Quantum computation capability verification protocol for noisy intermediate-scale quantum devices with the dihedral coset problem

Ruge Lin [1,2] and Weiqiang Wen [3]
[1]*Technology Innovation Institute, Abu Dhabi, United Arab Emirates*
[2]*Departament de Física Quàntica i Astrofísica and Institut de Ciències del Cosmos (ICCUB), Universitat de Barcelona, Martí i Franquès 1, 08028 Barcelona, Spain*
[3]*LTCL, Telecom Paris, Institut Polytechnique de Paris, France*

(Received 20 April 2022; accepted 11 July 2022; published 22 July 2022)

In this article, we propose an interactive protocol for one party (the verifier) holding a quantum computer to verify the quantum computation power of another party's (the prover) device via a one-way quantum channel. This protocol is referred to as the dihedral coset problem (DCP) challenge. The verifier needs to prepare quantum states encoding secrets (DCP samples) and send them to the prover. The prover is then tasked with recovering those secrets with a certain accuracy. Numerical simulation demonstrates that this accuracy is sensitive to errors in quantum hardware. Additionally, the DCP challenge serves as a benchmarking protocol for locally fully connected quantum architecture and aims to be performed on current and near-future quantum resources. We conduct a 4-qubit experiment on one of the IBM Q devices.

DOI: 10.1103/PhysRevA.106.012430

I. INTRODUCTION

In 2019, Google succeeded in reaching quantum supremacy with their Sycamore processor [1]. However, it remains a long way to a fully functioning quantum computer. At this moment, only noisy intermediate-scale quantum (NISQ) [2] devices are available, and a method is needed to verify their computing power.

Currently, instead of computation capability, random circuit sampling and cross-entropy benchmarking [3,4] are primarily concerned with testing the quantum property of the device. It is desirable to have a performance test on quantum hardware, proving to a verifier and unable to falsify. Recent works [5–8] demands a classical verifier. In particular, they rely on the hardness of the learning with errors (LWE) problem and needs thousands of qubits, which is not applicable to present quantum hardware.

This test should be designed based on two principles: dynamic enough to adapt various processors and friendly to NISQ devices, which can be directly applied in an experiment. To be dynamic, we focus on locally fully connected (LFC) quantum architecture. LFC means that the chip consists of m unit cells of $n+1$ qubits, with $m \geqslant 2$ and $n \geqslant 1$. Within each cell, $n+1$ qubits are fully connected, and each cell has a leader qubit, m leader qubits are fully connected. LFC shares many similarities with Chimera and Pegasus topologies in quantum annealing processor D-Wave [9]. Notice that, in reality, hardware for gate-based quantum computing rarely follow this geometry, but SWAP gates can be applied. A test based on LFC structure can cover any quantum chip with the number of qubits $\geqslant 4$ and not prime. Moreover, a quantum device should pass a test based on LFC architecture to demonstrate its potential for fully connected circuits, such as the Shor algorithm [10] and Grover algorithm [11]. Furthermore, for applying to NISQ devices, the test should contain only shallow circuits and not rely on quantum memory.

Nowadays, classical simulation programs for quantum circuits such as Cirq [12], Qiskit [13], and Qibo [14,15] can mimic noisy or noiseless quantum devices for up to dozens of qubits on classical hardware. It is hard to distinguish between a quantum device and a simulator around this scale. Therefore, we can consider introducing a quantum verifier. In previous works [16,17], the quantum verifier(s) is (are) asked to witness particular states generated by the prover. However, in Ref. [16], the target state is too complicated for NISQ devices. Also, the method provided in Ref. [17] is designed for sparse quantum chips with certain geometry restrictions.

This article presents the DCP challenge, a verification protocol of quantum computation capability, requiring a quantum verifier and a one-way quantum channel from the verifier to the prover. It is an interactive protocol for Alice, the verifier holding a $(n+1)$-qubit quantum device, to test the quantum computing power of Bob, the prover holding a $m(n+1)$-qubit device, which runs on the LFC architecture. In contrast to the method in Ref. [17], where the verifier needs more than half of the qubits of the prover, the DCP challenge only needs a fraction, implying a quantum channel with fewer qubits. In particular, Alice needs to provide simple quantum states (DCP samples) as a superposition of two possibilities, which can be easily verified by measurement, and send them to Bob, who solves the problem essentially using Quantum Fourier transform on n qubits. The advantage of the prover being the receiver of the quantum states is that the measurement error is also tested. We have also performed simulations of our protocol. On one side, we show that, in the error-free model, the quantum computing capability of the prover can

be successfully verified with overwhelming probability. On the other side, in the noisy setting simulation, our protocol is shown to be very sensitive to the presence of errors, while it is still shown to be robust up to some restricted errors. This property also makes the DCP challenge a promising benchmarking protocol when preparing samples and solving the problem are performed by the same quantum device.

II. PRELIMINARY

A. Dihedral coset problem

The dihedral coset problem has been a fundamental problem in studying the quantum hardness of the hidden subgroup problem over (non-Abelian) dihedral group in the last two decades [18–23]. Informally, it asks to recover the hidden subgroup of a dihedral group given random cosets of the hidden subgroup as superposition. A dihedral group is generated by reflections and rotations of a E-gon (regular polygon with E edges). The first part of the superposition encodes the reflection. From now on, we call it the reflection qubit. The second part encodes the rotation. Normalization is omitted for every equation in this article.

Definition 1. (Dihedral coset problem, DCP) The input of the DCP_E^ℓ with modulus E consists of ℓ samples. Each sample is a quantum state of the form

$$|\psi_{x,s}\rangle = |0\rangle\,|x\rangle + |1\rangle\,|(x+s) \bmod E\rangle\,, \tag{1}$$

stored in $1+\lceil \log_2 E\rceil$ qubits, where $x \in \{0,1,\ldots,E-1\}$ is randomly and uniformly selected for each sample and $s \in \{0,1,\ldots,E-1\}$ is fixed throughout all the states. The task is to output the secret s.

The problem is hypothesized to be unsolvable by direct measurement on the computational basis, which means the best-known classical solution is a random guess. We could not obtain x and $(x+s)$ mod E at the same time.

The DCP is known to be solvable in subexponential time while given a subexponential number of samples [24–26]. These solving algorithms were designed with different optimization targets. So far, Kuperberg's algorithm [24] achieves a smallest running-time $2^{O(\sqrt{\log_2(E)})}$ but requires $2^{O(\sqrt{\log_2(E)})}$ space while Regev's [25] variant requires only a polynomial (in $\log_2(E)$) space but its running time is slightly worse as $2^{O(\sqrt{\log_2(E)\log_2(\log_2(E))})}$.

Both of them start by running quantum Fourier transform on the given DCP samples (except the reflection qubit) and measure them, which naturally possess a LFC structure. The main drawback of these two algorithms is that some quantum states need to be maintained throughout the whole process.

In this work, given the constraints of current quantum computing devices (e.g., NISQ), the circuit depth and quantum memory required by both Kuperberg's and Regev's algorithms cannot be satisfied. Therefore, we consider a slightly different variant of the DCP problem and algorithm by minimizing circuit depth and limiting quantum registers.

Before introducing them, we first recall the quantum Fourier transform.

Definition 2. (Quantum Fourier transform, QFT) The quantum Fourier transform on the computational basis $|0\rangle,\ldots,|N-1\rangle$ of an n-qubit state is defined to be a linear operator with the following action on the basis states,

$$|j\rangle \mapsto \sum_{k=0}^{N-1} \omega_N^{jk}\,|k\rangle\,, \tag{2}$$

where $\omega_N = e^{2\pi i/N}$.

The evaluation time of QFT is $O(n^2)$ [27, Section 5.1].

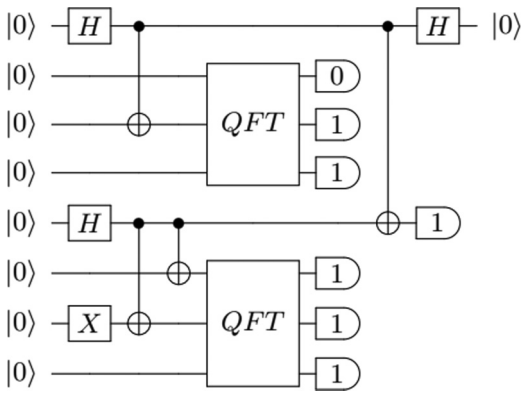

FIG. 1. A toy circuit for $m=2$, $n=3$, and $s=2$. The first four qubits correspond to the state when $x=0$, $|\psi_{0,2}\rangle = |0\rangle\,|000\rangle + |1\rangle\,|010\rangle$ and the second four qubits correspond to the state $x=2$, $|\psi_{2,2}\rangle = |0\rangle\,|010\rangle + |1\rangle\,|100\rangle$. The collision after QFT with $\hat{x}_1 = 3$ and $\hat{x}_2 = 7$ is chosen randomly.

B. New variant

Currently, NISQ devices have limited registers, low coherence time, low relaxation time, and imperfect gate implementation. They can only efficiently perform shallow circuits. Therefore, we slightly modify the DCP adapting this status. First, we set $E = N = 2^n$. Then, instead of solving the secret s, we ask to solve the parity of s, which represents the same order of complexity. Figures 1 and 2 are two example circuits of this new variant.

Alice can prepare the state $|\psi_{x,s}\rangle$ with only H, X and CNOT (which are the Clifford gates) and it takes $O(n)$ gates. She can

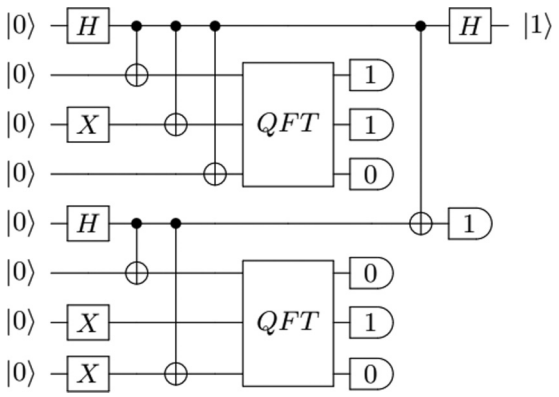

FIG. 2. A toy circuit for $m=2$, $n=3$, and $s=3$, first four qubits correspond to the state when $x=2$, $|\psi_{2,3}\rangle = |0\rangle\,|010\rangle + |1\rangle\,|101\rangle$ and the second four qubits correspond to the state $x=3$, $|\psi_{3,3}\rangle = |0\rangle\,|011\rangle + |1\rangle\,|110\rangle$. The collision after QFT with $\hat{x}_1 = 6$ and $\hat{x}_2 = 2$ is chosen randomly.

verify the accuracy of $|\psi_{x,s}\rangle$ by measuring it. Notice that, for a total of N^2 combinations of x and s, there are a total of N^2 combinations of X and CNOT gates. However, we do not have a direct relation between x, s and each of these gates.

To solve the parity of s within m cells of $n+1$ qubits, using the shallowest circuit currently known, we use a highly simplified version of Kuperberg's algorithm [24], and name it *ParitySolve*.

Bob performs QFT on the last n qubits and measures them. Here we highlight that he always needs more computation resources and operation steps than Alice; otherwise, it would not be a challenge.

After QFT is applied on the last n qubits of the DCP sample, the total state becomes

$$\sum_{k=0}^{N-1} \left(|0\rangle + \omega_N^{ks}\,|1\rangle\right)|k\rangle\,, \quad \omega_N = e^{2\pi i/N}. \tag{3}$$

Bob then checks the measurements after QFT. He needs a pair of measurements that the most significant qubit is different and the rest are identical. We call it a collision. If he does not have it, he resets all registers to $|0\rangle$ and starts another *ParitySolve*.

After the measurement, the reflection qubit becomes

$$|\phi_{\hat{x},s}\rangle = |0\rangle + \omega_N^{\hat{x}s}\,|1\rangle \tag{4}$$

for some uniform distributed random measured $\hat{x} \in \{0, 1, \ldots, N-1\}$. Assume that Bob has a collision, $\hat{x}_1$ and $\hat{x}_2$; then the tensor product between $|\phi_{\hat{x}_1,s}\rangle$ and $|\phi_{\hat{x}_2,s}\rangle$ gives

$$|0,0\rangle + \omega_N^{\hat{x}_1 s}\,|1,0\rangle + \omega_N^{\hat{x}_2 s}\,|0,1\rangle + \omega_N^{(\hat{x}_1+\hat{x}_2)s}\,|1,1\rangle\,. \tag{5}$$

Bob performs a CNOT gate on these two reflection qubits. The state becomes

$$|0,0\rangle + \omega_N^{\hat{x}_1 s}\,|1,1\rangle + \omega_N^{\hat{x}_2 s}\,|0,1\rangle + \omega_N^{(\hat{x}_1+\hat{x}_2)s}\,|1,0\rangle\,. \tag{6}$$

Then he measures the target qubits, with $\frac{1}{2}$ probability he can measure $|1\rangle$. If $|0\rangle$ is measured, he needs to reset all registers to $|0\rangle$ and start another *ParitySolve*. After $|1\rangle$ on the target qubit is measured, the controlled qubit becomes

$$|0\rangle + \omega_N^{(\hat{x}_1-\hat{x}_2)s}\,|1\rangle = |0\rangle + (-1)^s\,|1\rangle\,. \tag{7}$$

The equality holds because if $\hat{x}_1$ and $\hat{x}_2$ is a collision, then $\hat{x}_1 - \hat{x}_2 \mod N = \frac{N}{2}$.

Finally, the parity of s lies inside the phase of $|1\rangle$. Bob can solve it by applying an H gate on the remaining qubit and measuring it. If the result is $|0\rangle$, then s is even. He replies 0 to Alice. Otherwise, s is odd. He replies 1. The solution is completely correct if the quantum channel and devices are noiseless.

C. Measurement method

We have found a new method to solve the parity of s when $E = N$. As indicated previously, s is unsolvable by direct measurement on the computational basis. However, it is possible to have a minor advantage with single-qubit measurement on a different basis, equivalent to applying one layer of same single-qubit unitary gates before measuring on the computational basis, as shown in Fig. 3.

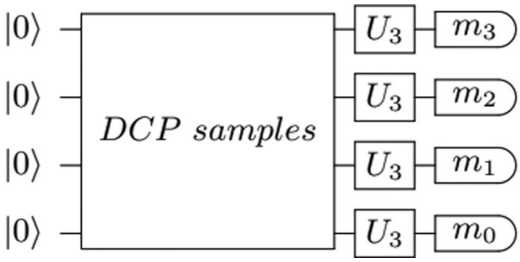

FIG. 3. A toy circuit for the measurement method for $n = 3$.

The third general unitary gate U_3 can be written into

$$U_3(a,b,c) = \begin{pmatrix} e^{-i(b+c)/2}\cos\left(\frac{a}{2}\right) & -e^{-i(b-c)/2}\sin\left(\frac{a}{2}\right) \\ e^{i(b-c)/2}\sin\left(\frac{a}{2}\right) & e^{i(b+c)/2}\cos\left(\frac{a}{2}\right) \end{pmatrix}, \tag{8}$$

with $a \in [0, \pi)$, $b \in [0, 4\pi)$, and $c \in [0, 2\pi)$. When $a = \frac{\pi}{2}$, $b = 0$, and $c = \pi$, U_3 is an H gate (with a global phase).

The parity of s can be distinguished with $a = \frac{\pi}{2}$, $c \in \{0, \pi\}$, and an arbitrary b. The measurement is read as $M = m_0 2^0 + m_1 2^1 + \cdots + m_n 2^n$. We denote as M_{non} the value not able to measure, M_{even} as the value that is only measurable when s is even, and M_{odd} as the value that is only measurable when s is odd. Therefore, M_{even} and M_{odd} can be considered as the feature values of the parity of s, which can be determined when one of them appears. When $c = 0$, $M_{\text{non}} = N - 1$, $M_{\text{even}} = 2N - 2$, and $M_{\text{odd}} = N - 2$. When $c = \pi$, $M_{\text{non}} = N$, $M_{\text{even}} = 1$, and $M_{\text{odd}} = N + 1$. The probability of measuring M_{even} or M_{odd} is $1/N$. For example, Bob can measure every DCP samples on an H basis, if any of them is 1, s is even, or if any of them is $N + 1$, then s is odd. This new technique is found by brute-force simulation for $n < 10$ and conjectured to be valid for lager n; it potentially leads to a solution of the DCP with only measurement.

III. PROTOCOL

In this section, we use an example to demonstrate the full protocol of the DCP challenge. A diagram is in Fig. 4. Alice is the verifier who has a quantum computer. Bob is the prover who declares having a quantum computer and wants to prove his quantum computation capability to Alice. To perform the challenge, Alice needs to have $n + 1$ qubits to prepare DCP samples, and Bob needs $m(n + 1)$ qubits to solve them. Before starting the challenge, they agree on the choice of m, n, the number of iterations t, and the number of repetitions r. In every repetition, there are t iterations.

In the first stage, Alice uniformly selects two numbers $x \in \{0, 1, \ldots, N-1\}$ and $s \in \{0, 1, \ldots, N-1\}$, both of which she keeps secret. She generates $|\psi_{x,s}\rangle$ with x and s. Alice sends m DCP samples one by one to Bob via a quantum channel in every iteration. Bob stores them in his m cells of registers and attempts to solve the parity of s using *ParitySolve*. In the first repetition, every sample has the same $s = s_1$ and a different random x. When Bob could not have a result after t iterations, he randomly guesses a 0 or 1. Here we have $\ell = mt$. Then Alice starts new repetitions, each time with a different s until she completes the challenge with a secret s_r. The number of repetitions r can be any number large enough to reflect Bob's probability of success, also called the accuracy **p**. In

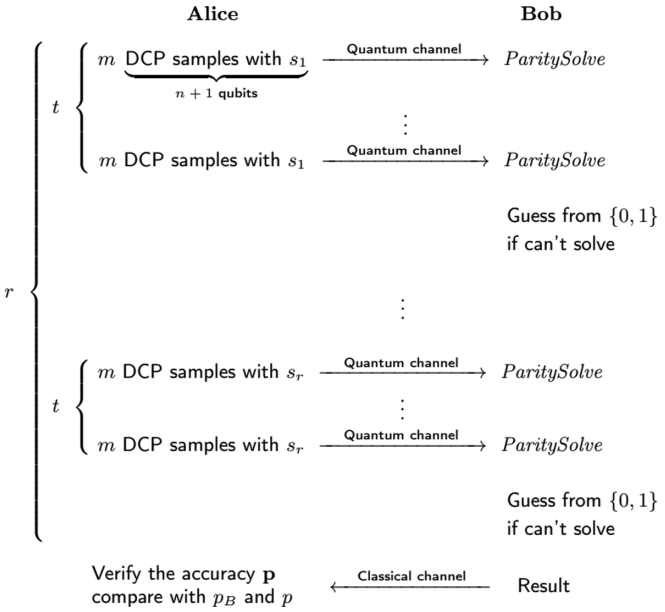

FIG. 4. The DCP challenge in a diagram.

this article, we use bold letters for a simulated or experimental outcome. Bob sends his results in a bit string back to Alice via a classical channel.

Finally, Alice verifies Bob's probability of success $\mathbf{p}$. If Bob has an error-free device, his accuracy is expected to be p, which can be calculated or simulated numerically. Furthermore, the choice of m, n, t depends on the number of qubits from both parties, the maximum transmission of the quantum channel, and the difference $p - p_B$, where p_B is the expected accuracy of performing the measurement method. Details are shown in Appendix A. Moreover, the presence of noise also implies that the loss of computing power will reflect in the accuracy. Since NISQ hardware is not likely to be error-free, we expect to have $p \geqslant \mathbf{p} \geqslant \frac{1}{2}$. The quantum computation capability of Bob's processor is verified with $\mathbf{p} > p_B$. When $\mathbf{p} \approx p$, the device is qualified for a stricter test.

The numerical simulation of this verification protocol can be found in Appendix B.

IV. POSSIBLE CHEATING METHODS

There is not any known method to cheat the DCP challenge without a quantum computer of better performance unless a new algorithm is found, reaching the expected accuracy p with a shallower circuit than *ParitySolve*. However, it is possible to cheat when having such a device and obtain $\mathbf{p} > p$ with even less than $m(n+1)$ qubits. Two methods are outlined below.

The first method assumes Bob's quantum computer has a longer relaxation time, such that those unmeasured qubits do not quickly return to $|0\rangle$. Instead of receiving m samples and erasing them all if he could not find a collision, he can erase one sample each time and receive another one. Once a collision is found, and after a CNOT gate, he measures $|0\rangle$, he can erase both of them and receive another two samples. This method wastes fewer DCP samples and leads to a larger probability of success.

The second method assumes that Bob's quantum computer has less noise to perform SWAP gates efficiently. Bob can move reflection qubits to the register of measured qubits after resetting them to $|0\rangle$; therefore, he has more room to store the reflection qubit of every sample. This method increases the probability of collision, thus increasing the accuracy.

Both methods rely on a more enhanced quantum computation capability, so they should not be considered cheating. Once quantum computers become powerful enough to "cheat" accurately, the "cheating method" can become the standard protocol. All Alice needs to do is to reduce t or raise p accordingly. There are more methods to reach $\mathbf{p} > p$ when Regev's [20,25] and Kuperberg's [24,26] complete algorithms can be performed.

Nonetheless, we can prohibit all these cheating methods by setting the time interval between iterations long enough to bypass the possible relaxation time for the near future but still short enough for an experiment. For example, we can set the interval as one second (the Sycamore processor is on the order of μs), so the device loses all memory of the previous iteration. In this way, Bob has no cheating method unless he has a quantum computer with an extremely longer relaxation time.

V. OTHER APPLICATIONS

The DCP challenge has more applications than a verification mechanism of quantum computation power. Here are some examples.

This protocol can be used to benchmark a quantum computer. It is a straightforward method for evaluating the performance of NISQ hardware. Gate-based quantum devices are usually manufactured by using various techniques and thus have distinct connection geometries and parameters. Even when they have the same number of qubits, direct comparison of their computation capability is difficult. The readout of the DCP challenge is the numerical accuracy $\mathbf{p}$ after applying a large amount of simple predefined circuits. It provides us a quantitative insight into a quantum computer, which can be regarded as a score. The numerical simulation of the use of the DCP challenge as a benchmarking protocol is in Appendix C. The smallest instance using the DCP challenge for benchmarking only requires four qubits in a line. In this case, QFT is an H gate. We perform this experiment on the first four qubits of 5-qubit IBM Q processor *ibmq_manila* [28], as detailed in Appendix D.

The DCP challenge helps benchmark a quantum channel. If Bob tests on his processor and has a probability of success $\mathbf{p}$. They should anticipate a comparable level of accuracy when Alice transmits the challenge to Bob. This protocol can also help to spot eavesdropping on a quantum channel. If Alice and Bob used to have a probability of success $\mathbf{p}$, suddenly the probability has dropped. If they are both certain there are no technical issues with the channel or their hardware, then perhaps Eve is intercepting. She steals some DCP samples from the channel, and when she puts some fake samples back, she is unaware of the parity of s. Even if Eve can also intercept the classical channel from Bob to Alice and change the result, she has no method to raise the probability unless she replaces Bob completely.

The DCP challenge is a very elemental puzzle game for NISQ devices. Its numerous potential uses remain unexplored.

VI. GENERALIZATION

Here we give a more general interactive verification protocol. The central assumption is a question encoding a secret in ℓ quantum states (samples), a quantum algorithm solves the secret with a probability p, a classical or measurement algorithm solves the secret with a baseline probability p_B. The key to verifying the quantum computing power lies in the inequality $p > p_B$. Alice sends a fixed amount ℓ of samples in each repetition. Bob needs to solve the secret and sends his result back to Alice, and she verifies the accuracy and compares it with p_B. By increasing the number of repetitions, Alice can confirm Bob's quantum computation capability. The protocol can be optimized by lowering the number of qubits and simplifying Alice's process of preparing the samples, creating a computation imbalance between the verifier and prover.

The DCP is chosen with the extra advantage that its current solutions naturally process a LFC structure. Moreover, even within the DCP framework, our readers are free to design new protocols for more advanced quantum computers; for example, Alice can ask about the full s instead of its parity or she can decide another $E < N$. New algorithm for solving the DCP or its different variants will be invented in the future and the protocol will be updated accordingly. However, the subexponential quantum complexity of the DCP remains relatively solid since it secures the hardness of the LWE problem [29].

VII. CONCLUSIONS

In this article, the DCP challenge has been proposed. Its computation has been shown, numerical simulations have been done, and different cheating strategies have been evaluated. Other applications have been described and a generalization has been produced. Our readers may perceive it as a quantum game rather than a methodology for confirming quantum processing capacity. Its rules are flexible and can be adapted to different situations.

The DCP challenge is designed for NISQ devices and is aimed to serve temporarily. One day, when quantum computers are powerful enough to outperform classical computers in various tasks such as factoring big integers, the protocol will lose its purpose as a proof of computation. Nevertheless, its other applications remain.

ACKNOWLEDGMENTS

We acknowledge Stavros Efthymiou and Sergi Ramos-Calderer for useful support in Qibo, and Ingo Roth for useful suggestions. Also, we acknowledge the use of IBM Quantum services for this work.

APPENDIX A: ANALYTICAL PROBABILITY

To obtain the ideal probability of success p, we need to determine $k_{\text{collision}}$, the probability of *not* having a collision in m cells among N possibilities. We can use the formula of the "Birthday Paradox" to provide its upper bound and lower bound.

k_{lower} is a direct application of the formula to calculate the probability of *not* having two identical elements when choosing m times out of N possibilities,

$$k_{\text{lower}} = \prod_{i=0}^{m-1} \frac{N-i}{N}. \tag{A1}$$

In order *not* to have a pair of identical elements, each choice must be different. However, in the case of *not* having a collision, we can keep the same choice. So it is slightly easier to have a collision than to have a pair of identical elements.

For calculating k_{upper}, we first consider the probability of *not* having two identical elements when choosing m times out of $\frac{N}{2}$ possibilities (for $n-1$ qubits except the most significant one). Then, we take into account that the last qubit is different. We have

$$k_{\text{upper}} = \frac{1}{2} + \frac{1}{2} \prod_{i=0}^{m-1} \frac{N/2-i}{N/2}. \tag{A2}$$

k_{upper} ignores the case of having more than one pair of identical elements when we are considering the first $n-1$ qubits.

We have

$$k_{\text{upper}} > k_{\text{collision}} > k_{\text{lower}}. \tag{A3}$$

But each collision has only a probability of $1/2$ to solve the parity of s, so the chance of *not* being able to solve after t iterations is

$$2 - 2p = \left(\frac{1 + k_{\text{collision}}}{2}\right)^t. \tag{A4}$$

Finally, when Bob is unable to solve, he has to randomly guess a result, which has the probability $1/2$ to be correct. So in total, his probability of success is

$$p = \left[2 - \left(\frac{1 + k_{\text{collision}}}{2}\right)^t\right] \Big/ 2. \tag{A5}$$

And we have the relation,

$$p_{\text{upper}} > p > p_{\text{lower}}, \tag{A6}$$

with

$$p_{\text{upper}} = \left[2 - \left(\frac{1 + k_{\text{lower}}}{2}\right)^t\right] \Big/ 2, \tag{A7}$$

and

$$p_{\text{lower}} = \left[2 - \left(\frac{1 + k_{\text{upper}}}{2}\right)^t\right] \Big/ 2. \tag{A8}$$

Although we do not have the analytical expression of p, we can obtain it numerically. We can generate m random bit strings of length n and search for a collision. By repeating this procedure, we can have the numerical $k_{\text{collision}}$ and use it for calculating p.

To have a given p, we can have an estimation of t,

$$t \sim \frac{\ln(2-2p)}{\ln\left(\frac{1+k_{\text{lower}}}{2}\right)}. \tag{A9}$$

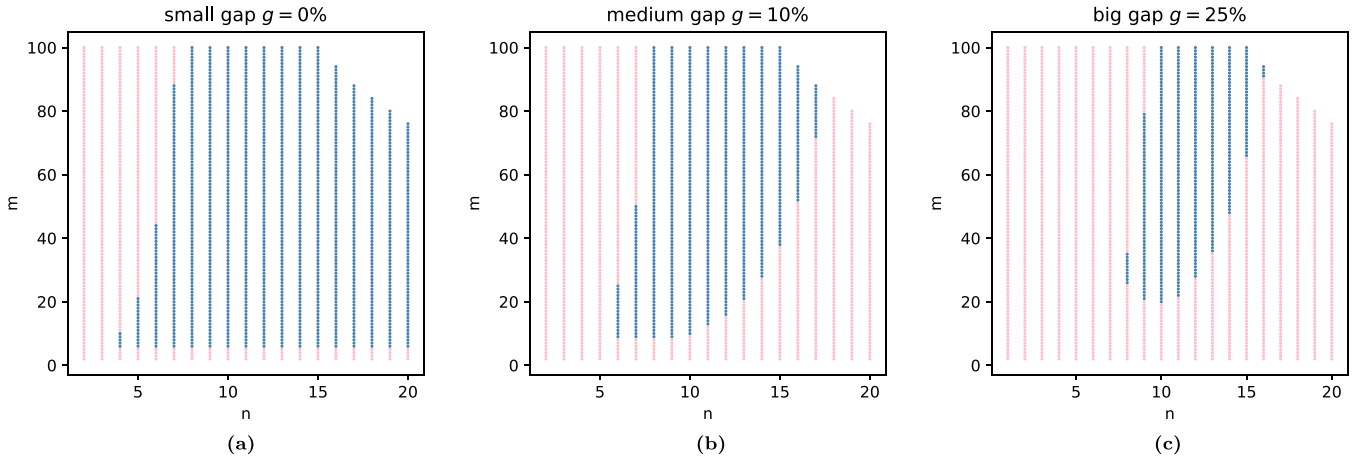

FIG. 5. Combinations of m and n that we can have $p_{\text{upper}} - p_B > g$ with a $t < 25$ are plotted in blue (darker). These three figures show the situation of a small gap $g = 0\%$, a medium gap $g = 10\%$, and a big gap $g = 25\%$ for less than 1600 qubits. (a) Smallest instance: $m = 6$, $n = 4, t = 1$ with $p_B = 66.05\%$ and $p_{\text{upper}} = 66.40\%$. (b) Smallest instance: $m = 9, n = 6, t = 4$ with $p_B = 71.64\%$ and $p_{\text{upper}} = 81.73\%$. and (c) Smallest instance: $m = 21, n = 9, t = 9$ with $p_B = 65.45\%$ and $p_{\text{upper}} = 90.66\%$.

We use k_{lower} instead of k_{upper} because it is numerically closer to $k_{\text{collision}}$. The choice of t is flexible, but setting it too large is not just a waste of resources: if Bob can solve the parity of s multiple times within t iterations and take the majority result, the protocol becomes less sensitive to error.

Here we consider only the case when measuring $|0\rangle$ after CNOT; we reset all registers and pass directly into the next iteration, ignoring the fact that there might be another collision in the same group. This is because the probability of having two collisions in the same group is significantly lower than having one, and this difference vanishes in p with the exponent t. We would like to keep the problem as simple as possible. Also, we would like to maintain the shallowest circuit. This situation is easy to simulate classically.

In some situations, especially when $N \gg m$, t can be too large (>1000) to fit in an experiment. In this case, we can set a lower p and increase r for preciseness. The rules of this protocol are adjustable. For a numerical indication, for a device with 1000 qubits, if we set $n = 19$ and $m = 50$, t should be $\geqslant 785$ to have $p > 80\%$ according to our protocol. Therefore, our protocol is still feasible for the advanced NISQ-era. At that time, quantum computers might be capable of "cheating" accurately (as in Sec. IV), we can even set a lower t.

There is a probability of $2 - 2p$ that a random guess needs to be made. Using the standard deviation formula, we can calculate the fluctuation from the expectation p:

$$\sigma_p = \sqrt{\frac{1-p}{2r}}. \tag{A10}$$

This formula of standard deviation is also valid for experimental $\mathbf{p}$. When $\mathbf{p} = \frac{1}{2}$, the fluctuation becomes the same as flipping a coin r times.

For the measurement method, the probability of measuring M_{even} or M_{odd} is $1/N$ and the probability of measuring both of them in the same repetition is zero. Therefore, the probability of measuring none of them also means not being able to solve the parity of s within mt samples and is

$$2 - 2p_B = \left(\frac{N-1}{N}\right)^{mt}. \tag{A11}$$

The expected accuracy is

$$p_B = \left[2 - \left(\frac{N-1}{N}\right)^{mt}\right] \Big/ 2. \tag{A12}$$

The standard deviation can also be calculated with the Eq. (A10).

From Eqs. (A7) and (A12), we can compare p_{upper}, which is numerically closer to p, and p_B for the difference m, n, and t. For less than 1600 qubits, a minor difference is shown in Fig. 5(a) for an indication. The number 1600 is the most up-to-date lower bound of a verification protocol with a classical verifier [8], ideally a quantum verifier is no longer

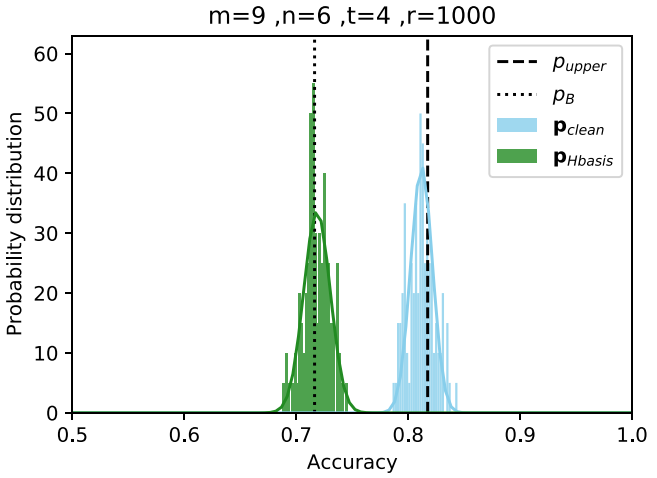

FIG. 6. Normalized probability distribution of $\mathbf{p}_{\text{clean}}$ and $\mathbf{p}_{\text{Hbasis}}$ over 100 trials, plotted with normal distribution $\mathcal{N}(\overline{\mathbf{p}_{\text{clean}}}, \sigma_{\mathbf{p}_{\text{clean}}})$ and $\mathcal{N}(\overline{\mathbf{p}_{\text{Hbasis}}}, \sigma_{\mathbf{p}_{\text{Hbasis}}})$.

TABLE I. Here we assume $m = n + 1$, so there are in total m^2 qubits, and t is the minimal number of iterations for $p_{\text{upper}} > 80\%$. $\mathbf{p}_{\text{clean}}$ and $\mathbf{p}_{\text{error}}$ are results of $r = 10\,000$.

m	3	4	5	6	7	8
t	3	3	4	5	7	9
p_{upper}	83.75%	82.47%	84.18%	83.22%	83.17%	80.62%
p_{lower}	78.90%	76.86%	79.38%	79.59%	80.61%	78.91%
$\mathbf{p}_{\text{clean}}$	81.60%	80.86%	83.05%	82.45%	82.38%	80.69%
$\mathbf{p}_{\text{error}}$	69.02%	65.38%	61.77%	58.74%	55.65%	53.30%

needed after this scale. A distinguishable difference is shown in Fig. 5(b), *ParitySolve* and the measurement method can be distinguished for $r \sim 1000$. A significant difference is shown in Fig. 5(c); in this case, the quantum computation capability of Bob can be verified even with moderated error.

APPENDIX B: NUMERICAL SIMULATION FOR VERIFICATION

One advantage of the DCP challenge is that it is effortless to simulate the whole process classically. Due to the LFC structure and the shallowness of the circuit, instead of simulating the entire circuit of $m(n + 1)$ qubits, we can simulate each DCP sample individually and store the measurement bit string and the state vector of the remaining qubit. Qibo can efficiently simulate a quantum circuit for up to 31 qubits on a laptop. Therefore, it can simulate a DCP circuit for up to $n = 30$ qubits.

If Bob has a quantum chip of $m = 9$ and $n = 6$, Alice can first choose with Fig. 5(b) that $t = 4$. Then she can calculate p_{upper} with Eq. (A7) and p_B with Eq. (A12), or even calculate p using Eq. (A5) with a numerical $k_{\text{collision}}$ to see the probability that she expects. She prepares $r = 1000$ repetitions to challenge Bob, each with $m = 9$ samples of $n + 1 = 7$ qubits and $t = 4$ iterations. In total, she needs to prepare 36 000 DCP samples, and Bob needs to perform about that many QFTs.

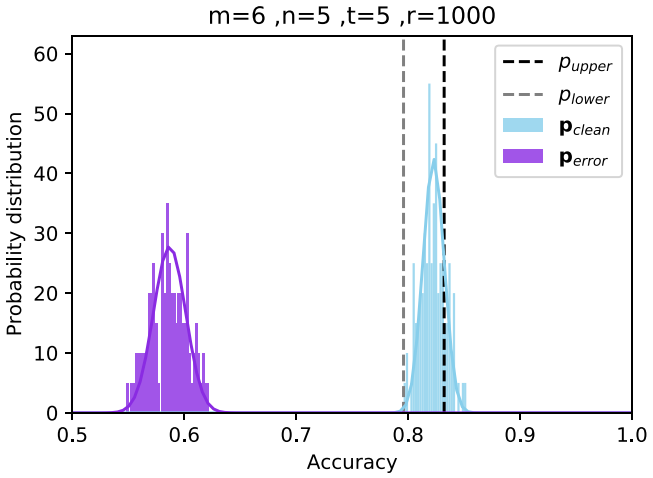

FIG. 7. Normalized probability distribution of $\mathbf{p}_{\text{clean}}$ and $\mathbf{p}_{\text{error}}$ over 100 trials, plotted with normal distribution $\mathcal{N}(\overline{\mathbf{p}_{\text{clean}}}, \sigma_{\mathbf{p}_{\text{clean}}})$ and $\mathcal{N}(\overline{\mathbf{p}_{\text{error}}}, \sigma_{\mathbf{p}_{\text{error}}})$.

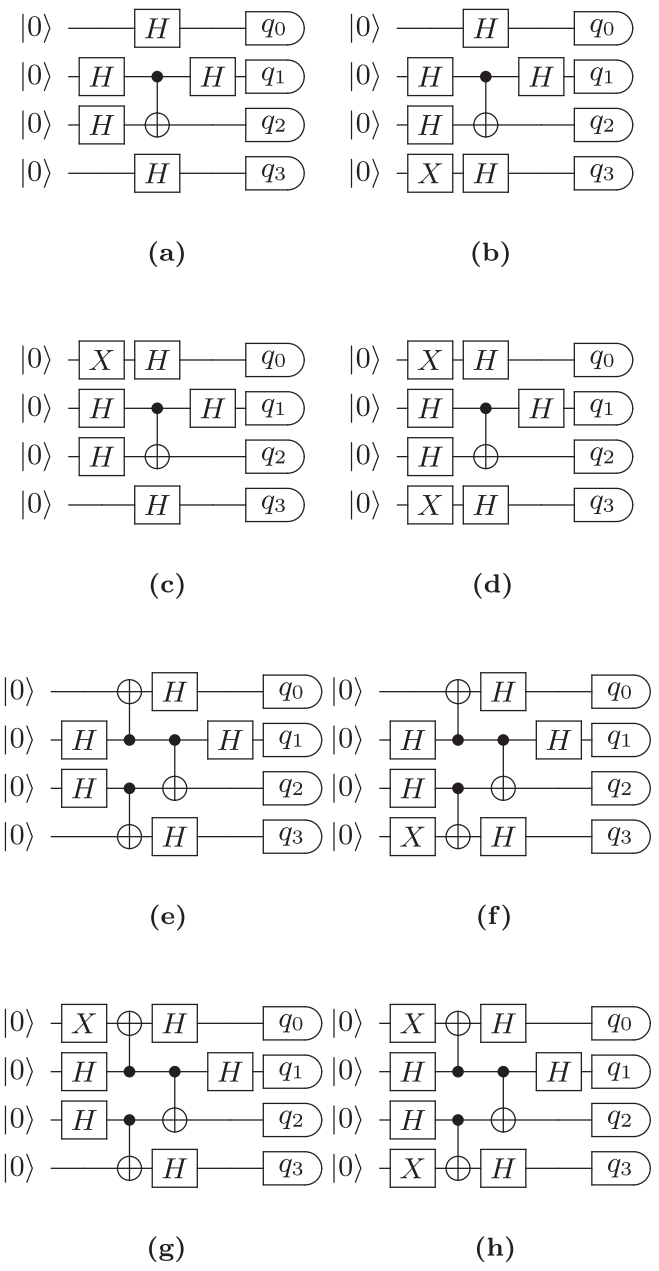

FIG. 8. Eight possible cases for two DCP samples of $n = 1$. (a) Case A, with $s = 0$, $x_0 = 0$ and $x_1 = 0$. (b) Case B, with $s = 0$, $x_0 = 0$ and $x_1 = 1$. (c) Case C, with $s = 0$, $x_0 = 1$ and $x_1 = 0$. (d) Case D, with $s = 0$, $x_0 = 1$ and $x_1 = 1$. (e) Case E, with $s = 1$, $x_0 = 0$ and $x_1 = 0$. (f) Case F, with $s = 1$, $x_0 = 0$ and $x_1 = 1$. (g) Case G, with $s = 1$, $x_0 = 1$ and $x_1 = 0$. and (h) Case H, with $s = 1$, $x_0 = 1$ and $x_1 = 1$.

Finally, Bob sends his 1000 answers back to Alice, verifying his accuracy. Figure 6 is a simulation with Qibo of $\mathbf{p}_{\text{clean}}$, the accuracy of the error-free simulation of *ParitySolve* and $\mathbf{p}_{\text{Hbasis}}$, the accuracy of solving by measuring on the H basis. We have $\overline{\mathbf{p}_{\text{clean}}} = p$ and $\overline{\mathbf{p}_{\text{Hbasis}}} = p_B$. The code is on Github [30]. We consider $r = 1000$ acceptable since it is trivial to distinguish the probability distribution of the clean circuit performing *ParitySolve* and the measurement method despite fluctuation.

TABLE II. Postselected measurements from IBM Q *ibmq_manila* processor. In total there are 5426 shots measuring $q_1 = 0$ and 4425 shots measuring $q_1 = 1$. There are >1000 shots per case, which allows us to reconstruct the DCP challenge with $r = 1000$.

$q_3q_2q_1q_0$	A	B	C	D	E	F	G	H
$\|0101\rangle$	517	638	624	642	89	73	61	78
$\|0111\rangle$	9	9	14	17	603	563	526	560
$\|1100\rangle$	682	575	632	583	56	50	51	75
$\|1110\rangle$	20	14	13	13	477	490	552	545
Error	2.4%	1.9%	2.1%	2.4%	11.8%	10.5%	9.4%	12.2%

APPENDIX C: NUMERICAL SIMULATION FOR BENCHMARKING

We use $p \approx 80\%$ for simulation. It is not too close to 50%, so we can notice the decrease of accuracy due to noise. Our readers can also choose $p \approx 90\%$, which means t needs to be ≈1.75 times greater according to Eq. (A9). Table I is the table of accuracy; a comparison between analytical p_{lower}, p_{upper}, error-free circuit simulation $\mathbf{p}_{\text{clean}}$, and noisy circuit simulation $\mathbf{p}_{\text{error}}$ [30]. For the noise map, we use 1% for bit error and phase error, 3% for measurement error, the choice of errors is inspired by Quantum Computer Datasheet [31] from Google.

In the table, $p_{\text{upper}} > \mathbf{p}_{\text{clean}} > p_{\text{lower}}$ for $m < 8$, which is what we expected. But p can be very close to p_{upper}. As we can see, when $m = 8$, $\mathbf{p}_{\text{clean}}$ is actually larger than p_{upper} because we obtain it through sampling and there is minor fluctuation.

To benchmark a quantum chip with $n = 5$ and $m = 6$, we can first predict with Eq. (A9) that $t = 5$, then calculate p_{upper} with Eq. (A7) and p_{lower} with Eq. (A8), or even calculate p using Eq. (A5) with a numerical $k_{\text{collision}}$, to see the probability that we expect. If we set $r = 1000$, we need to prepare 30 000 DCP samples in total and perform about that many QFTs. Figure 7 is a simulation of $\mathbf{p}_{\text{clean}}$ and $\mathbf{p}_{\text{error}}$ in this case. Comparing with $\mathbf{p}_{\text{clean}}$, the accuracy $\mathbf{p}_{\text{error}}$ shifts towards $1/2$ due to the noise. We consider $r = 1000$ acceptable since it is trivial to distinguish the probability distribution of the clean circuit and the noisy circuit despite fluctuation.

APPENDIX D: IBM Q EXPERIMENT

We benchmark the DCP challenge on four superconducting qubits provided by IBM Q quantum computers. Since the IBM Quantum Composer interface does not allow applying gates after measurement, the DCP challenge cannot be directly implemented. We need to perform multiple experiments on every possible configuration then use the output data to reconstruct the DCP challenge.

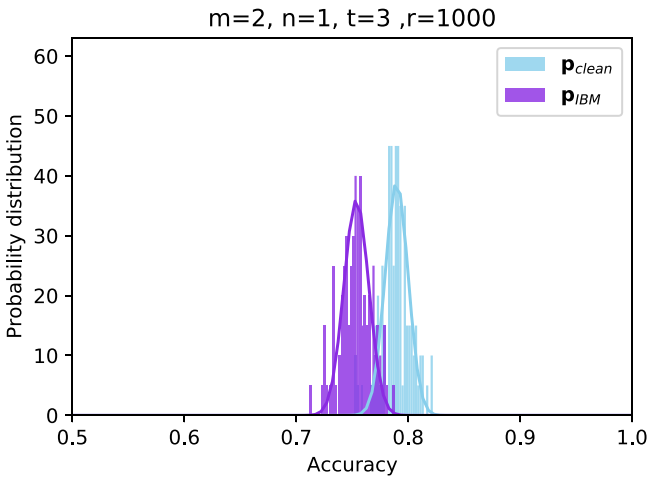

FIG. 9. Normalized probability distribution of $\mathbf{p}_{\text{clean}}$ and $\mathbf{p}_{\text{IBM}}$ (reconstructed from experimental data) over 100 trials, plotted with normal distribution $\mathcal{N}(\overline{\mathbf{p}_{\text{clean}}}, \sigma_{\mathbf{p}_{\text{clean}}})$ and $\mathcal{N}(\overline{\mathbf{p}_{\text{IBM}}}, \sigma_{\mathbf{p}_{\text{IBM}}})$. The accuracy $\mathbf{p}_{\text{IBM}}$ is very close to $\mathbf{p}_{\text{clean}}$. A larger r is needed to distinguish them. The quantum device is considered promising.

When $n = 1$ and $m = 2$, there are in total $2^3 = 8$ possible cases for two DCP samples, as shown in Fig. 8. Notice that the reflection qubits are in the center to avoid SWAP gates. We perform five tests of each case on the first four qubits of 5-qubit quantum processor *ibmq_manila*, which has a linear architecture. By default, each test consists of 1024 shots. Then we select the measurements that have a collision and the result is $|1\rangle$ on the target qubit of CNOT gate, $q_0 \neq q_3$ and $q_2 = 1$. If the device is noiseless, we should have $q_1 = s$.

The result is in Table II. We can see that the error when $s = 1$ is more significant since the circuit has more CNOT gates for preparing DCP samples. The difference will decrease with larger n. Eventually, the essential gate-error will be on the QFTs or SWAP gates depending on the structure of the device.

Furthermore, due to the imbalanced measurement error [31], it is more likely to measure $|0\rangle$ than $|1\rangle$ on a current quantum processor. The same situation can also be caused by the low relaxation time. If Alice chooses s uniformly, Bob is very likely to have more 0 than 1 in his result, and he can have a rough estimation of his performance. However, this extra information does not allow him to cheat since he does not know which 0 should be replaced by 1.

We use the erroneous data to reconstruct the DCP challenge [30], the result is shown in Fig. 9. The performance of *ibmq_manila* is not perfect but satisfying. Our reader can also use the DCP challenge to benchmark other processors of IBM Q, such as *ibmq_santiago*.

[1] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell *et al.*, Quantum supremacy using a programmable superconducting processor, Nature (London) **574**, 505 (2019).

[2] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[3] S. Mullane, Sampling random quantum circuits: A pedestrian's guide, arXiv:2007.07872.

[4] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J Bremner, J. M. Martinis, and H. Neven, Characterizing quantum supremacy in near-term devices, Nat. Phys. **14**, 595 (2018).

[5] O. Regev, On lattices, learning with errors, random linear codes, and cryptography, J. Assoc. Comput. Mach. **56**, 1 (2009).

[6] Z. Brakerski, P. Christiano, U. Mahadev, U. Vazirani, and T. Vidick, A cryptographic test of quantumness and certifiable randomness from a single quantum device, J. Assoc. Comput. Mach. **68**, 1 (2021).

[7] Z. Brakerski, V. Koppula, U. Vazirani, and T. Vidick, Simpler proofs of quantumness, arXiv:2005.04826.

[8] D. Zhu, G. D. Kahanamoku-Meyer, L. Lewis, C. Noel, O. Katz, B. Harraz, Q. Wang, A. Risinger, L. Feng, D. Biswas *et al.*, Interactive protocols for classically-verifiable quantum advantage, arXiv:2112.05156.

[9] N. Dattani, S. Szalay, and N. Chancellor, Pegasus: The second connectivity graph for large-scale quantum annealing hardware, arXiv:1901.07636.

[10] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM Rev. **41**, 303 (1999).

[11] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the Twenty-Eighth Annual ACM Symposium On Theory of Computing* (ACM Press, New York, 1996), pp. 212–219.

[12] Cirq, https://github.com/quantumlib/cirq.

[13] Qiskit, https://github.com/qiskit/qiskit.

[14] S. Efthymiou, S. Ramos-Calderer, C. Bravo-Prieto, A. Pérez-Salinas, D. García-Martín, A. Garcia-Saez, J. I. Latorre, and S. Carrazza, Qibo: A framework for quantum simulation with hardware acceleration, Quantum Sci. Technol. **7**, 015018 (2021).

[15] Qibo, https://github.com/qiboteam/qibo.

[16] J. F. Fitzsimons, M. Hajdušek, and T. Morimae, Post Hoc Verification of Quantum Computation, Phys. Rev. Lett. **120**, 040501 (2018).

[17] Y. Takeuchi, Y. Takahashi, T. Morimae, and S. Tani, Divide-and-conquer verification method for noisy intermediate-scale quantum computation, Quantum **6**, 758 (2022).

[18] M. Ettinger and P. Høyer, On quantum algorithms for noncommutative hidden subgroups, Adv. Appl. Math. **25**, 239 (2000).

[19] M. Grigni, L. Schulman, M. Vazirani, and U. Vazirani, Quantum mechanical algorithms for the non-Abelian hidden subgroup problem, in *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing* (ACM Press, New York, 2001), pp. 68–74.

[20] O. Regev, Quantum computation and lattice problems, SIAM J. Comput. **33**, 738 (2004).

[21] K. Friedl, G. Ivanyos, F. Magniez, M. Santha, and P. Sen, Hidden translation and orbit coset in quantum computing, in *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing* (ACM Press, New York, 2003), pp. 1–9.

[22] S. Hallgren, A. Russell, and A. Ta-Shma, Normal subgroup reconstruction and quantum computation using group representations, in *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing* (ACM Press, New York, 2000), pp. 627–635.

[23] M. Roetteler and T. Beth, Polynomial-time solution to the hidden subgroup problem for a class of non-Abelian groups, arXiv:quant-ph/9812070.

[24] G. Kuperberg, A subexponential-time quantum algorithm for the dihedral hidden subgroup problem, SIAM J. Comput. **35**, 170 (2005).

[25] O. Regev, A subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space, arXiv:quant-ph/0406151.

[26] G. Kuperberg, Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem, arXiv:1112.3333.

[27] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information, 10th Anniversary Edition* (Cambridge University Press, Cambridge, 2010).

[28] IBM Quantum, https://quantum-computing.ibm.com/.

[29] Z. Brakerski, E. Kirshanova, D. Stehlé, and W. Wen, Learning with errors and extrapolated dihedral cosets, in *Public-Key Cryptography–PKC 2018*, edited by M. Abdalla and R. Dahab, Lecture Notes in Computer Science Vol. 10770 (Springer, Cham, 2018), pp. 702–727.

[30] Ruge Lin, https://github.com/gogoko699/dcp-challenge.

[31] Google Quantum AI, quantum computer datasheet.