SCIENTIFIC REPORTS

natureresearch

Check for updates

OPEN

Circuit Depth Reduction for Gate-Model Quantum Computers

Laszlo Gyongyosi[1,2,3] ✉ & Sandor Imre[2]

Quantum computers utilize the fundamentals of quantum mechanics to solve computational problems more efficiently than traditional computers. Gate-model quantum computers are fundamental to implement near-term quantum computer architectures and quantum devices. Here, a quantum algorithm is defined for the circuit depth reduction of gate-model quantum computers. The proposed solution evaluates the reduced time complexity equivalent of a reference quantum circuit. We prove the complexity of the quantum algorithm and the achievable reduction in circuit depth. The method provides a tractable solution to reduce the time complexity and physical layer costs of quantum computers.

Gate-model quantum computers are realized by unitary operators (quantum gates) and quantum states[1–29]. As the technological limits of current semiconductor technologies will be reached within the next few years[30–40], fundamentally different solutions provided by quantum technologies will be significant in the experimental realization of future computations[15–18,31,32,41–72]. A quantum computer is set up with a quantum gate structure, that is, via a set of unitary operators. These quantum gates can realize different quantum operations and can be defined on different numbers of input quantum states[15–18,41–43,45,52,53]. In a quantum computer environment, the depth of the quantum gate structure refers to the number of time steps (time complexity) required for the quantum operations making up the circuit to run on the quantum hardware[15–18,41–43,45,52–59]. A crucial problem here is the time complexity reduction of the quantum gate structure of the quantum computer. Practically, this problem is such that an equivalent quantum state of the output quantum state of the original the reference quantum circuit (e.g., non-reduced time complexity circuit) can be obtained using a reduced time complexity quantum gate structure. Particularly, currently there exists no plausible and implementable solution for the time complexity reduction of quantum computers. Gate-model quantum computer implementations are affected by the problem of high time complexities and a universal (i.e., platform independent) and tractable solution for the time complexity reduction is essential. Relevant implication of this problem is the high economic cost of the physical apparatuses required for experimentally implementing practical quantum computation: specifically, the high economic cost of the high-precision quantum hardware elements required in the implementation of high-performance quantum circuits.

The quantum circuit of the quantum computer is modeled as an arbitrary quantum circuit with an arbitrary circuit depth formulated via a unitary sequence of L unitary operators. Each unitary is set via a particular Pauli operator and gate parameter (see also Section 2 for a detailed description). The input problem fed into the quantum computer is an arbitrary computational problem $\mathscr{P}$ with an objective function C. The C objective function is a subject of maximization via the quantum computer, i.e., via the unitaries of the circuit structure of the quantum computer. The objective function can model arbitrary combinatorial optimization problems[9,10,42], large-scale programming problems[10] such as the graph coloring problem, molecular conformation problem, job-shop scheduling problem, manufacturing cell formation problem, or the vehicle routing problem[10]. For a detailed description of input problems, we suggest[2–4,8–10,42–45].

Another important application of gate-model quantum computations is the near-term quantum devices of the quantum Internet[20,30,36–39,46–49,59,61,62,73–93].

Here, we define a quantum algorithm for the time complexity reduction of any quantum circuit of quantum computers set up with an arbitrary number of unitary gates. The aim of the proposed framework is to reduce the time complexity of an arbitrary reference quantum circuit and a maximization of the objective function of the computational problem fed into the quantum computer. The method defines the reduced time complexity equivalent of the reference quantum circuit and recovers the reference output quantum state via the reduced time complexity quantum circuit (Note: the terminology of quantum state refers to an input or output quantum system,

[1]School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK. [2]Department of Networked Systems and Services, Budapest University of Technology and Economics, Budapest, H-1117, Hungary. [3]MTA-BME Information Systems Research Group, Hungarian Academy of Sciences, Budapest, H-1051, Hungary. ✉e-mail: gyongyosi@hit.bme.hu

while the terminology of quantum gate refers to a unitary operator.). The reduced structures are determined via a pre-processing phase in the logical layer, and only the reduced time complexity quantum circuit and reduced quantum state are implemented in the physical layer. The pre-rocessing phase integrates a machine learning[94–97] unit for the parameter optimization. The high complexity reference quantum circuit and reference quantum input are characterized only in the pre-processing phase without any physical level implementation. The framework applies a quantum algorithm on the output of the reduced quantum gate structure to recover the equivalent quantum state of the output quantum state of the non-reduced reference structure. In particular, the proposed framework and the defined quantum algorithm are universal since they have no requirements for the structure of the reference (e.g., non-reduced) quantum circuit subject to be reduced, for the number of unitaries in the reference structure, for the size of the input quantum state of the reference quantum circuit, nor for the dimensions of the actual quantum state. The quantum algorithm is defined as a fixed, auxiliary hardware component for an arbitrary quantum computer environment, with a pre-determined constant computational complexity as an auxiliary cost of the application of the algorithm. Specifically, we prove that the auxiliary cost of the proposed quantum algorithm is orders lower than the reachable amount of the reduction in time complexity, and the computational cost of the quantum algorithm becomes negligible in practice. The method also allows significantly reducing the economic cost of physical layer implementations, since the required elements and high-cost hardware components can be reduced in an experimental setting.

The novel contributions of our manuscript are as follows:

1. We define a quantum algorithm for circuit depth reduction of quantum circuits of gate-model quantum computers.
2. We define the computational cost of the proposed quantum algorithm and prove that it is significantly lower than the gainable reduction in time complexity.
3. The algorithm provides a tractable solution to reduce circuit depth and the economic cost of implementing the physical layer quantum computer by reducing quantum hardware elements.
4. The results are useful for experimental gate-model quantum computations and near-term quantum devices of the quantum Internet.

This paper is organized as follows. Section 2 defines the system model. Section 3 proposes the quantum algorithm and proves the computational complexity. Section 4 discusses the performance of the algorithm. Finally, Section 5 concludes the results. Supplemental material is included in the Appendix.

System Model

Let QG_0 be a reference quantum circuit (quantum gate structure) with a sequence of L unitaries[42], defined as

$$U(\vec{\theta}) = U_L(\theta_L)U_{L-1}(\theta_{L-1})\ldots U_1(\theta_1), \tag{1}$$

where $\vec{\theta}$ is the L-dimensional vector of the gate parameters of the unitaries (gate parameter vector),

$$\vec{\theta} = (\theta_1, \ldots, \theta_L)^T, \tag{2}$$

and an i-th unitary gate $U_i(\theta_i)$ is evaluated as

$$U_i(\theta_i) = \exp(-i\theta_i P_i), \tag{3}$$

where P_i is a generalized Pauli operator acting on a few quantum states (qubits in an experimental setting) formulated by the tensor product of Pauli operators $\{\sigma_x, \sigma_y, \sigma_z\}$[42]. Note, that $U(\vec{\theta})$ in (1) identifies a unitary resulted from the serial application of the L unitary operators $U_L(\theta_L)U_{L-1}(\theta_{L-1}) \ldots U_1(\theta_1)$, and for an input quantum state $|\varphi\rangle$,

$$U(\vec{\theta})|\varphi\rangle = U_L(\theta_L)U_{L-1}(\theta_{L-1})\ldots U_1(\theta_1)|\varphi\rangle. \tag{4}$$

A qubit system example with a sequence of L unitaries is as follows. Let assume that the QG_0 structure of the quantum computer consists of g single-qubit and m two-qubit unitaries, $L = g + m$, such that a j-th single-qubit gate realizes an $X_j = \sigma_x^j$ operator, while a two-qubit gate realizes a $Z_jZ_k = \sigma_z^j\sigma_z^k$ operator (see also[42]). Then, at a particular objective function C of an arbitrary computational problem subject of a maximization via the quantum computer, the $U(\vec{\theta})$ sequence from (1) can be rewritten as

$$U(\vec{\theta}) = U(B, \vec{\beta})U(C, \vec{\gamma}), \tag{5}$$

where

$$U(B, \vec{\beta}) = \prod_j U(B_j, \beta_j), \tag{6}$$

where $\vec{\beta}$ is the gate parameter vector of the g single-qubit unitaries,

$$\vec{\beta} = (\beta_1, \ldots, \beta_g)^T, \tag{7}$$

while B is defined as

$$B = \sum_j X_j \tag{8}$$

with

$$X_j = \sigma_x^j, \tag{9}$$

and

$$U(B_j, \beta_j) = \exp(-i\beta_j X_j), \tag{10}$$

where $B_j = X_j$, while the two-qubit unitaries are defined as

$$U(C, \vec{\gamma}) = \prod_{\langle jk \rangle \in QG_0} U(C_{jk}, \gamma_{jk}), \tag{11}$$

where $\langle jk \rangle \in QG_0$ is a physical connection between qubits j and k in the hardware-level of the QG_0 structure of the quantum computer, $\vec{\gamma}$ is the gate parameter vector of the m two-qubit unitaries

$$\vec{\gamma} = (\gamma_1, \ldots, \gamma_m)^T, \tag{12}$$

while

$$C = \sum_{\langle jk \rangle \in QG_0} C_{jk}, \tag{13}$$

where C_{jk} is a component of the objective function, while unitary $U(C_{jk}, \gamma_{jk})$ for a given $\langle jk \rangle$ is defined as

$$U(C_{jk}, \gamma_{jk}) = U(Z_j Z_k, \gamma_{jk} C_{jk}) = \exp(-i\gamma_{jk} C_{jk} Z_j Z_k), \tag{14}$$

where

$$Z_j Z_k = \sigma_z^j \sigma_z^k. \tag{15}$$

At a particular physical connectivity of QG_0, the objective function C therefore can be written as

$$C(z) = \sum_{\langle jk \rangle \in QG_0} C_{jk}(z), \tag{16}$$

where $C_{jk}(z)$ is the objective function component evaluated for a given $\langle jk \rangle$, as

$$C_{\langle jk \rangle}(z) = \frac{1}{2}(1 - z_i z_j), \tag{17}$$

while z is an N-length input bitstring,

$$z = z_1 z_2 \ldots z_N, \tag{18}$$

where z_i identifies an i-th bit, $z_i \in \{-1, 1\}$.

For a given z, a $|z\rangle$ computational basis state is defined as

$$|z\rangle = |z_1 z_2 \ldots z_N\rangle, \tag{19}$$

and the $|\phi\rangle$ output system of QG_0 is as

$$|\phi\rangle = U(\vec{\theta})|z\rangle, \tag{20}$$

that can be evaluated further via (6) and (11), as

$$\begin{aligned} |\phi\rangle &= U(B, \vec{\beta})U(C, \vec{\gamma})|z\rangle \\ &= \prod_j U(B_j, \beta_j) \prod_{\langle jk \rangle \in QG_0} U(C_{jk}(z), \gamma_{jk})|z\rangle \\ &= \prod_j \exp(-i\beta_j X_j) \prod_{\langle jk \rangle \in QG_0} \exp(-i\gamma_{jk} C_{jk}(z) Z_j Z_k)|z\rangle. \end{aligned} \tag{21}$$

To achieve the quantum parallelism, the input system $|\varphi\rangle = |X\rangle$ of the quantum computer is set as an N-length d-dimensional ($d = 2$ for a qubit system) quantum state in the superposition of all possible d^N states. For a qubit system, it means that input $|X\rangle$ is an N-qubit quantum state in a superposition of all possible 2^N states $|0\rangle$ to $|2^N - 1\rangle$, and the computations are performed on 2^N states in parallel in the quantum computer.

Let $|X\rangle$ be a superposed input system of the non-reduced QG_0 gate structure:

$$|X\rangle = \frac{1}{\sqrt{d^N}} \sum_{x_i=1}^{d^N} |x_i\rangle, \tag{22}$$

where $|x_i\rangle$ is an i-th input state (represented as an N-length bit string for a qubit system), $i = 1, \ldots, n$, $n = d^N$, of the QG_0 structure of the quantum computer.

To describe the parallel processing of the n input vectors of $|X\rangle$ (see (22)), $\{|x_1\rangle, \ldots, |x_n\rangle\}$ of $|X\rangle$ (see (22)) in the quantum computer, let $\vec{\theta}_i$ be the gate parameter vector associated with a given $|x_i\rangle$ of $|X\rangle$:

$$\vec{\theta}_i = (\theta_{i,1}, \ldots, \theta_{i,L})^T. \tag{23}$$

Let X be the classical representation of $|X\rangle$ in (22) to get

$$X = \frac{1}{\sqrt{d^N}} (x_1, \ldots, x_n)^T, \tag{24}$$

where x_i is the classical representation of $|x_i\rangle$. (Note, that X and x_i are not accessible in the quantum computer, since the quantum algorithm operates in the quantum regime on quantum states. The classical representation is used only as an abstracted auxiliary representation to describe the steps of the algorithm in a plausible manner).

Then, let $U_0(\vec{\theta})$ be the non-reduced gate structure matrix of QG_0:

$$U_0(\vec{\theta}) = (U_0(\vec{\theta}_n) U_0(\vec{\theta}_{n-1}) \ldots U_0(\vec{\theta}_1)), \tag{25}$$

where

$$\vec{\theta} = (\vec{\theta}_1, \ldots, \vec{\theta}_n) \tag{26}$$

and $U_0(\vec{\theta}_i)$ is the unitary sequence associated with $|x_i\rangle$ in QG_0, defined as

$$U_0(\vec{\theta}_i) = U_L(\theta_{i,L}) U_{L-1}(\theta_{i,L-1}) \ldots U_1(\theta_{i,1}). \tag{27}$$

At an n-dimensional output vector

$$Y = \frac{1}{\sqrt{d^N}} (y_1, \ldots, y_n)^T, \tag{28}$$

and the $|Y\rangle$ output quantum state of the non-reduced QG_0 structure is

$$|Y\rangle = U_0(\vec{\theta})|X\rangle = \frac{1}{\sqrt{d^N}} \sum_{y_i} |y_i\rangle = \frac{1}{\sqrt{d^N}} \sum_{x_i=1}^{d^N} U_0(\vec{\theta}_i)|x_i\rangle. \tag{29}$$

To define the reduced gate structure, QG^*, it is necessary to find a reduced $U(\vec{\theta}_i^{'})$ with a reduced input $|\tilde{x}_i\rangle$, for all i.

Then, let $\widetilde{X}$ be the classical representation of the reduced quantum state $|\widetilde{X}\rangle$ fed into QG^*, as

$$\widetilde{X} = \frac{1}{\sqrt{n^*}} (\tilde{x}_1, \ldots, \tilde{x}_{n^*})^T \tag{30}$$

and

$$|\widetilde{X}\rangle = \frac{1}{\sqrt{n^*}} \sum_{\tilde{x}_i=1}^{d^{N^*}} |\tilde{x}_i\rangle, \tag{31}$$

where N^* is the number of d-dimensional (physical) quantum states that formulate $|\widetilde{X}\rangle$, $n^* = d^{N*}$, while the unitaries $U(\vec{\theta}\,')$ of QG^* are

$$U(\vec{\theta}\,') = (U(\vec{\theta}_{n^*}^{'}) U(\vec{\theta}_{n^*-1}^{'}) \ldots U(\vec{\theta}_1^{'})). \tag{32}$$

where

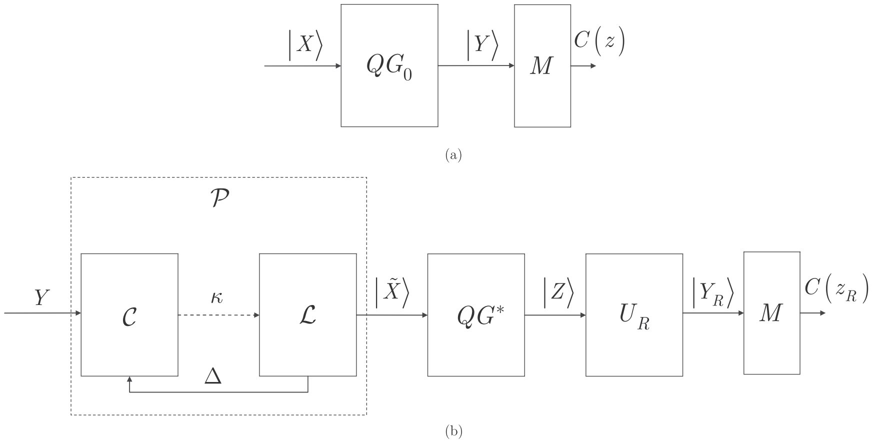
$$\vec{\theta}\,' = (\vec{\theta}_1^{'}, \ldots, \vec{\theta}_{n^*}^{'}) \tag{33}$$

and $U(\vec{\theta}_i^{'})$ is the reduced unitary sequence associated with $|\tilde{x}_i\rangle$, defined as

$$U(\vec{\theta}_i^{'}) = U_L(\tilde{\theta}_{i,L}) U_{L-1}(\tilde{\theta}_{i,L-1}) \ldots U_1(\tilde{\theta}_{i,1}). \tag{34}$$

The pre-processing phase determines output Z of QG^* as a classical representation

$$Z = U(\vec{\theta}\,')\widetilde{X}, \tag{35}$$



(a)

(b)

Figure 1. (**a**) The non-reduced time complexity quantum circuit QG_0 (reference circuit) with an input quantum state $|X\rangle$. The output of QG_0 is $|Y\rangle$. The state $|Y\rangle$ is measured via a measurement M to get the classical string z to evaluate the objective function $C(z)$. (**b**) System model of the time complexity reduction scheme. Pre-processing phase $\mathcal{P}$: the Y classical representation of output $|Y\rangle$ of QG_0 is pre-processed by the pre-processing unit $\mathscr{P}$. Unit $\mathscr{P}$ contains a $\mathscr{C}$ computational block that outputs a vector κ, fed into an $\mathscr{L}$ machine learning control unit for the Δ error feedback. Unit $\mathscr{P}$ outputs $\widetilde{X}$ and the gate parameters of the reduced structure that defines QG^*. Quantum phase: from $\widetilde{X}$ and the gate parameters, $|\widetilde{X}\rangle$ and QG^* are set up. System $|\widetilde{X}\rangle$ is fed into the reduced quantum circuit QG^*. The output of QG^* is $|Z\rangle$, which is fed into the U_R recovery quantum algorithm. The U_R quantum operation outputs the $|Y_R\rangle$ system, which is the reference output $|Y\rangle$ of the reference circuit QG_0. The state $|Y_R\rangle$ is measured via a measurement M to get the classical string z_R to evaluate objective function $C(z_R)$.

and the output quantum state $|Z\rangle$ of QG^* therefore yields

$$|Z\rangle = U(\vec{\theta}\,')|\widetilde{X}\rangle. \tag{36}$$

The notations of the system model are also summarized in Table A.1 of the Supplemental Information.

Problem statement.

Problems 1–3 summarize the problems to be solved.

Problem 1 *Find a classical pre-processing $\mathscr{P}$ for calculating the $|\widetilde{X}\rangle$ reduced input system and the gate parameters of the QG* reduced time complexity gate structure.*

Problem 2 *Find a universal (independent of the number L of the unitaries in QG_0) unitary operator U_R with a set $\mathscr{R}$ of quantum registers to recover output $|Y\rangle$ of the non-reduced QG_0 structure from output $|Z\rangle$ of the QG* reduced time complexity gate structure.*

Problem 3 *Determine the time complexity of U_R and the reduction in the overall time complexity of QG*.*

Theorems 1–3 give the resolutions of Problems 1–3.

The non-reduced time complexity quantum circuit QG_0 (reference circuit) with an input quantum state $|X\rangle$ is showed in Fig. 1(a). Figure 1(b) depicts the system model for the problem resolution. The method is realized via unitary U_R and $\mathscr{P}$ pre-processing, such that U_R is implemented in the physical layer, while $\mathscr{P}$ is only a logical-layer process. Only the reduced input quantum state $|\widetilde{X}\rangle$ and the reduced quantum gate structure QG^* must be built up in the physical layer to yield the reference output system $|Y\rangle$ of the reference circuit QG_0 via $|Y_R\rangle$. In both cases, the output states are measured via a measurement M to get a classical bitring for the objective function evaluation. As a next step, the gate parameter values of the unitaries of the circuits are calibrated until an optimal objective function value is not reached. The calibration of the gate parameters is a separate optimization procedure and its aim is fundamentally differ from the aim of $\mathscr{P}$, and therefore it is not part of the circuit depth reduction method. Note that existing algorithms can be utilized for this task (such as a the algorithms proposed in[19] and[20], or some gradient independent methods[98]).

Pre-processing. **Theorem 1** *There exists a $\mathscr{P}$ pre-processing to determine the $|\widetilde{X}\rangle$ input system and the $U(\vec{\theta}_i')$ gate parameters, $i =, \ldots, n$, for the reduced QG* gate structure for an arbitrary non-reduced QG_0 structure with $U(\vec{\theta}_i)$ and input $|X\rangle$.*

Proof. The $\mathscr{P}$ pre-processing phase can be decomposed as $\mathscr{P} = \mathscr{C}\mathscr{L}$, where $\mathscr{C}$ is a computational block, while $\mathscr{L}$ is a machine learning control block to calibrate the results of $\mathscr{C}$. We first define block $\mathscr{C}$, then discuss $\mathscr{L}$. The $\mathscr{P}$ pre-processing is a procedure to stabilize the output of the reduced quantum circuit. $\mathscr{P}$ is defined between the components $\mathscr{C}$ and $\mathscr{L}$ to evaluate $|\widetilde{X}\rangle$ and to set the gate parameters of the reduced quantum circuit structure QG^*

using the reference output $|Y\rangle$ of QG_0. Note, as the output $|Y_R\rangle$ is fed into an M measurement block, the measurement results provide a feedback to calibrate $\mathscr{P}$ in every subroutine of the protocol to produce a final saturated output. The Δ output of the $\mathscr{L}$ machine learning control unit is used as a feedback in unit $\mathscr{C}$. For the definition of Δ, see (116) in Algorithm 1.

In the $\mathscr{C}$ computational block, the reduced $U(\vec{\theta}_i^{\,\prime})$ and $\tilde{x}_i$ are determined for $\forall i$, in the following manner. Note, since $\mathscr{P}$ outputs the parameters of the reduced quantum gate structure, the extra complexity of a quantum structure can be replaced with classical complexity in the form of machine learning in the proposed framework.

Operation $\mathscr{C}$ sets a one-dimensional discrete cosine transform[99] in the reduction method, thus a matrix G is defined as a generator matrix to evaluate the output coefficients of $\mathscr{C}$, see later (45). The definition of $\mathscr{C}$ (see later in (40)) comes from the fact that any U unitary operator can be rewritten via the cos and sin functions, and using cosine functions rather than sine functions is critical for a compression[99]. In our setting, this is because fewer cosine functions are needed to approximate a particular U unitary operator.

Let x_i be the classical representation of $|x_i\rangle$, and $y_i = U(\vec{\theta}_i)x_i$ be the classical representation of $|y_i\rangle$. Using the sequences of the L unitaries in (29), define a matrix G with n coefficients a_i, $i = 1, \ldots, n$, as

$$\begin{aligned} G &= (a_1 \ \cdots \ a_n)^T \\ &= \begin{pmatrix} (\Sigma\,\theta_1) + x_1 \\ (\Sigma\,\theta_2) + x_2 \\ \vdots \\ (\Sigma\,\theta_n) + x_n \end{pmatrix} = \begin{pmatrix} (\Sigma_{j=1}^{L}\theta_{1,j}) + x_1 \\ (\Sigma_{j=1}^{L}\theta_{2,j}) + x_2 \\ \vdots \\ (\Sigma_{j=1}^{L}\theta_{n,j}) + x_n \end{pmatrix}, \end{aligned} \tag{37}$$

where

$$\Sigma\,\theta_i = \sum_{j=1}^{L}\theta_{i,j}, \tag{38}$$

where $\theta_{i,j}$ identifies the gate parameter of a j-th unitary $U_{i,j}(\theta)$ associated to an i-th input x_i, while unitary sequence $U_0(\vec{\theta}_i)$ to an i-th input x_i is

$$U_0(\vec{\theta}_i) = \exp(i\Sigma\,\theta_i P_i), \tag{39}$$

where P_i is a generalized Pauli operator.

First, the $\mathscr{C}$ operation (one-dimensional discrete cosine transform[99]) is applied to the input matrix G from (37),

$$\mathscr{C}: \left\{c_p, f_i\right\}, \ 0 \le p, i \le n-1, \tag{40}$$

where c_p is the p-th coefficient of $\mathscr{C}$,

$$c_p = \sum_{i=0}^{n-1} A_p f_i \cos\frac{\pi(2i+1)p}{2n}, \tag{41}$$

where

$$f_i = \sum_{i=0}^{n-1} A_p c_p \cos\frac{\pi(2i+1)p}{2n}, \tag{42}$$

and A_p is

$$A_p = \begin{cases} \dfrac{1}{\sqrt{n}}, & \text{if } p = 0 \\ \sqrt{\dfrac{2}{n}}, & \text{if } 1 < p < n \end{cases}. \tag{43}$$

The coefficients of $\mathscr{C}$ defines matrix γ as

$$\begin{aligned} \gamma &= \chi G \\ &= (\varsigma_1 \cdot G, \varsigma_2 \cdot G, \ldots, \varsigma_n \cdot G)^T \\ &= (c_1, c_2, \ldots, c_n)^T, \end{aligned} \tag{44}$$

where $\cdot$ is the inner product,

$$c_i = \varsigma_i \cdot G = \sum_{k=1}^{n} \varsigma_{i,k} \times a_k, \tag{45}$$

where coefficients a_k-s are given in (37), and χ is

$$\chi = \begin{pmatrix} (\varsigma_1)^T \\ \vdots \\ (\varsigma_{n-1})^T \\ (\varsigma_n)^T \end{pmatrix}, \tag{46}$$

where ς_i is an n-length vector

$$\varsigma_i = (\varsigma_{i,1}, \ldots, \varsigma_{i,n}) = A_i \left(\cos\frac{i\pi}{2n}, \ \cos\frac{3i\pi}{2n}, \ \ldots, \ \cos\frac{(2n-1)i\pi}{2n} \right)^T. \tag{47}$$

Then, the n-length output vector κ of $\mathscr{C}$ is defined as

$$\mathscr{C}(Y) = \kappa = \frac{1}{\sqrt{n}}(\Omega_1, \ldots, \Omega_n)^T, \tag{48}$$

where Y is given in (28), while Ω_i is as

$$\begin{aligned} \Omega_i &= U(\vec{\theta}_i^{\,\prime})\tilde{x}_i \\ &= U_L(\tilde{\theta}_{i,L})U_{L-1}(\tilde{\theta}_{i,L-1})\ldots U_1(\tilde{\theta}_{i,1})\tilde{x}_i. \end{aligned} \tag{49}$$

Then, using the coefficients (41), (42) and (43) of $\mathscr{C}$, $|\tilde{x}_p\rangle$ of the reduced state $|\widetilde{X}\rangle$ from (31) can be evaluated via the $\tilde{x}_p$ components of $\widetilde{X}$ of (30). A p-th input $|\tilde{x}_p\rangle$ for QG^* is defined via (49) as

$$\left|\tilde{x}_p\right\rangle = \left|\Omega_p U^{\dagger}(\vec{\theta}_p^{\,\prime})\right\rangle = \left|\sum_{i=0}^{n-1} A_p x_i \cos\frac{\pi(2i+1)p}{2n}\right\rangle, \tag{50}$$

and the reduced quantum gate sequence $U(\vec{\theta}_p^{\,\prime})$ of $|\tilde{x}_p\rangle$ in QG^*, as

$$U(\vec{\theta}_p^{\,\prime}) = \exp(i\Sigma\tilde{\theta}_p P_p), \tag{51}$$

where P_p is a generalized Pauli operator, and $\Sigma\tilde{\theta}_p$ is as

$$\Sigma\tilde{\theta}_p = \sum_{i=0}^{n-1} A_p(\Sigma\theta_i)\cos\frac{\pi(2i+1)p}{2n}. \tag{52}$$

Therefore, the quantum state $|Z\rangle$ of QG^* is

$$|Z\rangle = U(\vec{\theta}\,')|\widetilde{X}\rangle = \frac{1}{\sqrt{d^{N^*}}} \sum_{\tilde{x}_i=1}^{d^{N^*}} U(\vec{\theta}_i^{\,\prime})|\tilde{x}_i\rangle. \tag{53}$$

The description of the $\mathscr{L}$ machine learning control unit is as follows. Unit $\mathscr{L}$ uses the results of $\mathscr{C}$ to provide feedback for the pre-processing via supervised machine learning control.

The $\mathscr{L}$ machine learning algorithm for the pre-processing control is defined in Algorithm 1.

The steps of the $\mathscr{P}$ pre-processing method is given in Procedure 1.

■

Quantum Algorithm

Theorem 2 *The $|Y\rangle$ output of the non-reduced QG_0 structure can be recovered from the output $|Z\rangle$ of the reduced structure QG^* via a unitary operator U_R.*

Proof. Let $|\widetilde{X}\rangle$ be the input quantum state fed into the reduced structure QG^{*}, and let $|Z\rangle$ (see (53)) be the output of the reduced gate structure. The task here is therefore to recover $|Y\rangle = U_0(\vec{\theta})|X\rangle$ from $|Z\rangle$. The problem is solved via a unitary U_R, as follows.

Without loss of generality, in an i-th step, $i = 1, \ldots, n$, the goal of the U_R operation is to calculate the quantum state as

$$|\Phi_i\rangle = |\omega_i \cdot \kappa\rangle, \tag{54}$$

where κ is as in (48), while $\omega_i = (\omega_{i,1}, \ldots, \omega_{i,n})^T$ is an n-length vector defined for a given j, as

$$\omega_j = \left(e^{i(\Sigma\theta_1 - \Sigma\widetilde{\theta}_1)} A_j \cos\frac{j\pi}{2n}, \ldots, e^{i(\Sigma\theta_n - \Sigma\widetilde{\theta}_n)} A_j \cos\frac{(2n-1)j\pi}{2n} \right)^T, \tag{55}$$

where $\sum\theta_i$ is as given in

$$\Sigma\theta_i = \sum_{i=0}^{n-1} A_p(\Sigma\widetilde{\theta}_p)\cos\frac{\pi(2i+1)p}{2n}, \tag{56}$$

where $\Sigma\widetilde{\theta}_p$ is given in (52).

Then let

$$W = \begin{pmatrix} (\omega_1)^T \\ \vdots \\ (\omega_{n-1})^T \\ (\omega_n)^T \end{pmatrix}, \tag{57}$$

such that

$$W\kappa = (\omega_1 \cdot \kappa, \ldots, \omega_n \cdot \kappa)^T. \tag{58}$$

Applying U_R for all i, yields the recovered quantum state $|Y_R\rangle$ as

$$\begin{aligned} |Y_R\rangle &= \sum_{i=1}^{n} |\Phi_i\rangle = \sum_{i=1}^{n} |\omega_i \cdot \kappa\rangle \\ &= \frac{1}{\sqrt{n}} \sum_{x_i=1}^{n} e^{i(\Sigma\theta_i - \Sigma\widetilde{\theta}_i)} U(\vec{\theta}_i{}') |x_i\rangle \\ &= \frac{1}{\sqrt{d^N}} \sum_{x_i=1}^{d^N} U_0(\vec{\theta}_i) |x_i\rangle \\ &= U_0(\vec{\theta}) |X\rangle, \end{aligned} \tag{59}$$

where an i-th $|x_i\rangle$ is as

$$|x_i\rangle = \left| \sum_{i=0}^{n-1} A_p \widetilde{x}_p \cos\frac{\pi(2i+1)p}{2n} \right\rangle, \tag{60}$$

where $i \leq n-1$, and $p \geq 0$, and $\widetilde{x}_p$ is as given in (50); while the $U_0(\vec{\theta}_i)$ gate parameters (see (39)) of the L unitaries for a given i are evaluated as

$$\begin{aligned} U_0(\vec{\theta}_i) &= e^{i(\Sigma\theta_i - \Sigma\widetilde{\theta}_i)} U(\vec{\theta}_i') \\ &= e^{i(\Sigma\theta_i - \Sigma\widetilde{\theta}_i)} e^{i\Sigma\widetilde{\theta}_i P_i} \\ &= e^{i\Sigma\theta_i P_i}. \end{aligned} \tag{61}$$

The unitary U_R is defined via a set $\mathscr{R}$ of quantum registers as

$$U_R : \mathscr{R} = \{|R_1\rangle, \ldots, |R_7\rangle\}, \tag{62}$$

where $|R_i\rangle$ is the i-th quantum register. The registers are initialized via set $\mathscr{R}_0$ as

$$\mathscr{R}_0 = \begin{Bmatrix} |R_1\rangle = |\partial\rangle, |R_2\rangle = |\eta\rangle, |R_3\rangle = |i\rangle, |R_4\rangle = |\kappa\rangle, \\ |R_5\rangle = |0\rangle, |R_6\rangle = |0\rangle, |R_7\rangle = |0\rangle \end{Bmatrix}, \tag{63}$$

where κ is given in (48), while ∂ and η are initial parameters defined as

$$\partial = \frac{\|\kappa\|^2}{n}, \tag{64}$$

and

$$\eta = \|\kappa\|^2, \tag{65}$$

where

$$\|\kappa\|^2 = \kappa \cdot \kappa = \sum_{i=0}^{n-1} \frac{1}{n}(\Omega_i \times \overline{\Omega}_i) \tag{66}$$

where $\Omega_i = U(\vec{\theta}_i^{\prime})\widetilde{x}_i$, and

$$\partial \leq \frac{1}{n}(y_i \times y_i) \leq \eta, \tag{67}$$

where $y_i = U(\vec{\theta}_i)x_i$.

Then, unitary U_R is defined as

$$U_R = (2|\psi\rangle\langle\psi| - I)U_S, \tag{68}$$

where

$$|\psi\rangle = \frac{1}{\sqrt{n}}\sum_{i=0}^{n-1}|i\rangle, \tag{69}$$

and U_S is a unitary defined as

$$U_S = (U_0)^{-1}(O_{\Phi_i})^{-1}O_{f_i}O_{\Phi_i}U_0, \tag{70}$$

with eigenstate

$$|\Psi_S\rangle = |\partial\rangle|\eta\rangle|i\rangle|\kappa\rangle|0\rangle|0\rangle|0\rangle; \tag{71}$$

where U_0 is an initial unitary operator that prepares state $|R_5\rangle = |\omega_i\rangle$ for a given index state $|R_4\rangle = |i\rangle$, where ω_i is given in (55); from an initial $|R_4\rangle|R_5\rangle = |i\rangle|0\rangle$ as

$$U_0(|i\rangle|0\rangle) = |0 \oplus \omega_i\rangle \tag{72}$$

in the register set $\mathscr{R}_0$ (see (63)), where $\oplus$ is the CNOT operation, while O_{Φ_i} is an oracle applied on $\mathscr{R}_0$ to compute Φ_i (54), defined as

$$\begin{aligned} & O_{\Phi_i}(\mathscr{R}_0) = \mathscr{R}' \\ & : \begin{Bmatrix} |R_1\rangle = |\partial\rangle, |R_2\rangle = |\eta\rangle, |R_3\rangle = |i\rangle, |R_4\rangle = |\kappa\rangle, \\ |R_5\rangle = |\omega_i\rangle, |R_6\rangle = |0\rangle, |R_7\rangle = |(\Phi_i)^2\rangle \end{Bmatrix}, \end{aligned} \tag{73}$$

where $\mathscr{R}'$ is the resulting register set, while O_{f_i} is an oracle that outputs function f_i, as

$$O_{f_i} : f_i = \begin{cases} 1, & \text{if } \partial \leq \left(\frac{1}{\sqrt{n}}y_i\right)^2 \leq \eta \\ 0, & \text{otherwise} \end{cases}. \tag{74}$$

Specifically, note that (70) changes only the phase of the state as $(-1)^{f_i}$, where f_i is given in (74), while

$$(U_0)^{-1}(O_{\Phi_i})^{-1} = (O_{\Phi_i}U_0)^{-1}. \tag{75}$$

Applying (74) on (63) yields a register state $O_{f_i}(\mathscr{R}_0)$ as

$$O_{f_i}(\mathscr{R}_0) = (-1)^{f_i}\mathscr{R}_0, \tag{76}$$

where $(-1)^{f_i}$ is the eigenvalue of U_S in (70).

Then, using the register set (63), let $|\phi_0\rangle$ be the input state for U_R as

$$|\phi_0\rangle = \frac{1}{\sqrt{n}}\sum_{i=0}^{n-1} |\partial\rangle|\eta\rangle|i\rangle|\kappa\rangle|0\rangle|0\rangle|0\rangle. \tag{77}$$

Applying (68) k-times on (77) yields

$$|\phi_k\rangle = (U_R)^k|\phi_0\rangle. \tag{78}$$

The k iteration number in (78) is a random number, $k < c$, where $c = \min\{1.2 \cdot m, \sqrt{n}\}$, and m is initialized as $m = 1$[99].

Then let O_Z be an oracle defined on $\mathscr{R}'$ as

$$O_Z(\mathscr{R}') = \mathscr{R}^* : \begin{Bmatrix} |R_1\rangle = |\partial\rangle, |R_2\rangle = |\eta\rangle, |R_3\rangle = |i\rangle, |R_4\rangle = |\kappa\rangle, \\ |R_5\rangle = |\omega_i\rangle, |R_6\rangle = |\Phi_i\rangle, |R_7\rangle = |0\rangle \end{Bmatrix}. \tag{79}$$

Applying $O_Z U_0$ on (78), outputs system state

$$|\phi^*\rangle = O_Z U_0(|\phi_k\rangle). \tag{80}$$

In particular, in system state (80), the state of register $|R_6\rangle$ is

$$|R_6\rangle = \sum_{i=1}^{n} |\Phi_i\rangle = |Y_R\rangle, \tag{81}$$

therefore yields (59), such that

$$\partial \le (\Phi_i)^2 \le \eta \tag{82}$$

holds for all i of $|Y_R\rangle$, due to the conditions set in the pre-processing procedure $\mathscr{P}$ (see (67)).

Assuming that the input system (77) for U_R is prepared for R-times and the output register (81) is measured for R-times, i.e., U_R is applied for R times in overall, in an r-th repetition, $r = 1, \ldots, R$, the parameters of the procedure can be valuated as

$$\partial^{(r)} = \frac{\eta^{(r)}}{n - q^{(r)}}, \tag{83}$$

where

$$\eta^{(r)} = \|\kappa^{(r)}\|^2 - (\Phi_i^{(r)} \times \Phi_i^{(r)}), \tag{84}$$

where $\Phi_i^{(r)}$ is the measured value of $|\Phi_i\rangle$ in the r-th repetition of U_R, while $q^{(r)}$ is the number of coefficients have been already found[99].

The actual value of r requires no increment if the relation

$$\tau^{(r)} > \frac{1}{\|\kappa\|^2}\eta^{(r)}, \tag{85}$$

holds, where $\tau^{(r)}$ is a threshold value in the r-th iteration. Otherwise, the value of r can be increased, $r = r + 1$, as $r < R$.

The steps of the quantum algorithm U_R are given in Algorithm 2.

Distortion measure. As (81) is prepared in Step 4 of U_R, the state $|Y_R\rangle$ can be measured to get the classical string z_R to evaluate objective function $C(z_R)$, as follows. Measure register $|R_6\rangle$ of $\mathscr{R}$ via a measurement operator M to evaluate objective function $C(z_R)$, where z_R is a classical string resulted from the measurement of $|Y_R\rangle$, while C is an objective function of an arbitrary computational problem fed into the quantum computer.

The $\mathscr{D}$ distortion coefficient associated with the $|Y_R\rangle$ recovered quantum state (59) can be evaluated at a particular objective function C, associated to the computational problem fed into the quantum computer as

$$\mathscr{D} = |C(z) - C(z_R)|, \tag{86}$$

where z is a classical string resulting from the M measurement of $|Y\rangle$, while z_R is a classical string resulting from the M measurement of $|Y_R\rangle$.

Precisely, assuming R measurement rounds, an average of distortion yields

$$\mathscr{D}^{(R)} = \frac{1}{R}\sum_{r=1}^{R}|C^{(r)}(z) - C^{(r)}(z_R)|, \tag{87}$$

where $C^{(r)}(z)$ and $C^{(r)}(z_R)$ are the objective function values respectively associated with z and z_R in the r-th round, $r = 1, \ldots, R$.

Computational Complexity. **Theorem 3** *Quantum algorithm U_R can be implemented with time complexity $\mathcal{O}(\sqrt{n})$ for the time complexity reduction of any non-reduced QG_0 with an arbitrary number of L unitaries.*

Proof. Let

$$\mathscr{G}\{|i\rangle|b\rangle|c\rangle\} \tag{88}$$

be a global space spanned by $|i\rangle$, an n-dimensional vector $|b\rangle$, and by $|c\rangle$, which represents the inner product state.

Particularly, the U_R unitary in (68) applied on an input $|\varphi\rangle$ formulated via set $\mathscr{R}$ of quantum registers gives

$$\begin{aligned}|\varphi\rangle &= \frac{1}{\sqrt{n}}\sum_{i=0}^{n-1}|\Psi_i\rangle\\ &= \frac{1}{\sqrt{n}}\sum_{i=0}^{n-1}|\partial\rangle|\eta\rangle|i\rangle|\kappa\rangle|0\rangle|0\rangle|0\rangle,\end{aligned} \tag{89}$$

where

$$|\Psi_i\rangle = |\partial\rangle|\eta\rangle|i\rangle|\kappa\rangle|0\rangle|0\rangle|0\rangle; \tag{90}$$

thus U_R can be interpreted as a rotation on an n-dimensional subspace $\mathscr{S}\{|i\rangle\}$, $0 \le i < n$, i.e., on a span of all $|i\rangle$.

Let $\prod$ be the solution set with conditions (82) for all i of $\prod$,

$$\Pi = |\partial\rangle|\eta\rangle|i\rangle|\kappa\rangle|0\rangle|0\rangle|0\rangle, \tag{91}$$

and let $|\Upsilon\rangle \in \mathscr{S}\{|i\rangle\}$ be the superposition of all solutions:

$$\begin{aligned}|\Upsilon\rangle &= \frac{1}{\sqrt{|\Pi|}}\sum_{\xi\in\Pi}|\xi\rangle\\ &= \frac{1}{\sqrt{|\Pi|}}\sum_{\xi\in\Pi}|\partial\rangle|\eta\rangle\sum_{\Pi}|i\rangle|\kappa\rangle|0\rangle|0\rangle|0\rangle.\end{aligned} \tag{92}$$

The operation U_R on $|\varphi\rangle$ yields the state $|\phi^*\rangle \in \mathscr{S}\{|i\rangle\}$ (see (80)):

$$\begin{aligned}|\phi^*\rangle &= U_R(|\varphi\rangle)\\ &= |\partial\rangle|\eta\rangle\left[(2|\psi\rangle\langle\psi| - I)\sum_{i=0}^{n-1}(-1)^{f_i}\frac{1}{\sqrt{n}}|i\rangle\right]|\kappa\rangle|0\rangle|0\rangle|0\rangle\\ &= \frac{1}{\sqrt{n-|\Pi|}}\sum_{|\alpha\rangle\notin\Pi}|\alpha\rangle\\ &= \frac{1}{\sqrt{n-|\Pi|}}|\partial\rangle|\eta\rangle(\sum_{I-\Pi}|j\rangle)|\kappa\rangle|0\rangle|0\rangle|0\rangle,\end{aligned} \tag{93}$$

thus, U_R is a rotation on the subspace $\mathscr{S}\{|\phi^*\rangle, |\Upsilon\rangle\}$ by angle θ_{U_R} towards (92), as

$$\theta_{U_R} = 2\arcsin\sqrt{\frac{|\Pi|}{n}}, \tag{94}$$

where $|\prod|$ is the number of solutions (cardinality of solution set $\prod$).

U_R can be implemented as a rotation of θ_{U_R} on subspace $\mathscr{S}\{|i\rangle\}$ (instead of a rotation on global space (88)) via a generalized quantum searching[100] that yields time complexity $\mathcal{O}(\sqrt{n})$ for an arbitrarily large quantum circuit QG_0. ■

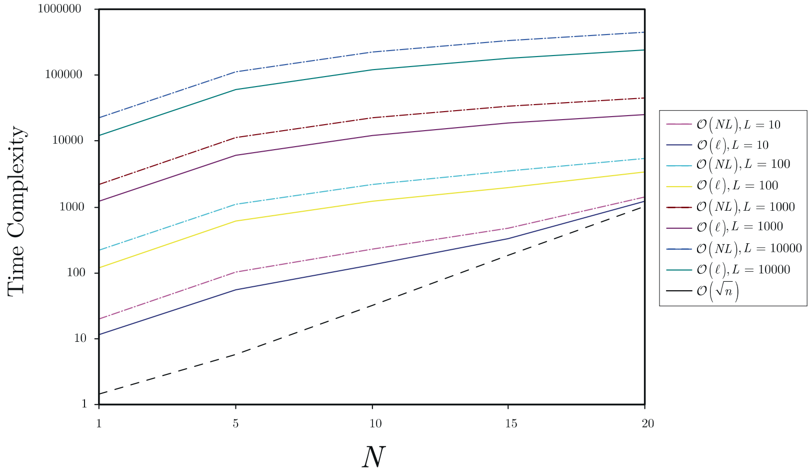


Figure 2. The time complexities (number of operations) for an N-qubit system, $d = 2$, $n = 2^N$, with an initial non-reduced gate structure QG_0 with L unitaries, $L = \{10, 100, 1000, 10000\}$. The time complexity of QG_0 is $\mathcal{O}(NL)$, while $\mathcal{O}(\ell)$ is an upper a bound on $\mathcal{O}(N^*L^*)$ of QG^*, $\mathcal{O}(\ell) = \mathcal{O}(NL - \sqrt{n})$.

Performance Evaluation

Assuming that the initial time complexity of the QG_0 non-reduced gate structure is

$$\mathcal{O}(NL), \tag{95}$$

where N is the number of d-dimensional (physical) quantum states in the superposed input system, and L is the number of unitaries in QG_0, the time complexity of the reduced QG^* structure is

$$\mathcal{O}(N^*L^*), \tag{96}$$

where N^* is the number of d-dimensional (physical) quantum states in the reduced superposed input system, and L^* is the number of unitaries in the reduced gate structure QG^*.

Since the complexity of the proposed scheme is

$$\mathcal{O}(\sqrt{n}), \tag{97}$$

the result of (96) is a reduced time complexity with respect to (95), as the relation

$$N^*L^* < NL - \sqrt{n}, \tag{98}$$

holds; thus

$$N^*L^* < NL - d^{N/2}. \tag{99}$$

The overall complexity of the QG^* reduced structure at the application of U_R is therefore

$$\mathcal{O}(N^*L^* + \sqrt{n}) = \mathcal{O}(N^*L^* + d^{N/2}). \tag{100}$$

Figure 2 depicts the resulting time complexities for a qubit implementation (N-qubit superposed input system, and qubit gate structure with L unitaries).

To achieve time complexity reduction using $|\widetilde{X}\rangle$ and QG^* instead of $|X\rangle$ and QG_0, the relation $\mathcal{O}(N^*L^*) < \mathcal{O}(\ell) = \mathcal{O}(NL - \sqrt{n})$ must be straightforwardly satisfied, i.e., the initial complexity $\mathcal{O}(NL)$ has to be reduced by more than $\mathcal{O}(\sqrt{n})$. Since the complexity of the procedure is independent from the actual size of the gate structure, the cost remains constant $\mathcal{O}(\sqrt{n})$ for an arbitrarily large L.

Conclusions

Gate-model quantum computers are equipped with a collection of quantum states and unitary quantum gates. The realization of the quantum circuit of a quantum computer requires high fidelity, high precision, and high-level control. Since both the timecomplexity (depth of the circuits) and the economic costs of these implementations are high in practical scenarios, a reduction of these costs is essential. Here, we defined a quantum algorithm for reducing the circuit depth of gate-model quantum computers. The method achieves a reduction in the physical layer allowing significantly reducing implementation costs. The framework is flexible and can be used for arbitrary circuit depths.

Submission note. Parts of this work were presented in conference proceedings[101].

Ethics statement. This work did not involve any active collection of human data.

Appendix

Algorithm 1

$\mathscr{L}$ supervised machine learning control of $\mathscr{C}$

Input: The Ω_i coefficients from block $\mathscr{C}$.

Output: Classification of the Ω_i coefficients, error Θ of $\mathscr{P}$; error Δ of $\mathscr{C}$, updated Ω_i coefficients.

Step 1. Let $\mathscr{S}_T$ be the training set of $\mathscr{L}$ as $\mathscr{S}_T = \langle(\kappa_1, l_1), \ldots, (\kappa_m, l_m)\rangle$, where κ_i is an i-th instance and an n-length vector (see (48)) drawn from the input space $\mathscr{X}$, while $l_i \in \mathscr{S}_\ell$ is the class label associated with κ_i, where $\mathscr{S}_\ell$ is the set of labels $\mathscr{S}_\ell = \{1, \ldots, k\}$.

Step 2. For $i = 1, \ldots, m$, set the initial $D_0(i)$ distribution of the i-th instance of $\mathscr{S}_T$ as $D_0(i) = \frac{1}{m}$.

Step 3. Set the R iteration number. For an r-th iteration, $r = 1, \ldots, R$, let D_r be distribution. Define hypothesis (implementable via a weak learning algorithm[102–105]) using the input distribution D_r as

$$h_r\colon \mathscr{X} \to \mathscr{S}_\ell \tag{101}$$

such that the ε_r training error (error of h_r) of $\mathscr{L}$ is

$$\varepsilon_r = \Pr_i[h_r(\kappa_i) \neq l_i] = \sum\nolimits_{i:h_r(\kappa_i)\neq l_i} D_r(i) \tag{102}$$

is minimized with respect to the distribution D_r.

Step 4. If $\varepsilon_r \leq 0.5$ goto Step 5, otherwise set $R = r - 1$, and stop.

Step 5. If $h_r(\kappa_i) = l_i$, set $D_{r+1}(i)$ as

$$D_{r+1}(i) = W_r \frac{D_r(i)}{\chi_r}, \tag{103}$$

where χ_r is a normalization term, while W_r is a weighting coefficient, as

$$W_r = \frac{\varepsilon_r}{1 - \varepsilon_r}. \tag{104}$$

If $h_r(\kappa_i) \neq l_i$, set

$$D_{r+1}(i) = \frac{D_r(i)}{\chi_r}. \tag{105}$$

Step 6. Repeat steps 3–5 for R times.

Step 7. From the R hypotheses $h_1, \ldots, h_R$, set the final hypothesis H to classify the Ω_i coefficients and the input system Y as

$$H = \underset{l \in \mathscr{S}_\ell}{\operatorname{argmax}} \sum_{r:h_r(\kappa)=l} \log \frac{1}{W_r}, \tag{106}$$

and calculate the $\varepsilon(H)$ error of H as

$$\varepsilon(H) = \sum_{i:H(\kappa_i)\neq l_i} D_H(i), \tag{107}$$

where D_H is the distribution associated to H.

Step 8. Initialize $\partial_{\mathscr{C}}$ and $\eta_{\mathscr{C}}$ parameters as

$$\partial_{\mathscr{C}} = \frac{\eta_{\mathscr{C}}}{n}, \tag{108}$$

where

$$\eta_{\mathscr{C}} = \|Y\|^2 = Y \cdot Y = \sum_{i=1}^{n} \frac{1}{n}(y_i \times y_i), \tag{109}$$

where Y is defined in (28), $y_i = U(\overrightarrow{\theta}_i)x_i$, and

$$\partial_{\mathscr{C}} \leq \left(\frac{1}{n}(\Omega_i \times \Omega_i)\right) \leq \eta_{\mathscr{C}}, \tag{110}$$

where $\Omega_i = U(\overrightarrow{\theta'}_i)\tilde{x}_i$.

Step 9. For an i-th coefficient Ω_i, set

$$\partial_{\mathscr{C},i} = \eta_{\mathscr{C}} - \left(\frac{1}{n}(\Omega_i \times \Omega_i)\right). \tag{111}$$

Step 10. Set σ_i threshold for an i-th coefficient as

$$\sigma_i = \frac{\partial_{\mathscr{C},i}}{\|Y\|^2}. \tag{112}$$

Repeat step 9, until

$$\sigma_i \geq \varsigma \tag{113}$$

where ς, $\varsigma > 0$ is defined via the relation

$$\varsigma \geq \gamma = \frac{1}{\|Y\|^2}\sum_{i=1}^{B}\frac{1}{n}(\Omega_i \times \Omega_i), \tag{114}$$

where B is the number of non-zero elements Ω_i of κ, $B \leq n$. If $\sigma_i < \varsigma$, goto step 11.
Step 11. Output the Θ error of $\mathscr{P}$ as

$$\Theta = \left(\sum_{i=1}^{B}|\frac{1}{\sigma_i}(y_i - \Omega_i)|^{\rho}\right)^{1/\rho} + \varepsilon(H), \tag{115}$$

where ρ is a constant, while $\varepsilon(H)$ is given in (107).
Step 12. Output the error Δ of $\mathscr{C}$ as

$$\Delta = \Theta - \varepsilon(H). \tag{116}$$

Step 13. According to (116), calibrate the Ω_i coefficients (defined in (49)) of output κ in (48) of $\mathscr{C}$, to yield $\Delta < \mathscr{E}$ in (115), where $\mathscr{E}$ is a threshold on the error of $\mathscr{P}$.

Procedure 1 Pre-processing $\mathscr{P}$
Input: Output Y (28) of the non-reduced QG_0 reference circuit.
Output: Input $|\widetilde{X}\rangle$ (31) of reduced quantum circuit QG^*.
Step 0. Characterize the $U_0(\vec{\theta})$ (25) unitaries of the QG_0 reference circuit via gate parameters (33), and the reference input $|X\rangle$ (22) via X (24). Evaluate $|Y\rangle$ (29) via Y (28).
Step 1. Fed Y into block $\mathscr{C}$ to determine κ (48), and control the results of $\mathscr{C}$ via $\mathscr{L}$.
Step 2. Output $\widetilde{X}$ via (30) and produce $|\widetilde{X}\rangle$ as (31).

Algorithm 2

Quantum algorithm U_R
Input: Output state $|Z\rangle$ (36) of QG^*.
Output: Quantum state $|Y_R\rangle$ (59).
Step 1. Prepare the reduced quantum state $|\widetilde{X}\rangle$ (30) using $\widetilde{X}$ (31) of $\mathscr{P}$.
Step 2. Prepare the unitaries $U(\vec{\theta}\,')$ of (32) via $\vec{\theta}\,'$ (33) of QG^*.
Step 3. Fed $|\widetilde{X}\rangle$ into QG^* to yield output system $|Z\rangle$ (36).
Step 4. Utilize the quantum register set $\mathscr{R}$ (63). Apply U_R on $\mathscr{R}$ to get $|Y_R\rangle$ (59) in register $|R_6\rangle$ of $\mathscr{R}$.

Received: 19 September 2019; Accepted: 13 May 2020;
Published online: 08 July 2020

References

1. Preskill, J. Quantum Computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018).
2. Arute, F. *et al*. Quantum supremacy using a programmable superconducting processor, *Nature*, Vol 574, https://doi.org/10.1038/s41586-019-1666-5 (2019).
3. Harrow, A. W. & Montanaro, A. Quantum Computational Supremacy. *Nature* **549**, 203–209 (2017).
4. Aaronson, S. & Chen, L. Complexity-theoretic foundations of quantum supremacy experiments. *Proceedings of the 32nd Computational Complexity Conference*, CCC '17, pages 22:1-22:67, (2017).
5. IBM. *A new way of thinking: The IBM quantum experience*, http://www.research.ibm.com/quantum (2017).
6. Alexeev, Y. *et al*. Quantum Computer Systems for Scientific Discovery, *arXiv:1912.07577* (2019).
7. Loncar, M. *et al*. Development of Quantum InterConnects for Next-Generation Information Technologies, *arXiv:1912.06642* (2019).
8. Foxen, B. *et al*. Demonstrating a Continuous Set of Two-qubit Gates for Near-term Quantum Algorithms, *arXiv:2001.08343* (2020).
9. Ajagekar, A., Humble, T. & You, F. Quantum Computing based Hybrid Solution Strategies for Large-scale Discrete-Continuous Optimization Problems. *Computers and Chemical Engineering* **132**, 106630 (2020).
10. Ajagekar, A. & You, F. Quantum computing for energy systems optimization: Challenges and opportunities. *Energy* **ume 179**, 76–89 (2019).
11. Harrigan, M. *et al*. Quantum Approximate Optimization of Non-Planar Graph Problems on a Planar Superconducting Processor, *arXiv:2004.04197v1* (2020).
12. Rubin, N. *et al*. Hartree-Fock on a superconducting qubit quantum computer, *arXiv:2004.04174v1* (2020).
13. Lloyd, S. Quantum approximate optimization is computationally universal, *arXiv:1812.11075* (2018).
14. Shor, P. W. Algorithms for quantum computation: discrete logarithms and factoring. In: *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (1994).
15. Debnath, S. *et al*. Demonstration of a small programmable quantum computer with atomic qubits. *Nature* **536**, 63–66 (2016).
16. Barends, R. *et al*. Superconducting quantum circuits at the surface code threshold for fault tolerance. *Nature* **508**, 500–503 (2014).

17. Ofek, N. *et al*. Extending the lifetime of a quantum bit with error correction in superconducting circuits. *Nature* **536**, 441–445 (2016).
18. Kielpinski, D., Monroe, C. & Wineland, D. J. Architecture for a large-scale ion-trap quantum computer. *Nature* **417**, 709–711 (2002).
19. Gyongyosi, L. Quantum State Optimization and Computational Pathway Evaluation for Gate-Model Quantum Computers, *Scientific Reports*, https://doi.org/10.1038/s41598-020-61316-4 (2020).
20. Gyongyosi, L. & Imre, S. Training Optimization for Gate-Model Quantum Neural Networks, *Scientific Reports*, https://doi.org/10.1038/s41598-019-48892-w (2019).
21. Gyongyosi, L. & Imre, S. Dense Quantum Measurement Theory, *Scientific Reports*, https://doi.org/10.1038/s41598-019-43250-2 (2019).
22. Gyongyosi, L. & Imre, S. State Stabilization for Gate-Model Quantum Computers, *Quantum Information Processing*, https://doi.org/10.1007/s11128-019-2397-0, (2019).
23. Gyongyosi, L. & Imre, S. Quantum Circuit Design for Objective Function Maximization in Gate-Model Quantum Computers, *Quantum Information Processing*, https://doi.org/10.1007/s11128-019-2326-2 (2019).
24. Brandao, F. G. S. L., Broughton, M., Farhi, E., Gutmann, S. & Neven, H. For Fixed Control Parameters the Quantum Approximate Optimization Algorithm's Objective Function Value Concentrates for Typical Instances, *arXiv:1812.04170* (2018).
25. Zhou, L.,Wang, S.-T., Choi, S., Pichler, H. & Lukin, M. D. Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices, arXiv:1812.01041 (2018).
26. Lechner, W. Quantum Approximate Optimization with Parallelizable Gates, *arXiv:1802.01157v2* (2018).
27. Crooks, G. E. Performance of the Quantum Approximate Optimization Algorithm on the Maximum Cut Problem, *arXiv:1811.08419* (2018).
28. Ho, W. W., Jonay, C. & Hsieh, T. H. Ultrafast State Preparation via the Quantum Approximate Optimization Algorithm with Long Range Interactions, *arXiv:1810.04817* (2018).
29. Song, C. *et al*. 10-Qubit Entanglement and Parallel Logic Operations with a Superconducting Circuit. *Physical Review Letters* **119**(no. 18), 180511 (2017).
30. Laurenza, R. & Pirandola, S. General bounds for sender-receiver capacities in multipoint quantum communications. *Phys. Rev. A* **96**, 032318 (2017).
31. Gyongyosi, L., Imre, S. & Nguyen, H. V. A Survey on Quantum Channel Capacities. *IEEE Communications Surveys and Tutorials* **99**, 1, https://doi.org/10.1109/COMST.2017.2786748 (2018).
32. Imre, S. & Gyongyosi, L. *Advanced Quantum Communications - An Engineering Approach*. Wiley-IEEE Press (New Jersey, USA), (2012).
33. Shor, P. W. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A* **52**, R2493–R2496 (1995).
34. Petz, D. *Quantum Information Theory and Quantum Statistics*, Springer-Verlag, Heidelberg, (2008).
35. Bacsardi, L. On the Way to Quantum-Based Satellite Communication. *IEEE Comm. Mag.* **51**(08), 50–55 (2013).
36. Gyongyosi, L. & Imre, S. Entanglement-Gradient Routing for Quantum Networks, *Sci. Rep*., Nature, https://doi.org/10.1038/s41598-017-14394-w (2017).
37. Gyongyosi, L. & Imre, S. Entanglement Availability Differentiation Service for the Quantum Internet, *Sci. Rep*., Nature, https://doi.org/10.1038/s41598-018-28801-3 (2018).
38. Gyongyosi, L. & Imre, S. Multilayer Optimization for the Quantum Internet, *Sci. Rep*., Nature, (2018).
39. Gyongyosi, L. & Imre, S. Decentralized Base-Graph Routing for the Quantum Internet, *Phys. Rev. A*, American Physical Society, (2018).
40. Gyongyosi, L. & Imre, S. A Survey on Quantum Computing Technology, *Computer Science Review*, Elsevier, https://doi.org/10.1016/j.cosrev.2018.11.002, ISSN: 1574-0137, (2018).
41. Farhi, E., Goldstone, J. & Gutmann, S. A Quantum Approximate Optimization Algorithm. *arXiv:1411.4028*. (2014).
42. Farhi, E., Goldstone, J., Gutmann, S. & Neven, H. Quantum Algorithms for Fixed Qubit Architectures. *arXiv:1703.06199v1* (2017).
43. Farhi, E. & Neven, H. Classification with Quantum Neural Networks on Near Term Processors, *arXiv:1802.06002v1* (2018).
44. Farhi, E., Goldstone, J., Gutmann, S. & Zhou, L. The Quantum Approximate Optimization Algorithm and the Sherrington-Kirkpatrick Model at Infinite Size, *arXiv:1910.08187* (2019).
45. Farhi, E., Goldstone, J. & Gutmann, S. A Quantum Approximate Optimization Algorithm Applied to a Bounded Occurrence Constraint Problem. *arXiv:1412.6062*. (2014).
46. Pirandola, S. End-to-end capacities of a quantum communication network. *Commun. Phys.* **2**, 51 (2019).
47. Pirandola, S., Laurenza, R., Ottaviani, C. & Banchi, L. Fundamental limits of repeaterless quantum communications. *Nature Communications* **8**, 15043, https://doi.org/10.1038/ncomms15043 (2017).
48. Pirandola, S. *et al*. Theory of channel simulation and bounds for private communication. *Quantum Sci. Technol.* **3**, 035009 (2018).
49. Van Meter, R. *Quantum Networking*, John Wiley and Sons Ltd, ISBN 1118648927, 9781118648926 (2014).
50. Biamonte, J. *et al*. Quantum Machine Learning. *Nature* **549**, 195–202 (2017).
51. LeCun, Y., Bengio, Y. & Hinton, G. Deep Learning. *Nature* **521**, 436–444 (2014).
52. Monz, T. *et al*. Realization of a scalable Shor algorithm. *Science* **351**, 1068–1070 (2016).
53. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning*. MIT Press. Cambridge, MA, (2016).
54. Rebentrost, P., Mohseni, M. & Lloyd, S. Quantum Support Vector Machine for Big Data Classification. *Phys. Rev. Lett.* **113** (2014).
55. Lloyd, S. The Universe as Quantum Computer, *A Computable Universe: Understanding and exploring Nature as computation*, Zenil, H. ed., World Scientific, Singapore, 2012, *arXiv:1312.4455v1* (2013).
56. Lloyd, S., Mohseni, M. & Rebentrost, P. Quantum algorithms for supervised and unsupervised machine learning, *arXiv:1307.0411v2* (2013).
57. Lloyd, S., Garnerone, S. & Zanardi, P. Quantum algorithms for topological and geometric analysis of data. *Nat. Commun.*, **7**, *arXiv:1408.3106* (2016).
58. Lloyd, S. *et al*. Infrastructure for the quantum Internet. *ACM SIGCOMM Computer Communication Review* **34**, 9–20 (2004).
59. Lloyd, S., Mohseni, M. & Rebentrost, P. Quantum principal component analysis. *Nature Physics* **10**, 631 (2014).
60. Schuld, M., Sinayskiy, I. & Petruccione, F. An introduction to quantum machine learning. *Contemporary Physics* 56, pp. 172–185. *arXiv: 1409.3097* (2015).
61. Krisnanda, T. *et al*. Probing quantum features of photosynthetic organisms. *npj Quantum Inf.* **4**, 60 (2018).
62. Krisnanda, T. *et al*. Revealing Nonclassicality of Inaccessible Objects. *Phys. Rev. Lett.* **119**, 120402 (2017).
63. Krisnanda, T. *et al*. Observable quantum entanglement due to gravity. *npj Quantum Inf.* **6**, 12 (2020).
64. Krisnanda, T. *et al*. Detecting nondecomposability of time evolution via extreme gain of correlations. *Phys. Rev. A* **98**, 052321 (2018).
65. Shannon, K., Towe, E. & Tonguz, O. On the Use of Quantum Entanglement in Secure Communications: A Survey, *arXiv:2003.07907*, (2020).
66. Amoretti, M. & Carretta, S. Entanglement Verification in Quantum Networks with Tampered Nodes, *IEEE Journal on Selected Areas in Communications*, https://doi.org/10.1109/JSAC.2020.2967955 (2020).
67. Cao, Y. *et al*. Multi-Tenant Provisioning for Quantum Key Distribution Networks with Heuristics and Reinforcement Learning: A Comparative Study, *IEEE Transactions on Network and Service Management*, https://doi.org/10.1109/TNSM.2020.2964003 (2020).

68. Cao, Y. *et al.* Key as a Service (KaaS) over Quantum Key Distribution (QKD)-Integrated Optical Networks, *IEEE Comm. Mag.*, https://doi.org/10.1109/MCOM.2019.1701375 (2019).
69. Farhi, E., Gamarnik, D. & Gutmann, S. The Quantum Approximate Optimization Algorithm Needs to See the Whole Graph: A Typical Case, *arXiv:2004.09002v1* (2020).
70. Farhi, E., Gamarnik, D. & Gutmann, S. The Quantum Approximate Optimization Algorithm Needs to See the Whole Graph: Worst Case Examples, *arXiv:2005.08747* (2020).
71. Brown, K. A. & Roser, T. Towards storage rings as quantum computers. *Phys. Rev. Accel. Beams* **23**, 054701 (2020).
72. Sax, I. *et al.* Approximate Approximation on a Quantum Annealer, *arXiv:2004.09267* (2020).
73. Pirandola, S. & Braunstein, S. L. Unite to build a quantum internet. *Nature* **532**, 169–171 (2016).
74. Wehner, S., Elkouss, D. & Hanson, R. Quantum internet: A vision for the road ahead. *Science* **362**, 6412 (2018).
75. Pirandola, S. Bounds for multi-end communication over quantum networks. *Quantum Sci. Technol.* **4**, 045006 (2019).
76. Pirandola, S. Capacities of repeater-assisted quantum communications, *arXiv:1601.00966* (2016).
77. Pirandola, S. *et al.* Advances in Quantum Cryptography, *arXiv:1906.01645* (2019).
78. Cacciapuoti, A. S. *et al.* Quantum Internet: Networking Challenges in Distributed Quantum Computing, *arXiv:1810.08421* (2018).
79. Caleffi, M. End-to-End Entanglement Rate: Toward a Quantum Route Metric, *2017 IEEE Globecom*, https://doi.org/10.1109/GLOCOMW.2017.8269080, (2018).
80. Caleffi, M. Optimal Routing for Quantum Networks, *IEEE Access*, Vol 5, https://doi.org/10.1109/ACCESS.2017.2763325 (2017).
81. Caleffi, M., Cacciapuoti, A. S. & Bianchi, G. Quantum Internet: from Communication to Distributed Computing, *aXiv:1805.04360* (2018).
82. Castelvecchi, D. The quantum internet has arrived, *Nature*, News and Comment, https://www.nature.com/articles/d41586-018-01835-3 (2018).
83. Cuomo, D., Caleffi, M. & Cacciapuoti, A. S. Towards a Distributed Quantum Computing Ecosystem, *arXiv:2002.11808v1* (2020).
84. Chakraborty, K., Rozpedeky, F., Dahlbergz, A. & Wehner, S. Distributed Routing in a Quantum Internet, *arXiv:1907.11630v1* (2019).
85. Khatri, S., Matyas, C. T., Siddiqui, A. U. & Dowling, J. P. Practical figures of merit and thresholds for entanglement distribution in quantum networks. *Phys. Rev. Research* **1**, 023032 (2019).
86. Kozlowski, W. & Wehner, S. Towards Large-Scale Quantum Networks, *Proc. of the Sixth Annual ACM International Conference on Nanoscale Computing and Communication*, Dublin, Ireland, *arXiv*:1909.08396 (2019).
87. Pathumsoot, P., Matsuo, T., Satoh, T., Hajdusek, M., Suwanna, S. & Van Meter, R. Modeling of Measurement-based Quantum Network Coding on IBMQ Devices. *Phys. Rev. A* **101**, 052301 (2020).
88. Pal, S., Batra, P., Paterek, T. & Mahesh, T. S. Experimental localisation of quantum entanglement through monitored classical mediator, *arXiv:1909.11030v1* (2019).
89. Miguel-Ramiro, J. & Dur, W. Delocalized information in quantum networks, *New J. Phys*, https://doi.org/10.1088/1367-2630/ab784d (2020).
90. Pirker, A. & Dur, W. A quantum network stack and protocols for reliable entanglement-based networks, *arXiv:1810.03556v1* (2018).
91. Miguel-Ramiro, J., Pirker, A. & Dur, W. Genuine quantum networks: superposed tasks and addressing, *arXiv:2005.00020v1* (2020).
92. Amer, O., Krawec, W. O. and Wang, B. Efficient Routing for Quantum Key Distribution Networks, *arXiv:2005.12404* (2020).
93. Liu, Y. Preliminary Study of Connectivity for Quantum Key Distribution Network, *arXiv:2004.11374v1* (2020).
94. Hosen, M. A., Khosravi, A., Nahavandi, S. & Creighton, D. *IEEE Transactions on Industrial Electronics* **62**(no. 7), 4420–4429 (2015).
95. Precup, R.-E., Angelov, P., Costa, B. S. J. & Sayed-Mouchaweh, M. An overview on fault diagnosis and nature-inspired optimal control of industrial process applications. *Computers in Industry* **74**, 75–94 (2015).
96. Saadat, J., Moallem, P. & Koofigar, H. Training echo state neural network using harmony search algorithm. *International Journal of Artificial Intelligence* **15**(no. 1), 163–179 (2017).
97. Vrkalovic, S., Lunca, E.-C. & Borlea, I.-D. Model-free sliding mode and fuzzy controllers for reverse osmosis desalination plants. *International Journal of Artificial Intelligence* **16**(no. 2), 208–222 (2018).
98. Guerreschi, G. G. & Smelyanskiy. M. Practical optimization for hybrid quantum-classical algorithms, *arXiv:1701.01450* (2017).
99. Pang, C.-Y., Zhou, Z.-W. & Guo, G.-C. Quantum Discrete Cosine Transform for Image Compression, *arXiv:quant-ph/0601043* (2006).
100. Grover, L. K. A fast quantum mechanical algorithm for database search, *Proceedings of the 28th ACM symposium on Theory of Computing (STOC)*, pp.212–218 (1996).
101. Gyongyosi, L. A Universal Quantum Algorithm for Time Complexity Reduction of Quantum Computers, *Proceedings of Quantum Information Processing 2019* (QIP 2019), University of Colorado Boulder, USA (2019).
102. Roy, A. & Chakraborty, R. S. Camera Source Identification Using Discrete Cosine Transform Residue Features and Ensemble Classifier, *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 10.1109/CVPRW.2017.231 (2017).
103. Freund, Y. & Schapire, R. E. Experiments with a new boosting algorithm, *ICML*, vol. 96, pp. 148–156. 4 (1996).
104. Freund, Y. & Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences.* **55**, 119 (1997).
105. Criminisi, A., Shotton, J. & Konukoglu, E. Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning, *Microsoft Research Cambridge, Tech. Rep.* MSRTR-2011-114, vol. 5, no. 6, p. 12, 3 (2011).

Acknowledgements

Open access funding provided by Budapest University of Technology and Economics (BME). The research reported in this paper has been supported by the Hungarian Academy of Sciences (MTA Premium Postdoctoral Research Program 2019), by the National Research, Development and Innovation Fund (TUDFO/51757/2019-ITM, Thematic Excellence Program), by the National Research Development and Innovation Office of Hungary (Project No. 2017-1.2.1-NKP-2017-00001), by the Hungarian Scientific Research Fund - OTKA K-112125 and in part by the BME Artificial Intelligence FIKP grant of EMMI (Budapest University of Technology, BME FIKP-MI/SC).

Author contributions

L.GY. designed the protocol and wrote the manuscript. L.GY. and S.I. analyzed the results. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41598-020-67014-5.

Correspondence and requests for materials should be addressed to L.GY.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.