, whose value is 2. Therefore, U_2's internal front cutting coordinate is [(2,4)], and its external front cutting coordinate is [(2,6)]. Next, we determine U_2's internal rear cutting coordinates. U_2's rear cutting coordinates are [(4,4), (4,6)], and U_1's front cutting coordinates are []. Because there are no shared qubit indices between U_2's rear cutting coordinates and U_1's front cutting coordinates, the internal rear cutting coordinates for U_2 are []. Accordingly, the external rear cutting coordinates are [(4,4), (4,6)]. As before, internal cutting coordinates were used for reconstruction calculations, whereas external cutting coordinates were used as the basis for grouping. Based on the external front cutting coordinate [(2,6)] and the external rear cutting coordinates [(4,4), (4,6)], we regroup the 144 basis state probability distributions of U_2 provided by the client into 36 groups. Take [*00*, *ZZ*, CPD_1],[*10*, *ZZ*, CPD_2],[*+0*, *ZZ*, CPD_3],[*i0*, *ZZ*, CPD_4] as an example: in this case, we fix Z-basis measurements on qubit indices [4,5], and prepare 0-states on qubit indices [6], when qubit 4 is prepared in the quantum states $|0\rangle$, $|1\rangle$, $|+\rangle$, and $|i\rangle$, respectively,

Algorithm 3. N-Subcircuit n-qubit cutting-based reconstruction.

Input: P_{scs}
Output: P_{recon}
1: Let P_{sc}^{pre} be the first element of P_{scs}
2: **for** $P_{sc} \in P_{scs}[1:]$ **do**
3: $EFC, ERC = \text{Compute}(P_{sc}^{pre}, P_{sc})$
4: **for** $i \in EFC$ **do**
5: **for** $j \in ERC$ **do**
6: Add $\text{GMCR}(P_{SC_i}^{pre}, P_{sc_j})$ to P_{sc}^{cur}
7: **end for**
8: **end for**
9: $P_{sc}^{pre} = P_{sc}^{cur}$
10: **end for**
11: $P_{recon} = P_{sc}^{pre}$

resulting in four group of distinct outcomes. Four outcomes were prepared for use in the reconstruction formula. The remaining 35 groups were subjected to the same logic. Hence, we reorganized the client's 144 original datasets into a new set of 36 grouped datasets.

(3) For the combined subcircuit of U_1 and U_2, a total of $4^1 \times 3^4 = 324$ operations are required to obtain all possible probability distributions. For example, to perform reconstruction using the grouped basis state probability distribution ['', *ZZZ*, CPD_1],['', *ZZX*, CPD_1],['', *ZZY*, CPD_1] from U_1 and [*00*, *ZZ*, CPD_1],[*10*, *ZZ*, CPD_2],[*+0*, *ZZ*, CPD_3],[*i0*, *ZZ*, CPD_4] from U_2. The result computed through the reconstruction formula corresponds to the output of the new composite subcircuit formed by combining U_1 and U_2, where the quantum state at the front cut qubit index 6 is initialized to $|0\rangle$, and the measurement bases at the rear cut qubit indices [1, 3, 4, 6] are all set to the Z-basis. The remaining 323 computations follow the same logic. Through these 324 rounds of computation, we obtain the complete dataset required for the next round of reconstruction. The new composite subcircuit involves qubit indices [1,2,3,4,5,6], with front cutting coordinate [(2,6)] and rear cutting coordinates [(6,1), (4,3), (4,4), (4,6)]. Thus, all the data required for the next round of computation are fully prepared.

Repeat steps (1) through (3) until the reconstruction reaches the final subcircuit. At that point, the basis state probability distribution of the original quantum circuit is successfully obtained.

The detailed reconstruction procedure is shown in algorithm 3. The inputs of the algorithm include all the probability distributions of the subcircuits in special configurations $P_{\text{scs}} = \{P_{\text{sc}_1}, P_{\text{sc}_2}, \ldots\}$. Where P_{sc} denotes the probability distribution of a subcircuit in special configurations,the probability distribution of the combined subcircuit computed from the previous reconstruction calculation $P_{\text{sc}}^{\text{pre}}$, the probability distribution of the combined subcircuit computed from the current reconstruction calculation $P_{\text{sc}}^{\text{cur}}$, the internal front cutting coordinate set of the subcircuit for the current reconstruction $[(x_1^{\text{IF}}, q_1^{\text{IF}}), (x_2^{\text{IF}}, q_2^{\text{IF}}), (x_3^{\text{IF}}, q_3^{\text{IF}}), \ldots]$ IFC, the internal rear cutting coordinate set of the subcircuit for the current reconstruction $[(x_1^{\text{IR}}, q_1^{\text{IR}}), (x_2^{\text{IR}}, q_2^{\text{IR}}), (x_3^{\text{IR}}, q_3^{\text{IR}}), \ldots]$ IRC, the external front cutting coordinate set of the subcircuit for the current reconstruction $[(x_1^{\text{EF}}, q_1^{\text{EF}}), (x_2^{\text{EF}}, q_2^{\text{EF}}), (x_3^{\text{EF}}, q_3^{\text{EF}}), \ldots]$ EFC, the external rear cutting coordinate set of the subcircuit for the current reconstruction $[(x_1^{\text{ER}}, q_1^{\text{ER}}), (x_2^{\text{ER}}, q_2^{\text{ER}}), (x_3^{\text{ER}}, q_3^{\text{ER}}), \ldots]$ ERC.The outputs of the algorithm includes: the reconstructed probability distribution P_{recon}.

In the algorithm 3, first, we set $P_{\text{sc}}^{\text{pre}}$ be the first element of P_{scs}. Next, we retrieved the subsequent subcircuit from the set and entered the iteration loop. In line 3, the external front and rear cutting coordinate sets (EFC and ERC) are computed. We denote this computation process as function Compute(), which takes two subcircuits as input and returns the corresponding EFC and ERC. In lines 4–8, we perform a nested iteration over the elements in *EFC* and ERC, corresponding to the grouping operation previously described. For each pair from the grouped sets, a reconstruction computation, $\text{GMCR}(P_{\text{SC}_i}^{\text{pre}}, P_{\text{sc}_j})$ is performed. Each computation yielded a probability distribution corresponding to the newly composed subcircuit under specific conditions. The results were successively stored in the set $P_{\text{sc}}^{\text{cur}}$. After completing the current round of reconstruction, set $P_{\text{SC}_i}^{\text{pre}}$ is updated to set $P_{\text{SC}_i}^{\text{cur}}$ (as shown in line 9), and the next round begins. This procedure was repeated until all the elements in the set P_{scs} were processed. In line 11, P_{recon} is updated to $P_{\text{SC}_i}^{\text{pre}}$, thus, the reconstructed probability distribution of the original circuit is obtained.

In GMCR scheme, density operator is used to represent the output state of the original circuit and the state at the cutting points. Combined with AS algorithm QCR-AS, density operator can adapt to complex cutting schemes which are verified in the quantum circuits in section 5. Compared with the approaches in the existing literatures, the proposed GMCR-based DQC include complex cutting schemes, e.g. U-type cutting (discussed in section 4.3). These schemes are obtained by optimizing for less non-local quantum gates, less execution rounds, and less swap operations.

5. Benchmarks and discussion

5.1. Circuits

We selected five quantum circuits to evaluate the proposed algorithms. They encode circuits for the 7-qubit Steane code, circuit of Shor's algorithm, quantum supremacy circuit, quantum circuit of BV algorithm, and circuit of approximate quantum Fourier transform (AQFT), respectively. For convenience, we have numbered them as $C_1, C_2, \ldots, C_5$, as listed in table 1.

Figure 16 presents circuit C_1, a prototypical example of the Calderbank-Shor-Steane (CSS) class of quantum error-correcting codes. This sophisticated encoding scheme implements a redundancy-based approach to quantum information protection, where a single logical qubit is fault-tolerant and encoded into a block of seven physical qubits. In this circuit, qubit q_0 serves as the input qubit carrying the logical information, whereas qubits q_1 through q_6 are initialized in the $|0\rangle$ state and act as ancillary qubits to assist in the encoding process. The encoded state enables the detection and correction of arbitrary single-qubit errors, including bit-flip (X), phase-flip (Z), and their coherent combinations (Y errors), thereby preserving quantum coherence against decoherence processes. The encoding circuit, as illustrated, employs a carefully designed sequence of fundamental quantum operations, primarily consisting of Hadamard gates and CNOT gates arranged in a specific topological configuration. This circuit architecture implements the stabilizer formalism of the Steane code, where the logical state is prepared in the simultaneous $+1$ eigenspace of the stabilizer generators of the code. The judicious arrangement of these quantum gates ensures the creation of the necessary entanglement structure among the physical qubits, while maintaining fault tolerance during the encoding process.

When users import the QASM file of the quantum circuit into a program, the system first converts the quantum gates in the QASM file into a coordinate-based mathematical representation. This conversion process fully preserves the topological relationships and temporal sequence information of quantum gates.

The program then employs algorithm 1 in section 3.3 for optimal partitioning, with its key parameters rigorously optimized: an initial temperature of 1000 K to ensure sufficient search space, a termination temperature of 1 K to guarantee the algorithm convergence, a cooling coefficient of 0.99 to balance convergence speed with optimization quality, and 50 iterations at each temperature to ensure thorough exploration.

During the partitioning process, the following constraints are applied to each newly generated partition in every iteration: the number of subcircuits must not exceed that of the initial scheme, and each sub-circuit can contain no more than five qubits. These constraints ensure the physical feasibility of partitioned quantum circuits, while maintaining their computational integrity.

5.2. Metrics

We select the following metrics to evaluate the performance of the proposed algorithm with respect to quantum circuit cutting:

(1) Number of edges cut *Nc*. This metric determinates the number of rounds of reconstruction, that is the optimization function f_1, as defined in section 3.3.
(2) Number of subcircuits *Ns*. This number is the number of quantum computers related to the cost of the DQC.
(3) Maximal depth of subcircuits *d_max*. The coherence time of the qubit limits the maximum depth of a quantum circuit. The lower the circuit depth, the higher fidelity of the circuit.
(4) Number of nonlocal gates *Nng*. The nonlocal gate is based on a quantum-entangled state. The greater the number of nonlocal gates, the more entangled the state is consumed.
(5) Total rounds of quantum computation *Rqc*. This metric determine the computation complexity, which is an important metric.
(6) Number of swap operations *Nswap*. In the process of qubit mapping from logic qubits to physical qubits, is a key function for non-adjacent qubits, as described in Sex.

Table 1. A summary of five benchmark quantum circuits used to evaluate the performance of the proposed iMOSA-DQC and GMCR algorithms, including the 7-qubit Steane code, Shor's algorithm, and others.

No	Quantum circuits
C_1	Encoding circuit for the 7-qubit Steane code
C_2	The circuit of Shor's algorithm
C_3	Quantum supremacy circuit
C_4	The quantum circuit of Bernstein–Vazirani (BV) algorithm
C_5	The circuit of approximate quantum Fourier transform (AQFT)

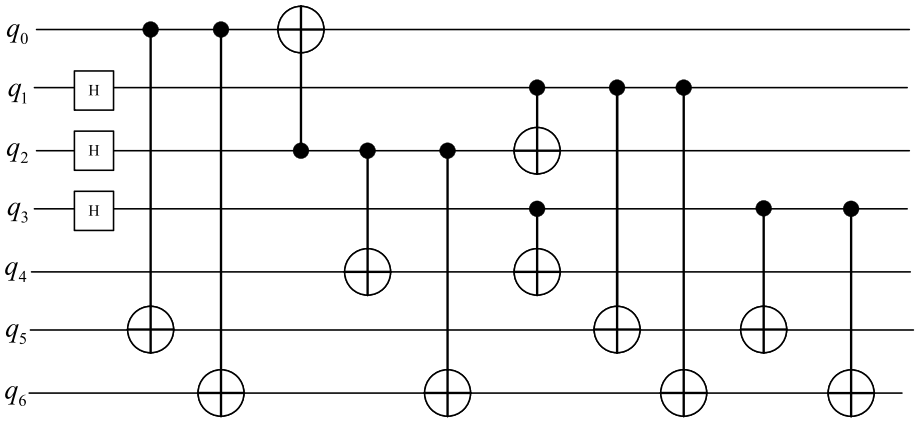

Figure 16. Encoding circuit of 7-qubit Steane code. Reproduced from [34]. CC BY 4.0.

We chose the following metrics to evaluate the performance of the proposed algorithm with respect to the uantum Circuit Reconstruction AS:

(1) The U-type structure support *Us*. This metric indicates whether the algorithm supports U-type circuit cutting, a generalized form of cutting strategy as defined in section 4.2.
(2) Multi-qubit cutting support *Mqc*. This metric assesses whether the original circuit can be partitioned by cutting two or more qubits, thereby enabling flexible decomposition strategies.
(3) Multi-circuit support *Mc*. This evaluates the ability of the algorithm to decompose the original quantum circuit into more than two subcircuits, which is essential for large-scale DQC.
(4) Reconstruction Accuracy *Ra*. This metric examines whether the reconstruction of the original output is exact or approximate, reflecting the fidelity of the decomposition-reconstruction process.
(5) Targeting a specific circuit *Tsc*. This assesses whether the algorithm is designed for general-purpose circuit decomposition or is tailored exclusively to specific types of circuits.
(6) Time complexity *Tc*. This metric refers to the computational complexity of the algorithm, indicating its efficiency and scalability with respect to circuit size.

5.3. Results and discussion

As a reference, we applied Tang *et al*'s MIP algorithm [18] to circuit C_1, with the following parameter configuration: the initial circuit input consisted of seven qubits, each subcircuit was constrained to a maximum of five input qubits, the maximum number of cuts was set to 10, and the allowable range for the subcircuit quantity was [3,4].

The results show that the cutting scheme obtained through the MIP method requires a minimum of 939 computation rounds for reconstruction, whereas the iMOSA-DQC approach achieves the same objective in 721 rounds without employing non-local gates. The MIP method performed six cuts, generating three subcircuits, with a specific distribution of cut points and subcircuits, as illustrated in figure 17. Meanwhile, iMOSA-DQC executed six cuts while also producing three subcircuits, although with distinct cutting positions as detailed in figure 18. This demonstrates that while both methods yield the same number of subcircuits, MOSA-DQC's cutting strategy substantially reduces the required number of computation rounds.

Table 2 presents a comprehensive comparison of iMOSA with the MOSA and MIP methods. The results of the MIP method (first row) correspond to the cutting scheme shown in figure 17. Moreover, compared with the MOSA partitioning scheme, iMOSA achieves fewer reconstruction rounds and

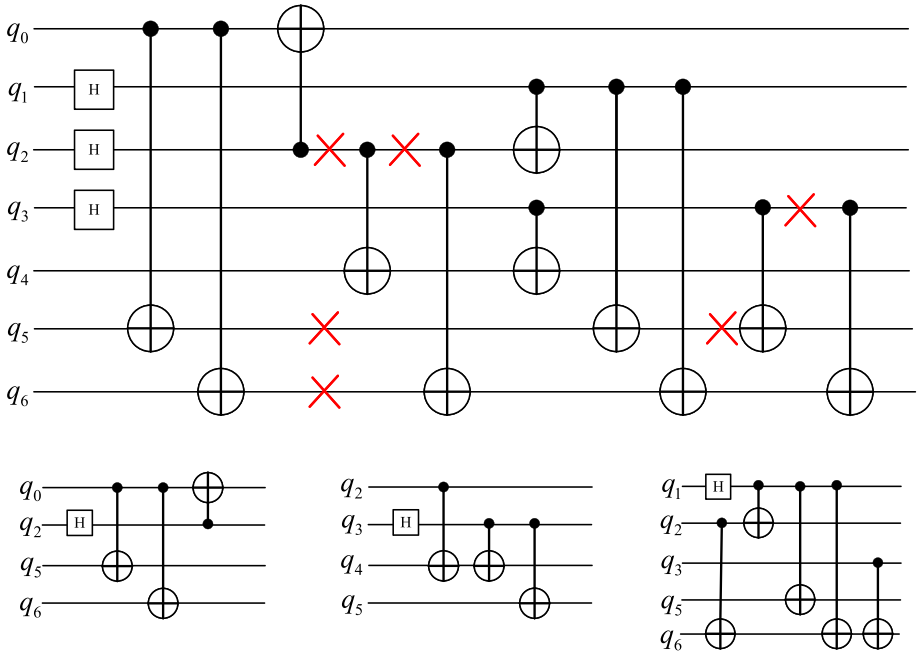

Figure 17. The MIP-based partitioning of the 7-qubit Steane code circuit, resulting in three subcircuits and requiring 939 reconstruction rounds.

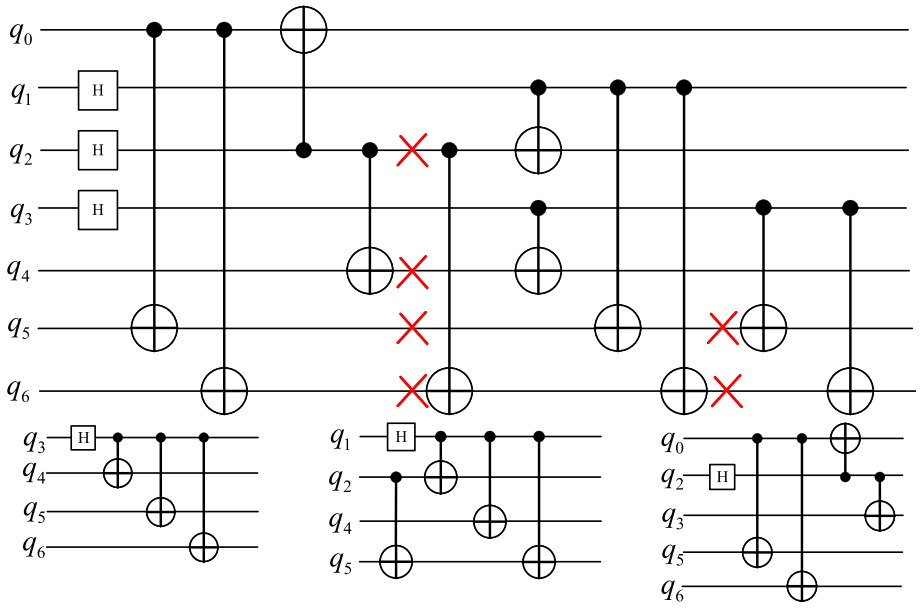

Figure 18. The iMOSA-DQC partitioning of the same circuit, also yielding three subcircuits but with only 721 reconstruction rounds, demonstrating improved efficiency.

requires fewer SWAP gate operations, given the same number of non-local gates. These results demonstrate that the introduction of the new objective function achieves further optimization while providing users with a broader selection of partitioning schemes.

The experimental results for the remaining four quantum circuits are presented in tables 3–6. Notably, the number of cutting schemes generated by the iMOSA exceeds that of MOSA by more than a factor of two. From these results, one or two representative schemes with identical objective functions were selected for tabular comparison. For the Shor's algorithm circuit, under the condition of identical numbers of nonlocal gates and reconstruction rounds, iMOSA utilizes fewer SWAP gate operations than MOSA. In the case of the supremacy circuit, with the reconstruction rounds fixed at one, iMOSA uses two fewer SWAP gate operations but incurs one additional nonlocal gate compared with MOSA. A similar trend is observed in the BV circuit. Regarding the AQFT circuit, although iMOSA

Table 2. Performance comparison of iMOSA, MOSA, and MIP algorithms on the 7-qubit Steane code circuit, showing improvements in reconstruction rounds and SWAP operations.

Algorithm	*Nc*	*Ns*	*d_max*	*Nng*	*Rqc*	*Nswap*
MIP	6	3	5	0	939	13
$MOSA_1$	6	3	6	0	721	13
$iMOSA_1$	6	3	6	0	721	12
$MOSA_2$	4	3	6	2	337	10
$iMOSA_2$	4	3	7	2	156	7
$MOSA_3$	2	2	10	3	12	7
$iMOSA_3$	2	2	10	3	12	7

Table 3. Comparison results for Shor's algorithm circuit, demonstrating iMOSA's ability to reduce SWAP operations under identical nonlocal gate and reconstruction round constraints.

Algorithm	*Nc*	*Ns*	*d_max*	*Nng*	*Rqc*	*Nswap*
MIP	4	3	9	0	96	7
$MOSA_1$	4	3	9	0	88	11
$iMOSA_1$	4	3	9	0	88	11
$MOSA_2$	3	3	9	1	84	7
$iMOSA_2$	3	3	10	1	84	4
$MOSA_3$	0	3	19	6	1	4
$iMOSA_3$	0	2	17	6	1	2

Table 4. Performance evaluation on a quantum supremacy circuit, showing trade-offs between SWAP operations and nonlocal gate counts.

Algorithm	*Nc*	*Ns*	*d_max*	*Nng*	*Rqc*	*Nswap*
MIP	4	3	10	0	103	5
$MOSA_1$	4	3	10	0	103	5
$iMOSA_1$	4	3	10	0	103	5
$MOSA_2$	0	2	22	2	1	4
$iMOSA_2$	0	3	22	4	1	2

Table 5. Results for the Bernstein–Vazirani algorithm circuit, highlighting iMOSA's optimization in multi-objective partitioning.

Algorithm	*Nc*	*Ns*	*d_max*	*Nng*	*Rqc*	*Nswap*
MIP	2	3	5	0	19	5
$MOSA_1$	2	3	5	0	19	5
$iMOSA_1$	2	3	5	0	19	5
$MOSA_2$	2	3	5	1	7	6
$iMOSA_2$	2	3	6	2	7	4

Table 6. Comparison on the approximate quantum Fourier transform circuit, showing significant reduction in reconstruction rounds despite a slight increase in nonlocal gates.

Algorithm	*Nc*	*Ns*	*d_max*	*Nng*	*Rqc*	*Nswap*
MIP	7	3	14	0	2065	23
$MOSA_1$	7	3	14	0	2065	23
$iMOSA_1$	7	3	14	0	2065	23
$MOSA_2$	6	3	14	4	1228	15
$iMOSA_2$	4	3	14	5	337	15

and MOSA exhibit the same number of SWAP gate operations 15, iMOSA achieves a reconstruction round count of only 337, which is significantly lower than that of MOSA 1228. However, iMOSA incurs one more nonlocal gate than MOSA in this circuit, and it can be observed that iMOSA is capable of reducing the number of SWAP gates in certain circuits while keeping the other two objective functions unchanged. However, in other circuits, trade-offs among multiple objectives are required, making it difficult to optimize all the performance metrics simultaneously.

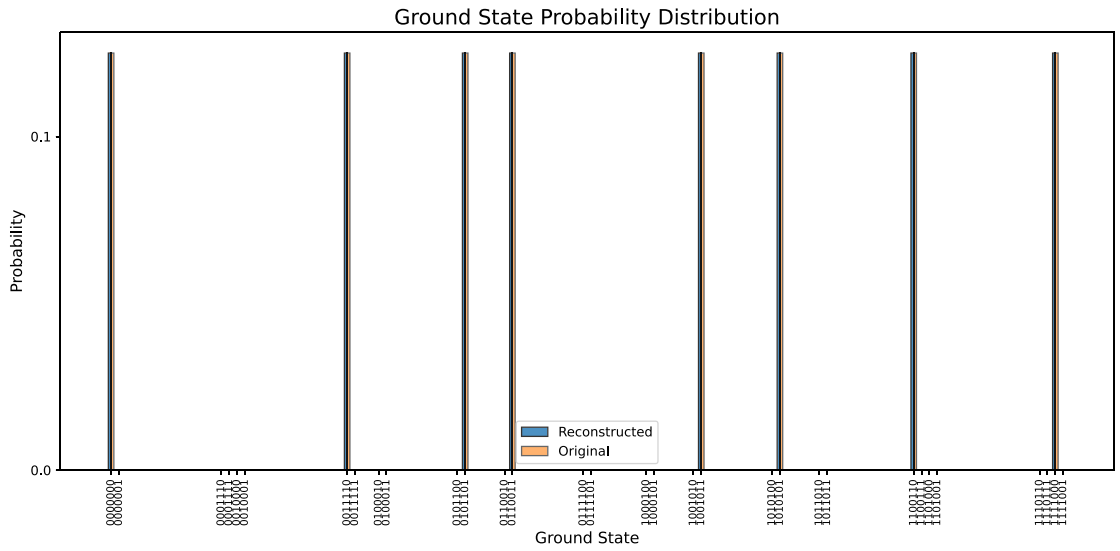

Figure 19. A comparative analysis was conducted between the reconstructed probability distribution and the original output distribution of the quantum circuit.

Table 7. Comprehensive comparison of reconstruction algorithms (DD, ARA, FRA, GMCR) across multiple criteria including support for U-type cutting, multi-qubit cutting, and reconstruction accuracy.

Algorithm	Us	Mqc	Mc	Ra	Tsc	Tc
DD	No	Yes	Yes	High	Yes	High
ARA	No	Yes	Yes	Low	Yes	Low
FRA	No	No	No	Low	Yes	Low
GMCR	Yes	Yes	Yes	High	No	High

In section 4, we derive a generalized reconstruction formula for circuit cutting. During the reconstruction process, users can either employ their own customized cutting strategies or adopt the iMOSA-DQC cutting scheme proposed in this study.

Because our reconstruction approach is based on circuit cutting, only those cutting solutions with zero nonlocal gates are directly applicable for reconstruction. The subcircuits that include the nonlocal gate can be constructed as a cluster. In the reconstruction, this type of cluster is taken as a subcircuit for the use of the generalized method.

We employed Qiskit's state vector-simulator for the subcircuit simulations. This simulator computes the exact mathematical representation of the quantum state (i.e. the state vector), thereby avoiding measurement-induced randomness and enabling ideal-state analysis and circuit verification.

In our experiment, we reconstructed a circuit based on the cutting scheme shown in figure 18. The reconstruction results, as depicted in figure 19, perfectly match the output of the original circuit, indicating a high-precision reconstruction. For comparison, we also evaluated the DD query-based reconstruction method proposed by Tang *et al* [18], which achieved exact reconstruction in this case.

However, it is worth noting that the cutting pattern in figure 17 corresponds to the 'inverted L-shape' structure, as defined earlier, which is compatible with both reconstruction approaches. In experiments of quantum circuit C_2, the 7-qubit Shor algorithm, we employed a cutting scheme with a 'U-shape' structure. In this configuration, the DD method failed to produce valid reconstruction results, whereas our method successfully reconstructed the circuit.

We also compared the Metropolis-Hastings algorithm(MH) proposed by Chen *et al* [26] and the Fast Reconstruction Algorithm (FRA) proposed by Lian *et al* [27]. Reconstruction experiments were conducted on the five aforementioned quantum circuits, and the results are summarized in table 7.

From table 7, it is evident that the primary advantage of the FRA and approximate reconstruction algorithm (ARA) over DD and GMCR lies in their lower time complexity, enabling rapid acquisition of reconstruction results. However, the reconstruction outcomes of the FRA and ARA exhibit significant deviations from the original circuit outputs, representing a fuzzy reconstruction. Moreover, FRA is limited to single-qubit cutting reconstruction, and has not been extended to circuits involving multi-qubit cuts, which are further constrained to specific circuit types, such as QAOA. DD fails to produce valid reconstruction results for circuits involving U-type cuts and is primarily applicable to circuits with

output probability distributions characterized by pronounced sparsity or density, thereby restricting its general applicability. In contrast, although the GMCR reconstruction algorithm proposed herein entails a higher time complexity, it imposes no restrictions on the cutting schemes of circuits and can perform high-fidelity reconstruction utilizing all qubit cutting configurations, demonstrating superior generality and adaptability.

The quantum circuits examples in this section show that the proposed GMCR-and-nonlocal-quantum-gate-based DQC can accept various circuit cutting schemes and can be applied to other DQC cases of large-scale quantum circuits.

6. Conclusion

In this study,we propose a GMCR for a DQC based on circuit cutting. It can handle complex cutting patterns, for example, one with a U-type structure, which cannot be processed by existing methods. To decrease the overhead of subsequent qubit mapping in the quantum compiling procedure, we add a new objective function: the number of required SWAP operations. Additionally, the solution space of the domination set increases. Five quantum circuits are used to verify the propose model and algorithm. The results show significant performance improvement, for example,in quantum circuits such as the encoding circuit of the 7-qubit Steane code, our approach effectively reduces the number of SWAP gate operations without increasing the number of nonlocal gates or reconstruction rounds, thereby further optimizing the overall performance of the cutting schemes.

However, our experiments also revealed inherent trade-offs in multi-objective optimization. In circuits such as BV and AQFT, when one objective is held constant, the remaining two cannot be optimized simultaneously, necessitating careful balancing among competing performance metrics. It is worth mentioning that a high time cost is associated with reconstructing the results of the original circuit, particularly for large-scale circuits. The use of a high-performance computing (HPC) platform can help reduce this cost, which remains a primary focus of future work.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors. The dataset is also available at GitCode (QuautumCut-Reconstruction - GitCode).

Acknowledgments

The project is supported by the National Natural Science Foundation of China (Grant No. 62471350); Foundation of Shaanxi Key Laboratory of Information Communication Network and Security(ICNS201802).

Conflict of interests

The authors declare no competing financial interests.

Appendix. Coordinate transformation of quantum circuits

Coordinate representation of quantum gates: Single-qubit gates (such as U_2, P) are represented by a single coordinate, in the format [(x, y), 'gate_name', params]. For example, $U_2(0, \pi)$ acting on qubit q_2 can be represented as [[(0, 2), 'u2', [0, π]]]. Two-qubit gates (such as CX) are represented by two coordinates for the control and target qubits, in the format [(x_ctrl, y_ctrl), (x_targ, y_targ), 'gate_name', params]. For example, CX q_2], q_1 can be represented as [(2, 2), (2, 1), 'cx', []]. Multi-qubit gates (such as Toffoli/CCX) are represented by three coordinates, in the format [(x_ctrl1, y_ctrl1), (x_ctrl2, y_ctrl2), (x_targ, y_targ), 'gate_name', params]. For example, CCX q_1, q_2, q_0 can be represented as [(5, 2), (5, 1), (5, 0), 'ccx', []]. The entire quantum circuit can be represented as a list containing the coordinate-based description of all quantum gates, i.e.:[[(0, 2), 'u2', [0, π]], [(1, 2), 'p', [π/4]], [(2, 2), (2, 1), 'cx', []], [(3, 1), 'p', [-π/4]], [(4, 0), (4, 1), 'cx', []], [(5, 2), (5, 1), (5, 0), 'ccx', []]].

However, during the process of cutting and reconstruction, we are more concerned with the spatial distribution of the quantum gates than with the specific parameters. In this case, we can extract only the coordinate information, forming a more concise representation, as follows: [(0, 2), (1, 2), (2, 2), (2, 1), (3, 1), (4, 0), (4, 1), (5, 2), (5, 1), (5, 0)]. This lightweight representation allows us to quickly obtain spatial information of the subcircuit without carrying additional redundant information, making it more

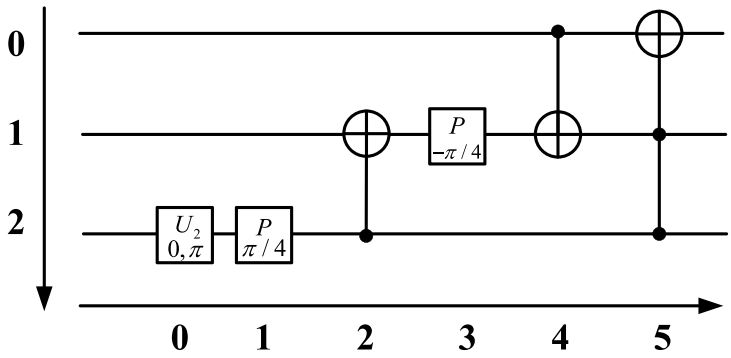

Figure A1. Coordinate transformation of quantum circuits.

suitable for structural analysis during the cutting and reconstruction processes. However, the complete coordinate representation is more suitable for accurately restoring the sub-circuit obtained from cutting for data preparation during reconstruction.

ORCID iDs

Yi Sun 0009-0003-9303-0080
Changhua Zhu 0000-0001-9267-7817
Yuan Zhao 0009-0007-8639-5005
Guangwu Hou 0009-0008-5149-1344

References

[1] Feynman R P 1982 *Int. J. Theor. Phys.* **21** 467–88
[2] Einstein A, Podolsky B and Rosen N 1935 *Phys. Rev.* **47** 777–80
[3] Shor P W 1994 Algorithms for quantum computation: discrete logarithms and factoring *Proc. 35th Annual Symp. on Foundations of Computer Science* (IEEE) pp 124–34
[4] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 *Nature* **549** 195–202
[5] Lanyon B P *et al* 2010 *Nat. Chem.* **2** 106–11
[6] Shor P W 1999 *SIAM Rev.* **41** 303–32
[7] Lee R S T 2020 Future trends in quantum finance *Quantum Finance: Intelligent Forecast and Trading Systems* (Springer) pp 399–405
[8] Fowler A G, Mariantoni M, Martinis J M and Cleland A N 2012 *Phys. Rev. A* **86** 032324
[9] Preskill J 2018 *Quantum* **2** 79
[10] Li X *et al* 2022 *Strat. Study Chinese Acad. Eng.* **24** 133–44
[11] Cuomo D, Caleffi M and Cacciapuoti A S 2020 *IET Quantum Commun.* **1** 3–8
[12] Grover L K 1997 arXiv:quant-ph/9704012
[13] Tang W and Martonosi M 2024 *Computer* **57** 131–6
[14] Caleffi M, Amoretti M, Ferrari D, Illiano J, Manzalini A and Cacciapuoti A S 2024 *Comput. Netw.* **254** 110672
[15] Barral D *et al* 2025 *Comput. Sci. Rev.* **57** 100747
[16] Peng T, Harrow A W, Ozols M and Wu X 2020 *Phys. Rev. Lett.* **125** 150504
[17] Perlin M A, Saleem Z H, Suchara M and Osborn J C 2021 *npj Quantum Inf.* **7** 64
[18] Tang W, Tomesh T, Suchara M, Larson J and Martonosi M 2021 Cutqc: using small quantum computers for large quantum circuit evaluations *Proc. 26th ACM Int. Conf. on Architectural Support for Programming Languages and Operating Systems* pp 473–86
[19] Guo X *et al* 2023 *Phys. Rev. Appl.* **19** 034044
[20] Ufrecht C, Herzog L S, Scherer D D, Periyasamy M, Rietsch S, Plinge A and Mutschler C 2024 *Phys. Rev. A* **109** 052440
[21] Liu W Q and Wei H R 2025 *Phys. Rev. Appl.* **23** 014064
[22] Hou G, Zhu C and Sun Y 2024 *Phys. Scr.* **99** 115108
[23] Liu L and Dou X 2021 Qucloud: A new qubit mapping mechanism for multi-programming quantum computing in cloud environment *2021 IEEE Int. Symp. on High-Performance Computer Architecture (HPCA)* (IEEE) pp 167–78
[24] Bäumer E, Tripathi V, Wang D S, Rall P, Chen E H, Majumder S, Seif A and Minev Z K 2024 *PRX Quantum* **5** 030339
[25] Li G, Ding Y and Xie Y 2019 Tackling the qubit mapping problem for nisq-era quantum devices *Proc. 24th Int. Conf. on Architectural Support for Programming Languages and Operating Systems* pp 1001–14
[26] Chen D, Baheri B, Chaudhary V, Guan Q, Xie N and Xu S 2022 Approximate quantum circuit reconstruction *2022 IEEE Int. Conf. on Quantum Computing and Engineering (QCE)* (IEEE) pp 509–15
[27] Lian H, Xu J, Zhu Y, Fan Z, Liu Y and Shan Z 2023 *Sci. Rep.* **13** 17773
[28] Perlin M, Tomesh T, Pearlman B, Tang W, Alexeev Y and Suchara M 2019 Parallelizing simulations of large quantum circuits *Proc. Int. Conf. for High Performance Computing, Networking, Storage and Analysis (SC'19)*

[29] Ayral T, Le Régent F M, Saleem Z, Alexeev Y and Suchara M 2020 Quantum divide and compute: Hardware demonstrations and noisy simulations *2020 IEEE Computer Society Annual Symp. on VLSI (ISVLSI)* (IEEE) pp 138–40
[30] Lowe A, Medvidović M, Hayes A, O'Riordan L J, Bromley T R, Arrazola J M and Killoran N 2023 *Quantum* **7** 934
[31] Saleem Z H, Tomesh T, Perlin M A, Gokhale P and Suchara M 2021 arXiv:2107.07532
[32] Eisert J, Jacobs K, Papadopoulos P and Plenio M B 2000 *Phys. Rev. A* **62** 052317
[33] Jiang L, Taylor J M, Sørensen A S and Lukin M D 2007 *Phys. Rev. A* **76** 062323
[34] Quan D, Liu C, Lv X and Pei C 2022 *Entropy* **24** 1107