[04/09/2023]

Centro de Investigación en Tecnologías de la Información y las Comunicaciones (CITIC)

Universidade da Coruña (UDC)

A review of Quantum Computing Benchmarking

Lista de autores

Diego Andrade Canosa
Juan Touriño Domínguez
María José Martín Santamaría
Basilio Fraguela Rodríguez

XUNTA DE GALICIA

Unión Europea

Unha maneira de facer Europa.

CESGA
GALICIA SUPERCOMPUTING CENTER

Despregamento dunha infraestructura baseada en tecnoloxías cuánticas da información que permita impulsar a I+D+i en Galicia.
Apoiar a transición cara a una economía dixital.
Operación financiada pola Unión Europea, a través do FONDO EUROPEO DE DESENVOLVEMENTO REXIONAL (FEDER) como parte da resposta da Unión á pandemia da COVID-19

DATA	AUTHOR	CHANGES	VERSION
15 / 11 / 2022	Diego Andrade Canosa	Initial version and structure	0.1
22 / 11 / 2022	Diego Andrade Canosa	First complete initial version	1.0
23 / 11 / 2022	María José Martín Santamaría	Review of the document	1.1
24 / 11 / 2022	Basilio Fraguela Rodríguez	Review of the document	1.2
25 / 11 / 2022	Juan Touriño Domínguez	Review of the document	1.3

Table of contents

List of figures

List of tables

List of acronyms

QC	*Quantum Computing*
QDC	*Quantum Digital Computer*
NISQ	*Noisy Intermediate-State Quantum*
IEEE	*Institute of Electrical and Electronics Engineers*
TQF	*Total Quantum Factor*
QAQA	*Quantum Approximate Optimization Algorithm*
CLOPS	*Circuit Layer Operations per Second*
IBM	*International Business Machines Corporation*
CLOPS	*Circuit Layer Operations per Second*
CB	*Cycle Benchmarking*
SPAM	*State Preparation and Measurement*
RC	*Randomized Compiling*
QCVV	*Quantum Characterization, Verification, and Validation*
QPU	*Quantum Processing Unit*
RACBEM	*Random Circuit Block-Encoding*
MQT	*Munich Quantum Toolkit*
API	*Application Programming Interface*
OpenQASM	*Open Quantum Assembly Language*

1 Introduction

The concept of Quantum Computing (QC) was coined at the beginning of the 80's decade of the 20th century. This concept encloses several possible variants or implementations of the same concept among which, the quantum digital computer (QDC) is the most popular one. These computers are based on a set of qubits that can be initialized to a certain initial ground state and that can be operated through a set of well-know unitary operators (gates) that usually involve one or two of these qubits. The system is programmed through a circuit composed of a certain combination of these gates, designed to perform the required operation. Soon, QC will cooperate with classical one in hybrid platforms where QC will play the role of an accelerator for some highly demanding kernels. The development of real Quantum platforms nowadays is highly influenced by the existence of high levels of noise that interfere their operation, current quantum platforms are called Noisy Intermediate-State Quantum (NISQ).

This document presents a review of the state of the art work on benchmarking of QC platforms. The document begins with a short overview of the main challenges faced by QC benchmarking (Sec.2). Then, we cover the main performance metrics introduced by the community (Sec. 3). Then, we review the main QC benchmarking works classified as: hardware-level (Sec.4), compiler level (Sec.5), full-stack level (Sec.6) and application level (Sec.7). We finish with a short discussion 8.

2 Quantum Computing Benchmarking Overview

QC community is currently facing the challenging task of setting a benchmarking methodology for a developing technology that is changing rapidly and where there is a big uncertainty regarding the direction towards it will evolve. The previous experience with benchmarking in classical computing shows that you can address this task targeting different levels of the software/hardware stack. This pattern is being repeated with QC technology, but given its early stage of development, most of the benchmarking approaches for QC target the hardware levels, that is, the quality of the qubits. We have performed a revision of the state-of-the-art of QC benchmarking. There are several works where QC benchmarking is correlated with classical computing benchmarking. The authors in [23] discuss the challenges faced by a QC benchmark. The four most important factors are:

- Quantum platforms are more diverse than classical one. Firstly, there are different definitions of QC (from adiabatic to fault-tolerant digital quantum computing). Secondly, the physical substrate used for their implementation is diverse (transmon, ions, neutral atoms, etc.).
- Quantum hardware is less developed than classical one. Most systems have only a few qubits and cannot perform useful applications. Current typical qubit count is less than 100, although there are already systems with up to 433 qubits. Yet, it is not clear whether such high numbers of qubits will be actually useful to address real-world tasks; indeed, some of the most common quantum algorithms require up to millions of qubits [9].
- Quantum algorithms are still under development, and it is unknown what applications will be the most useful.
- Quantum noise is not well understood, and it is difficult to simulate, thus, its characterization is particularly challenging. There are many types of noise in QC, from the one related to single qubit operations to cross-correlation or reading errors. Each device or technology is affected by different kinds of errors, which can therefore have different impacts. In fact, as [23], maybe a Quantum Processors should not be "considered as a monolithic system and their error rates cannot be considered independently".

This last work [23] also study how to effectively assess the impact of quantum noise in a QC platform, using different synthetic benchmarks (as Adder, QFT, quantum classifier or optimisation) with different metrics (depending on the benchmark, it could be fidelity, accuracy or expectation value), and conclude that "quantum noise is more complex and difficult to model than is often assumed and has profound effects everywhere, and can be felt significantly even at higher levels of the system stack. This complicates the task of benchmarking, which is already challenging and full of subtlety for classical systems". A general analysis of QC benchmarking is also performed in [3]. They claim that this topic is currently a research topic, being too soon to establish metrics or standard benchmarks, as we instead already have in classical computing. But these must be defined wisely, because "a bad benchmarking can be worse that no benchmarking at all" (citing expert in classical benchmarking, Mr. Dongarra). Additionally, IEEE has formed a work group [12] to define a quantum benchmark framework, and a draft of their proposal is scheduled for 2023 .

3 Performance Metrics

Every benchmarking methodology uses one or several performance metrics to evaluate different aspects of the performance of a target platform. In the field of QC, some of the metrics that have been defined are:

- Total Quantum Factor: This metric (introduced by [24]) is formulated as

$$TQF = T_1/t_g n_q \tag{1}$$

  T_1 being the average coherence time for all qubits in the processor, t_g the longest gate time in the universal set, and n_q the number of qubits in the chip. They claim that this metric estimates the size of a quantum circuit that can be run in the processor before being affected by decoherence phenomena. However, this metric does not include the topology of the qubits or the fidelity of the gates, making its interest limited for the purpose of comparing the performance of NISQ platforms.

- Q-score: This metric (introduced by [14]) assesses the behaviour of QAOA (a variational quantum approximation algorithm) for MaxCut on a class of random instances. The results are normalized in a smart way to factor out the purely random behaviour and thus compute the true deviation of the algorithm with respect to a simple coin toss. The final score corresponds to the number of qubits up to which the quantum processor's behaviour is significantly different from a random uniform sampling.

- qBAS: This metric (introduced by [1]) targets variational algorithms (it based on a data-driven quantum circuit learning). It uses the classical Bars-and-Stripes test for classical generative models. qBAS measures the capacity of a circuit to learn these patterns from an (n,m) pixels, while it does not with non-BAS patters. The metric is based on the classical F1 metric of classifiers.

- CLOPS: Circuit Layer Operations per Second. This metric (introduced by IBM© [5])) measures the largest depth and number of qubits that can be executed by a quantum computer. It uses synthetic benchmarks that try to assess the capacity of quantum computers to execute random synthetic circuits. The main problem of this metric is that the result should be validated against a classical simulation of such circuits, which can be unfeasible for large number of qubits.

4 Hardware-level Benchmarks

There is a family of benchmarking methodology that focus on the physical low-level details of the Quantum hardware. For instace, the authors in [17] present an early benchmarking suite which comprises a set of small low-level programs as quantum adders or identity operations. They also present the process to execute them and remark the need of having a benchmarking methodology.

The authors of [8] develops cycle benchmarking, "a rigorous and practically scalable protocol for characterizing local and global errors across multi-qubit quantum processors". Cycle Benchmarking (CB) is a protocol for estimating the process fidelity of a global noise process affecting a quantum device that occur when a cycle of operations is applied to a quantum register. It measures the process fidelity of a Pauli-matrix based Clifford cycle, comparing an ideal G and a noisy G, executing these unitary operations m_1 and m_2 times such that $G^{m_1} = G^{m_2} = I$. Because it compares the ideal output with the noisy implementation, CB needs to calculate this ideal result using classical methods. However, the authors claim that this step can be done efficiently.

CB is proved as a robust protocol to SPAM (State Preparation and Measurement) errors. Furthermore, the number of measurements required to estimate the process fidelity to a fixed precision is approximately independent of the number of qubits. Fidelity corresponds to the effective error rate under RC (randomized compiling, a technique that introduces independent random single-qubits gates into the logic circuit, keeping the logic of the circuit [26]). The performance of the same operation in a circuit without RC can differ significantly from the estimated fidelity of its constituents due to the addition or cancellation of coherent errors. This is a general issue with performance metrics for quantum operations and RC has been designed to eliminate these coherent errors. The protocol also provides insight into how noise scales within a fixed architecture. Cycle benchmarking data validates that the error rate per single-qubit gate and per two-qubit coupling does not increase with increasing system size. Also, they have studied the variation with time of the fidelity due to drifts in the computer, providing a method to calculate the necessary frequency for the calibration to the purpose of keeping the fidelity within a 1% error.

Randomized Model Circuits[5] are presented as an evolution of IBM volumetric benchmarks. This means that, in volumetric benchmarks, the performance of a Quantum platform is assessed through a set of predefined volumetric benchmarks. In this case, they use what it is called random synthetic circuits. The aim of this benchmarking is to assess the effects of quantum noise. For this, the authors create a methodology to generate random synthetics circuits whose output can be validated against a model generated one. One important difference of this paper with respect to our intended approach is that this benchmark is not application-focused as it attempts to stress physical and functional aspects of quantum platforms through synthetic benchmarks but not using real-life applications

pyGSTi[20] is a toolset defined to make quantum characterization, verification, and validation (QCVV) of processors. It characterises the qubits and can execute several benchmarks, as the randomized benchmark. It helps to characterize the qubits, the errors and the drift, including a tool to visualise and analyse the results. A brief review of certification is also done by (Eisert, et al., 2019), which also studies some benchmarks. They state that a measure of the quality of a processor would be defined in terms of the difference (or measure) between the desired final state and the real achieved result.

5 Quantum Compilers Benchmarks

Other benchmarks focus on assessing the effects of quantum compilers. For instance, Arline (automated benchmarking platform for quantum compilers)[1] is a framework to evaluate the capabilities of the compilers. Since compilers are a key module of the stack, indeed they should provide an efficient circuit for the platform, they deserve a specific benchmark. This framework executes and reports a set of cases, producing some metrics such as the compression factor (which measures the ratio of final count of gates and the initial number), the final gate count, the circuit depth, the compiler runtime, etc. The framework is automated and produces a report which currently permits to compare three compilers: IBM Qiskit, pytket (from Cambridge Quantum Computing) and Google Circ. The code is open-source and produces an automatic report with the plots to compare the results of the different metrics.

6 Full-stack Benchmarks

There is a family of benchmarks that attempt to integrate the effects of the full software/hardware stack of QC platforms in a benchmarking methodology.

In [18], the authors define what they call “holistic benchmarks” to stress quantum computing systems as a whole, not each component individually. That is why this work can be classified in this category of the state of the art. This paper is also inspired by the IBM volumetric benchmarks. The proposed methodology is based on modifying the variable components of the stack one by one and measure the effect of this variation in the overall performance of the platform.

For this, the authors propose three figures of merit that measures how close the solution is to the correct one: heavy output generation probability, cross entropy difference and l1-norm distance. These three figures of merit compare distribution between the quantum platform output and a classical simulation. This is possible as benchmarks restrict to use Clifford gates, where classical simulation is feasible to up to 1000 qubits.

Other authors [19] perform a cross-technology hardware/software assessment of the design of a Quantum platform. They assess the effect on performance and reliability of the design choices taken at different levels of the stack, e.g. fundamental gates, their software visibility or communication topologies.

As in the previous reviewed work, they focus on testing the influence on performance and success rate of the optimization of one specific component of the stack. The intention of this analysis is to probe that a tool where these characteristics are configurable can improve performance portability in Quantum platforms.

In [2], the authors propose a benchmark framework that generalise the Quantum Volume, moving from square circuits (depth and width are equal) to rectangular synthetic circuits, where depth and width could be different. Also, they describe how these kind of circuits can be used to replace other benchmarks (as the randomized benchmarks). And even they speculate that Grover or Trotterized-based application benchmarks could be replaced by this rectangular circuits. Finally, the authors also discuss briefly the different possibilities for defining the success criteria for a circuit, so the test can be marked as “passed”, and propose a visualisation method to analyse the results.

The authors of [21] define a full methodology that permits the evaluation of the performance of any computer for a specific application. They transform a general circuit in a set of mirror circuits

[1] https://www.airline.io

(which add an approximate inverse of original circuits) that can be easily verified. That circuit is finally equivalent to a circuit composed only of product of Paulis. This kind of circuit is easily simulated for 1000's of qubits. The methodology includes the possibility of restricting connectivity and parallelism.

7 Application-level Benchmarks

Finally, there are benchmarks that focus on the application level. They evaluate the performance measuring the capability of a platform to execute fast and accurately a given application.

While the previous set of benchmarks are important to evaluate the QPUs and the stacks, for some areas, the execution of a specific type of applications is more informative. Following this path [10] introduces an application-based benchmarking relying on the Ising model. It simulates the time evolution of a Ising model with four sites and Open Boundary Conditions that can be expressed in terms of sums of σ_z and of products of σ_x.

The authors define a specific metric (the gross averaged discrepancy) to compare results. Such metric calculates the difference between the measured site occupations, and the exact value, for all site occupations in the case where the measured value is above a certain threshold. They also show that Quantum Volume is not a good metric to measure this kind of results, although further research has to be pursued to understand this fact. [4] and [6] propose similar problems to evaluate quantum computers, because the one-dimensional Ising or Fermionic Fermi-Hubbard models have an exact analytic solution and, as a consequence, the produced result can be compared against them. [27] defines a set of tests inspired by applications in chemistry, such as, packaging-related quantum computing algorithms. These tests can measure the evolution of the computers and can compare it with other classical methods.

QPack is another application-oriented benchmark[16] based on QAOA. They have included three types of problems that can be solved by this optimisation algorithm (MaxCut, Traveling salesman problem and Dominating set problem), so the manufacturers or vendors cannot tune the system for a single problem. The main metrics are the time of execution for a set of steps and the accuracy of the solution. In the case of the time, because this is a hybrid algorithm, they propose several times to measure:

- Overall run-time.
- Run-time classical hardware.
- Connection between classical and quantum hardware.
- Preprocessing/placing and routing.
- Run-time quantum hardware.

Quantum Machine Learning ([11],[28]) is another application where benchmarks have been defined. One specific measure was presented earlier in this document (qBAS). It emulates the classical Bars and Stripes problem. This set of proposals use classical metrics to compare the results, as the Kullback-Leiber divergence (also known as relative entropy) between discrete probability distributions P and Q defined upon the same space (X):

$$D_{LK}(P\|Q) = -\sum_{x\in X} P(x)\log\left(\frac{Q(x)}{P(x)}\right) \tag{2}$$

or F1 score.

[7] propose a benchmark based on Random Circuit Block-Encoding (RACBEM), which is similar to the dense matrix used on the LINPACK classical benchmarks. Block-Encoding is a technique that allows to encode any n-qubit Hermitian matrix into a unitary (n+m)-qubits operator. Reversing the process, this means, firstly a random unitary operator is generated and later an encoded matrix is found, it is possible to define a Quantum Linear System Problem to solve, which is used as benchmarking. The advantage is that these circuits can be easily defined in a quantum computer and can be used to test the full quantum stack.

[15] focus their proposal on chemistry calculations. The quantum results are compared against classical accurate and but expensive models. The proposed benchmark is centred exclusively on chemistry applications, so the utility out of this area could be minor.

QasmBench [13] is a low-level benchmark suite implemented with OpenQasm. In order to asses if the suite covers the whole spectrum of QC application, they propose a set of metrics to classify the candidate applications: circuit width and depth, gate density, retention lifespan, measurement density, entanglement, susceptibility to NISQ error and potential gain from machine-specific optimizations. The selected benchmarks are divided into: small (2-5 qubits), medium (6-15) and large scale (≥ 15).

They measure the accuracy of the QC execution using Hellinger distance between the reconstructed density matrix of the resultant mixed state in a real quantum device observed through density matrix tomography, and the density matrix of the pure state by running the same circuit in a noiseless classical simulator.

MQT Bench [22] is part of the Munich Quantum Toolkit (MQT). It is based on four design principles: (1) cross-level support for different abstraction levels, (2) accesibility, as it provides an easy web interface and a Python API, (3) generalizabilty as it provides a broad set of benchmarks, and (4), exentedabilty to future algoritms, gate-sets and hardware architecture. The benchmarks count is around 50.000 benchmarks ranging from 2 to 130 qubits. The selection criteria for the benchmarks is to select those that they consider de-facto standards. They have a special section for the variational quantum algorithms from IBM Qiskit application level.

Supermarq [25] is an application-based benchmark suite composed of 8 benchmarks that are implemented in OpenQASM. A set of features vectors are used to take the decision of which benchmarks are included in the suite. This is used to make sure that they cover all the spectrum of QC applications. These metrics cover aspects that characterize a quantum workload such as: the amount of measurement or entanglement, the shape of the circuit, and the percentage of 2-qubit gates, etc...

8 Discussion

At the end of this review, we can extract some conclusions about the state of QC benchmarking. First, QC benchmarking methodologies follow two main trends: (1) Those that attempt to asses the ability of the platform to deal with physical aspects of QC, like noise or different circuit configurations, and (2) those that attempt to measure how fast a platform can execute typical quantum workloads. Metrics of the first class of methodologies focus on measuring those physical aspects that are assessed by the methodology or its impact on the accuracy of the quantum computation. Metrics of the some class are divided among those targeting to measure the accuracy of the computation, and those measuring the time to solution.

Second, in application level benchmarks there are also two additional aspects to consider, the scalability of the benchmarks and the criteria used to select them. The first aspect is solved by some authors proposing different benchmarks for different circuits size, and by others proposing parameterized benchmarks that can be instantiated for any circuit size.

The second aspect, the selection criteria, is solved by some authors with a short discussion about the general criteria used for the selection, usually focusing on the coverage of aspects like generality, complexity or circuit shape. Other authors use a systematic approach based on selection metrics that guarantee that the full spectrum of applications is covered by the suite.

Bibliography

[1] Marcello Benedetti, Delfina Garcia-Pintos, Oscar Perdomo, Vicente Leyton-Ortega, Yunseong Nam, and Alejandro Perdomo-Ortiz. A generative modeling approach for benchmarking and training shallow quantum circuits. npj Quantum Information, 5(1):1–9, 2019.

[2] Robin Blume-Kohout and Kevin C Young. A volumetric framework for quantum computer benchmarks. Quantum, 4:362, 2020.

[3] Robin J Blume-Kohout and Kevin Young. Metrics and benchmarks for quantum processors: State of play. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2019.

[4] Alba Cervera-Lierta. Exact ising model simulation on a quantum computer. Quantum, 2:114, 2018.

[5] Andrew W Cross, Lev S Bishop, Sarah Sheldon, Paul D Nation, and Jay M Gambetta. Validating quantum computers using randomized model circuits. Physical Review A, 100(3):032328, 2019.

[6] Pierre-Luc Dallaire-Demers, Michał Stęchły, Jerome F Gonthier, Ntwali Toussaint Bashige, Jonathan Romero, and Yudong Cao. An application benchmark for fermionic quantum simulations. arXiv preprint arXiv:2003.01862, 2020.

[7] Yulong Dong and Lin Lin. Random circuit block-encoded matrix and a proposal of quantum linpack benchmark. Physical Review A, 103(6):062412, 2021.

[8] Alexander Erhard, Joel J Wallman, Lukas Postler, Michael Meth, Roman Stricker, Esteban A Martinez, Philipp Schindler, Thomas Monz, Joseph Emerson, and Rainer Blatt. Characterizing large-scale quantum computers via cycle benchmarking. Nature communications, 10(1):1–7, 2019.

[9] Craig Gidney and Martin Ekerå. How to factor 2048 bit rsa integers in 8 hours using 20 million noisy qubits. Quantum, 5:433, 2021.

[10] Erik Gustafson, Patrick Dreher, Zheyue Hang, and Yannick Meurice. Benchmarking quantum computers for real-time evolution of a $(1 + 1)$ field theory with error mitigation. arXiv preprint arXiv:1910.09478, 2019.

[11] Kathleen E Hamilton, Eugene F Dumitrescu, and Raphael C Pooser. Generative model benchmarks for superconducting qubits. Physical Review A, 99(6):062323, 2019.

[12] IEEE. QUANTUM COMPUTING BENCHMARKING WORKING GROUP P7131. https://sagroups.ieee.org/7131/, 2023. [Online; accessed 28-nov-2023].

[13] Ang Li, Samuel Stein, Sriram Krishnamoorthy, and James Ang. Qasmbench: A low-level qasm benchmark suite for nisq evaluation and simulation. arXiv preprint arXiv:2005.13018, 2020.

[14] Simon Martiel, Thomas Ayral, and Cyril Allouche. Benchmarking quantum coprocessors in an application-centric, hardware-agnostic, and scalable way. IEEE Transactions on Quantum Engineering, 2:1–11, 2021.

[15] Alexander J McCaskey, Zachary P Parks, Jacek Jakowski, Shirley V Moore, Titus D Morris, Travis S Humble, and Raphael C Pooser. Quantum chemistry as a benchmark for near-term quantum computers. npj Quantum Information, 5(1):1–8, 2019.

[16] Koen Mesman, Zaid Al-Ars, and Matthias Möller. Qpack: Quantum approximate optimization algorithms as universal benchmark for quantum computers. arXiv preprint arXiv:2103.17193, 2021.

[17] Kristel Michielsen, Madita Nocon, Dennis Willsch, Fengping Jin, Thomas Lippert, and Hans De Raedt. Benchmarking gate-based quantum computers. Computer Physics Communications, 220:44–55, 2017.

[18] Daniel Mills, Seyon Sivarajah, Travis L Scholten, and Ross Duncan. Application-motivated, holistic benchmarking of a full quantum computing stack. Quantum, 5:415, 2021.

[19] Prakash Murali, Norbert Matthias Linke, Margaret Martonosi, Ali Javadi Abhari, Nhung Hong Nguyen, and Cinthia Huerta Alderete. Full-stack, real-system quantum computer studies: Architectural comparisons and design insights. In 2019 ACM/IEEE 46th Annual International Symposium on Computer Architecture (ISCA), pages 527–540. IEEE, 2019.

[20] Erik Nielsen, Kenneth Rudinger, Timothy Proctor, Antonio Russo, Kevin Young, and Robin Blume-Kohout. Probing quantum processor performance with pygsti. Quantum science and technology, 5(4):044002, 2020.

[21] Timothy Proctor, Kenneth Rudinger, Kevin Young, Erik Nielsen, and Robin Blume-Kohout. Measuring the capabilities of quantum computers. Nature Physics, 18(1):75–79, 2022.

[22] Nils Quetschlich, Lukas Burgholzer, and Robert Wille. Mqt bench: Benchmarking software and design automation tools for quantum computing. arXiv preprint arXiv:2204.13719, 2022.

[23] Salonik Resch and Ulya R Karpuzcu. Benchmarking quantum computers and the impact of quantum noise. ACM Computing Surveys (CSUR), 54(7):1–35, 2021.

[24] Eyob A Sete, William J Zeng, and Chad T Rigetti. A functional architecture for scalable quantum computing. In 2016 IEEE International Conference on Rebooting Computing (ICRC), pages 1–6. IEEE, IEEE, 2016.

[25] Teague Tomesh, Pranav Gokhale, Victory Omole, Gokul Subramanian Ravi, Kaitlin N Smith, Joshua Viszlai, Xin-Chuan Wu, Nikos Hardavellas, Margaret R Martonosi, and Frederic T Chong. Supermarq: A scalable quantum benchmark suite. In 2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA), pages 587–603. IEEE, 2022.

[26] Joel J Wallman and Joseph Emerson. Noise tailoring for scalable quantum computation via randomized compiling. Physical Review A, 94(5):052325, 2016.

[27] Kübra Yeter-Aydeniz, Bryan T Gard, Jacek Jakowski, Swarnadeep Majumder, George S Barron, George Siopsis, Travis S Humble, and Raphael C Pooser. Benchmarking quantum chemistry computations with variational, imaginary time evolution, and krylov space solver algorithms. Advanced Quantum Technologies, 4(7):2100012, 2021.

[28] Daiwei Zhu, Norbert M Linke, Marcello Benedetti, Kevin A Landsman, Nhung H Nguyen, C Huerta Alderete, Alejandro Perdomo-Ortiz, Nathan Korda, A Garfoot, Charles Brecque, et al. Training of quantum circuits on a hybrid quantum computer. Science advances, 5(10):eaaw9918, 2019.