HamPerf: A Hamiltonian-Oriented Approach to Quantum Benchmarking

Anastasiia Butko
Lawrence Berkeley National Lab
Berkeley, CA
abutko@lbl.gov

Katherine Klymko
Lawrence Berkeley National Lab
Berkeley, CA
kklymko@lbl.gov

Daan Camps
Lawrence Berkeley National Lab
Berkeley, CA
dcamps@lbl.gov

Nicolas Sawaya
Azulene Labs
Berkeley, CA
nicolas@azulenelabs.com

ABSTRACT

Quantum computing technologies are undergoing rapid development. The different qubit modalities being considered for quantum computing each have their strengths and weaknesses, making it challenging to compare their performance relative to each other and the state-of-the-art in classical high-performance computing. To better understand the utility of a given quantum processor and to assess when and how it will be able to advance the frontiers of computational science, researchers need a robust approach to quantum benchmarking. A variety of approaches have been proposed, many of which characterize the presence of noise in current quantum devices. These efforts include component-level performance metrics, such as randomized benchmarking and gate set tomography; high-level application-dependent metrics; and device-level metrics, such as the Quantum Volume. However, it remains unclear how low-level metrics, such as fidelities and decoherence times, and global device metrics, such as Quantum Volume, relate to the computational utility and practical limitations of quantum processors to solve useful problems. In this paper, we describe our Hamiltonian-oriented approach to quantum benchmarking called HamPerf. Where previous application-dependent approaches specify a suite of benchmarking circuits inspired by applications, we place the problem Hamiltonian at the center. Our strategy allows us to probe the computational performance of a quantum processor on standardized and relevant problem sets, agnostic of the algorithms and hardware used to solve them; it also provides fundamental insights into how device characteristics correlate with computational utility.

CCS CONCEPTS

• **Theory of computation → Quantum computation theory**; • **Hardware → Quantum computation**; • **Computing methodologies → Simulation evaluation**.

CF '24 Companion, May 7–9, 2024, Ischia, Italy

ACM ISBN 979-8-4007-0492-5/24/05
https://doi.org/10.1145/3637543.3653431

KEYWORDS

quantum computing, benchmarking, Hamiltonian, software stack, hardware characteristics.

ACM Reference Format:
Anastasiia Butko, Katherine Klymko, Daan Camps, and Nicolas Sawaya. 2024. HamPerf: A Hamiltonian-Oriented Approach to Quantum Benchmarking. In *Proceedings of the 21st ACM International Conference on Computing Frontiers Workshops and Special Sessions (CF '24 Companion), May 7–9, 2024, Ischia, Italy*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3637543.3653431

1 INTRODUCTION

In classical supercomputing, benchmarking has found a broad range of use cases. The LINPACK [19] benchmark for solving dense linear algebra is the de facto standard for ranking supercomputers in the TOP500 list [7]. Benchmarks can also aid in understanding a system's performance for certain key applications, e.g., for machine learning [42, 44]. Furthermore, benchmark suites that span the full range of the system architecture are indispensable for guiding investment and procurement decisions about next-generation supercomputers [3].

The field of high-performance computing has matured over the past decades and developed a deep understanding of the most important indicators, such as memory bandwidth, energy usage, and fp64 performance, that affect the application performance. This deep understanding is still missing in the quantum computing space for a variety of reasons. Firstly, quantum computing technologies are comparatively new, immature, and undergoing rapid development cycles, which means that,

- there hasn't been sufficient opportunity and time for users to gain experience with this new technology;
- it is currently not yet possible to run benchmarks over a broad range of applications and problem sizes; and,
- what was true yesterday might no longer be valid today.

At present, there exist very few end-to-end applications for which a scientist may expect benefits from running them on a quantum accelerator other than for fundamental research in the field of quantum information science. This problem represents two sides of the same coin: the hardware is not yet of the quality and size to provide substantial computational advantages and at the same time the use cases, applications, and algorithms are not fully understood because

the hardware to run them at scale is not available. Secondly, there is no consensus yet on what is the best qubit hardware platform or even the computational model to build quantum computers and design quantum algorithms. On the hardware side, superconducting qubits are currently state-of-the-art according to many metrics, but ion traps and neutral atom qubits outperform them in terms of coherence times and operational requirements (e.g. no dilution fridge). On the application and algorithms side, there exist gaps between,

- algorithms for the fault-tolerant large-scale quantum computer era (e.g. Shor's algorithm, Quantum Phase Estimation (QPE) [32], Hamiltonian Simulation [37]) for which convincing evidence of utility exists *and* near-term algorithms for noisy quantum computers (e.g. VQA [41], Quantum Subspace Methods [33]) for which the advantage is much less understood and debated;
- different models of quantum computation, such as the circuit model, adiabatic computation, analog computation, measurement-based computations, etc..

This makes it a major challenge to devise quantum benchmarks that are (i) broadly applicable across the different technologies and computational models, and (ii) future-proof for large-scale quantum computers while still meaningful in the noisy quantum computing era. Current approaches to quantum benchmarking still fall short of these goals and can be roughly subdivided in three tiers, see Fig. 1:

(1) **Component level benchmarks**—such as randomized benchmarking [34] and gate set tomography [11] allow for low-level, accurate characterization of quantum devices by means of gate fidelities and decoherence times. While they provide accurate, low-level information, it is unclear how they relate to the overall computational utility of the device. Furthermore, they often incur an exponential overhead and can thus be costly to evaluate at scale.
(2) **Device level benchmarks**—such as quantum volume (QV) [15] attempt to capture the performance in a single metric and are in that sense the quantum analog of LINPACK. The main shortcomings are: (1) random square circuits might not be representative of an actual workload, and (2) they are not future-proof as the classical verification that is required does not scale beyond a few 10s of qubits.
(3) **Application level benchmarks**—such as SupermarQ [50], QUARK [21], and the QED-C suite [38] are closest in philosophy to HamPerf. They define a suite of quantum circuits based on applications, such as VQE, and a feature vector of metrics designed to capture the performance.

These existing approaches remain largely siloed and fail to accurately correlate application-level performance with component-level metrics [50]. In our approach, we want to take meaningful and important strides toward breaking down these silos and achieving this ambitious goal.

We identify that the commonality between all qubit technologies and computational models is that they solve a computational problem through Hamiltonian evolution and *encode* the problem in the Hamiltonian. This connection can be very explicit, as in the case of adiabatic and analog quantum computation, or more subtle as for the circuit model. Furthermore, the problem encoding can be 1:1,

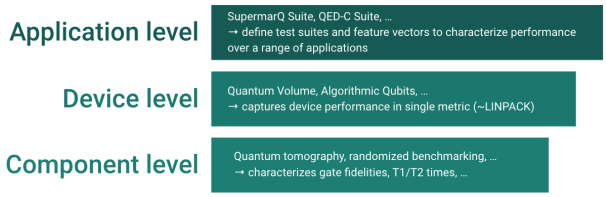

Figure 1: There currently exist three tiers of quantum benchmarks. At the *component level* protocols like gate set tomography [11] and randomized benchmarking [34] allow for characterization of individual gate or qubit fidelities. *Device level* metrics such as quantum volume (QV) [15] attempt to capture the performance of a particular device in a single quantity, while *application level* benchmark suites are designed to measure device performance over a representative workload.

when the problem Hamiltonian is naturally expressed in the Pauli algebra, it can be very close to 1:1, for example, the Jordan-Wigner transformation for fermionic Hamiltonians, or it can make use of embeddings such as for example binary, unary (one-hot), or Gray codes [47, 48].

Because of the intricate connection between the Hamiltonian and the problem on hand, and the Hamiltonian and the quantum system on the other hand, we propose to develop a Hamiltonian-oriented approach to quantum benchmarking. The umbrella name for our project is *HamPerf*, in analogy to MLPerf [42]. Compared to existing application level benchmarks, we design HamPerf to be (1) applicable to multiple models of computation, including classical, (2) able to map the crossover regime from classically easy to hard, (3) flexible in correlating hardware characteristics to computational utility, and (4) scalable.

The rest of the paper is organized as follows: we describe the contents of HamLib and the goal of having a standardized dataset for future algorithm development; we then describe the goals of HamPerf as a benchmarking framework for evaluating quantum application performance and the steps needed to allow quantum algorithm and hardware evaluation to reach a similar level as classical computing.

2 PROPOSED RESEARCH AND METHODS

In this section, we describe what research and methods we include in our benchmarking approach.

2.1 HamLib

Our approach to HamPerf starts from HamLib [46], a new and large dataset of qubit-based quantum Hamiltonians designed for benchmarking purposes. HamLib contains datasets from the physical sciences and combinatorial optimization, as depicted in Fig. 2. The collection contains problems on 2 qubits up to 1,000 qubits and thus allows for benchmarking of current, small-scale quantum computers while at the same time being future-proof. Many of the problem instances can also be further extended to larger systems where required. Furthermore, the dataset contains problems of different complexity, from classically easy to hard, for all problem domains.

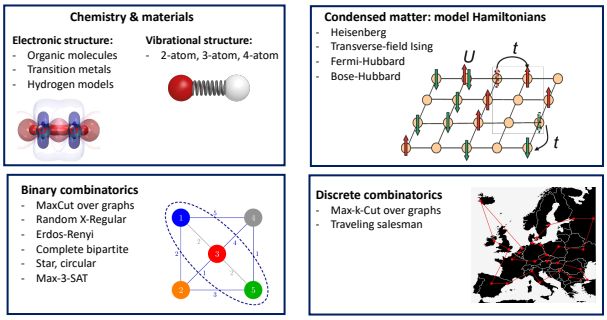

Figure 2: An overview of the HamLib [46] dataset of Hamiltonians for quantum benchmarking which contains problems from physics and chemistry, as well as combinatorial optimization. The figure used from [46].

This diversity will allow us to design benchmarks on top of HamLib that can map out the intermediate cross-over regime where quantum computers can start to deliver potential computational advantages and correlate this regime to device characteristics.

Furthermore, during the development of HamPerf, we continue to evaluate and update the HamLib data set in order to cover the breadth of applications that are of interest to researchers. Other science domains of interest to include problem data sets from HamLib and HamPerf are, for example, high-energy physics, nuclear physics, and linear algebra.

We note that another Hamiltonian dataset was recently released by Pennylane [5] (Xanadu's quantum software library) with a similar goal, containing example Hamiltonians from chemistry, lattice models, and other areas. They have also recently incorporated parts of HamLib into Pennylane's data module such that the Hamiltonians can easily be used with their software.

2.2 HamPerf

HamPerf is focused on building a transparent and unbiased evaluation of the performance of different types of quantum hardware in executing tasks of importance to the scientific community. The benchmarking framework is designed in a modular and integrated way according to the organization as presented in Fig. 3.

2.2.1 Correlating Hardware Traits to Device Utility. Existing quantum devices and control hardware vary tremendously in their technology and implementation details. A comprehensive and universal understanding of the relationship between low-level hardware metrics and computational and algorithmic utility in quantum computers is crucial for developing useful quantum devices.

Each high-level metric can be decomposed into low-level hardware characteristics. For example, an overall time-to-solution can be represented as per-component profiling that can give valuable insights into how an existing quantum system can be scaled and where the major improvements and developments must be made. Energy cost is another crucial metric that has been driving green HPC research for decades [2]. We propose to analyze the energy cost per component using the HamPerf approach. Moreover, as the number of qubits grows the trends in these metrics and characteristics change.

In our previous research [12], we demonstrated how high-level circuit characterization can be used to analyze and quantify quantum control hardware giving insights into its capabilities and limitations per qubit number. It allowed us to predict when circuit and experiment size will no longer be feasible due to contestants in control hardware speed. This demonstration shows the importance of using quantum benchmarks in not only quantum device characterization but also in the evaluation of the support hardware. While in classical computing a low IPS rate will result in a longer delay to obtain computational results, in quantum computing it can cause the results of an experiment to be completely unusable. Quantifying these limitations will allow us to better understand the utility of quantum devices for specific use cases and under the continuing growth of the system size and complexity.

Here, we study which physical properties of quantum devices and the classical control eco-system constrain the computational power for different problem domains. We create a hardware evaluation methodology using the proposed Hamiltonian approach. HamPerf will also allow us to understand which algorithms and problem domains may be more or less robust against noise and provide insight into which combination of hardware, algorithm, and problem instance will lead to quantum acceleration.

2.2.2 Colliding Worlds: Quantum vs. Classical. In order to assess whether a quantum computer will be useful for an application, we must understand whether it outperforms a classical computer for the same task. One aspect of HamPerf involves studying the scaling and resource requirement of classical software packages, to compare against our analyses of quantum programs.

We plan to make such comparisons through (a) where possible, collecting existing scaling and resource data from the literature [36, 54]; and (b) where necessary, running well-developed advanced classical software packages [8, 18, 22, 26, 31, 49, 52, 55] to understanding scaling and resources. These scaling analyses can be run on HPC compute resources to ensure a fair comparison between the quantum backends and state-of-the-art classical computing. We intend to make such quantum-classical comparisons for most of the Hamiltonians in HamLib.

We make a distinction between two different problem categories:

(i) Physics and Chemistry problems—In the realm of classical simulation of quantum physics (i.e. chemistry, materials, and condensed matter), our view is that quantum computers are most likely to be useful when very high accuracy results for strongly correlated systems are required. Because of this, it is especially important to study quantum algorithm performance with respect to these three algorithm classes: quantum Monte Carlo [23], tensor network methods [40], and coupled cluster theory [9] (the latter is applicable primarily to chemistry).

We plan to run benchmark tests for all three of these classical algorithm classes, in order to compare against our estimated quantum resource requirements. Though it would not be reasonable to consider every major software package in this area, it is important to ensure that the software package chosen for these comparisons is reasonably fast. We consider packages NWChem [52], pySCF [49], pyQMC [55], QMCPack [31], Quimb [26], and iTensor [22], as these are packages that are known to be reasonably fast and with which the authors are already familiar. We will do spot checks on a few

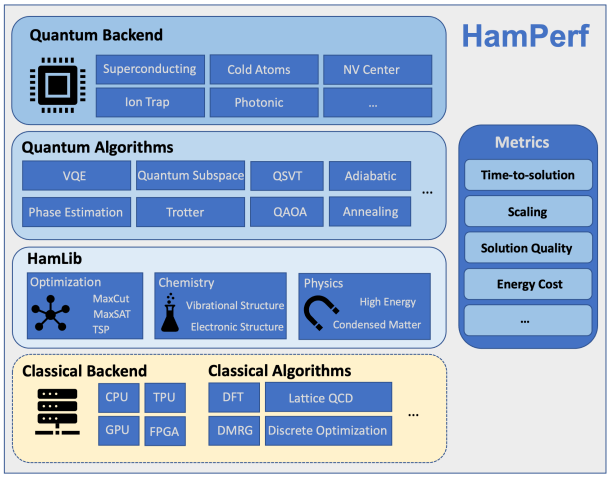

Figure 3: HamPerf connects a dataset of relevant problem Hamiltonians (HamLib [46]) with applicable combinations of quantum algorithms and quantum backends to cover the spectrum of quantum technologies under development.

Hamiltonians to determine which package to use in each Hamiltonian class and use the best-performing package for our reported times-to-solution.

(ii) Combinatorial Optimization problems— There are several existing classical solvers for classical combinatorial problems, and these will be used to compare against our quantum resource estimates. HamLib is set up such that it allows for scaling comparisons for a variety of classical problem instances that are somewhat related to the real world. For instance, we are able to use HamLib to make quantum-to-classical comparisons of real-world traveling salesman problem [43] instances as well as industrially relevant instances of graph problems [27].

In order to limit the scope of work, we focus primarily on SDP (semidefinite program) solvers to study classical scaling. We will consider both time-to-solution and quality of solution using existing well-developed packages CVXPY [18] and MOSEK [8].

To determine whether and in which contexts quantum computers will be useful, it is necessary to estimate resource counts and how they scale with respect to problem size. In turn, this often simply translates to studying how *each component* of the problem scales. Certainly this is the case for hybrid quantum-classical algorithms [10, 20], where each component of the algorithm may be studied separately. A hybrid quantum-classical algorithm typically includes a parametrized quantum circuit ansatz, which encodes the problem cost function that is evaluated on the quantum computer through measurements. Next, the circuit parameters are updated to improve the trial solution wavefunction. This process is repeated iteratively until a sufficiently good approximation is obtained.

To the best of our knowledge, no standard benchmarking framework has been developed for hybrid algorithms, which are the primary algorithms we expect to demonstrate quantum advantage in the near term, and thus our work fills an important and timely gap in the field.

Our efforts are focused on developing benchmarks for two crucial components of hybrid algorithms, quantum measurements, and ansatzae.

(i) Measurement efficiency— Evaluating one instance of a cost function (e.g. the energy of a molecule) is often costly, requiring many circuit repetitions. Several methods have recently been developed to vastly reduce the number of "shots" required to estimate this cost function, and this aspect of hybrid algorithms is an important aspect of algorithm scaling, as the time-to-solution is directly proportional to the required number of shots.

Many aspects of these measurement efficiency schemes relate directly to a given hardware set. For example, some near- and medium-term hardware may not allow for measurement efficiency routines that require longer circuit depth. Further, differences in qubit connectivity between quantum computers may be the sole variable leading to one method being favorable over another. Finally, the quality of a single-shot measurement will not be perfect for near-term hardware and will vary considerably between different devices; this variability should be explicitly taken into account in HamPerf.

HamPerf develops a benchmark based partly on the following state-of-the-art measurement efficiency methods. First, there are what we call the "commuting measurement" methods, for which the many-term Hamiltonian is partitioned, and each measurement shot provides information about many terms in the Hamiltonian [14, 25, 53]. These are in some sense the conceptually simplest methods, though they also tend to be straightforward to code and require the shortest circuit depths. The HamPerf framework would also include code for the following three methods, which require more circuit depth to implement, but for certain Hamiltonian classes will require fewer measurements: fermion-specific change-of-basis methods [29], quantum subspace methods [33], and shadow tomography [28].

(ii) Circuit ansatzae— In order to link quantum algorithm performance with a given hardware (and in turn whether a quantum computer will find utility), it is important to consider the quantum circuit ansatz used for a hybrid algorithm. This ansatz is a consequential decision when designing quantum algorithms, and an ansatz that is optimal on one hardware might be inefficient for another type of hardware [10, 24, 30]. Especially relevant to this topic is qubit connectivity. Our hope is that an extensive analysis of circuit ansatzae may even allow us to make more general comments about hardware types. For instance, it could be that circuit ansatzae are more naturally expressed on ion traps and are more favorable for a particular Hamiltonian class.

2.2.3 Flexible Software Stack for Benchmarking. A big part of the HamPerf development involves the implementation and standardization of the benchmarking procedures in the HamPerf software stack. We design this software stack to bring together the different components shown in Fig. 3 in a flexible, lightweight, sufficiently general, and user-friendly way. We ensure that the classical part of the HamPerf software stack can be deployed on HPC compute resources, while the quantum algorithms and quantum backends will be supported using emerging industry standards such as OpenQASM 3.x [4], Qiskit [51], Cirq [17], etc. We work to integrate HamPerf with quantum backends that are available to the research

team, for example, the Advanced Quantum Testbed [1] and QuEra hardware [6]. Furthermore, we explore the use of emulated quantum backends, such as the cuQuantum SDK [16], in HamPerf. This allows us to: (1) lay the groundwork for end-to-end quantum benchmarks on actual quantum backends through numerical emulation, (2) additionally, benchmark *quantum-inspired algorithms* against purely classical implementations.

The HamPerf software stack brings together the different components shown in Fig. 3. We approach this using modern software engineering practices such as continuous integration/continuous deployment (CI/CD). This approach has the advantage that we can immediately start developing the HamPerf software stack, while still allowing for a high degree of flexibility to adapt the design to the requirements set by the other thrusts of the project. For example, the integration of new performance metrics, quantum algorithms, or classical codes defined by the other thrusts can set new requirements for the core HamPerf stack. Our CI/CD approach means that such changes will be made efficiently and with minimal risk of introducing errors.

2.2.4 Verification, reproducibility and scalability. We anticipate that over the next five years, there will be some experiments demonstrating quantum advantage (i.e. solving a problem that would be intractable with only classical resources) for problems of interest to the scientific community. With this in mind, we construct HamPerf as a scalable toolset that can be adapted for that period and be useful for benchmarking systems/algorithms on 40+ qubits. Therefore, we focus on verification, by which we mean confirming that the results from the applications run on the quantum computer are correct, reproducibility, meaning that we can recover the same quality of results from different quantum runs, and scalability, meaning our approach will be useful as quantum computers grow in qubit number. To accomplish these goals we include exactly solvable models in our set of applications. For example, there are certain problems of interest in condensed matter physics that are classically solvable even for a large number of qubits, such as free fermionic systems. The transverse field Ising model in one dimension (included in HamLib for a range of system sizes) is an example of a free fermionic system that is of great importance for understanding quantum phase transitions [45] and is used regularly as a model for quantum hardware demonstrations of new algorithms [33, 39]. The authors of this paper have experience developing efficient classical software to process these model systems at scale [13, 35] and have demonstrated results up to 16 qubits on quantum hardware and a few 1,000s of qubits classically. One approach would be to use these lower complexity, exactly solvable problems as a starting point for verification of HamPerf benchmarks at scale. Complexity could slowly added by for example turning on extra terms in the Hamiltonian perturbatively. A similar approach can be used for chemical systems, where molecules that are not strongly correlated and therefore amenable to classical solvers could be the starting point for verifying/using HamPerf at scale. We emphasize that HamLib has been constructed such that a range of problem complexities are included for each subcategory of the dataset and thus it is well suited for this approach.

3 ACKNOWLEDGMENTS

This work was supported by the Office of Advanced Scientific Computing Research for the Computational Research Division, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

REFERENCES

[1] [n. d.]. Advanced Quantum Testbed @ LBNL. https://aqt.lbl.gov/. [Online; accessed April 2023].
[2] [n. d.]. GREEN500 The List. https://www.top500.org/lists/green500/. [Online; accessed April 2023].
[3] [n. d.]. NERSC-10 Benchmark Suite. https://www.nersc.gov/systems/nersc-10/benchmarks/. [Online; accessed April 2023].
[4] [n. d.]. OpenQASM 3.x Live Specification. https://openqasm.com/. [Online; accessed April 2023].
[5] [n. d.]. Quantum Data (beta). https://pennylane.ai/datasets/.
[6] [n. d.]. QuEra NERSC R&D partnership. https://www.quera.com/press-releases/quera-to-bring-expanded-quantum-capability-to-nersc. [Online; accessed April 2023].
[7] [n. d.]. TOP500 The List. https://www.top500.org/. [Online; accessed April 2023].
[8] Erling D. Andersen and Knud D. Andersen. 2000. *The Mosek Interior Point Optimizer for Linear Programming: An Implementation of the Homogeneous Algorithm.* Springer US, Boston, MA, 197–232. https://doi.org/10.1007/978-1-4757-3216-0_8
[9] Rodney J Bartlett and Monika Musiał. 2007. Coupled-cluster theory in quantum chemistry. *Reviews of Modern Physics* 79, 1 (2007), 291.
[10] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S Kottmann, Tim Menke, et al. 2022. Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics* 94, 1 (2022), 015004.
[11] Robin Blume-Kohout, John King Gamble, Erik Nielsen, Kenneth Rudinger, Jonathan Mizrahi, Kevin Fortier, and Peter Maunz. 2017. Demonstration of qubit operations below a rigorous fault tolerance threshold with gate set tomography. *Nature Communications* 8 (4 2017), 14485. Issue 1. https://doi.org/10.1038/ncomms14485
[12] Anastasiia Butko, George Michelogiannakis, Samuel Williams, Costin Iancu, David Donofrio, John Shalf, Jonathan Carter, and Irfan Siddiqi. 2020. Understanding Quantum Control Processor Capabilities and Limitations through Circuit Characterization. In *2020 International Conference on Rebooting Computing (ICRC).* 66–75. https://doi.org/10.1109/ICRC2020.2020.00011
[13] Daan Camps, Efekan Kokcu, Lindsay Bassman Oftelie, Wibe A De Jong, Alexander F Kemper, and Roel Van Beeumen. 2022. An algebraic quantum circuit compression algorithm for hamiltonian simulation. *SIAM J. Matrix Anal. Appl.* 43, 3 (2022), 1084–1108.
[14] Ophelia Crawford, Barnaby van Straaten, Daochen Wang, Thomas Parks, Earl Campbell, and Stephen Brierley. 2021. Efficient quantum measurement of Pauli operators in the presence of finite sampling error. *Quantum* 5 (Jan. 2021), 385. https://doi.org/10.22331/q-2021-01-20-385
[15] Andrew W. Cross, Lev S. Bishop, Sarah Sheldon, Paul D. Nation, and Jay M. Gambetta. 2019. Validating quantum computers using randomized model circuits. *Physical Review A* 100 (9 2019), 032328. Issue 3. https://doi.org/10.1103/PhysRevA.100.032328
[16] The cuQuantum development team. 2023. *cuQuantum.* https://doi.org/10.5281/zenodo.7806810 If you use this software, please cite it as below..
[17] Cirq Developers. 2022. *Cirq.* https://doi.org/10.5281/zenodo.7465577 See full list of authors on Github: https://github .com/quantumlib/Cirq/graphs/contributors.
[18] Steven Diamond and Stephen Boyd. 2016. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research* 17, 83 (2016), 1–5.
[19] Jack J. Dongarra, Piotr Luszczek, and Antoine Petitet. 2003. The LINPACK Benchmark: past, present and future. *Concurr. Comput. Pract. Exp.* 15, 9 (2003), 803–820. http://dblp.uni-trier.de/db/journals/concurrency/concurrency15.html#DongarraLP03
[20] Dmitry A Fedorov, Bo Peng, Niranjan Govind, and Yuri Alexeev. 2022. VQE method: a short survey and recent developments. *Materials Theory* 6, 1 (2022), 1–21.
[21] Jernej Rudi Finžgar, Philipp Ross, Leonhard Hölscher, Johannes Klepsch, and Andre Luckow. 2022. QUARK: A Framework for Quantum Computing Application Benchmarking. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE).* 226–237. https://doi.org/10.1109/QCE53715.2022.00042
[22] Matthew Fishman, Steven R. White, and E. Miles Stoudenmire. 2022. The ITensor Software Library for Tensor Network Calculations. *SciPost Phys. Codebases* (2022), 4. https://doi.org/10.21468/SciPostPhysCodeb.4
[23] WMC Foulkes, Lubos Mitas, RJ Needs, and Guna Rajagopal. 2001. Quantum Monte Carlo simulations of solids. *Reviews of Modern Physics* 73, 1 (2001), 33.

[24] Bryan T Gard, Linghua Zhu, George S Barron, Nicholas J Mayhall, Sophia E Economou, and Edwin Barnes. 2020. Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm. *npj Quantum Information* 6, 1 (2020), 10.

[25] Pranav Gokhale, Olivia Angiuli, Yongshan Ding, Kaiwen Gui, Teague Tomesh, Martin Suchara, Margaret Martonosi, and Frederic T. Chong. 2020. $O(N^3)$ Measurement Cost for Variational Quantum Eigensolver on Molecular Hamiltonians. *IEEE Transactions on Quantum Engineering* 1 (2020), 1–24. https://doi.org/10.1109/TQE.2020.3035814

[26] Johnnie Gray. 2018. quimb: a python library for quantum information and many-body calculations. *Journal of Open Source Software* 3, 29 (2018), 819. https://doi.org/10.21105/joss.00819

[27] Christopher Hojny, Imke Joormann, Hendrik Lüthen, and Martin Schmidt. 2021. Mixed-integer programming techniques for the connected max-k-cut problem. *Mathematical Programming Computation* 13, 1 (2021), 75–132.

[28] Hsin-Yuan Huang, Richard Kueng, and John Preskill. 2020. Predicting many properties of a quantum system from very few measurements. *Nature Physics* 16, 10 (June 2020), 1050–1057. https://doi.org/10.1038/s41567-020-0932-7

[29] William J. Huggins, Jarrod R. McClean, Nicholas C. Rubin, Zhang Jiang, Nathan Wiebe, K. Birgitta Whaley, and Ryan Babbush. 2021. Efficient and noise resilient measurements for quantum chemistry on near-term quantum computers. *npj Quantum Information* 7, 1 (Feb. 2021). https://doi.org/10.1038/s41534-020-00341-7

[30] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. 2017. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *nature* 549, 7671 (2017), 242–246.

[31] Jeongnim Kim, Andrew D Baczewski, Todd D Beaudet, Anouar Benali, M Chandler Bennett, Mark A Berrill, Nick S Blunt, Edgar Josué Landinez Borda, Michele Casula, David M Ceperley, Simone Chiesa, Bryan K Clark, Raymond C Clay, Kris T Delaney, Mark Dewing, Kenneth P Esler, Hongxia Hao, Olle Heinonen, Paul R C Kent, Jaron T Krogel, Ilkka Kylänpää, Ying Wai Li, M Graham Lopez, Ye Luo, Fionn D Malone, Richard M Martin, Amrita Mathuriya, Jeremy McMinis, Cody A Melton, Lubos Mitas, Miguel A Morales, Eric Neuscamman, William D Parker, Sergio D Pineda Flores, Nichols A Romero, Brenda M Rubenstein, Jacqueline A R Shea, Hyeondeok Shin, Luke Shulenburger, Andreas F Tillack, Joshua P Townsend, Norm M Tubman, Brett Van Der Goetz, Jordan E Vincent, D ChangMo Yang, Yubo Yang, Shuai Zhang, and Luning Zhao. 2018. QMCPACK: an open source ab initio quantum Monte Carlo package for the electronic structure of atoms, molecules and solids. *Journal of Physics: Condensed Matter* 30, 19 (apr 2018), 195901. https://doi.org/10.1088/1361-648X/aab9c3

[32] A. Yu. Kitaev. 1995. Quantum measurements and the Abelian Stabilizer Problem. arXiv:quant-ph/9511026 [quant-ph]

[33] Katherine Klymko, Carlos Mejuto-Zaera, Stephen J. Cotton, Filip Wudarski, Miroslav Urbanek, Diptarka Hait, Martin Head-Gordon, K. Birgitta Whaley, Jonathan Moussa, Nathan Wiebe, Wibe A. de Jong, and Norm M. Tubman. 2022. Real-Time Evolution for Ultracompact Hamiltonian Eigenstates on Quantum Hardware. *PRX Quantum* 3 (May 2022), 020323. Issue 2. https://doi.org/10.1103/PRXQuantum.3.020323

[34] E. Knill, D. Leibfried, R. Reichle, J. Britton, R. B. Blakestad, J. D. Jost, C. Langer, R. Ozeri, S. Seidelin, and D. J. Wineland. 2008. Randomized benchmarking of quantum gates. *Physical Review A* 77 (2008), 012307. Issue 1. https://doi.org/10.1103/PhysRevA.77.012307

[35] Efekan Kökcü, Daan Camps, Lindsay Bassman Oftelie, James K Freericks, Wibe A de Jong, Roel Van Beeumen, and Alexander F Kemper. 2022. Algebraic compression of quantum circuits for Hamiltonian evolution. *Physical Review A* 105, 3 (2022), 032420.

[36] Dimitrios G Liakos, Yang Guo, and Frank Neese. 2019. Comprehensive benchmark results for the domain based local pair natural orbital coupled cluster method (DLPNO-CCSD (T)) for closed-and open-shell systems. *The Journal of Physical Chemistry A* 124, 1 (2019), 90–100.

[37] Seth Lloyd. 1996. Universal Quantum Simulators. *Science* 273, 5278 (1996), 1073–1078. https://doi.org/10.1126/science.273.5278.1073 arXiv:https://www.science.org/doi/pdf/10.1126/science.273.5278.1073

[38] Thomas Lubinski, Sonika Johri, Paul Varosy, Jeremiah Coleman, Luning Zhao, Jason Necaise, Charles H. Baldwin, Karl Mayer, and Timothy Proctor. 2021. Application-Oriented Performance Benchmarks for Quantum Computing. (2021). https://doi.org/10.48550/arxiv.2110.03137

[39] Mario Motta, Chong Sun, Adrian TK Tan, Matthew J O'Rourke, Erika Ye, Austin J Minnich, Fernando GSL Brandão, and Garnet Kin-Lic Chan. 2020. Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution. *Nature Physics* 16, 2 (2020), 205–210.

[40] Román Orús. 2019. Tensor networks for complex quantum systems. *Nature Reviews Physics* 1, 9 (2019), 538–550.

[41] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. 2014. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications* 5, 1 (July 2014). https://doi.org/10.1038/ncomms5213

[42] Vijay Janapa Reddi, Christine Cheng, David Kanter, Peter Mattson, Guenther Schmuelling, Carole-Jean Wu, Brian Anderson, Maximilien Breughe, Mark Charlebois, William Chou, Ramesh Chukka, Cody Coleman, Sam Davis, Pan Deng, Greg Diamos, Jared Duke, Dave Fick, J. Scott Gardner, Itay Hubara, Sachin Idgunji, Thomas B. Jablin, Jeff Jiao, Tom St. John, Pankaj Kanwar, David Lee, Jeffery Liao, Anton Lokhmotov, Francisco Massa, Peng Meng, Paulius Micikevicius, Colin Osborne, Gennady Pekhimenko, Arun Tejusve Raghunath Rajan, Dilip Sequeira, Ashish Sirasao, Fei Sun, Hanlin Tang, Michael Thomson, Frank Wei, Ephrem Wu, Lingjie Xu, Koichi Yamada, Bing Yu, George Yuan, Aaron Zhong, Peizhao Zhang, and Yuchen Zhou. 2020. MLPerf Inference Benchmark. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*. 446–459. https://doi.org/10.1109/ISCA45697.2020.00045

[43] Gerhard Reinelt. 1991. TSPLIB—A traveling salesman problem library. *ORSA journal on computing* 3, 4 (1991), 376–384.

[44] Albert Reuther, Peter Michaleas, Michael Jones, Vijay Gadepally, Siddharth Samsi, and Jeremy Kepner. 2019. Survey and Benchmarking of Machine Learning Accelerators. In *2019 IEEE High Performance Extreme Computing Conference (HPEC)*. 1–9. https://doi.org/10.1109/HPEC.2019.8916327

[45] Subir Sachdev. 1999. Quantum phase transitions. *Physics world* 12, 4 (1999), 33.

[46] Nicolas PD Sawaya, Daniel Marti-Dafcik, Yang Ho, Daniel P Tabor, David E Bernal Neira, Alicia B Magann, Shavindra Premaratne, Pradeep Dubey, Anne Matsuura, Nathan Bishop, Wibe A de Jong, Simon Benjamin, Ojas D Parekh, Norm Tubman, Katherine Klymko, and Daan Camps. 2023. HamLib: A library of Hamiltonians for benchmarking quantum algorithms and hardware. *arXiv e-prints*, Article arXiv:2306.13126 (June 2023), arXiv:2306.13126 pages. https://doi.org/10.48550/arXiv.2306.13126 arXiv:2306.13126 [quant-ph]

[47] Nicolas PD Sawaya, Tim Menke, Thi Ha Kyaw, Sonika Johri, Alán Aspuru-Guzik, and Gian Giacomo Guerreschi. 2020. Resource-efficient digital quantum simulation of d-level systems for photonic, vibrational, and spin-s Hamiltonians. *npj Quantum Information* 6, 1 (2020), 49.

[48] Nicolas PD Sawaya, Albert T Schmitz, and Stuart Hadfield. 2022. Encoding trade-offs and design toolkits in quantum algorithms for discrete optimization: coloring, routing, scheduling, and other problems. *arXiv preprint arXiv:2203.14432* (2022).

[49] Qiming Sun, Timothy C. Berkelbach, Nick S. Blunt, George H. Booth, Sheng Guo, Zhendong Li, Junzi Liu, James D. McClain, Elvira R. Sayfutyarova, Sandeep Sharma, Sebastian Wouters, and Garnet Kin-Lic Chan. 2018. PySCF: the Python-based simulations of chemistry framework. *WIREs Computational Molecular Science* 8, 1 (2018), e1340. https://doi.org/10.1002/wcms.1340 arXiv:https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wcms.1340

[50] T. Tomesh, P. Gokhale, V. Omole, G. Ravi, K. N. Smith, J. Viszlai, X. Wu, N. Hardavellas, M. R. Martonosi, and F. T. Chong. 2022. SupermarQ: A Scalable Quantum Benchmark Suite. In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE Computer Society, Los Alamitos, CA, USA, 587–603. https://doi.org/10.1109/HPCA53966.2022.00050

[51] Matthew Treinish, Jay Gambetta, Soolu Thomas, Paul Nation, qiskit bot, Paul Kassebaum, Diego M. Rodríguez, Salvador de la Puente González, Jake Lishman, Shaohan Hu, Luciano Bello, Kevin Krsulich, Jim Garrison, Jessie Yu, Manoel Marques, Julien Gacon, David McKay, Juan Gomez, Lauren Capelluto, Travis-S-IBM, Abby Mitchell, Ashish Panigrahi, lerongil, Rafey Iqbal Rahman, Steve Wood, Toshinari Itoko, Alex Pozas-Kerstjens, Christopher J. Wood, Divyanshu Singh, and Drew Risinger. 2023. *Qiskit/qiskit: Qiskit 0.41.0.* https://doi.org/10.5281/zenodo.7591922

[52] M. Valiev, E.J. Bylaska, N. Govind, K. Kowalski, T.P. Straatsma, H.J.J. Van Dam, D. Wang, J. Nieplocha, E. Apra, T.L. Windus, and W.A. de Jong. 2010. NWChem: A comprehensive and scalable open-source solution for large scale molecular simulations. *Computer Physics Communications* 181, 9 (2010), 1477–1489. https://doi.org/10.1016/j.cpc.2010.04.018

[53] Vladyslav Verteletskyi, Tzu-Ching Yen, and Artur F. Izmaylov. 2020. Measurement optimization in the variational quantum eigensolver using a minimum clique cover. *The Journal of Chemical Physics* 152, 12 (03 2020). https://doi.org/10.1063/1.5141458 arXiv:https://pubs.aip.org/aip/jcp/article-pdf/doi/10.1063/1.5141458/15573883/124114_1_online.pdf 124114.

[54] Rajat Walia and Jun Yang. 2021. From benchmarking ab-initio DMRG investigations to theoretical modeling of new singlet fission systems. In *International Conference on Photochemistry 2021 (ICP 2021)*.

[55] William A. Wheeler, Shivesh Pathak, Kevin G. Kleiner, Shunyue Yuan, João N. B. Rodrigues, Cooper Lorsung, Kittithat Krongchon, Yueqing Chang, Yiqing Zhou, Brian Busemeyer, Kiel T. Williams, Alexander Muñoz, Chun Yu Chow, and Lucas K. Wagner. 2023. PyQMC: An all-Python real-space quantum Monte Carlo module in PySCF. *The Journal of Chemical Physics* 158, 11 (03 2023). https://doi.org/10.1063/5.0139024 114801.