




Distributed quantum architecture search

Haozhen Situ ¹, Zhimin He,² Shenggen Zheng ³, and Lvzhou Li ^{4,3,*}

¹College of Mathematics and Informatics, *South China Agricultural University*, Guangzhou 510642, China

²School of Electronic and Information Engineering, *Foshan University*, Foshan 528000, China

³Quantum Science Center of Guangdong-Hong Kong-Macao Greater Bay Area (Guangdong), Shenzhen 518045, China

⁴Institute of Quantum Computing and Software, School of Computer Science and Engineering, *Sun Yat-sen University*, Guangzhou 510006, China



(Received 4 April 2024; accepted 8 July 2024; published 1 August 2024)

Variational quantum algorithms, inspired by neural networks, have become a novel approach in quantum computing. However, designing efficient parameterized quantum circuits remains a challenge. Quantum architecture search tackles this by adjusting circuit structures along with gate parameters to automatically discover high-performance circuit structures. In this study, we propose an end-to-end distributed quantum architecture search framework, where we aim to automatically design distributed quantum circuit structures for interconnected quantum processing units with specific qubit connectivity. We devise a circuit generation algorithm which incorporates TeleGate and TeleData methods to enable nonlocal gate implementation across quantum processing units. While taking into account qubit connectivity, we also incorporate qubit assignment from logical to physical qubits within our quantum architecture search framework. A two-stage progressive training-free strategy is employed to evaluate extensive circuit structures without circuit training costs. Through numerical experiments on three VQE tasks, the efficacy and efficiency of our scheme is demonstrated. Our research into discovering efficient structures for distributed quantum circuits is crucial for near-term quantum computing where a single quantum processing unit has a limited number of qubits. Distributed quantum circuits allow for breaking down complex computations into manageable parts that can be processed across multiple quantum processing units.

DOI: [10.1103/PhysRevA.110.022403](https://doi.org/10.1103/PhysRevA.110.022403)

I. INTRODUCTION

Variational quantum algorithms (VQAs) [1] represent a novel approach to algorithm design that has emerged in recent years within the realm of quantum computing research. Drawing inspiration from machine learning methodologies, VQAs leverage parameterized quantum circuits (PQCs), also known as variational quantum circuits (VQCs), as analogs to neural networks. These PQCs undergo iterative updates to their gate parameters aimed at optimizing an objective function. Building upon the foundation of VQCs, a range of intriguing quantum machine learning models have been proposed. For instance, a hybrid quantum-classical neural network with deep residual learning has been proposed [2]. Quantum generative adversarial networks [3–5] integrate both quantum and classical generators and discriminators, facilitating the generation of quantum or classical data. Quantum attention networks [6–8] have been developed incorporating the attention mechanisms to better capture intricate interconnections among features within high-dimensional data. Quantum counterparts of denoising diffusion probabilistic models [9–11] have been devised to generate quantum state ensembles. Quantum analogs of capsule networks [12] have been developed. Deep quantum neural networks [13] and

quantum convolutional neural networks [14] have been realized on superconducting processors.

Drawing inspiration from neural architecture search (NAS) [15], which focuses on automatically designing neural network architectures, various schemes for quantum architecture search (QAS) have emerged. In QAS, not only are the gate parameters adjustable but the structures of quantum circuits, comprising gate types and positions, are also learnable. This adaptability yields more compact PQCs tailored to specific tasks and hardware-specific qubit connectivity.

A machine learning approach has been employed to discover quantum algorithms for computing the overlap $\text{Tr}(\rho\sigma)$ between two quantum states ρ and σ [16]. Although the swap test serves as a standard algorithm for this purpose across various applications, the learned algorithms exhibit significantly reduced depths, with one achieving constant depth. Various aspects of QAS have been explored, including modeling the search space [17,18], refining the search strategy [19,20], and enhancing the evaluation methods [21,22].

In this work, our focus lies within the domain of distributed quantum computing [23]. By interconnecting and coordinating multiple small-scale quantum processing units (QPUs), we aim to leverage a greater number of qubits to tackle larger-scale problems, thereby presenting a promising architecture for near-term quantum computing. Our objective is to automatically devise distributed quantum circuit structures tailored for specific VQA tasks. We thus endeavor to employ QAS to design distributed quantum circuits.

*Contact author: lilvzh@mail.sysu.edu.cn

Concretely, we first propose a circuit generation algorithm. This algorithm begins by transforming the graph describing the distributed system into a virtual connectivity graph. Subsequently, it iteratively samples gate types and positions randomly. Two distinct methods facilitating the implementation of nonlocal gates across various QPUs are integrated into our algorithm. Besides taking into account the qubit connectivity of QPUs, our distributed QAS framework also incorporates the qubit assignment process into its workflow, which is often overlooked in previous QAS literature. To circumvent the training costs associated with VQCs and enable exploration across a broader spectrum of circuit structures, we adopt the training-free evaluation methodology [24]. This methodology streamlines vast circuit structures through two stages, utilizing a path-based proxy and an expressibility-based proxy as filters. We evaluate our distributed QAS framework through three VQE tasks, demonstrating both the efficacy and efficiency of our approach through numerical experimentation.

The proposed scheme, which adjusts both circuit structures and gate parameters, shares similarities with the VAns (variable ansatz) method [25] and the adaptive derivative-assembled pseudo-Trotter Ansatz variational quantum eigensolver (ADAPT-VQE) algorithm [26]. The VAns method applies a set of rules to both add and remove quantum gates from the circuit in an informed manner during the optimization, while the ADAPT-VQE algorithm incrementally expands the circuit by adding gates that implement fermionic operators selected from a pool of single and double excitation operators. Compared with these methods, the proposed distributed QAS framework offers a distinct approach. Instead of using a nested optimization loop where the outer loop optimizes the circuit structure and the inner loop optimizes the gate parameters, the proposed distributed QAS employs training-free proxies to evaluate circuit structures without optimizing the gate parameters. Only a filtered selection of promising candidate circuit structures then undergo the gate parameter optimization process. Additionally, the proposed scheme features the integration of qubit assignment and the implementation of nonlocal gates, resulting in a comprehensive end-to-end distributed QAS framework.

The remainder of this article is organized as follows: In Sec. II, some related work on QAS are reviewed. Then, in Sec. III, we present our distributed QAS framework including qubit assignment, methods for nonlocal gate implementation, virtual connective graph construction, distributed circuit generation, and the search strategy. Section IV is dedicated to the evaluation of our proposed framework across three VQE tasks. Section V discusses some issues related to our framework. Finally, Sec. VI concludes this article.

II. RELATED WORK

A neural network based predictor has been used as the evaluation policy for QAS [21]. Rather than training quantum circuits to assess their performance, this method trains a neural predictor to directly gauge the performance of quantum circuits using only their structures. This predictor is then integrated into the QAS workflow to accelerate the search process. A graph self-supervised methodology was introduced

to improve predictor-based QAS [27]. A graph encoder is pre-trained using a well-designed pretext task on a large number of unlabeled quantum circuits, aiming to generate meaningful representations of quantum circuits. Subsequently, the downstream predictor is trained on a small set of quantum circuit representations paired with their labels. Once the encoder is trained, it becomes applicable to various downstream tasks.

A quantum neuro-evolution algorithm was introduced to autonomously find near-optimal quantum neural networks for different machine learning tasks [17]. This algorithm establishes a one-to-one mapping between quantum circuits and directed graphs, reducing the problem of finding the appropriate gate sequences to a task of searching suitable paths in the corresponding graph as a Markovian process. The non-sorted genetic algorithm II (NSGA-II) was employed to automatically generate optimal *ad hoc Ansatz* for classification tasks utilizing quantum support vector machines [19]. This multi-objective genetic algorithm enables the simultaneous maximization of accuracy and minimization of *Ansatz* size. A genome-length-adjustable evolutionary algorithm was utilized to design a robust VQA circuit that is optimized over variations of both circuit *Ansätze* and gate parameters, without any prior assumptions on circuit structure or depth [20]. Evolutionary quantum neural architecture search (EQNAS), an evolutionary QAS algorithm designed for image classification, was introduced [28]. This algorithm initiates a quantum population following quantum image encoding and further refines it through the application of quantum rotation gates and entirety interference crossover operations.

A quantum circuit architecture optimization algorithm leveraging Monte Carlo tree (MCT) search was proposed [18]. This algorithm first models the search space with an MCT that can be regarded as a supernet. During MCT training, the weight sharing strategy is utilized to reduce computation cost. Training results are stored in MCT nodes for future decisions, and hierarchical node selection is applied to obtain an optimal *Ansatz*. An algorithmic framework, combining nested MCT search with the combinatorial multi-armed bandit model, was introduced for the automatic design of quantum circuits [29].

A QAS scheme incorporating supernet and weight sharing was introduced [22]. This method establishes a supernet defining the *Ansatz* pool, parametrizing each *Ansatz* via weight sharing strategy. Subsequently, it iteratively samples an *Ansatz* from the pool and optimize its parameters. Following evaluation across a number of *Ansätze*, the top-performing candidate is selected and fine tuned with few iterations. QuantumNAS is a comprehensive framework for noise-adaptive co-search of variational circuits and qubit mapping [30]. Initially, a supercircuit is constructed and trained through iteratively sampling and updating the subcircuits. Then, an evolutionary co-search of subcircuit and its qubit mapping is deployed. Finally, iterative gate pruning and fine-tuning procedures are executed to eliminate redundant gates.

A general framework of differentiable QAS was proposed [31]. This approach involves the relaxation of the discrete search space of quantum circuit structures onto a continuous and differentiable domain, enabling optimization through gradient descent. QuantumDARTS is a differentiable QAS based on Gumbel-Softmax [32]. This algorithm distinguishes itself from existing methods that typically demand extensive circuit

sampling and evaluation. It introduces a micro search strategy to infer the subcircuit structure from a small-scale problem and then transfer it to a large-scale context. A gradient-based QAS algorithm enhanced with meta-learning was introduced [33]. This approach learns good initialization heuristics of the architecture, along with the meta-parameters of quantum gates from a number of training tasks. By doing so, it enables the algorithm to swiftly adjust to new tasks with minimal gradient updates, facilitating fast learning on new tasks.

QAS algorithms require calculating the performances for a large number of circuits during the search process, which incurs substantial computational costs due to the iterative updating of gate parameters. The predictor-based approach alleviates the training cost of PQCs by using a predictor to approximate the circuit performance. However, the predictor is trained using a supervised learning approach, which necessitates calculating the ground-truth performances for the circuits in the training set. The quantity of circuit-performance pairs is crucial for the predictor's generalization ability. In the weight sharing approach, the performance of a sub-circuit is estimated using parameters inherited from the super-circuit. Nevertheless, there is no guarantee that the performance of a quantum circuit with inherited parameters will strongly correlate with the performance when trained individually.

A training-free QAS approach, utilizing two proxies to rank quantum circuits in place of the expensive circuit training, was introduced [24]. Initially, directed acyclic graphs are utilized for circuit representation, and a zero-cost proxy based on the number of paths in the directed acyclic graph effectively filters out a substantial portion of unpromising circuits. Subsequently, an expressibility-based proxy, finely reflecting circuit performance, is employed to identify high-performance circuits from the remaining candidates. However, like most QAS literature, qubit assignment is not considered.

To fill the gap in automatically designing quantum circuits for distributed quantum computing, we propose a distributed QAS framework, which incorporates the latest training-free QAS approach to avoid expensive circuit training cost. The qubit assignment overlooked by previous literature is integrated into the optimization process, to better utilize the qubit topology property of the quantum device. Additionally, we incorporate two methods for implementing nonlocal gates, TeleGate and TeleData, resulting in a flexible distributed QAS framework.

III. DISTRIBUTED QUANTUM ARCHITECTURE SEARCH

A distributed quantum computing system comprises multiple QPUs interconnected by quantum links. It can be characterized by a graph $G = (Q, E, L)$, where Q represents the set of qubits, E represents the set of coupling edges between qubits on the same QPU, and L represents the set of quantum links, each connecting two communication qubits from different QPUs. There are two types of qubits on QPUs, the data qubits Q_d and communication qubits Q_c . The data qubits are dedicated to computation, while the communication qubits are reserved for facilitating nonlocal operations, which

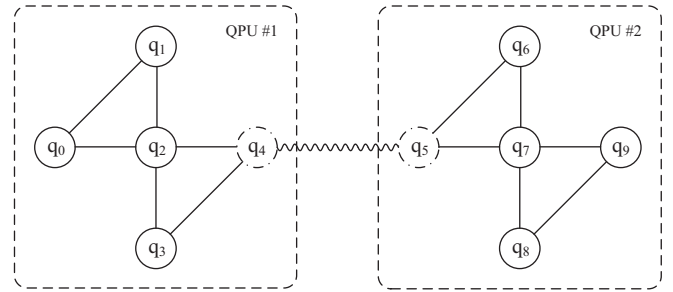


FIG. 1. Two IBM Yorktown quantum processors are interconnected via a quantum link. Circles with solid lines represent data qubits, while circles with dash-dotted lines represent communication qubits. Solid lines denote local couplings, while the wavy line represents the quantum link for distributing pairs of entangled qubits to two communication qubits.

will be elaborated later. Figure 1 depicts a distributed quantum computing system where two five-qubit QPUs are interconnected by a quantum link. The data qubit set Q_d consists of $\{q_0, q_1, q_2, q_3, q_6, q_7, q_8, q_9\}$, while the communication qubit set Q_c consists of $\{q_4, q_5\}$. The total number of data qubits available for assignment of logical qubits is eight, implying a maximum problem size of eight qubits.

Without loss of generality, we assume the edges between qubits are undirected, enabling a controlled-NOT (CNOT) gate to be applied on a pair of adjacent qubits regardless of which one serves as the control qubit. Additionally, we assume that each data qubit is connected to at most one communication qubit.

A. Qubit assignment

As the connectivity of physical qubits depends on specific QPU, certain qubits exhibit more connections to others, while some have less. Thus, a well-suited qubit assignment becomes crucial for effectively addressing computation tasks. Qubit assignment is a mapping from logical qubits to data qubits, which can be defined as

$$f_q : Q_l \longrightarrow Q_d, \quad (1)$$

where Q_l and Q_d denote logical qubits and data qubits, respectively. If there are more data qubits available than logical qubits required, some data qubits are reserved to accommodate teleported qubits. These reserved qubits are termed as empty qubits.

It's worth noting that previous research on QAS has rarely included qubit assignment f_q as a part of the optimization process, despite considering qubit connectivity. In this study, we integrate the optimization of f_q into our QAS framework to learn an optimal f_q .

B. Nonlocal gate implementation

Without the use of quantum links, only single-qubit and local two-qubit gates can be executed, while a nonlocal two-qubit gate involving two qubits from different QPUs is not feasible. In this work, we assume the native gate set includes single-qubit U gates, two-qubit CNOT gates, and SWAP gates,

defined as follows:

$$U(\theta, \phi, \lambda) = \begin{bmatrix} \cos(\theta/2) & -e^{i\lambda} \sin(\theta/2) \\ e^{i\phi} \sin(\theta/2) & e^{i(\phi+\lambda)} \cos(\theta/2) \end{bmatrix},$$

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \text{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2)$$

We use superscripts and subscripts to denote the control and target qubits of CNOT gates. For instance, CNOT_b^a indicates that qubit a serves as the control and qubit b serves as the target.

In quantum distributed computing, two widely used methods for implementing nonlocal gates are the TeleGate method and the TeleData method [23]. Both methods leverage shared entangled qubits and classical communication to realize nonlocal gates.

1. TeleGate method

We first describe the TeleGate method. Suppose the state of data qubits is represented by $|\psi\rangle$. By treating all the qubits as a single composite system, we can express its state as

$$|\psi\rangle(|00\rangle + |11\rangle)_{ab} = (|0\rangle_c |\psi_0\rangle + |1\rangle_c |\psi_1\rangle)(|00\rangle + |11\rangle)_{ab}, \quad (3)$$

where the subscript c denotes the control qubit of the nonlocal CNOT gate, while subscripts a and b denote the communication qubits sharing a Bell state. Note that $|\psi_0\rangle$ and $|\psi_1\rangle$ are unnormalized for brevity. Qubits c and a are situated within one QPU and connected by an edge, while qubit b is located on another QPU.

To implement nonlocal CNOT gates, the TeleGate method initially employs a ‘‘cat-entangler’’ primitive operation [34], which transforms the quantum state of the entire system to

$$(|00\rangle_{cb} |\psi_0\rangle + |11\rangle_{cb} |\psi_1\rangle). \quad (4)$$

The details of how this transformation works are given in the Appendix. We can see that after the cat-entangler, the control qubit c on one QPU and the communication qubit b on another QPU become entangled. Consequently, a local CNOT gate controlled by qubit b is equivalent to a nonlocal CNOT gate controlled by qubit c . For ease of explanation later on, we refer to qubit c as being in ‘‘control mode.’’

In the example depicted in Fig. 1, if a cat-entangler operation is performed on $q_2q_4q_5$, the resulting state becomes

$$|00\rangle_{q_2q_5} |\psi_0\rangle + |11\rangle_{q_2q_5} |\psi_1\rangle. \quad (5)$$

When using q_5 as the control qubit, local $\text{CNOT}_{q_6}^{q_5}$ is equivalent to nonlocal $\text{CNOT}_{q_6}^{q_2}$. Analogously, local $\text{CNOT}_{q_7}^{q_5}$ is equivalent to nonlocal $\text{CNOT}_{q_7}^{q_2}$. In this way, two nonlocal CNOT gates become possible. At the same time, qubit q_2 can still participate in local CNOT gates, such as $\text{CNOT}_{q_0}^{q_2}$, $\text{CNOT}_{q_1}^{q_2}$, and $\text{CNOT}_{q_3}^{q_2}$.

The nonlocal gates mentioned above remains feasible as long as qubit c remains in control mode, but it cannot act as the target of other gates. To transition qubit c out of control mode, the ‘‘cat-disentangler’’ primitive operation [34] is required. The cat-disentangler transforms Eq. (4) to

$$|0\rangle_c |\psi_0\rangle + |1\rangle_c |\psi_1\rangle. \quad (6)$$

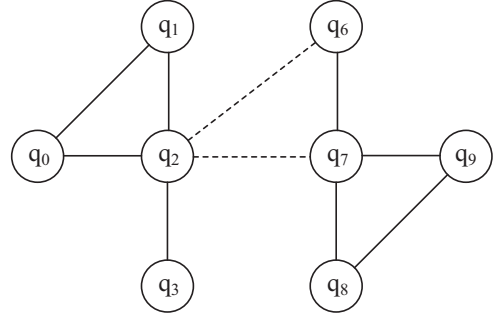


FIG. 2. The virtual connectivity graph is derived from the distributed architecture depicted in Fig. 1. Local two-qubit gates can be directed performed on qubits connected by solid lines. The two dashed lines represent virtual edges added through the cat-entangler primitive operation on qubits $q_2q_4q_5$.

The details of how this transformation works are given in the Appendix. After the cat-disentangler operation, qubit c transitions out of control mode and can participate in all local gates. Since communication qubits a and b have been measured, the shared entanglement is depleted. A new pair of entangled qubits must be distributed among qubits a and b . We say that the cat-entangler and the cat-disentangler cost one ebit.

In order to hide the intricacies of cat-entangler and cat-disentangler operations within our QAS framework, we can ignore the presence of communication qubits. We dynamically add or remove ‘‘virtual edges’’ whenever these primitive operations occur. Figure 2 depicts the virtual connectivity graph corresponding to the data qubits in Fig. 1. When a cat-entangler operation is performed on $q_2q_4q_5$, virtual edges (q_2, q_6) and (q_2, q_7) are inserted with q_2 entering control mode. With the presence of these virtual edges and the knowledge that q_2 is in control mode, two nonlocal CNOT gates CNOT_6^2 and CNOT_7^2 become feasible. Similarly, a cat-entangler operation on $q_3q_4q_5$ results in the insertion of virtual edges (q_3, q_6) and (q_3, q_7) , with q_3 entering control mode. A cat-entangler operation on $q_6q_5q_4$ leads to the insertion of virtual edges (q_6, q_2) and (q_6, q_3) , with q_6 entering control mode. A cat-entangler operated on $q_7q_5q_4$ results in the insertion of virtual edges (q_7, q_2) and (q_7, q_3) , with q_7 entering control mode. When a cat-disentangler operation occurs, the associated virtual edges are removed, and the corresponding qubit in control mode transitions out of control mode.

2. TeleData method

When there are more data qubits available than logic qubits required for the computation task, empty qubits are present. To transmit a qubit from one QPU to the empty qubit of another QPU, the teleportation protocol [35] is employed, as direct qubit transmission is susceptible to decoherence. Teleportation is a well-known quantum protocol, and we’ll omit its detailed workings here. In essence, a data qubit adjacent to a communication qubit of QPU 1 and an empty qubit adjacent to a communication qubit of QPU 2 can be swapped if the two communication qubits share a pair of entangled qubits. In the example depicted in Fig. 1, in order to perform a nonlocal CNOT gate on q_2q_8 , we can teleport q_2 to q_5 , swap q_5 and q_7 (which must be empty), and then perform a CNOT gate on q_7q_8 .

The TeleData method incurs the same cost of one ebit as the TeleGate method. If the communication qubits required for teleportation are already occupied by the cat-entangler, no entangled qubits are available for teleportation. In such cases, the cat-disentangler can be invoked to release the communication qubits, allowing for the distribution of a new pair of entangled qubits between them.

C. Virtual connectivity graph

Before generating random circuits, we first deduce a virtual connectivity graph from the graph description of the distributed quantum computing system, denoted as $G = (Q, E, L)$. The virtual connectivity graph only includes data qubits $Q_d \subset Q$, excluding edges involving communication qubits. The permissible positions of local gates are defined by

$$S_l \leftarrow E - \{(a, b) \in E \mid a \notin Q_d \vee b \notin Q_d\}. \quad (7)$$

SWAP gates that do not alter qubit connectivity are prohibited. For instance, $\text{SWAP}(q_0, q_1)$ in Fig. 2 is not allowed because neighbours of q_0 and q_1 remain unchanged after the SWAP gate. The permissible positions of SWAP gates are defined by

$$S_s \leftarrow \{(a, b) \in S_l \mid R(a) - \{b\} \neq R(b) - \{a\}\}, \quad (8)$$

where $R(x) = \{y \mid (x, y) \in S_l\}$. The TeleGate method results in adding virtual edges to the virtual connectivity graph. The permissible virtual edges are defined by

$$S_{lg} \leftarrow \cup_{(a,b) \in L} R'(a) \times R'(b), \quad (9)$$

where $R'(x) = \{y \mid (x, y) \in E\}$. The permissible positions of nonlocal gates implemented through the TeleData method are defined by

$$S_{td} \leftarrow \cup_{(a,b) \in L} [(R'[R'(a)] - \{a\}) \times R'(b)] \cup [R'(a) \times (R'[R'(b)] - \{b\})]. \quad (10)$$

D. Circuit generation

In QAS, a large number of circuit structures are explored. The algorithm for generating random distributed quantum circuit structures is provided in Algorithm 1. Given the graph description of the distributed quantum computing system, our algorithm initially identifies the permissible positions of local gates and nonlocal gates. It then proceeds iteratively, adding gates by randomly selecting their types and positions. Our algorithm ensures that empty qubits cannot be the objectives of U and CNOT gates, and at least one of the participating qubits in a SWAP gate is nonempty. The proportion of local and nonlocal two-qubit gates is predefined. The TeleGate and TeleData procedures are invoked when a nonlocal gate needs to be added.

Three categories of distributed circuits can be generated: (1) circuits with nonlocal gates implemented through the TeleGate method, (2) circuits with nonlocal gates implemented through the TeleData method, and (3) circuits with nonlocal gates implemented through either TeleGate or TeleData.

Before adding a gate, some requirements are checked to prevent redundancy. Some examples of redundant gates are illustrated in Fig. 3. In order to generate more efficient circuits, we ensure that (1) two consecutive gates cannot be of

ALGORITHM 1. Generation of distributed quantum circuits.

Input: $G = (Q, E, L)$: a graph description of the distributed quantum computing system; N_g : the number of gates in a circuit; P_g : the distribution of gate types; p_{nl} : the probability of nonlocal gates; M : the method to implement nonlocal gates.

Output: C : a distributed quantum circuit; N_e : the number of ebits

- 1: Deduce the virtual connectivity graph according to Eqs. (7)–(10).
- 2: $C \leftarrow \emptyset$
- 3: $N_e \leftarrow 0$
- 4: **while** $|C| < N_g$ **do**
- 5: select a random gate type $g \in \{U, \text{CNOT}, \text{SWAP}\}$ according to P_g
- 6: **if** $g = U$ **then**
- 7: select a random qubit q from nonempty qubits
- 8: **if** $\text{redundant}(g, q) = \text{False}$ **then**
- 9: **if** $\text{control_mode}(q) = \text{True}$ **then**
- 10: remove virtual edges related to q
- 11: $\text{control_mode}(q) \leftarrow \text{False}$
- 12: $C \leftarrow C \cup \{(g, q)\}$
- 13: **else if** $g = \text{CNOT}$ **then**
- 14: pick a random value $a \in (0, 1)$
- 15: **if** $a > p_{nl}$ **then**
- 16: randomly choose the control c and the target t from S_l satisfying neither c nor t is empty
- 17: **if** $\text{redundant}(g, c, t) = \text{False}$ **then**
- 18: **if** $\text{control_mode}(t) = \text{True}$ **then**
- 19: remove virtual edges related to t
- 20: $\text{control_mode}(t) \leftarrow \text{False}$
- 21: $C \leftarrow C \cup \{(g, c, t)\}$
- 22: **else**
- 23: call NonlocalGate (procedure 2)
- 24: **else if** $g = \text{SWAP}$ **then**
- 25: select a random position (a, b) from S_s satisfying either a or b is non-empty
- 26: **if** $\text{redundant}(g, a, b) = \text{False}$ **then**
- 27: **if** $\text{control_mode}(a) = \text{True}$ **then**
- 28: remove virtual edges related to a
- 29: $\text{control_mode}(a) \leftarrow \text{False}$
- 30: **if** $\text{control_mode}(b) = \text{True}$ **then**
- 31: remove virtual edges related to b
- 32: $\text{control_mode}(b) \leftarrow \text{False}$
- 33: $C \leftarrow C \cup \{(g, a, b)\}$
- return** C, N_e

the same type because two U gates can be combined into one, and two CNOT gates with the same control and target cancel each other out. (2) After $\text{SWAP}(a, b)$, another $\text{SWAP}(a, b)$ is

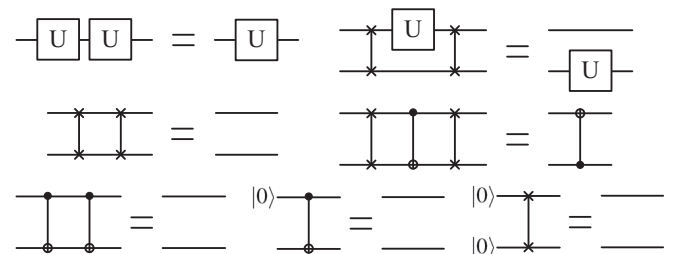


FIG. 3. Examples of redundant gates.

PROCEDURE 2. NonlocalGate.

```

1: if  $M = \text{TeleGate}$  then
2:    $S'_{tg} \leftarrow \{(a, b) \in S_{tg} | \text{nonempty}(a) = \text{True} \wedge \text{nonempty}(b) = \text{True}\}$ 
3:   if  $S'_{tg} \neq \emptyset$  then
4:     randomly select the control  $c$  and target  $t$  from  $S'_{tg}$ 
5:     if  $\text{redundant}(g, c, t) = \text{False}$  then
6:       if  $\text{control\_mode}(c) = \text{True}$  then
7:          $C \leftarrow C \cup \{(g, c, t)\}$ 
8:       else
9:         call TeleGate (procedure 3)
10:  else if  $M = \text{TeleData}$  then
11:     $S'_{td} \leftarrow \{(a, b) \in S_{td} | \text{nonempty}(a) = \text{True} \wedge \text{nonempty}(b) = \text{True}\}$ 
12:    if  $S'_{td} \neq \emptyset$  then
13:      randomly select the control  $c$  and target  $t$  from  $S'_{td}$ 
14:      if  $\text{redundant}(g, c, t) = \text{False}$  then
15:        if empty qubit for teleportation exists then
16:          call TeleData (procedure 4)
17:  else if  $M = \text{Both}$  then
18:     $S' \leftarrow \{(a, b) \in S_{tg} \cup S_{td} | \text{nonempty}(a) = \text{True} \wedge \text{nonempty}(b) = \text{True}\}$ 
19:    if  $S' \neq \emptyset$  then
20:      randomly select the control  $c$  and target  $t$  from  $S'$ 
21:      if  $\text{redundant}(g, c, t) = \text{False}$  then
22:        if  $g \in S_{tg}$  then
23:          if  $\text{control\_mode}(c) = \text{True}$  then
24:             $C \leftarrow C \cup \{(g, c, t)\}$ 
25:          else
26:            if  $g \in S_{td} \wedge$  empty qubit for teleportation exists then
27:              call TeleGate (procedure 3) or TeleData (procedure 4) with equal probability
28:            else
29:              call TeleGate (procedure 3)
30:            else
31:              if empty qubit for teleportation exists then
32:                call TeleData (procedure 4)

```

permitted if a or b has participated in a two-qubit gate (but not the same two-qubit gate) or has been involved in the TeleData procedure. (3) If the initial state of the circuit is $|0\rangle$, a CNOT gate cannot be the first gate of its control qubit. Additionally, $\text{SWAP}(a, b)$ is not allowed if a and b have not been affected by other gates except when one of them is an empty qubit.

PROCEDURE 3. TeleGate.

```

1: if required quantum link in use then
2:   remove associated virtual edges to release the link
3:   let the corresponding control qubit exit control mode
4: add virtual edges related to  $c$ 
5:  $\text{control\_mode}(c) \leftarrow \text{True}$ 
6:  $C \leftarrow C \cup \{(g, c, t)\}$ 
7:  $N_e \leftarrow N_e + 1$ 

```

PROCEDURE 4. TeleData.

```

1: from qubits  $c$  and  $t$ , select the qubit to be teleported and denote it as  $a$ 
2: if required quantum link in use then
3:   remove associated virtual edges to release the link
4:   let the corresponding control qubit exit control mode
5: if  $\text{control\_mode}(t) = \text{True}$  then
6:   remove virtual edges related to  $t$ 
7:    $\text{control\_mode}(t) \leftarrow \text{False}$ 
8: teleport qubit  $a$ 
9:  $C \leftarrow C \cup \{(g, c, t)\}$ 
10:  $N_e \leftarrow N_e + 1$ 

```

E. Search strategy

In this work, we employ a two-stage progressive strategy free from training requirements [24] to navigate the expansive search space effectively. This strategy utilizes two training-free proxies to evaluate the performance of numerous circuits efficiently. It filters out unpromising circuits, retaining only a select few candidates. Subsequently, these candidate circuits are trained on specific VQA task to ascertain the optimal circuit.

Following the generation of K_a distributed quantum circuits, the number of paths between the input and output nodes of the directed acyclic graph representation of each circuit is calculated. The circuits are then sorted in descending order based on their number of paths, and the top K_p circuits are selected. Because of its low computational cost and its capability to capture the topological complexity of quantum circuits, the path-based proxy can effectively serve as a preliminary filtering mechanism to eliminate poor-performance circuits.

Next, the expressibility [36] of each of the K_p circuits is evaluated as follows

$$\mathcal{E}(C) = D_{\text{KL}}(P_C(F) || P_{\text{Haar}}(F)), \quad (11)$$

where $P_C(F)$ represents the distribution of $F = |\langle 0|V^\dagger(\theta)|V(\theta')|0\rangle|^2$ generated by sampling random gate parameters θ and θ' , $P_{\text{Haar}}(F)$ represents the distribution of $F = |\langle \psi|\psi'\rangle|^2$ with $|\psi\rangle$ and $|\psi'\rangle$ being Haar random states, and D_{KL} denotes the Kullback-Leibler divergence. A smaller expressibility value $\mathcal{E}(C)$ indicates better expressibility of the circuit C . These circuits are once again sorted in ascending order of their expressibility values, and the top K_e circuits are chosen to constitute the candidate circuit set. Expressibility reflects the quantum circuits' capacity to uniformly reach the entire Hilbert space, making it a more precise filtering proxy. Finally, a small subset of candidate circuits that survive the two-stage filtering process undergo tailored training specific to the problem at hand.

IV. EVALUATION

We evaluate our distributed quantum architecture search framework using three six-qubit VQE tasks: the BeH_2 molecule, the Heisenberg model and the transverse-field Ising model (TFIM). The Hamiltonian for BeH_2 at its lowest-energy interatomic distance (bond distance) is constructed according to Ref. [37]. The Hamiltonian of the

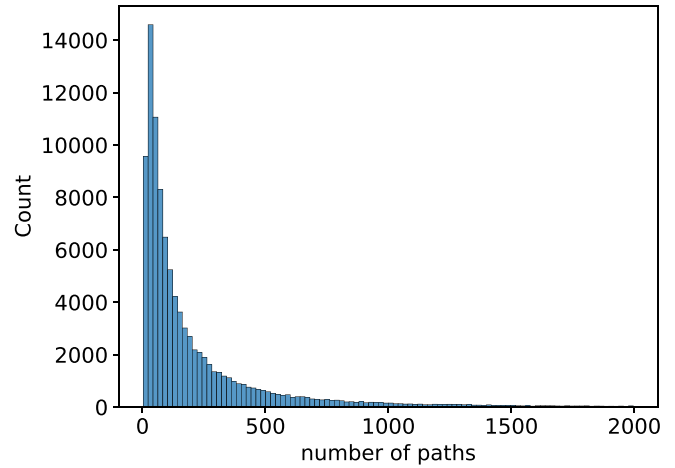
Heisenberg model with periodic boundary condition is given by $H_{\text{Heisenberg}} = \sum_i X_i X_{i+1} + Y_i Y_{i+1} + Z_i Z_{i+1} + Z_i$. The Hamiltonian of the TFIM with periodic boundary condition is $H_{\text{TFIM}} = \sum_i Z_i Z_{i+1} + X_i$. The goal of VQE tasks is to determine the ground state energy of these Hamiltonians, denoted $E_g = \min_{\theta} \langle 0 | V^\dagger(\theta) H V(\theta) | 0 \rangle$, where $V(\theta)$ represents a variational quantum circuit with gate parameters denoted by θ . A solution is considered optimal if it falls within the chemical accuracy threshold of 0.0016 of the ground-state energy.

The generated circuits adhere strictly to the topological constraints illustrated in Fig. 1. To enhance the diversity of generated circuits, the distribution of gate types P_g for each circuit is chosen randomly from three predefined distributions: (0.4, 0.2, 0.4), (0.5, 0.25, 0.25), and (0.6, 0.3, 0.1), where the three probabilities in each distribution correspond to U, CNOT and SWAP gates, respectively. Additionally, the probability of nonlocal gates p_{nl} for each circuit is randomly selected from {0.1, 0.2, 0.3, 0.4}. We maintain that the number of U gates is not less than the number of CNOT gates, and the number of nonlocal gates does not exceed the number of local gates.

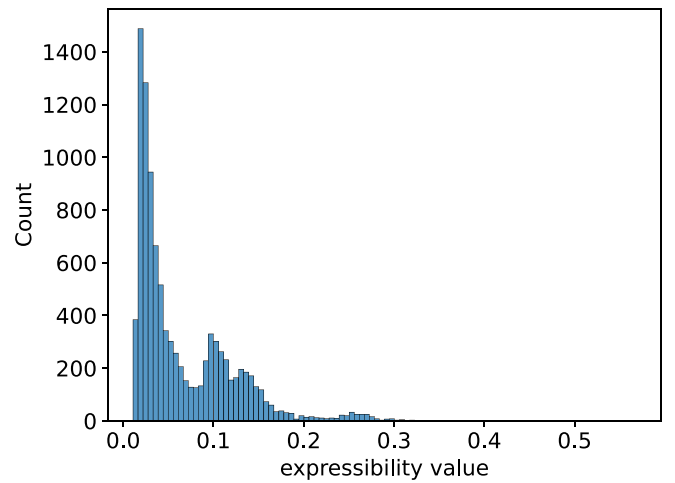
In the numerical simulations, we set the values of K_a , K_p , and K_e to 100 000, 10 000, and 1000, respectively. The hyperparameters K_a , K_p , and K_e determine the number of circuits whose number of paths, expressibility, and performance will be calculated, respectively. We choose their values by balancing the need for a sufficient number of samples with acceptable computation time. The K_e circuits, sorted based on their expressibility, serve as candidate circuits for the VQE tasks. These candidate circuits are trained individually, following the order of expressibility from better to worse. We refer to the training of each candidate circuit as a query. During each query, the candidate circuit undergoes training with 10 random initializations of parameters until convergence, achievement of chemical accuracy, or reaching the maximum number of iterations 10 000. The minimal energy found within 10 runs is regarded as the performance of a PQC. For simulation and training of quantum circuits, we utilize the TensorCircuit framework [38]. We employ the Adam optimizer with a learning rate of 0.01 to optimize the parameters of the circuits.

Figure 4 illustrates the distribution of the number of paths among the generated K_a circuits, as well as the expressibility distribution of K_p circuits featuring the highest number of paths. These circuits are generated using $N_g = 50$ gates and the TeleGate method ($M = \text{TeleGate}$). It can be observed that the distribution of circuits is concentrated in the region with fewer paths, and as the number of paths increases, the distribution exhibits a long-tail phenomenon. Given the correlation between the number of paths and expressibility elucidated in Ref. [24], we select the top- K_p circuits with the highest number of paths and calculate their expressibility. The distribution of expressibility among these selected circuits is concentrated in the region of low expressibility values, corresponding to better expressibility, thus ensuring promising performance in the VQE tasks. Similar distributions can be observed when varying N_g and M .

The distributions of the number of ebits for circuits generated with $N_g = 40, 50, 60$ gates using the TeleGate method is depicted in Fig. 5. The distributions approximate normal distributions. As the two-stage filtering process progresses, the



(a)



(b)

FIG. 4. Histograms depicting the number of paths and expressibility of generated circuits with 50 gates using the TeleGate method. (a) The number of paths for all K_a circuits. (b) The expressibility of K_p circuits filtered based on the number of paths.

mean of the normal distribution shifts towards higher values. A higher value of ebits corresponds to more nonlocal gates, thereby increasing the expressibility of the circuit. Similar distributions can be observed when employing the TeleData method or a combination of TeleGate and TeleData.

After filtering the generated circuits by the number of paths and then the expressibility, the remaining circuits are queried for VQE tasks, following the order from better to worse expressibility. Figure 6 illustrates the achieved lowest energy for the BeH₂ problem among all K_e candidate circuits when using the TeleGate method. As the number of gates increases, the distribution gradually shifts towards the optimal solution. With $N_g = 40$, the gap between the found solution and the optimal one is 0.0030, slightly larger than the chemical accuracy threshold. As N_g increases to 50, 13 optimal solutions are found, and this number rises to 79 when N_g increases to 60.

To further investigate the efficiency of finding optimal solutions, we depict the variation of the lowest energy achieved

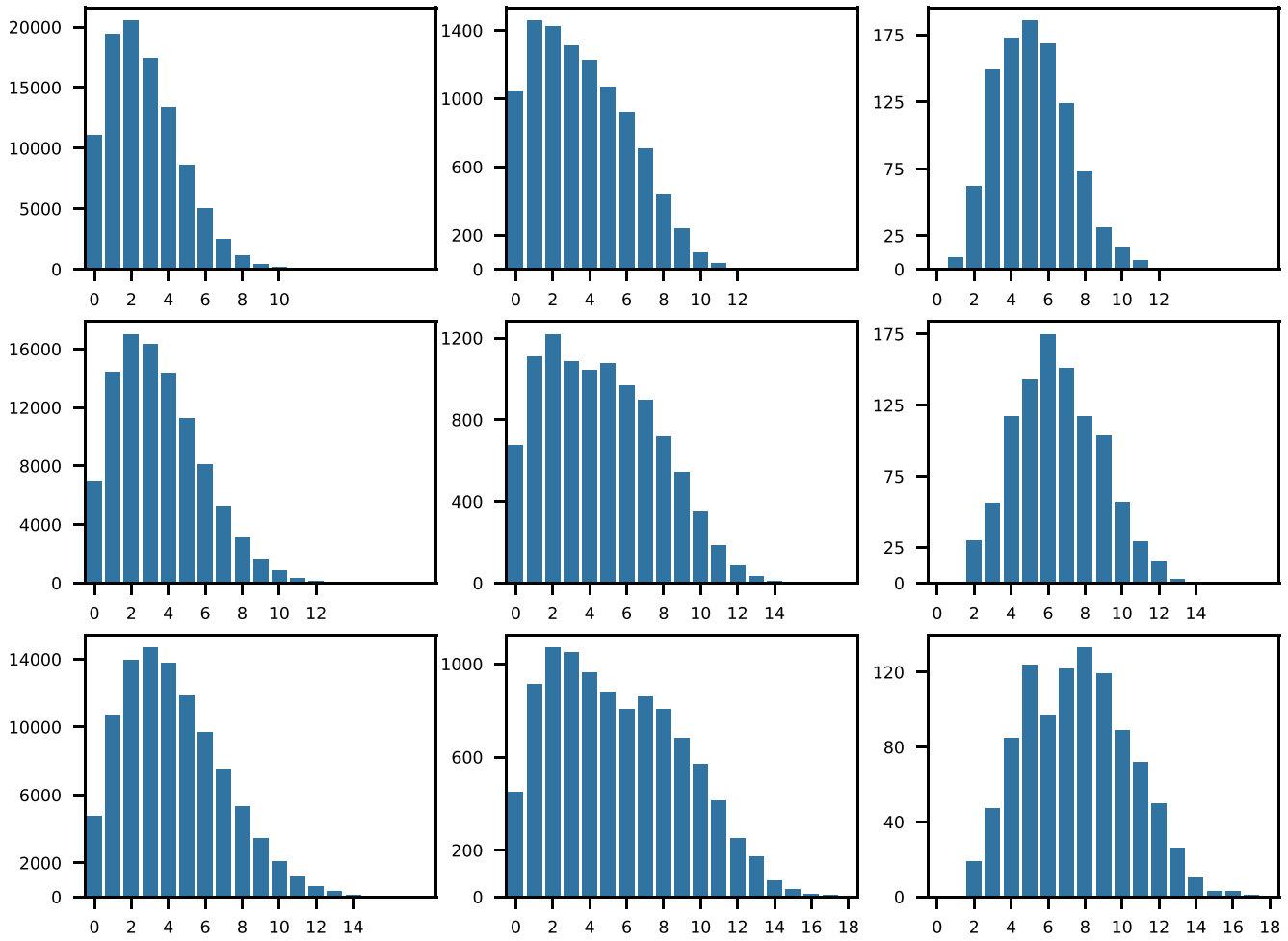


FIG. 5. Histograms depicting the number of ebits in generated circuits using the TeleGate method. The left, middle, and right column represent all K_a circuits, K_p circuits filtered by the number of paths, and K_e circuits filtered by expressibility, respectively. The top, middle, and bottom row represent circuits with 40, 50, and 60 gates, respectively.

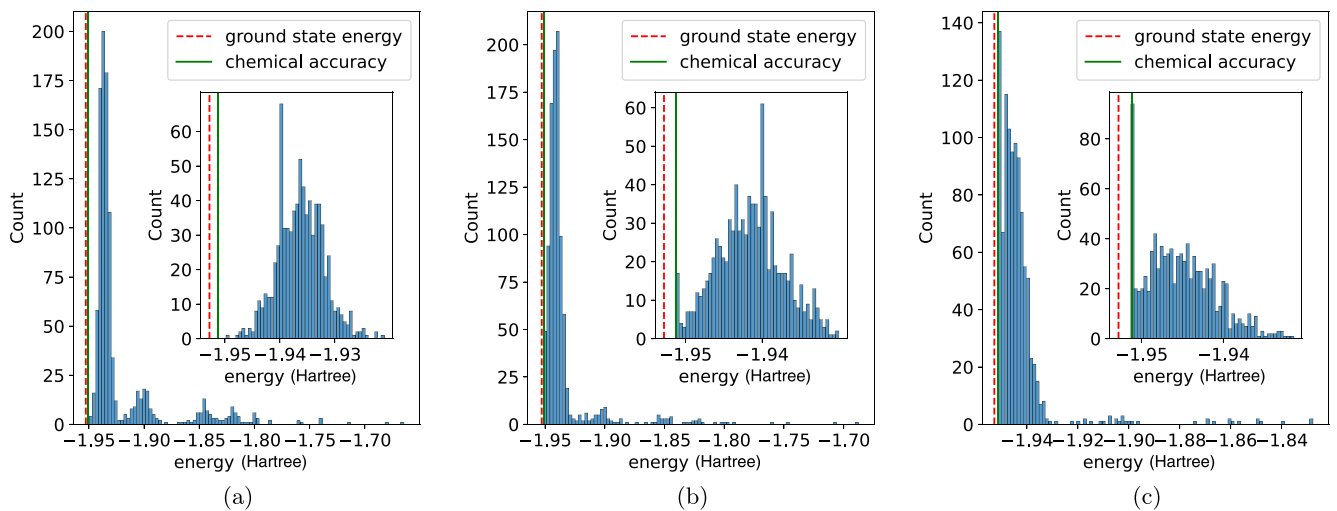
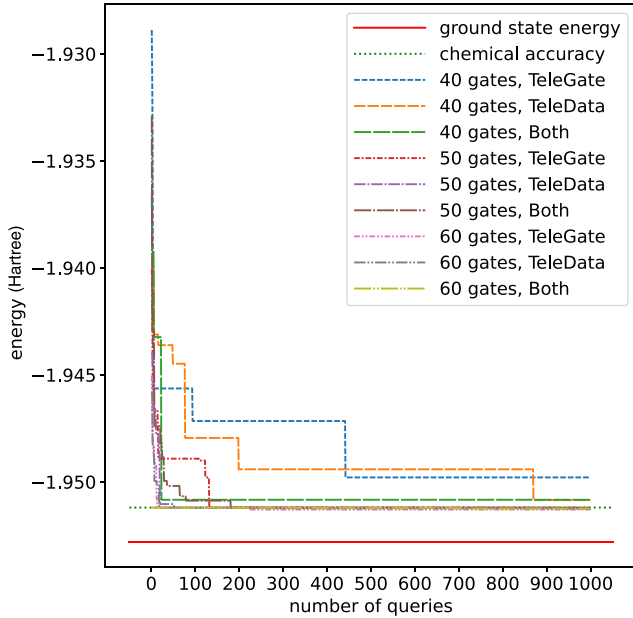
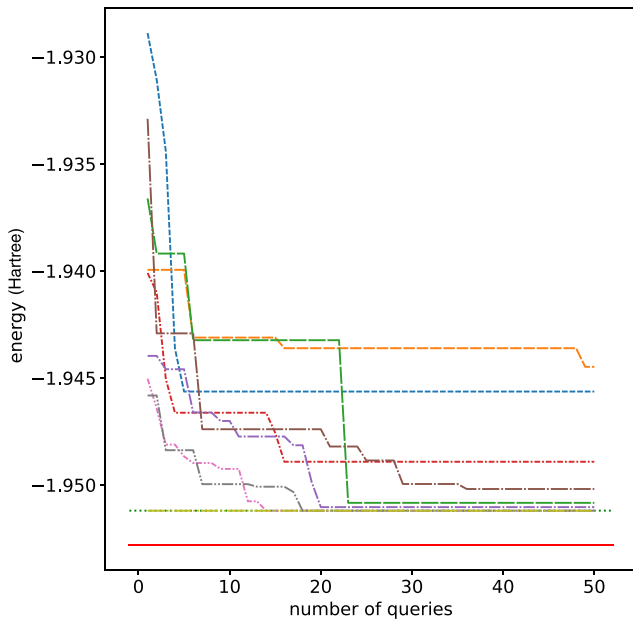


FIG. 6. Histograms depicting the lowest energy achieved for the BeH_2 problem by K_e candidate circuits using the TeleGate method. Plots (a)–(c) represent circuits with 40, 50, and 60 gates, respectively. The red dashed line denotes the ground-state energy of BeH_2 , while the green solid line indicates the energy within the chemical accuracy threshold of 0.0016.



(a)



(b)

FIG. 7. Lowest energy achieved versus number of queries for BeH_2 problem. Panel (b) is a magnification of the first 50 queries in panel (a).

with the number of queries in Fig. 7. As more candidate circuits are queried, the lowest energy achieved decreases. Generally, circuits with more gates correspond to better solution and faster convergence to optimal solutions. For instance, with $N_g = 60$ and $M = \text{Both}$, the first query successfully finds the optimal solution. Similarly, the TeleGate and TeleData methods also discover the optimal solution within 20 queries. When N_g decreases to 50, optimal solutions can be found within 200 queries. No optimal solution can be found when $N_g = 40$. The overall query process is efficient as only a small

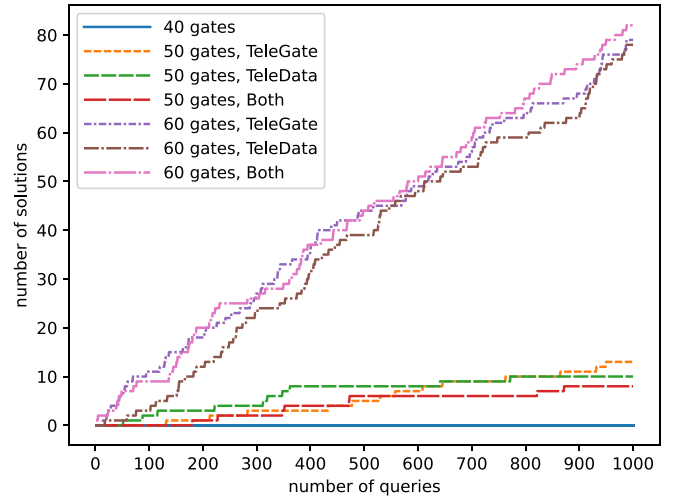


FIG. 8. Number of optimal solutions versus number of queries for BeH_2 problem.

number of candidate circuits need to be queried before finding the optimal solution. Figure 8 illustrates the relation between the number of optimal solutions and the number of queries. It can be seen that as the number of queries increases, there is a gradual increase in the number of optimal solutions.

Table I presents the simulation results for the six-qubit BeH_2 molecule, Heisenberg model, and TFIM, respectively. It is evident that circuits comprising 40 gates fail to achieve the optimal solution. Notably, the gap between the minimal energy found and the ground state energy is already very close to the chemistry accuracy threshold of 0.0016 in the case of the six-qubit BeH_2 problem. Upon increasing the number of gates to 50, optimal solutions are achieved in all three problems with the combination of the TeleGate and TeleData method. Even when only TeleGate or TeleData is utilized and the optimal solution is not immediately found, the gap has significantly diminished. With the number of gates increasing to 60, using the TeleGate method and the TeleData method alone are capable of finding the optimal solutions, leading to an increase in the number of successful circuits. The minimal costs of ebits remains relatively consistent across different methods. For the BeH_2 problem using the TeleData method, and the Heisenberg problem using either the TeleGate or Both method, the consumed ebits decrease from four to two when the number of gates increases from 50 to 60. This points toward the possibility that increasing the number of gates may help discover circuits that consume fewer ebits.

We also compare the results obtained from distributed QAS with those from the hardware efficient Ansatz (HEA), commonly used in VQAs. The HEA circuits are constructed by alternately placing odd and even layers. An odd layer is constructed as $V_{\text{odd}} = \text{CNOT}_{q_2}^{q_1} \text{CNOT}_{q_6}^{q_3} \text{CNOT}_{q_8}^{q_7} U_{q_1} U_{q_2} U_{q_3} U_{q_6} U_{q_7} U_{q_8}$, while an even layer is constructed as $V_{\text{even}} = \text{CNOT}_{q_3}^{q_2} \text{CNOT}_{q_7}^{q_6} U_{q_1} U_{q_2} U_{q_3} U_{q_6} U_{q_7} U_{q_8}$. For instance, the circuit HEA-3 has 3 layers, constructed as $V_{\text{odd}} V_{\text{even}} V_{\text{odd}}$. The nonlocal $\text{CNOT}_{q_6}^{q_3}$ can be implemented using the TeleGate method at the cost of one ebit. The comparison results in terms of the number of ebits, the number of parameters and the number of CNOT gates are presented in

TABLE I. Simulation results for the six-qubit BeH_2 molecule, Heisenberg model, and TFIM. “#Solution” indicates the number of candidate circuits that achieve the optimal solution. “Gap” denotes the difference between the minimal energy found and the ground-state energy. “#Ebit” indicates the minimum number of ebits used in the circuits that achieves the optimal solutions.

		$N_g = 40$		$N_g = 50$			$N_g = 60$	
		#Solution	Gap	#Solution	Gap	#Ebit	#Solution	#Ebit
BeH ₂	TeleGate	0	0.0030	13		3	79	3
	TeleData	0	0.0019	10		4	78	2
	Both	0	0.0020	8		3	82	2
Heisenberg	TeleGate	0	0.2114	2		4	35	2
	TeleData	0	0.3768	0	0.0095		29	3
	Both	0	0.0720	2		4	35	2
	TeleGate	0	0.0704	0	0.0128		5	3
TFIM	TeleData	0	0.0647	0	0.0108		3	3
	Both	0	0.0464	1		3	2	3

Table II. For each problem, we first list the optimal circuits discovered by distributed QAS using the TeleGate, TeleData, and Both methods, followed by the optimal HEA circuit. The differences between the lowest energy achieved by these optimal circuits and the ground state energy are less than the chemical accuracy threshold of 0.0016. Additionally, the HEA circuit with the same number of ebits as the circuits found by distributed QAS and a comparable number of parameters is also listed, along with the gap between the lowest energy achieved and the ground state energy. From Table II, we can see that, compared with HEA circuits, the circuits found by distributed QAS achieve optimal solutions using only 40%–60% of the ebits, 44%–57% of the parameters, and a comparable number of CNOT gates. This demonstrates that distributed QAS has a clear advantage in saving ebits, which are the most expensive resources in the context of distributed quantum computing.

TABLE II. Comparison results between distributed QAS and HEA *Ansätze*. HEA- l indicates hardware efficient *Ansätze* with l layers. “#Ebit,” “#Parameter,” and “#CNOT” indicate the number of ebits, the number of parameters, and the number of CNOT gates used in the circuits, respectively. “Gap” denotes the difference between the minimal energy found and the ground-state energy.

		#Ebit	#Parameter	#CNOT	Gap
BeH ₂	TeleGate	3	72	19	
	TeleData	2	90	24	
	Both	2	90	25	
	HEA-9	5	162	23	
	HEA-6	3	108	15	0.0122
Heisenberg	TeleGate	2	84	26	
	TeleData	3	96	23	
	Both	2	90	24	
	HEA-10	5	180	25	
	HEA-6	3	108	15	0.3538
TFIM	TeleGate	3	93	24	
	TeleData	3	84	24	
	Both	3	78	20	
	HEA-9	5	162	23	
	HEA-6	3	108	15	0.0402

V. DISCUSSION

A major challenge in training PQC is the barren plateau phenomenon [39]. When a PQC encounters a barren plateau, its optimization landscape becomes mostly flat as the size of the system increases. This causes the gradients to vanish exponentially, necessitating an exponentially large number of measurements to determine the parameter update direction. A fundamental relationship between expressibility and trainability has been derived [40], indicating that highly expressive *Ansätze* exhibit flatter cost landscapes and therefore will be harder to train. In the proposed distributed QAS, the two-stage filtering process chooses circuits with best expressibility. These circuits are then trained to optimize the loss function of the specific problem. However, due to the impact of barren plateaus, circuits with poor trainability may converge to nonoptimal solutions or have their training suspended once the number of training iterations reaches a predefined threshold. To mitigate this, we use multiple initializations of parameters when training a PQC, similar to the approach taken in Ref. [21], thus increasing the likelihood that at least one initialization will avoid barren plateaus. A more effective solution is to consider trainability during the filtering process. Therefore, in future work, we plan to design a training-free proxy capable of reflecting the trainability of a PQC. This will help filter out circuits with poor trainability, thereby avoiding the waste of time on training such circuits.

Our distributed QAS framework adopts a training-free approach, utilizing low-computation-cost metrics to evaluate quantum circuits. This approach enables us to explore a greater number of quantum circuits compared to training-based methods, thereby increasing the likelihood of finding optimal circuits. To better address larger-scale quantum circuit search problems, we can incorporate elements of the VAns method [25] into our framework. Specifically, by following the gate insertion rule of VAns, we can add blocks of gates to expand the circuit during the quantum circuit generation process, rather than inserting one quantum gate at a time. This enables the generation of larger-scale quantum circuits without increasing computational costs. Additionally, placing blocks of quantum gates reduces the need to select qubit positions for each individual gate, thereby controlling the size of the search space and preventing exponential growth in the

required number of circuit samples as the number of qubits increases. Meanwhile, the gate simplification rule of VAns can be employed to remove unnecessary or redundant gates, or gates with minimal impact on cost, thereby managing the growth in the number of gates.

Two commonly used nonlocal gate implementation methods are incorporated into our distributed QAS framework. In general, the TeleGate method may benefit the effective utilization of ebits. When a qubit enters control mode, multiple nonlocal or local CNOT gates using it as the control can be implemented, until the qubit becomes the target of certain gate and exits control mode. On the other hand, the TeleData method may reduce the number of nonlocal gates. After a qubit is teleported to another QPU, all gates acting on it and the native qubits of that QPU become local gates. Allowing both methods increases the flexibility of distributed quantum circuits. As shown in Table II, for each problem, the ebit consumption using both methods is no more than the consumption when using only the TeleGate method or only the TeleData method.

VI. CONCLUSION

In distributed quantum computing, the entire distributed system is composed of all qubits of multiple interconnected QPUs, and nonlocal gates across different QPUs can be implemented by methods like TeleGate or TeleData. Although various aspects of QAS have been investigated in recent years, how to design distributed quantum circuit structures automatically remained unexplored. This problem poses a more complex challenge because it requires optimization not only of gate types and positions, but also of the implementation method for nonlocal gates.

In this work, we have proposed a distributed QAS framework for multiple interconnected QPUs with specific qubit connectivity. The integration of TeleGate and TeleData, as well as the qubit assignment from logical to physical qubits, makes our QAS framework very flexible in exploring diverse circuit structures. The training-free evaluation methodology also benefits the exploration of huge amount of quantum structures, enhancing the likelihood of discovering resource-efficient circuits. Considering the qubit connectivity of two IBM quantum processors, we use the proposed framework to find distributed quantum circuits for computing the ground state energy of the BeH₂ molecule, the Heisenberg model and the transverse-field Ising model. The optimal solutions have been achieved for these problems, although only a small set of circuits are trained during the entire QAS workflow.

ACKNOWLEDGMENTS

This work was supported by the Guangdong Basic and Applied Basic Research Foundation (Grants No. 2022A1515140116 and No. 2021A1515011985), the Jihua Laboratory Scientific Project (No. X210101UZZ210), the Innovation Program for Quantum Science and Technology (Grant

No. 2021ZD0302901), the Guangdong Provincial Quantum Science Strategic Initiative (Grant No. GDZX2303007), and the National Natural Science Foundation of China (Grant No. 62272492).

APPENDIX: THE CAT-ENTANGLER AND CAT-DISENTANGLER

In this Appendix, we provide a detailed description of the cat-entangler and cat-disentangler primitive operations [34], which support the TeleGate method described in Sec. III B 1.

The quantum state before the cat-entangler operation is

$$|\psi\rangle(|00\rangle + |11\rangle)_{ab} = (|0\rangle_c|\psi_0\rangle + |1\rangle_c|\psi_1\rangle)(|00\rangle + |11\rangle)_{ab}. \quad (\text{A1})$$

The cat-entangler comprises (1) a CNOT_a^c, (2) a measurement on qubit *a*, and (3) a conditional *X* gate on qubit *b* based on the measurement outcome. Following the first step, the quantum state transforms to

$$\begin{aligned} & \text{CNOT}_a^c|\psi\rangle(|00\rangle + |11\rangle)_{ab} \\ &= |0\rangle_c|\psi_0\rangle(|00\rangle + |11\rangle)_{ab} + |1\rangle_c|\psi_1\rangle(|10\rangle + |01\rangle)_{ab}. \end{aligned} \quad (\text{A2})$$

Afterwards, qubit *a* is measured in the computational basis. If the measurement outcome is zero, the state collapses to

$$|00\rangle_{cb}|\psi_0\rangle + |11\rangle_{cb}|\psi_1\rangle, \quad (\text{A3})$$

otherwise it collapses to

$$|01\rangle_{cb}|\psi_0\rangle + |10\rangle_{cb}|\psi_1\rangle. \quad (\text{A4})$$

In the third step, a *X* gate is applied to qubit *b* only if the measurement outcome is one. Therefore, regardless of the measurement outcome, the state after the cat-entangler operation is always

$$|00\rangle_{cb}|\psi_0\rangle + |11\rangle_{cb}|\psi_1\rangle. \quad (\text{A5})$$

The cat-disentangler comprises (1) a Hadamard gate *H* on qubit *b*, (2) a measurement on qubit *b*, and (3) a conditional *Z* gate on qubit *c* based on the measurement outcome. After the first step, the quantum state becomes

$$|0\rangle_c(|0\rangle + |1\rangle)_b|\psi_0\rangle + |1\rangle_c(|0\rangle - |1\rangle)_b|\psi_1\rangle. \quad (\text{A6})$$

Then qubit *b* is measured in the computational basis. If the measurement outcome is zero, the state collapses to

$$|0\rangle_c|\psi_0\rangle + |1\rangle_c|\psi_1\rangle, \quad (\text{A7})$$

otherwise it collapses to

$$|0\rangle_c|\psi_0\rangle - |1\rangle_c|\psi_1\rangle. \quad (\text{A8})$$

In the third step, a *Z* gate is performed on qubit *c* only if the measurement outcome is one. Thus, regardless of the measurement outcome, the state after the cat-disentangler is always

$$|0\rangle_c|\psi_0\rangle + |1\rangle_c|\psi_1\rangle. \quad (\text{A9})$$

[1] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J.

Coles, Variational quantum algorithms, *Nat. Rev. Phys.* **3**, 625 (2021).

- [2] Y. Liang, W. Peng, Z.-J. Zheng, O. Silvén, and G. Zhao, A hybrid quantum–classical neural network with deep residual learning, *Neural Netw.* **143**, 133 (2021).
- [3] H. Situ, Z. He, Y. Wang, L. Li, and S. Zheng, Quantum generative adversarial network for generating discrete distribution, *Inf. Sci.* **538**, 193 (2020).
- [4] N.-R. Zhou, T.-F. Zhang, X.-W. Xie, and J.-Y. Wu, Hybrid quantum–classical generative adversarial networks for image generation via learning discrete distribution, *Signal Process.: Image Commun.* **110**, 116891 (2023).
- [5] Y. Wang, S. Xue, Y. Wang, Y. Liu, J. Ding, W. Shi, D. Wang, Y. Liu, X. Fu, G. Huang, A. Huang, M. Deng, and J. Wu, Quantum generative adversarial learning in photonics, *Opt. Lett.* **48**, 5197 (2023).
- [6] J. Shi, R.-X. Zhao, W. Wang, S. Zhang, and X. Li, QSAN: A near-term achievable quantum self-attention network, [arXiv:2207.07563](https://arxiv.org/abs/2207.07563).
- [7] R.-X. Zhao, J. Shi, and X. Li, QKSAN: A quantum kernel self-attention network, [arXiv:2308.13422](https://arxiv.org/abs/2308.13422).
- [8] R.-X. Zhao, J. Shi, and X. Li, GQHAN: A Grover-inspired quantum hard attention network, [arXiv:2401.14089](https://arxiv.org/abs/2401.14089).
- [9] M. Parigi, S. Martina, and F. Caruso, Quantum-noise-driven generative diffusion models, *Adv. Quant. Technol.*, 2300401 (2024).
- [10] B. Zhang, P. Xu, X. Chen, and Q. Zhuang, Generative quantum machine learning via denoising diffusion probabilistic models, *Phys. Rev. Lett.* **132**, 100602 (2024).
- [11] C. Chen, Q. Zhao, M. Zhou, Z. He, Z. Sun, and H. Situ, Quantum generative diffusion model: A fully quantum-mechanical model for generating quantum state ensemble, [arXiv:2401.07039](https://arxiv.org/abs/2401.07039).
- [12] Z. Liu, P.-X. Shen, W. Li, L.-M. Duan, and D.-L. Deng, Quantum capsule networks, *Quantum Sci. Technol.* **8**, 015016 (2023).
- [13] X. Pan, Z. Lu, W. Wang, Z. Hua, Y. Xu, W. Li, W. Cai, X. Li, H. Wang, Y.-P. Song, C.-L. Zou, D.-L. Deng, and L. Sun, Deep quantum neural networks on a superconducting processor, *Nat. Commun.* **14**, 4006 (2023).
- [14] J. Herrmann, S. M. Llima, A. Remm, P. Zapletal, N. A. McMahon, C. Scarato, F. Swiadek, C. K. Andersen, C. Hellings, S. Krinner, N. Lacroix, S. Lazar, M. Kerschbaum, D. C. Zanz, G. J. Norris, M. J. Hartmann, A. Wallraff, and C. Eichler, Realizing quantum convolutional neural networks on a superconducting quantum processor to recognize quantum phases, *Nat. Commun.* **13**, 4144 (2022).
- [15] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, X. Chen, and X. Wang, A comprehensive survey of neural architecture search: Challenges and solutions, *ACM Comput. Surv.* **54**, 1 (2021).
- [16] L. Cincio, Y. Subaşı, A. T. Sornborger, and P. J. Coles, Learning the quantum algorithm for state overlap, *New J. Phys.* **20**, 113022 (2018).
- [17] Z. Lu, P.-X. Shen, and D.-L. Deng, Markovian quantum neuroevolution for machine learning, *Phys. Rev. Appl.* **16**, 044039 (2021).
- [18] F.-X. Meng, Z.-T. Li, X.-T. Yu, and Z.-C. Zhang, Quantum circuit architecture optimization for variational quantum eigensolver via Monte Carlo tree search, *IEEE Trans. Quantum Eng.* **2**, 1 (2021).
- [19] S. Altares-López, A. Ribeiro, and J. J. García-Ripoll, Automatic design of quantum feature maps, *Quantum Sci. Technol.* **6**, 045015 (2021).
- [20] Y. Huang, Q. Li, X. Hou, R. Wu, M.-H. Yung, A. Bayat, and X. Wang, Robust resource-efficient quantum variational ansatz through an evolutionary algorithm, *Phys. Rev. A* **105**, 052414 (2022).
- [21] S.-X. Zhang, C.-Y. Hsieh, S. Zhang, and H. Yao, Neural predictor based quantum architecture search, *Mach. Learn.: Sci. Technol.* **2**, 045027 (2021).
- [22] Y. Du, T. Huang, S. You, M.-H. Hsieh, and D. Tao, Quantum circuit architecture search for variational quantum algorithms, *npj Quantum Inf.* **8**, 62 (2022).
- [23] M. Caleffi, M. Amoretti, D. Ferrari, D. Cuomo, J. Illiano, A. Manzalini, and A. S. Cacciapuoti, Distributed quantum computing: a survey, [arXiv:2212.10609](https://arxiv.org/abs/2212.10609).
- [24] Z. He, M. Deng, S. Zheng, L. Li, and H. Situ, Training-free quantum architecture search, in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38 (AAAI, Washington, DC, 2024), pp. 12430–12438.
- [25] M. Bilkis, M. Cerezo, G. Verdon, P. J. Coles, and L. Cincio, A semi-agnostic ansatz with variable structure for variational quantum algorithms, *Quantum Mach. Intell.* **5**, 43 (2023).
- [26] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, An adaptive variational algorithm for exact molecular simulations on a quantum computer, *Nat. Commun.* **10**, 3007 (2019).
- [27] Z. He, M. Deng, S. Zheng, L. Li, and H. Situ, GSQAS: Graph self-supervised quantum architecture search, *Physica A* **630**, 129286 (2023).
- [28] Y. Li, R. Liu, X. Hao, R. Shang, P. Zhao, and L. Jiao, EQNAS: Evolutionary quantum neural architecture search for image classification, *Neural Netw.* **168**, 471 (2023).
- [29] P. Wang, M. Usman, U. Parampalli, L. C. Hollenberg, and C. R. Myers, Automated quantum circuit design with nested Monte Carlo tree search, *IEEE Trans. Quantum Eng.* **4**, 3100620 (2023).
- [30] H. Wang, Y. Ding, J. Gu, Y. Lin, D. Z. Pan, F. T. Chong, and S. Han, QuantumNAS: Noise-adaptive search for robust quantum circuits, in *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)* (IEEE, 2022), pp. 692–708.
- [31] S.-X. Zhang, C.-Y. Hsieh, S. Zhang, and H. Yao, Differentiable quantum architecture search, *Quantum Sci. Technol.* **7**, 045023 (2022).
- [32] W. Wu, G. Yan, X. Lu, K. Pan, and J. Yan, QuantumDARTS: Differentiable quantum architecture search for variational quantum algorithms, in *International Conference on Machine Learning* (Microtome Publishing, Brookline, MA, 2023), pp. 37745–37764.
- [33] Z. He, C. Chen, L. Li, S. Zheng, and H. Situ, Quantum architecture search with meta-learning, *Adv. Quantum Technol.* **5**, 2100134 (2022).
- [34] A. Yimsiriwattana and S. J. Lomonaco, Jr., Generalized GHZ states and distributed quantum computing, [arXiv:quant-ph/0402148](https://arxiv.org/abs/quant-ph/0402148).
- [35] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels, *Phys. Rev. Lett.* **70**, 1895 (1993).

- [36] S. Sim, P. D. Johnson, and A. Aspuru-Guzik, Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms, *Adv. Quantum Technol.* **2**, 1900070 (2019).
- [37] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, *Nature* **549**, 242 (2017).
- [38] S.-X. Zhang, J. Allcock, Z.-Q. Wan, S. Liu, J. Sun, H. Yu, X.-H. Yang, J. Qiu, Z. Ye, Y.-Q. Chen, C.-K. Lee, Y.-C. Zheng, S.-K. Jian, H. Yao, C.-Y. Hsieh, and S. Zhang, TensorCircuit: a quantum software framework for the NISQ era, *Quantum* **7**, 912 (2023).
- [39] M. Larocca, S. Thanasilp, S. Wang, K. Sharma, J. Biamonte, P. J. Coles, L. Cincio, J. R. McClean, Z. Holmes, and M. Cerezo, A review of barren plateaus in variational quantum computing, [arXiv:2405.00781](https://arxiv.org/abs/2405.00781).
- [40] Z. Holmes, K. Sharma, M. Cerezo, and P. J. Coles, Connecting ansatz expressibility to gradient magnitudes and barren plateaus, *PRX Quantum* **3**, 010313 (2022).