






Variational secure cloud quantum computingYuta Shingu ^{1,2,*}, Yuki Takeuchi ^{3,†}, Suguru Endo,⁴ Shiro Kawabata,^{2,5} Shohei Watabe ¹, Tetsuro Nikuni ^{1,‡}, Hideaki Hakoshima ² and Yuichiro Matsuzaki^{2,5,§}¹*Department of Physics, Faculty of Science Division I, Tokyo University of Science, Shinjuku, Tokyo 162-8601, Japan*²*Research Center for Emerging Computing Technologies, National Institute of Advanced Industrial Science and Technology (AIST), 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan*³*NTT Communication Science Laboratories, NTT Corporation, 3-1 Morinosato Wakamiya, Atsugi, Kanagawa 243-0198, Japan*⁴*NTT Computer and Data Science Laboratories, NTT corporation, Musashino, Tokyo 180-8585, Japan*⁵*NEC-AIST Quantum Technology Cooperative Research Laboratory, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Ibaraki 305-8568, Japan*

(Received 16 July 2021; accepted 21 December 2021; published 3 February 2022)

Variational quantum algorithms (VQAs) have been considered to be useful applications of noisy intermediate-scale quantum (NISQ) devices. Typically, in VQAs, a parametrized ansatz circuit is used to generate a trial wave function, and the parameters are optimized to minimize a cost function. On the other hand, blind quantum computing (BQC) has been studied in order to provide a quantum algorithm with security by using cloud networks. A client with a limited ability to perform quantum operations hopes to have access to a quantum computer of a server, and BQC allows the client to use the server's computer without leakage of the client's information (such as input, running quantum algorithms, and output) to the server. However, BQC is designed for fault-tolerant quantum computing, and this requires many ancillary qubits, which may not be suitable for NISQ devices. Here, we propose an efficient way to implement the NISQ computing with guaranteed security for the client. In our architecture, only $N + 1$ qubits are required, under an assumption that the form of ansätze is known to the server, where N denotes the necessary number of the qubits in the original NISQ algorithms. The client only performs single-qubit measurements on an ancillary qubit sent from the server, and the measurement angles can specify the parameters for the ansätze of the NISQ algorithms. The no-signaling principle guarantees that neither parameters chosen by the client nor the outputs of the algorithm are leaked to the server. This work paves the way for new applications of NISQ devices.

DOI: [10.1103/PhysRevA.105.022603](https://doi.org/10.1103/PhysRevA.105.022603)**I. INTRODUCTION**

Quantum devices have the potential to offer significant advantages over classical devices. Especially, quantum computation, quantum cryptography, and quantum metrology are considered promising applications of quantum devices [1–19]. Recently, great efforts have been devoted to the hybridization between quantum computation, quantum cryptography, and quantum metrology [20–36].

Blind quantum computation (BQC) is an idea to combine quantum computation and quantum cryptography [37–42], where the concept of measurement-based quantum computation (MBQC) [43–45] is adopted. Suppose that a client who does not have a sophisticated quantum device hopes to access a server that has a scalable fault-tolerant quantum computer. The BQC provides a client with a way to access the server's quantum computer in a secure way where the client's information such as input, output, and algorithm is not leaked to the server. The server sends a cluster state, which is a resource of the entanglement, to the client. On the other

hand, the client performs the single-qubit measurements on the cluster state. Importantly, the client needs to change angles of the single-qubit measurements depending on the algorithm, while the form of the cluster state generated by the server does not depend on the choice of the algorithm. Therefore, the server does not obtain any information of either the details or output of the algorithm set by the client, and the no-signaling principle guarantees the security of the protocol [38,46].

Recently, many theoretical and experimental works have been devoted to developing quantum devices in the noisy intermediate-scale quantum (NISQ) era. The NISQ device could involve tens to thousands of qubits with a gate error rate of around 10^{-3} [47]. The NISQ computing typically requires only a shallow circuit to implement quantum algorithms. Variational quantum algorithms (VQAs) are the typical application of the NISQ computing [48–54]. In the VQA, one generates a trial wave function from a parametrized ansatz circuit that is typically shallow. To optimize a cost function tailored to a problem, one updates the parameters with classical computation to generate a new trial wave function. One can search exponentially large Hilbert space with the parametrized quantum circuit via the repetition of such hybrid quantum-classical operations and thus could find a solution to a given problem.

A natural question is whether one can implement the NISQ computing in the blind architecture. If one adopts the BQC

*shingu.yuta@aist.go.jp

†yuki.takeuchi.yt@hco.ntt.co.jp

‡nikuni@rs.kagu.tus.ac.jp

§matsuzaki.yuichiro@aist.go.jp

with the MBQC, one can in principle perform any gate-type quantum computation including NISQ computing. However, to implement the BQC with the MBQC on the cluster state, the necessary number of the qubits is around $3N$ [43–45], where N is the number of the qubits required in the original NISQ algorithm. Since the number of the qubits in the blind architecture with the MBQC is much larger than that in the original algorithm without blind properties [37–41], such a scheme may not be implementable with the NISQ device with a limited number of qubits.

Here, we propose an efficient scheme to implement the variational secure cloud quantum computing. The purpose of our scheme is that the client accesses the quantum computer of the server to implement the NISQ computing in a secure way where the information of the ansatz circuit's parameters and output of the algorithm are not leaked to the server. This is essential for security, because the ansatz circuit's parameters could contain important information such as private data, especially when we perform machine learning with NISQ devices [55–59]. Importantly, our scheme requires only $N + 1$ qubits while MBQC on the cluster state requires around $3N$ qubits. The key idea of our scheme is to use an ancillary qubit for the implementation of the quantum gates on register qubits of the server. The server performs only a limited set of gate operations with fixed angles, namely, Hadamard operations and controlled- Z gates on the register qubits, while the client performs arbitrary single-qubit measurements on the ancillary qubit.

A key idea of our scheme is the use of ancilla-driven quantum computation (ADQC) [60–63]. While the ADQC was originally discussed as one of the novel ways to perform the gate-type quantum computation, we adopt the ancilla-driven architecture for NISQ computing with security inbuilt. In our architecture, the server couples an ancillary qubit to a register qubit via a fixed two-qubit gate at the server side, and the ancillary qubit is sent to the client. Then the client implements a single-qubit measurement on the ancillary qubit either to specify a parameter for the NISQ computing or to readout a computational result. Importantly, since the client does not send any qubits or classical signals to the server, the information about the parameters and output of the NISQ algorithm cannot be leaked to the server due to the no-signaling principle [38,46].

The paper is structured as follows: In Secs. II and III, we review the ADQC and NISQ algorithm, respectively. In Sec. IV, we describe our architecture of the NISQ computing with security inbuilt. In Sec. V, we conclude our results.

II. ANCILLA-DRIVEN QUANTUM COMPUTATION

In the ADQC [60], we define register qubits to execute algorithms and also define an ancillary qubit that can be spatially transferred from one place to another. The basic idea of the ADQC is to entangle the register qubit and ancillary qubit, and the ancillary qubit is sent to another place for the measurement at a specific angle. These operations allow one to perform a universal set of operations. For the implementation with the physical systems, register qubits can be solid-state systems that can interact with photons, and the ancillary qubit can be an optical photon that is transmitted to a distant place.

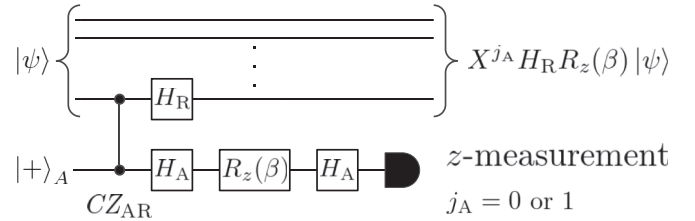


FIG. 1. The circuit for implementing the ancilla-driven quantum computation (ADQC). The upper horizontal lines (the lower line) represents register qubits (an ancillary qubit), where CZ_{AR} denotes the controlled- Z gate between one of the register qubits and the ancillary qubit, H_R (H_A) denotes the Hadamard gate for the register (ancillary) qubit, and $R_z(\beta)$ denotes a parametrized rotation around the z axis with any value β . We prepare the initial states $|\psi\rangle$ and $|+\rangle_A$ for the register qubits and the ancillary qubit, respectively, where $|\psi\rangle$ denotes an arbitrary input state. After implementing the unitary operation $H_A R_z(\beta) E_{AR}$ and measuring the ancillary qubit in the z basis, where $E_{AR} \equiv H_A H_R CZ_{AR}$, we obtain $X^{j_A} H_R R_z(\beta) |\psi\rangle$ at the register qubits, where X denotes the Pauli X gate and $j_A = 0$ or 1 is the result of the measurement.

A. Single-qubit rotation on a register qubit

We explain a realization of single-qubit rotation along the z axis as follows (see Fig. 1):

(1) We prepare a state $|+\rangle_A \equiv \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ of an ancilla qubit (which we call qubit A) and any state $|\psi\rangle$ of register qubits (which we call qubits R).

(2) The ancillary qubit A is coupled with one of the register qubits R via a controlled- Z gate CZ_{AR} , and, subsequently, we implement two Hadamard gates H_A and H_R to the qubit A and the qubit R, respectively. Thus, we have a unitary operation of $E_{AR} \equiv H_A H_R CZ_{AR}$.

(3) A rotation about the z axis $R_z(\beta)$ and a Hadamard gate are implemented on the ancillary qubit, where β is an arbitrary rotation angle.

(4) Measuring the ancillary qubit in the z basis projects the state of the register qubit onto $X^{j_A} H_R R_z(\beta) |\psi\rangle$, where $j_A = 0$ or 1 is the result of the measurement on the ancillary qubit.

The third and final steps can be unified into a single measurement step if an arbitrary-angle single-qubit measurement can be implemented on the ancillary qubit. The details of performing an arbitrary single-qubit rotation and two-qubit gates with ADQC are explained in Appendix A.

III. VARIATIONAL QUANTUM ALGORITHM FOR NISQ DEVICE

Variational quantum algorithms (VQAs) perform a required task by preparing a parametrized wave function on a quantum circuit $|\psi(\vec{\theta})\rangle$ with the variational parameters $\vec{\theta}$ to be optimized by minimizing a cost function $C(\vec{\theta})$ tailored to a problem. The parametrized wave function can be generally described as $|\psi(\vec{\theta})\rangle = U_{AN}(\vec{\theta})|\bar{0}\rangle$ with $|\bar{0}\rangle \equiv \bigotimes_{i=1}^N |0\rangle$, where the ansatz quantum circuit is represented as a repetition of parametrized quantum gates and fixed quantum gates as $U_{AN}(\vec{\theta}) = V_{L+1} U_L(\theta_L) V_L U_{L-1}(\theta_{L-1}) \cdots U_1(\theta_1) V_1$. Here, L is the number of parameters, $U_k(\theta_k)$ and V_k are the k th parametrized and fixed gates, respectively, and θ_k is the k th

component of the parameter set $\vec{\theta}$. As an example of the cost function, in the celebrated variational quantum eigensolver (VQE) [48,51], one uses the expectation value of the Hamiltonian H , i.e., $\langle \psi(\vec{\theta}) | H | \psi(\vec{\theta}) \rangle$. Typically, the parameters at $(j+1)$ th step $\vec{\theta}[j+1]$ is obtained by optimizing the cost function at the j th step $C(\vec{\theta}[j])$ by using, e.g., gradient descent methods. The total number of iteration steps to update the parameters is defined as M . The other example of VQAs is variational quantum simulations (VQSs), which are used to simulate quantum dynamics such as the Schrödinger equation [53,54]. By using the variational principles, it is possible to minimize the distance between the ideal state in the exact evolution and the parametrized trial state, which provides us with the feasible update rule of parameters.

In variational algorithms, we should implement not only the original quantum circuit but also variant types of the original circuit. For example, in many variational algorithms, derivatives of quantum states, i.e., $\frac{\partial |\psi(\vec{\theta})\rangle}{\partial \theta_k}$ are used. They are generated from a different quantum circuit from the original ansatz circuit. To discuss these cases in a general form, we denote the set of variational quantum circuits used in the algorithm as $\{U_{\text{AN}}^{(i)}(\vec{\theta})\}_{i=1}^G \equiv \{V_{L+1}^{(i)} U_L^{(i)}(\theta_L) V_L^{(i)} U_{L-1}^{(i)}(\theta_{L-1}) V_{L-1}^{(i)} \cdots U_1^{(i)}(\theta_1) V_1^{(i)}\}_{i=1}^G$, where G is the number of variational quantum circuits including the original and variants. Accordingly, we denote the set of the observables measured in these quantum circuits as $\{\hat{A}_1^{(i)}, \hat{A}_2^{(i)}, \dots, \hat{A}_{K^{(i)}}^{(i)}\}_{i=1}^G$, where $\hat{A}^{(i)}$ is a Pauli matrix (or an operator made up of tensor products of the Pauli matrices), and $K^{(i)}$ is the number of observables measured in the i th quantum circuits. We use these notation throughout this paper. We show a prescription about how to implement the conventional variational algorithms with these notation in Appendix B.

IV. VARIATIONAL SECURE CLOUD QUANTUM COMPUTING

We explain our protocol of the variational secure cloud quantum computing. Suppose that a client who has the ability to perform only single-qubit measurements hopes to access the NISQ computer of the server in a secure way. The main purpose of our scheme is to hide the information of the ansatz parameters $\vec{\theta}$ set by the client and output of the algorithm. In our scheme, the ansatz circuit to be implemented by the server is publicly announced beforehand. Our scheme is efficient for the NISQ device that has a limited resource, because our scheme requires only a single ancillary qubit independently of the number of qubits needed in the original NISQ algorithm. This is in stark contrast with the original BQC. In the BQC, all information of the choice of the client is hidden [37–41], while $3N$ qubits are approximately required to execute an algorithm using N qubits.

Throughout our paper, we assume that the client has his or her own private space, and any information in the private space is not leaked to the outside. This is the standard assumption in the quantum key distribution [64]

The key of our protocol is to use the concept of the ADQC when the server runs the NISQ computing algorithm. We assume that the server has register qubits, and an ancillary qubit can be sent from the server to the client. When the

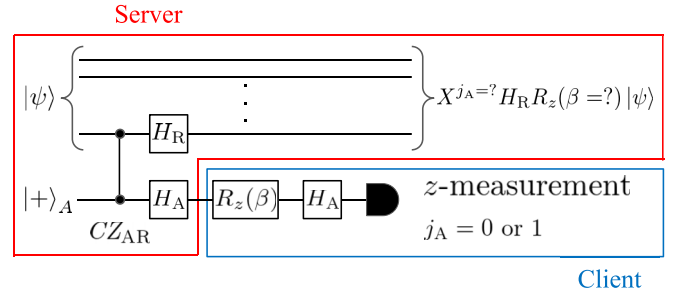


FIG. 2. A quantum circuit to implement a single-qubit rotation by the client in our scheme. The circuit is the same as that in Fig. 1. First, the server entangles one register qubit with an ancillary qubit by the unitary operation $E_{\text{AR}} = H_{\text{A}} H_{\text{R}} C Z_{\text{AR}}$. Second, the server sends the ancillary qubit to the client. Third, the client performs a single-qubit rotation of $H_{\text{A}} R_z(\beta)$ on the ancillary qubit, where β is determined only by the client. Finally, after the client measures the ancillary qubit in the z basis, $X^{j_{\text{A}}} H_{\text{R}} R_z(\beta) |\psi\rangle$ is generated for the register qubit. Since the client does not send any signals to the server, the server does not have any information about the rotation angle β and a measurement result j_{A} , which is guaranteed by the no-signaling principle. By repeating this process several times, arbitrary single-qubit rotations on a register qubit can be implemented.

server needs to implement a single-qubit operation based on the ansatz, the server uses the single-qubit rotation scheme of the ADQC as shown in Fig. 2. More specifically, the server performs a two-qubit gate E_{AR} between the register qubit (that we want to perform the single-qubit rotation) and the ancillary qubit, and sends the client the ancillary qubit to be measured by the client side. The angle and axis of the single-qubit rotation are determined by the client. With three sets of the rotation, an arbitrary single-qubit rotation can be achieved in a register qubit (see Appendix A).

We explain how the information of the parameters and the output is hidden from the server and define that our scheme is secure in this case. During the implementations of the gates in our scheme, the gate operations executed by the server do not depend on the ansatz parameters. Moreover, the client does not send any information to the server during our protocol. Therefore, the server cannot find the parameters of the ansatz circuit set by the client. This discussion is based on the no-signaling principle [38,46].

Moreover, by performing a single-qubit rotation on every register qubit in our scheme, we have byproduct operators of $X^{j_1 + j_3} Z^{j_2}$ on every register qubit as shown in Eq. (A1).

It is known that, when Pauli matrices or an identity operator are randomly implemented on a quantum state (see Sec. 8.3.4 in Ref. [65]), the state becomes completely mixed. This means that the byproduct operators make the state completely mixed for the server. Due to this property, any measurements on the register qubits provide random outcomes if the server side does not have any information of the client's dataset, which is helpful for the client to hide the output of the algorithm. In our scheme, we assume that the server and the client perform the ancilla-driven single-qubit rotation on each qubit at least once during the protocol. We show that, even if the server does not obey the instructions from the client, the client can still

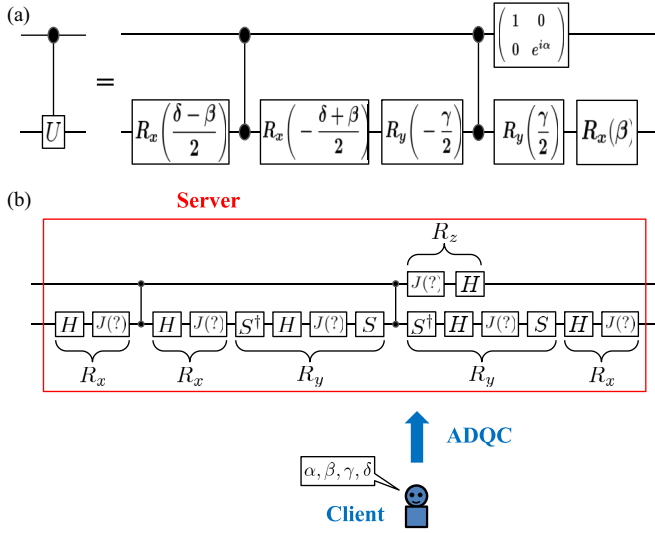


FIG. 3. An implementation of two-qubit gates in our scheme. (a) An equivalent circuit with a controlled-gate operation. We can decompose an arbitrary two-qubit gate into several gates such as single-qubit rotations (including parameters) and controlled-Z gates, where we need to choose appropriate parameters of α , β , γ , and δ for the equivalence. (b) A quantum circuit to implement an arbitrary controlled-gate operation by the client while the rotation parameters are hidden to the server in our scheme. The basic structure of the circuit is the same as that in panel (a), where S denotes a phase gate. The Hadamard, the controlled-Z, and the phase gates are implemented by the server in the register qubits. An important point is that every single-qubit rotation in the circuit should be performed by the client in the same way as described in Fig. 2. In this case, the no-signaling principle guarantees that the rotation parameters (α , β , γ , and δ) cannot be inferred by any operation on the server.

hide the information of the ansatz parameters and output in the Appendix C.

When the server needs to perform a two-qubit gate based on the ansatz with a specific angle, we adopt a quantum circuit shown in Fig. 3(a). The point is that an arbitrary two-qubit gate can be decomposed by arbitrary single-qubit gates and controlled-Z gates. We combine the single-qubit rotations in the ADQC with two controlled-Z gates as shown in Fig. 3(b). In this case, the angles of the two-qubit gates can be determined by the client because the angle of the single-qubit gate can be specified just by the client. Similar to the case of the single-qubit gates, the no-signaling principle guarantees that the server does not obtain any information about the ansatz parameters during the implementation of the two-qubit gates.

The combinations of the single-qubit gates and two-qubit gates in our architecture are shown in Fig. 4. The server performs only Hadamard gates, phase gates, and controlled-Z gates, which are clifford gates. Therefore, when the server measures the observables in the register qubits and sends the measurement results to the client, the client can effectively remove the effect of the byproduct operators by changing the interpretation of the measurement results (see Appendix A).

Before the client performs the secure cloud NISQ computation, the server publicly announces the set of unitary operators $\{U_{AN}^{(i)}\}_{i=1}^G$, the set of the observables

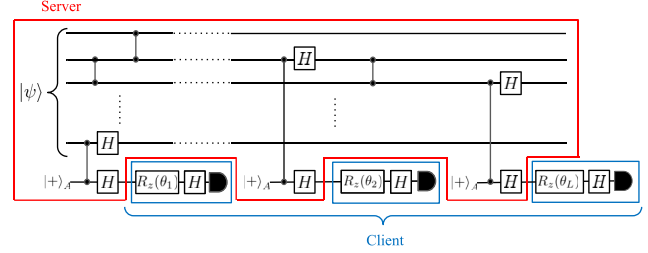


FIG. 4. A quantum circuit to implement our variational secure cloud quantum computing. The NISQ algorithm requires the parameters $\{\theta_j\}_{j=1}^L$ to change the ansatz circuit in a variational way. The server implements gate operations that do not depend on the parameters and sends the ancillary qubit to the client. On the other hand, the client can specify the parameters by changing the measurement angles on the ancillary qubits sent from the server. Importantly, in our scheme, the client does not send any signal to the server, and thus the server does not know the parameters set by the client, due to the no-signaling principle.

$\{\hat{A}_1^{(i)}, \hat{A}_2^{(i)}, \dots, \hat{A}_{K^{(i)}}^{(i)}\}_{i=1}^G$, the repetition numbers $\{N^{(i)}\}_{i=1}^G$ for sampling with the quantum circuits, initial states $\{|\psi^{(i)}(\vec{\theta}[0])\rangle\}_{i=1}^G$, the number of variational parameters L , the total number of iteration steps for VQAs M , and the number of variants of variational quantum circuits G , as shown in Fig. 5.

We summarize our scheme in Fig. 6 as follows:

(1) Adopting the quantum circuits of $\{U_{AN}^{(i)}\}_{i=1}^G$, the server and client implement these unitary operations to generate the trial wave functions $\{|\psi^{(i)}(\vec{\theta}[1])\rangle\}_{i=1}^G$. Here, parametrized single- and two-qubit gates should be implemented in the specific ways as described in Figs. 2 and 3(b), respectively. More specifically, the server performs operations, such as the Hadamard, the controlled-Z, and the phase gates [Fig. 6(1.a)], while the client specifies the measurement angles [Fig. 6(1.b)].

We do not need to prepare $\{|\psi^{(i)}(\vec{\theta}[1])\rangle\}_{i=1}^G$ simultaneously by using G quantum computers, but we can prepare

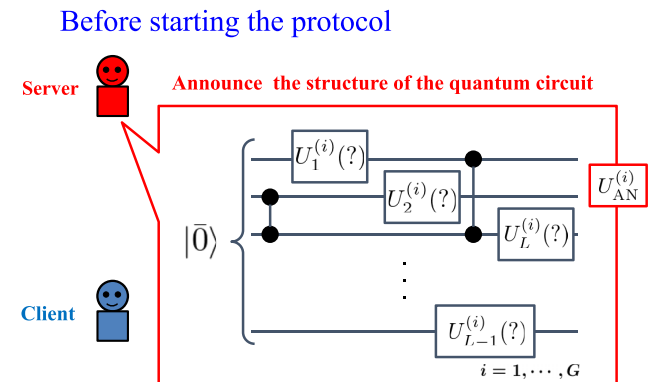


FIG. 5. Before the client starts the protocol, the server broadcasts the information about their quantum circuit. This includes the set of unitary operations $\{U_{AN}^{(i)}\}_{i=1}^G$, the set of the observables $\{\hat{A}_1^{(i)}, \hat{A}_2^{(i)}, \dots, \hat{A}_{K^{(i)}}^{(i)}\}_{i=1}^G$ to be measured, the repetition numbers $\{N^{(i)}\}_{i=1}^G$ for the quantum circuits, initial states $\{|\psi^{(i)}(\vec{\theta}[0])\rangle\}_{i=1}^G$, and the total number M of iteration steps to update the parameters. The client implements the NISQ algorithm based on this information.

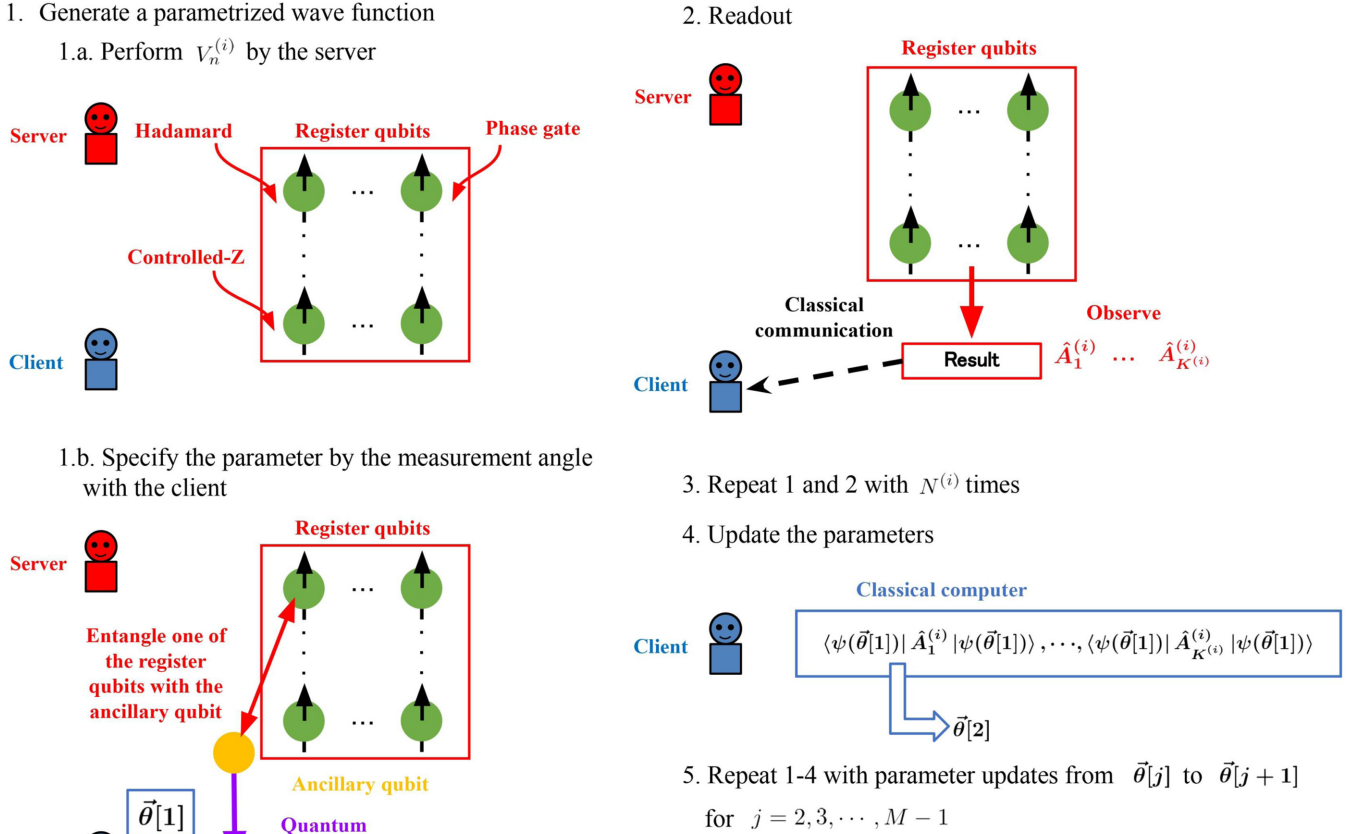


FIG. 6. The sequence of our scheme to implement a NISQ algorithm with a parameter set of $\vec{\theta}$ in a variational secure cloud quantum computing. (1) The server sequentially performs the unitary operations $\{U_{AN}^{(i)}\}_{i=1}^G$ for the register qubits, where G denotes the number of the quantum circuits to be performed. (1.a) The server implements a unitary (non-parametrized) operation $V_n^{(i)}$ for $n = 1, 2, \dots, L + 1$; a Hadamard or a controlled-Z, or a phase gate, on the register qubits. (1.b) The server entangles a register qubit with an ancillary qubit and sends the ancillary qubit to the client in the same way as Fig. 2. The client measures the ancillary qubit sent from the server, where the client specifies a measurement angle based on the initial parameters $\vec{\theta}[1]$. (2) The server measures $\hat{A}_1^{(i)}, \hat{A}_2^{(i)}, \dots,$ and $\hat{A}_{K^{(i)}}^{(i)}$ for $i = 1, 2, \dots, G$ and sends the results to the client by the classical communication. (3) For each $\{U_{AN}^{(i)}\}_{i=1}^G$, the server and the client repeat these two steps $\{N^{(i)}\}_{i=1}^G$ times, and then the client obtains expectation values of $\{\hat{A}_1^{(i)}, \hat{A}_2^{(i)}, \dots, \hat{A}_{K^{(i)}}^{(i)}\}_{i=1}^G$ with $\{|\psi^{(i)}(\vec{\theta}[1])\rangle\}_{i=1}^G$. (4) The client updates the parameters as $\vec{\theta}[2]$ by processing the measurement data with classical computation. (5) The server and the client repeat these four steps $M - 2$ times updating the parameters from $\vec{\theta}[j]$ to $\vec{\theta}[j + 1]$ for $j = 2, 3, \dots, M - 1$, and the client obtains the output. Since the client does not send any signals to the server during the computation, the server cannot obtain any information about $\vec{\theta}[1], \vec{\theta}[2], \dots, \vec{\theta}[M]$, because of the no-signaling principle.

and measure these in sequence by using a single quantum computer, similar to the standard VQA for NISQ devices (see Appendix B).

(2) The server measures the states of the register qubits with $\{\hat{A}_1^{(i)}, \hat{A}_2^{(i)}, \dots, \hat{A}_{K^{(i)}}^{(i)}\}_{i=1}^G$, and sends the results to the client with classical communications.

(3) For the sampling, the server and client repeat the first and the second steps with $\{N^{(i)}\}_{i=1}^G$ times for each state $\{|\psi^{(i)}(\vec{\theta}[1])\rangle\}_{i=1}^G$ so that the client should obtain the expectation values of $\{\hat{A}_1^{(i)}, \hat{A}_2^{(i)}, \dots, \hat{A}_{K^{(i)}}^{(i)}\}_{i=1}^G$.

When the observables are measured, the effect of the byproduct operators can be canceled out by the client (see Appendix A).

(4) By processing the measurement results with a classical computer at the client side, the client updates the parameters and obtains $\vec{\theta}[2] = (\theta_1[2], \dots, \theta_L[2])^T$ for the next step.

(5) The client and the server repeat the steps 1–4 ($M - 2$) times with $\{U_{AN}^{(i)}\}_{i=1}^G$ and $\vec{\theta}[j]$, where classical computation

based on the results at the j th step provides the client with the updated parameters of $\vec{\theta}[j + 1]$ for $j = 2, 3, \dots, M - 1$. The client finally obtains the desired results in a secure way from the server.

As a physical implementation, the register qubits can be the solid-state systems that interact with a photon, and the ancillary qubit can be an optical photon that transmits to a distant place. We implicitly assumed that the photon loss would be negligible during the transmission in the discussion above.

We discuss the effect of photon loss on our scheme. When the server sends the client an ancillary qubit that corresponds to an optical photon, there is a possibility that the photon can be lost during the transmission. In principle, if the server and the client have quantum memories, they can share a Bell pair under the effect of photon loss by repeating the entanglement generation process until success [66], and they can use the Bell pairs to perform our gate operations in a

deterministic way. In this case, the client needs to ask the server to send the photons again and again, depending on how many times the photon is lost [66]. However, in order to apply the no-signaling principle, the client is not allowed to send the server any information. This means that the client cannot ask the server to send the photon again. So we cannot adopt the repeat-until-success strategy with quantum memories.

Thus, we assume that the client adopts the observation results of the readout of the register qubits by the server only when all photons are successfully transmitted to the client during the computation. In this case, the probability of no photon loss during the computation exponentially decreases as the number of sending photons increases. The number of required photons sent to the client can be determined by the number of the tunable parameters used in the ansatz circuit. When $U_{\text{AN}}^{(i)}$ is composed of $n_{\text{single}}^{(i)}$ single-qubit operations and $n_{\text{two}}^{(i)} = L - n_{\text{single}}^{(i)}$ two-qubit operations, the necessary number $N_{\text{ph}}^{(i)}$ of the photons to send the client is at most $N_{\text{ph}}^{(i)} = 3n_{\text{single}}^{(i)} + 6n_{\text{two}}^{(i)}$ as shown in Eq. (A1) and Figs. 2 and 3(b). The probability for all the photons to be detected by the client is $(1 - p_{\text{loss}})^{N_{\text{ph}}^{(i)}}$, where p_{loss} is a photon loss probability for a single transmission. Therefore, the repetition number $N^{(i)}$ with the photon loss should be set to be much larger than $N_{\text{ideal}}^{(i)}/(1 - p_{\text{loss}})^{N_{\text{ph}}^{(i)}}$, where $N_{\text{ideal}}^{(i)}$ denotes the required number of repetition with no photon loss. To keep $N^{(i)}$ within a reasonable amount, p_{loss} should be smaller than 1% under the assumption that $N_{\text{ph}}^{(i)}$ is around a few hundreds.

We could overcome such a problem due to the recent experimental and theoretical developments of quantum repeating technology. The best single-photon detector in optics has 99% efficiency [67–69]. A microwave quantum repeater with a short distance such as 100 m has been proposed [70], and a qubit can catch a microwave photon with 99.4% absorption efficiency in the microwave regime [71]. Also, there are proposals to physically move the solid-state qubit [72,73] for distributed quantum computation or a quantum repeater. Through the combination of these protocols and a long-lived quantum memory such as a nuclear spin [74,75], the ancillary solid-state qubits might be carried to the client without the problems of the photon loss.

In our scheme, the depth of the quantum circuit increases compared with the conventional NISQ algorithms without security. For example, if the client and the server implement single-qubit rotations for every register qubit with those variational parameters in the ansatz circuit, our scheme requires $\Theta(N)$ steps, where N denotes the number of register qubits. However, we can reduce the depth of the quantum circuit in our scheme as follows: Suppose that the client has N photon detectors and can perform single-qubit rotations on N photonic qubits in parallel. In this case, the server can interact with each solid-state qubit (register qubit) with a photonic qubit (ancillary qubit) and emit N photons to the client, who can measure these photons in parallel. This scheme allows the client to implement the single-qubit measurements on all qubits with N variational parameters at the same time, and therefore the depth of the quantum circuit becomes shorter.

Finally, we present some future works. First, in principle, we could also reduce the depth of the quantum circuit in another approach as follows: In our scheme, we assume that

the client tries to hide all variational parameters in the ansatz circuit from the server. However, if the client just wants to hide a part of the variational parameters, the depth of the quantum circuit in our scheme should be shorter. Although this approach seems to be important for the NISQ devices using a short-depth quantum circuit, further research is needed to assess the feasibility. Second, our scheme requires quantum communications while some previous blind quantum computation protocols for the fault tolerant architecture with graph states require only classical communications [76,77]. It is important to investigate whether the client can implement the NISQ algorithm with the server's quantum computer by using classical communications while the information of the ansatz parameters and output is hidden. Third, if the server is adversary to perform some POVM measurements by deviating from the instructions, the client may not be able to obtain correct calculation results. It would be interesting if one could find a scheme to check whether the server obeys the client's instruction so that the client can verify the calculated results. We leave these points for further research.

V. CONCLUSION

In conclusion, we proposed a noisy intermediate-scale quantum (NISQ) computing with security inbuilt. The main targets of our scheme are variational quantum algorithms (VQAs), which involve parameters of an ansatz to be optimized by minimizing a cost function. We considered a circumstance that a client with a limited ability to perform quantum operations hopes to access a NISQ device possessed by a server and the client tries to avoid leakage of the information about the quantum algorithm that he or she runs. Importantly, the naive application of the previously known blind quantum computation (BQC) [38] requires around $3N$ qubits [43–45], where N denotes the number of the qubits to run the quantum algorithm in the original architecture. That may not be suitable for the NISQ devices with the limited number of qubits. Our proposal is more efficient in the sense that we use a single ancillary qubit and N register qubits required in the original NISQ algorithm. In VQAs, we use a parametrized trial wave function, and our scheme prevents the information about the parameters from the leakage to the server. We rely on the no-signaling principle to guarantee security. Our scheme paves the way for new applications of the NISQ devices.

ACKNOWLEDGMENTS

This work was supported by Leading Initiative for Excellent Young Researchers MEXT Japan and JST presto (Grant No. JPMJPR1919) Japan. This work was supported by MEXT Quantum Leap Flagship Program (MEXT Q-LEAP) (Grants No. JPMXS0120319794 and No. JPMXS0118068682), JST ERATO (Grant No. JPMJER1601), JST (Moonshot R&D–MILLENNIA Program) Grant No. JPMJMS2061, MEXT Quantum Leap Flagship Program (MEXT Q-LEAP) Grant No. JPMXS0118067394, and S.W. was supported by Nanotech CUPAL, National Institute of Advanced Industrial Science and Technology (AIST). This paper was partly based on results obtained from the Project No. JPNP16007,

commissioned by the New Energy and Industrial Technology Development Organization (NEDO), Japan.

Y.S. and Y.T. contributed equally to this work.

APPENDIX A: DETAILED ANCILLA-DRIVEN QUANTUM COMPUTATION

1. Arbitrary single-qubit rotation

We describe a way to implement an arbitrary single-qubit rotation. Any single-qubit rotation U can be represented by $U = R_z(\beta')R_x(\gamma')R_z(\delta')$, where R_x denotes a rotation about the x axis, and β' , γ' , and δ' denote the rotation angles about the corresponding axis. Defining $J(\beta) \equiv HR_z(\beta)$, one can rewrite U as $U = J(\beta)J(\gamma)J(\delta)$, where we choose β , γ , and δ to satisfy $R_z(\beta)R_x(\gamma)R_z(\delta) = HU$. As we explained, one can implement the single-qubit rotation of $X^j H_R R_z(\beta) |\psi\rangle$ on the register qubit by the coupling with an ancillary qubit and a subsequent measurement. Therefore, three sequential operations of this type of the single-qubit rotation provide us with the following operation:

$$\begin{aligned} & \{X^{j_3} H_R R_z[(-1)^{j_2} \beta]\} X^{j_2} H_R R_z[(-1)^{j_1} \gamma] \{X^{j_1} H_R R_z(\delta)\} \\ &= X^{j_3} J[(-1)^{j_2} \beta] X^{j_2} J[(-1)^{j_1} \gamma] X^{j_1} J(\delta) \\ &= (-1)^{j_1 \cdot j_2} X^{j_1 + j_2} J(\beta) J(\gamma) J(\delta), \end{aligned} \quad (\text{A1})$$

where j_i denotes the result of the i th measurement on the ancillary qubits. For the implementation of this operation, we change the rotation angle of the ancillary qubit depending on the previous measurement results. Equation (A1) involves the byproduct operator $X^{j_1 + j_2} Z^{j_2}$. However, as long as we measure the qubit in a computational basis for the readout, the byproduct operators just flip the measurement result from 0 to 1 or vice versa, and so we can effectively remove the byproduct operators from the states by changing the interpretation of the measurement results.

2. Two-qubit gate between the register qubits

We explain a way to perform the controlled-Z gate on the two register qubits R and R' in the ADQC. First, we implement E_{AR} on the ancillary qubit (prepared in the state $|+\rangle_A$) and the register qubit R and subsequently perform $E_{AR'}$ on the ancillary qubit and the other register qubit R' . Second, one measures the ancillary qubit in the y basis. These operations are equivalent to the controlled-Z gate, up to local operations.

When we perform several single-qubit gates and two-qubit gates, the byproduct operators are applied as $U_\Sigma U_{\text{ideal}} |\bar{0}\rangle$, where U_Σ denotes the total byproduct operators and U_{ideal} denotes the unitary operations that we aim to implement. Again, when one measures observables of Pauli matrices (or a tensor product of Pauli matrices), one can effectively remove the byproduct operators from the states by changing the interpretation of the measurement results.

APPENDIX B: VQA FOR NISQ DEVICES

We show a prescription about how to implement the conventional variational algorithms with our notation. We prepare a parametrized wave function on a quantum circuit $|\psi(\vec{\theta})\rangle$

with the variational parameters $\vec{\theta}$ to be optimized by minimizing a cost function $C(\vec{\theta})$ tailored to a problem. First, with the quantum circuits of $\{U_{\text{AN}}^{(i)}\}_{i=1}^G$, we realize parametrized wave functions of N qubits $\{|\psi^{(i)}(\vec{\theta}[1])\rangle\}_{i=1}^G$, where $|\psi^{(i)}(\vec{\theta}[1])\rangle \equiv V_{L+1}^{(i)} U_L^{(i)}(\theta_L[1]) V_L^{(i)} \cdots U_1^{(i)}(\theta_1[1]) V_1^{(i)} |\bar{0}\rangle$ where $|\bar{0}\rangle \equiv \bigotimes_{i=1}^N |0\rangle$ denotes the wave function, $\vec{\theta}[1] = (\theta_1[1], \dots, \theta_L[1])^T$ is a vector of the parameters, and $\{|\psi^{(i)}(\vec{\theta}[0])\rangle\}_{i=1}^G$ are initial states, and we measure the state of the wave function with observables of $\{\hat{A}_1^{(i)}, \hat{A}_2^{(i)}, \dots, \hat{A}_{K^{(i)}}^{(i)}\}_{i=1}^G$.

Second, for the sampling, we repeat the first step to obtain expectation values of $\{\hat{A}_1^{(i)}, \hat{A}_2^{(i)}, \dots, \hat{A}_{K^{(i)}}^{(i)}\}_{i=1}^G$ with $\{|\psi^{(i)}(\vec{\theta}[1])\rangle\}_{i=1}^G$. Third, based on the expectation values, we implement a classical algorithm so that we can obtain updated parameters $\vec{\theta}[2]$ for the next quantum circuits, where we typically use a gradient method to make the cost function smaller. For example, we use $\vec{\theta}[j+1] = \vec{\theta}[j] - \alpha \text{grad} C(\vec{\theta}[j])$ for the gradient method.

Finally, we repeat the first, second, and third steps $M-2$ times with $\{U_{\text{AN}}^{(i)}\}_{i=1}^G$ and $\vec{\theta}[k]$, where classical computation based on the results at the k th step provides the updated parameters of $\vec{\theta}[k+1]$ for $k = 2, 3, \dots, M-1$. These processes provide us with an output of the algorithm.

APPENDIX C: SECURITY PROOF

In this Appendix, we show that even when the server is adversary to perform positive operator-valued measure (POVM) measurements at his or her own will at any stage of our protocol in order to know the information of the client, the server cannot obtain any information about the client's output and measurement angles chosen by the client. Our proof is based on the previous work of Morimae and Fujii [38]. We assume that the server cannot guess the choice of the client's measurement angles from the form of the ansatz. Also, due to the byproduct operators acting on every qubit, the server cannot guess the final outputs by directly measuring the output quantum states obtained from the ansätze. By considering these, we have to show that our scheme satisfies C1 and C2:

C1: Given all the classical information obtained by the server during our protocol and the measurement results of any POVM measurements implemented by the server at any stage of our protocol, a conditional probability distribution of *the client's measurement angles* is defined. This conditional probability distribution is equal to its prior probability distribution, which is a uniform probability distribution among all possible angles.

C2: Given all the classical information obtained by the server during our protocol and the measurement results of any POVM measurements implemented by the server at any stage of our protocol, a conditional probability distribution of *the final output of the algorithm for the client* is. This probability distribution is equal to its prior probability distribution, which is a uniform probability distribution among all possible outputs.

C1 and C2 mean that the server cannot obtain any information of the ansatz parameters and the output for the client by implementing any POVM measurements.

Theorem 1. Our scheme satisfies C1.

Proof. No-signaling principle provides the following relationship between probabilities [38,46],

$$\begin{aligned} P(M_S = m_S | \Theta = \vec{\theta}, S = s) \\ = P(M_S = m_S | \Theta = \vec{\theta}', S = s) \end{aligned}$$

for all m_S , $\vec{\theta}$, $\vec{\theta}'$, and s where Θ denotes the random variable representing the client's measurement angles, S denotes the random variable representing the type of the POVM measurement performed at the server side, M_S denotes the random variable representing the result of the POVM measurement, m_S denotes the result of the POVM measurement, $\vec{\theta}$ denotes the variational parameters, $\vec{\theta}'$ denotes another set of the variational parameters, and s denotes a choice of the POVM measurement. We assume $P(\Theta = \vec{\theta} | S = s) = P(\Theta = \vec{\theta}' | S = s)$, because we take an average over all possible client's choice, which is the same assumption as adopted in Ref. [38].

By using this equality, we show that the operations at the server side do not affect the probability distribution of the client's measurement angles, as follows:

$$\begin{aligned} P(\Theta = \vec{\theta} | S = s, M_S = m_S) \\ &= \frac{P(M_S = m_S | \Theta = \vec{\theta}, S = s) P(\Theta = \vec{\theta}, S = s)}{P(S = s, M_S = m_S)} \\ &= \frac{P(M_S = m_S | \Theta = \vec{\theta}, S = s) P(\Theta = \vec{\theta} | S = s) P(S = s)}{P(S = s, M_S = m_S)} \\ &= \frac{P(M_S = m_S | \Theta = \vec{\theta}', S = s) P(\Theta = \vec{\theta}' | S = s) P(S = s)}{P(S = s, M_S = m_S)} \\ &= P(\Theta = \vec{\theta}' | S = s, M_S = m_S). \end{aligned}$$

Thus, the server cannot obtain any information of the client's measurement angles.

Theorem 2. Our scheme satisfies C2.

Proof. No-signaling principle provides the following relationship between probabilities [38,46],

$$\begin{aligned} P(M_S = m_S | O = o, S = s) \\ = P(M_S = m_S | O = o', S = s) \end{aligned}$$

for all m_S , o , o' , and s where O denotes the random variable representing the output of the algorithm for the client, S denotes the random variable representing the type of the POVM measurement which the server performs at the server side, M_S denotes the random variable representing the result of the POVM measurement, m_S denotes the result of the POVM measurement, o denotes the output of the algorithm for the client that the client obtains, o' denotes another output, and s denotes a choice of the POVM measurement. We assume $P(O = o | S = s) = P(O = o' | S = s)$ because we take an average over all possible client's choice, which is the same assumption as adopted in Ref. [38]. By using this equality, we show that the operations at the server side do not affect the probability distribution of the measurement result, as follows:

$$\begin{aligned} P(O = o | S = s, M_S = m_S) \\ &= \frac{P(M_S = m_S | O = o, S = s) P(O = o, S = s)}{P(S = s, M_S = m_S)} \\ &= \frac{P(S = s, M_S = m_S)}{P(M_S = m_S | O = o, S = s) P(O = o | S = s) P(S = s)} \\ &= \frac{P(M_S = m_S | O = o', S = s) P(O = o' | S = s) P(S = s)}{P(S = s, M_S = m_S)} \\ &= P(O = o' | S = s, M_S = m_S). \end{aligned}$$

Thus, the server cannot obtain any information of the outputs of the algorithm for the client.

-
- [1] P. W. Shor, *SIAM J. Comput.* **26**, 1484 (1997).
[2] L. K. Grover, *Phys. Rev. Lett.* **79**, 325 (1997).
[3] A. W. Harrow, A. Hassidim, and S. Lloyd, *Phys. Rev. Lett.* **103**, 150502 (2009).
[4] L. M. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, *Nature (London)* **414**, 883 (2001).
[5] C. H. Bennett and G. Brassard, in *Proceedings of the Conference on Computers, Systems and Signal Processing* (IEEE Piscataway, NJ, 1984), p. 175.
[6] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin, *J. Cryptol.* **5**, 3 (1992).
[7] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, *Rev. Mod. Phys.* **74**, 145 (2002).
[8] A. Broadbent, J. Fitzsimons, and E. Kashefi, in *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science, Atlanta, WA* (IEEE, Piscataway, NJ, 2009), pp. 517–526.
[9] J. P. Dowling and G. J. Milburn, *Philos. Trans. R. Soc., A* **361**, 1655 (2003).
[10] T. P. Spiller, W. J. Munro, S. D. Barrett, and P. Kok, *Contemp. Phys.* **46**, 407 (2005).
[11] C. L. Degen, F. Reinhard, and P. Cappellaro, *Rev. Mod. Phys.* **89**, 035002 (2017).
[12] D. Budker and M. Romalis, *Nat. Phys.* **3**, 227 (2007).
[13] G. Balasubramanian, I. Chan, R. Kolesov, M. Al-Hmoud, J. Tisler, C. Shin, C. Kim, A. Wojcik, P. R. Hemmer, A. Krueger *et al.*, *Nature (London)* **455**, 648 (2008).
[14] J. R. Maze, P. L. Stanwix, J. S. Hodges, S. Hong, J. M. Taylor, P. Cappellaro, L. Jiang, M. G. Dutt, E. Togan, A. Zibrov *et al.*, *Nature (London)* **455**, 644 (2008).
[15] P. Neumann, I. Jakobi, F. Dolde, C. Burk, R. Reuter, G. Waldherr, J. Honert, T. Wolf, A. Brunner, J. H. Shim *et al.*, *Nano Lett.* **13**, 2738 (2013).
[16] D. J. Wineland, J. J. Bollinger, W. M. Itano, F. L. Moore, and D. J. Heinzen, *Phys. Rev. A* **46**, R6797 (1992).
[17] S. F. Huelga, C. Macchiavello, T. Pellizzari, A. K. Ekert, M. B. Plenio, and J. I. Cirac, *Phys. Rev. Lett.* **79**, 3865 (1997).
[18] Y. Matsuzaki, S. C. Benjamin, and J. Fitzsimons, *Phys. Rev. A* **84**, 012103 (2011).
[19] A. W. Chin, S. F. Huelga, and M. B. Plenio, *Phys. Rev. Lett.* **109**, 233601 (2012).

- [20] E. M. Kessler, I. Lovchinsky, A. O. Sushkov, and M. D. Lukin, *Phys. Rev. Lett.* **112**, 150802 (2014).
- [21] W. Dür, M. Skotiniotis, F. Frowis, and B. Kraus, *Phys. Rev. Lett.* **112**, 080801 (2014).
- [22] G. Arrad, Y. Vinkler, D. Aharonov, and A. Retzker, *Phys. Rev. Lett.* **112**, 150801 (2014).
- [23] D. A. Herrera-Martí, T. Gefen, D. Aharonov, N. Katz, and A. Retzker, *Phys. Rev. Lett.* **115**, 200501 (2015).
- [24] T. Unden, P. Balasubramanian, D. Louzon, Y. Vinkler, M. B. Plenio, M. Markham, D. Twitchen, A. Stacey, I. Lovchinsky, A. O. Sushkov, M. D. Lukin, A. Retzker, B. Naydenov, L. P. McGuinness, and F. Jelezko, *Phys. Rev. Lett.* **116**, 230502 (2016).
- [25] Y. Matsuzaki and S. Benjamin, *Phys. Rev. A* **95**, 032303 (2017).
- [26] B. L. Higgins, D. W. Berry, S. D. Bartlett, H. M. Wiseman, and G. J. Pryde, *Nature (London)* **450**, 393 (2007).
- [27] G. Waldherr, J. Beck, P. Neumann, R. Said, M. Nitsche, M. Markham, D. Twitchen, J. Twamley, F. Jelezko, and J. Wrachtrup, *Nat. Nanotechnol.* **7**, 105 (2012).
- [28] P. Komar, E. M. Kessler, M. Bishof, L. Jiang, A. S. Sørensen, J. Ye, and M. D. Lukin, *Nat. Phys.* **10**, 582 (2014).
- [29] T. J. Proctor, P. A. Knott, and J. A. Dunningham, *Phys. Rev. Lett.* **120**, 080501 (2018).
- [30] Z. Eldredge, M. Foss-Feig, J. A. Gross, S. L. Rolston, and A. V. Gorshkov, *Phys. Rev. A* **97**, 042337 (2018).
- [31] V. Giovannetti, S. Lloyd, and L. Maccone, *J. Opt. B: Quantum Semiclassical Opt.* **4**, S413 (2002).
- [32] V. Giovannetti, S. Lloyd, and L. Maccone, *Phys. Rev. A* **65**, 022309 (2002).
- [33] G. Chiribella, L. Maccone, and P. Perinotti, *Phys. Rev. Lett.* **98**, 120501 (2007).
- [34] Z. Huang, C. Macchiavello, and L. Maccone, *Phys. Rev. A* **99**, 022314 (2019).
- [35] Y. Takeuchi, Y. Matsuzaki, K. Miyanishi, T. Sugiyama, and W. J. Munro, *Phys. Rev. A* **99**, 022325 (2019).
- [36] P. Yin, Y. Takeuchi, W.-H. Zhang, Z.-Q. Yin, Y. Matsuzaki, X.-X. Peng, X.-Y. Xu, J.-S. Xu, J.-S. Tang, Z.-Q. Zhou, G. Chen, C. F. Li, and G. C. Guo, *Phys. Rev. Appl.* **14**, 014065 (2020).
- [37] A. Broadbent, J. Fitzsimons, and E. Kashefi, in *2009 50th Annual IEEE Symposium on Foundations of Computer Science (IEEE, 2009)*, pp. 517–526.
- [38] T. Morimae and K. Fujii, *Phys. Rev. A* **87**, 050301(R) (2013).
- [39] Y. Takeuchi, K. Fujii, R. Ikuta, T. Yamamoto, and N. Imoto, *Phys. Rev. A* **93**, 052307 (2016).
- [40] S. Barz, E. Kashefi, A. Broadbent, J. F. Fitzsimons, A. Zeilinger, and P. Walther, *Science* **335**, 303 (2012).
- [41] C. Greganti, M.-C. Roehsner, S. Barz, T. Morimae, and P. Walther, *New J. Phys.* **18**, 013020 (2016).
- [42] W. Li, S. Lu, and D.-L. Deng, *Sci. China: Phys., Mech. Astron.* **64**, 100312 (2021).
- [43] R. Raussendorf and H. J. Briegel, *Phys. Rev. Lett.* **86**, 5188 (2001).
- [44] R. Raussendorf, D. E. Browne, and H. J. Briegel, *Phys. Rev. A* **68**, 022312 (2003).
- [45] P. Walther, K. J. Resch, T. Rudolph, E. Schenck, H. Weinfurter, V. Vedral, M. Aspelmeyer, and A. Zeilinger, *Nature (London)* **434**, 169 (2005).
- [46] S. Popescu and D. Rohrlich, *Found. Phys.* **24**, 379 (1994).
- [47] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan, *J. Phys. Soc. Jpn.* **90**, 032001 (2021).
- [48] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O’Brien, *Nat. Commun.* **5**, 4213 (2014).
- [49] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, *Nature (London)* **549**, 242 (2017).
- [50] N. Moll, P. Barkoutsos, L. S. Bishop, J. M. Chow, A. Cross, D. J. Egger, S. Filipp, A. Fuhrer, J. M. Gambetta, M. Ganzhorn *et al.*, *Quantum Sci. Technol.* **3**, 030503 (2018).
- [51] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, *New J. Phys.* **18**, 023023 (2016).
- [52] E. Farhi, J. Goldstone, and S. Gutmann, [arXiv:1411.4028](https://arxiv.org/abs/1411.4028).
- [53] Y. Li and S. C. Benjamin, *Phys. Rev. X* **7**, 021050 (2017).
- [54] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. C. Benjamin, *Quantum* **3**, 191 (2019).
- [55] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, *Phys. Rev. A* **98**, 032309 (2018).
- [56] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, *Phys. Rev. A* **101**, 032308 (2020).
- [57] E. Farhi and H. Neven, [arXiv:1802.06002](https://arxiv.org/abs/1802.06002) (2018).
- [58] C. Zoufal, A. Lucchi, and S. Woerner, *Quantum Mach. Intell.* **3**, 7 (2021).
- [59] Y. Shingu, Y. Seki, S. Watabe, S. Endo, Y. Matsuzaki, S. Kawabata, T. Nikuni, and H. Hakoshima, *Phys. Rev. A* **104**, 032413 (2021).
- [60] J. Anders, D. K. L. Oi, E. Kashefi, D. E. Browne, and E. Andersson, *Phys. Rev. A* **82**, 020301(R) (2010).
- [61] A. Bocharov, M. Roetteler, and K. M. Svore, *Phys. Rev. Lett.* **114**, 080502 (2015).
- [62] D. Browne and H. Briegel, *Quantum Inf.: Found. Quantum Technol. Appl.* **2**, 449 (2016).
- [63] A. Paetznick and K. M. Svore, *Quantum Inf. Comput.* **14**, 1277 (2014).
- [64] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, *Rev. Mod. Phys.* **81**, 1301 (2009).
- [65] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, Cambridge, UK, 2010).
- [66] S. C. Benjamin, D. E. Browne, J. Fitzsimons, and J. J. Morton, *New J. Phys.* **8**, 141 (2006).
- [67] A. E. Lita, B. Calkins, L. Pellouchoud, A. J. Miller, and S. Nam, in *Advanced Photon Counting Techniques IV* (International Society for Optics and Photonics, 2010), Vol. 7681, p. 76810D.
- [68] D. Fukuda, G. Fujii, T. Numata, K. Amemiya, A. Yoshizawa, H. Tsuchida, H. Fujino, H. Ishii, T. Itatani, S. Inoue *et al.*, *Opt. Express* **19**, 870 (2011).
- [69] A. Kuzanyan, A. Kuzanyan, and V. Nikoghosyan, *J. Contemp. Phys. (Arm. Acad. Sci.)* **53**, 338 (2018).
- [70] Z.-L. Xiang, M. Zhang, L. Jiang, and P. Rabl, *Phys. Rev. X* **7**, 011035 (2017).
- [71] J. Wenner, Y. Yin, Y. Chen, R. Barends, B. Chiaro, E. Jeffrey, J. Kelly, A. Megrant, J. Y. Mutus, C. Neill, P. J. J. O’Malley, P. Roushan, D. Sank, A. Vainsencher, T. C. White, A. N. Korotkov, A. N. Cleland, and J. M. Martinis, *Phys. Rev. Lett.* **112**, 210501 (2014).
- [72] M. Schaffry, E. M. Gauger, J. J. L. Morton, and S. C. Benjamin, *Phys. Rev. Lett.* **107**, 207210 (2011).

- [73] S. J. Devitt, A. D. Greentree, A. M. Stephens, and R. Van Meter, *Sci. Rep.* **6**, 36163 (2016).
- [74] K. Saeedi, S. Simmons, J. Z. Salvail, P. Dluhy, H. Riemann, N. V. Abrosimov, P. Becker, H.-J. Pohl, J. J. Morton, and M. L. Thewalt, *Science* **342**, 830 (2013).
- [75] N. Aslam, M. Pfender, P. Neumann, R. Reuter, A. Zappe, F. F. de Oliveira, A. Denisenko, H. Sumiya, S. Onoda, J. Isoya *et al.*, *Science* **357**, 67 (2017).
- [76] A. Gheorghiu and T. Vidick, in *Proceedings of the 2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)* Baltimore, MA (IEEE, Piscataway, NJ, 2019), pp. 1024–1033.
- [77] A. Cojocaru, L. Colisson, E. Kashefi, and P. Wallden, in *Advances in Cryptology – ASIACRYPT 2019*, edited by S. D. Galbraith and S. Moriai (Springer International Publishing, Cham, 2019), pp. 615–645.