HamLib: A library of Hamiltonians for benchmarking quantum algorithms and hardware

Nicolas PD Sawaya [*1], Daniel Marti-Dafcik[2], Yang Ho[3], Daniel P Tabor[4], David E Bernal Neira[5], Alicia B Magann[3], Shavindra Premaratne[6], Pradeep Dubey[1], Anne Matsuura[6], Nathan Bishop[7], Wibe A de Jong[8], Simon Benjamin[9], Ojas D Parekh[3], Norm M. Tubman[5], Katherine Klymko [†10], and Daan Camps [‡10]

[1] *Intel Labs, Santa Clara, CA 94115, USA*
[2] *Physical & Theoretical Chemistry Lab, Univ. of Oxford, Oxford, OX1 3QZ, UK*
[3] *Sandia National Laboratories, Albuquerque, NM 87185, USA*
[4] *Department of Chemistry, Texas A&M University, College Station, TX 77843, USA*
[5] *NASA Ames Research Center, Moffett Field, CA 94035, USA*
[6] *Intel Labs, Hillsboro, OR 97124, USA*
[7] *Intel Corporation, Hillsboro, OR 97124, USA*
[8] *Lawrence Berkeley National Lab, Berkeley, California 94720*
[9] *Department of Materials, University of Oxford, Oxford OX1 3PH, UK*
[10] *National Energy Research Scientific Computing Center (NERSC), Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*

Abstract

In order to characterize and benchmark computational hardware, software, and algorithms, it is essential to have many problem instances on-hand. This is no less true for quantum computation, where a large collection of real-world problem instances would allow for benchmarking studies that in turn help to improve both algorithms and hardware designs. To this end, here we present a large dataset of qubit-based quantum Hamiltonians. The dataset, called HamLib (for Hamiltonian Library), is freely available online and contains problem sizes ranging from 2 to 1000 qubits. HamLib includes problem instances of the Heisenberg model, Fermi-Hubbard model, Bose-Hubbard model, molecular electronic structure, molecular vibrational structure, MaxCut, Max-k-SAT, Max-k-Cut, QMaxCut, and the traveling salesperson problem. The goals of this effort are (a) to save researchers time by eliminating the need to prepare problem instances and map them to qubit representations, (b) to allow for more thorough tests of new algorithms and hardware, and (c) to allow for reproducibility and standardization across research studies.

Contents

[*] nicolas.sawaya@intel.com
[†] kklymko@lbl.gov
[‡] dcamps@lbl.gov

1 Scope and Preliminaries

1.1 Motivation

Large datasets of problem instances have long been useful for the analysis of computer hardware, software, and algorithms. For instance, ImageNet [DDS+09] is a massive repository of images that has facilitated the testing of many deep learning packages. Another example is computational chemistry and materials science, where extensive datasets (e.g. the Protein Data Bank [BWF+00], the Materials Project [JOH+13] and QM9 [RDRVL14]) improve ease of testing new algorithms and software approaches, while providing standardization across the field. Well-defined datasets or problem instances may in turn be used to define benchmarking suites such as MLPerf [MRC+20] and LINPACK [DLP03, MB18].

Though there has been progress in introducing benchmarks in the quantum computing community [PWS+16, CFG+22, LSKA, LJV+21, CBG21, TGO+22], there is not yet a topically broad database of problem instances. Having such a dataset would be convenient for many reasons. For instance, when researchers wish to test a novel Hamiltonian simulation algorithm [Llo96, WBAG11, LC17, CST+21] for chemistry, they must first go through the tedious and non-trivial process of preparing a set of chemical Hamiltonians on their own. It would be useful for the researcher to have these preparatory steps done ahead of time, so that they may spend more of their efforts on algorithm or hardware design.

There are three primary motivations behind creating a dataset of Hamiltonians with broad coverage in application area and in problem difficulty. First, such a library can save substantial labor time. For example, a researcher wanting to test their new quantum chemistry algorithms will not have to learn the minutia of electronic structure, install and run various packages, choose a representative test set, and debug inevitable software difficulties. In turn, this allows for resources and time being devoted to to the more interesting aspects of algorithm and software development.

Second, a large Hamiltonian library allows for more thorough testing. For example, if one is performing numerical tests on a new Hamiltonian simulation algorithm, immediately being able to run it on a very broad class of problems (as opposed to only e.g. a toy spin model and two to three molecules) allows for a stronger understanding of when the algorithm performs well.

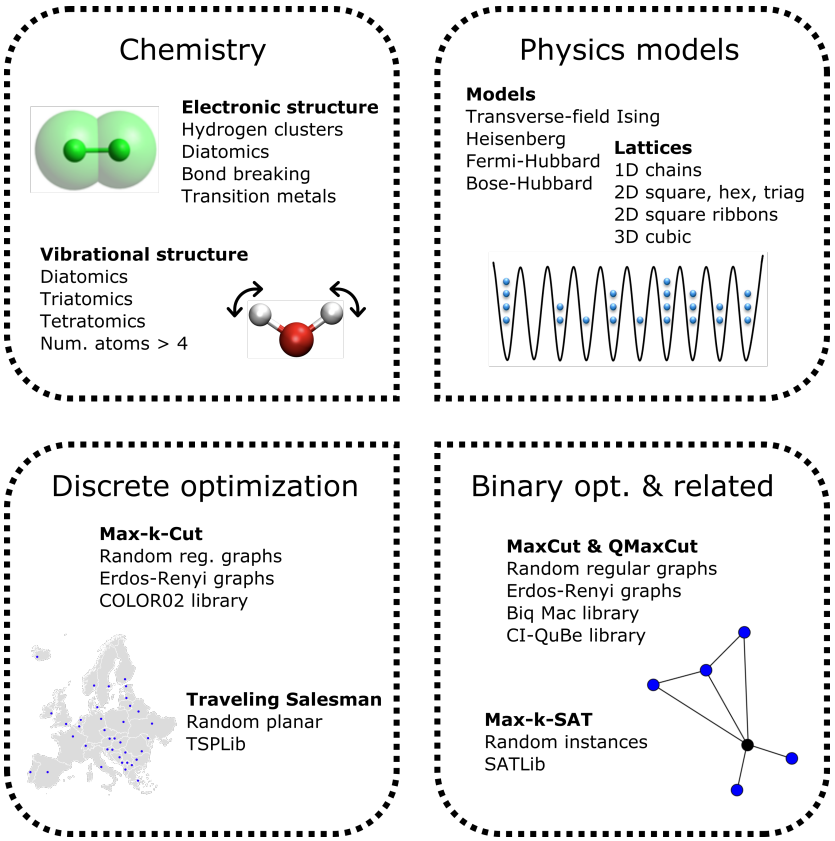

Figure 1: The four categories of qubit Hamiltonians included in HamLib. In the area of chemistry we include real-world accurate Hamiltonians for both electronic and vibrational structure. The condensed matter dataset includes four commonly studied models defined on a variety of lattice topologies, with and without periodic boundary conditions. We include Max-k-SAT and MaxCut in the binary combintorial problems, as well as QMaxCut, a quantum analogue with interesting mathematical properties. For discrete combinatorics we include both Max-k-Cut and the traveling salesperson problem.

Third, such a library allows for reproducibility and standardization across research studies. It is more straightforward to make fair comparisons between two algorithms if they are benchmarked using the exact same problem sets.

In order to facilitate practical benchmarking of quantum algorithms, software, and hardware, in this work we introduce an extensive dataset of many quantum *problem Hamiltonians* from a variety of fields related to condensed matter physics, chemistry, and classical optimization. We call the dataset HamLib (for Hamiltonian Library) version 1. We focus primarily on presenting a wide range of Hamiltonians that may be used in various contexts; the definition of proper benchmarks—which generally require defining both a problem and an algorithm *in addition to* a dataset—will not be our primary goal. However, we include broad discussion of benchmarking in Section 3.

1.2 Attributes of this library

There were several principles that guided the curation of this library. Although choosing the contents for any dataset is somewhat arbitrary, we believe that HamLib's contents are well-motivated, while showing exceptional breadth in terms of problem size, complexity, mathematical properties, and field of study.

We strive to create datasets that were "well-spaced" in terms of qubit counts. This is important for multiple reasons. First, strong simulation of quantum algorithms cannot be performed past 40 to 50 qubits; hence in order to properly study scaling, one should be studying problem instances for every few qubits. Even for cases where tensor networks are used to simulate 100s of qubits, or where quantum compilers are benchmarked on even more qubits, problem sizes in the context of early

quantum computing are still much smaller than typical classical datasets. For instance, the classical benchmark Graph 500 [MMB+10] contains graph problem instances of more than 10^{40} vertices. Second, because current quantum hardware is still limited in size, small spacings are necessary in order to test as much of the device as possible. For instance, if an experimental quantum computer has exactly 24 qubits, we want to ensure that several problem instances from HamLib can be used to test the full device.

A main focus of HamLib is the curation of *real-world* problems, with parameters and properties as similar as possible to those encountered in some real scientific or industrial applications. We include high-quality and meticulously prepared problems in molecular electronic structure and vibrational structure, while also curating real-world datasets for combinatorial problems such as routing, by using distance matrices from real cities [Rei91]. It is our suspicion that the reason many quantum algorithms are often tested only on simpler condensed matter models or on random graphs is simply that it is time-consuming to prepare problems that are more representative of the real world. However, lattice models and random instances are equally vital for benchmarking and scientific understanding, and as such much of our dataset consists of toy problems as well.

Another essential facet of HamLib is that all problem instances have already been mapped to qubits, i.e. they have already been mapped to a Pauli representation of the form

$$H_{\text{encoded}} = \sum_i c_i \bigotimes_k \{\sigma_{ik}\} \tag{1}$$

where σ_{ik} is a one-qubit Pauli or identity operator, i.e. $\sigma_{ik} \in \{I, X, Y, Z\}$, and c_i is a real scalar. It is our hope that this eliminates substantial labor for some researchers. The user can simply download the qubit representations and immediately use them in a qubit-based computer or simulator. However, we caution against a black box approach. It will often be important to understand the difference between (for example) the various fermion-to-qubit mappings or the various integer-to-qubit mappings, before interpreting the results of a particular benchmark. The original "unencoded" Hamiltonians are included where appropriate, so that the user can implement alternative encoding approaches not considered here. Further, some auxiliary information is included, such as approximate ground states or operators that aid in implementing certain quantum algorithms.

We attempt to achieve broad coverage in terms of the mathematical properties of the Hamiltonians. A broad range of localities (i.e. Pauli weights) are present, as the many mapping choices for fermionic, vibrational, bosonic, and combinatorial problems lead to localities of anywhere from 2 to N, where N is the number of qubits. Qubit connectivites vary widely as well, as we consider many graph types for combinatorial problems, and several grid dimensionalities and types in condensed matter. Finally, we note that a broad range of Hamiltonian norms are present—this is important because the formal complexity of many algorithms (especially in Hamiltonian simulation [CST+21, LSTT23]) depend explicitly on such norms.

A main use case we have in mind for this dataset is comparing quantum approaches against each other, not just comparing quantum algorithms/computers to classical ones. Hence there are entire problem classes for which the task is either certain or likely to scale polynomially on a classical computer (e.g. ground state finding for the 1D transverse-field Ising model)—such problem instances are still useful for quantum-to-quantum comparisons for quantum algorithms, quantum compilers, and quantum computers. However, for many portions of HamLib, it would certainly be a reasonable use of the dataset to compare scaling behavior in quantum versus classical methods, and we encourage this as use case as well.

Some comments are merited on why certain problem areas are *not* included in HamLib. We have focused exclusively on problems that are commonly represented *as Hamiltonians*, largely for the purpose of saving time for researchers. For instance, quantum machine learning [CVH+22] for classical data is not included, because most implementations of such algorithms do not in fact first map the data to a qubit-based Hamiltonian. This is an important research areas that will require its own curated high-quality datasets, and some work has already been done in this area [PYF21].

1.3 Related work

Benchmarking for classical computing has matured over many decades. Perhaps the most prominent large-scale linear algebra benchmark is LINPACK [DLP03, MB18], though widely used classical

benchmarking standards exist, including MLPerf [MRC+20], SPEC [Dix91], HPCG [DLH13], Green500 [FC07], and Graph500 [MMB+10]. More analogous to the current work are past efforts constructing large *datasets* that can be used in various benchmarking tasks. For instance, as mentioned above, ImageNet [DDS+09] contains millions of images intended for use in image analysis and deep learning, while similarly large datasets exist for simulating chemistry [RDRVL14] and materials science [JOH+13].

In recent years there have been several proposals for benchmarking quantum algorithms and quantum computers. Perhaps most similar to previous classical benchmarking efforts are the proposed benchmark suites consisting of common quantum subroutines, such as quantum phase estimation and quantum adders [PWS+16, CFG+22, LSKA, LJV+21, CBG21, TGO+22]. Benchmarks have been proposed specifically for near-term hybrid quantum-classical algorithms [YAGJ+21, MPJ+19, WRV+23], with some benchmarking work concentrating on more specific topics, such as particular classical optimization problems [MAAM21, FRH+22, LCM+23], benchmarking the optimizer used in the classical step of hybrid algorithms [MRBAG16, GS17, SMM22], a dataset used for quantum machine learning [PYF21], and the dataset of curated Hamiltonians available in the software package Pennylane [BIS+18]. Narrower benchmarks have been proposed for very specific applications as well [KLMT00, HWPC05, ZKE+19, YADM+19, EHW+20, DDSG+20, RK21, DL21, NMP+23]. Other proposals include those related to quantum volume [CBS+19, MGE12], random benchmarking [KLR+08], and related techniques [EWP+19, PRY+22]; such efforts are not associated with specific quantum algorithms but are an essential aspect of hardware characterization and design.

Despite the above-mentioned progress, there is not yet a large publicly available dataset of quantum Hamiltonians (i.e. cost functions) for benchmarking algorithmic implementations e.g. QPE, adiabatic quantum computing, VQE, and QAOA. The purpose of the current work is to fill this gap.

1.4 Availability and dataset structure

At the time of publication, the HamLib dataset is downloadable from the following URL:

https://portal.nersc.gov/cfs/m888/dcamps/hamlib/

The Hamiltonians are organized into the following high-level categories:

- **Binary-variable optimization and related problems**. Subcategories are Max-k-SAT, Max-Cut, and QMaxCut.
- **Discrete-variable optimization problems** (discrete variables taking > 2 values). Subcategories are Max-k-Cut and the traveling salesperson problem.
- **Condensed matter** physics models commonly studied by theoretical physicists. Subcategories are the transverse-field Ising, Heisenberg, Fermi-Hubbard, and Bose-Hubbard models.
- **Chemistry** Hamiltonians that use curated or calculated real-world parameters. Subcategories are electronic structure and vibrational structure.

As stated above, a major purpose of this work is to provide a large set of Hamiltonians that have already been mapped to a qubit representation. All encoded Hamiltonians are represented using OpenFermion's QubitOperator class [MRS+20] and stored in the HDF5 format. These are compressed to ZIP format. *Please note*, it will not be unusual for the decompression to yield an HDF5 file that is an order of magnitude larger than the ZIP file. We provide a few Python code snippets useful to interact with the HamLib dataset in Appendix A.

It is important to point out that for some of these Hamiltonians, minimization (e.g. minimum eigenvalue finding) is a typical goal, while for others maximization (e.g. maximum eigenvalue finding) is more natural. We point this out this for any researcher considering running the entire dataset through a set of algorithms in a black-box manner.

In addition to the qubit representations of the operators herein, we include files for the original "pre-encoded" (before having been encoded into qubit representation) Hamiltonians where appropriate. This may take the format of a mat2qubit, OpenFermion, or SciPy array object as specified in the relevant sections of this document. Where appropriate, other auxiliary data is included, e.g. graph instances or approximate ground state values. If such auxiliary data is present for a given subset then this is mentioned in the relevant portion of Section 2.

In Hamiltonians with non-binary and non-fermionic variables, the correspondence between the qubit indices and the original variable indices can be determined as follows. Smaller variables indices in the original variables space correspond to smaller variable indices in qubit space, and within an encoded discrete variable the least significant figures correspond to the smaller qubit index. For example, assume we have variables x_0, x_1, and x_2, all with cardinalities 4 and encoded with standard binary. The state $x_0 = 2 \to 10_2$ (subscript denoting binary representation) corresponds to the quantum basis states `01****` where asterisks denote arbitrary values on x_1 and x_2. For the unary (one-hot) representation for which $x_0 = 2 \to 0100_2$, the quantum basis states are `0010********`.

1.5 Modifications to and extensions of HamLib

We believe that, for the purposes of this project, it is best to maintain *static* datasets. This allows comparisons between research papers to be as valid as possible over the years. However, there are of course many other problem types that could have been included in this dataset, including e.g. those related to high energy physics or nuclear structure. Hence future modifications to HamLib will likely take the form of additional datasets, whether e.g. adding new distinct subsets of electronic structure Hamiltonians, or adding subsets for entirely new problem areas. Additionally there is some chance that some modifications will be made to the first version of the dataset.

The first version of the dataset is denoted HamLib v1.0. If modifications are made to HamLib, we will assign a new version number to the changes, and include a full description of the changes or additions in the present section of this manuscript. HamLib versions:

- **v1.0** - Initial release of HamLib.

2 Problem instances

2.1 Binary-variable optimization and related problems

This section includes combinatorial problems defined over binary variables. We consider MaxCut, Max-k-SAT, and QMaxCut, a local Hamiltonian problem that is a quantum analogue of MaxCut. We make a point of using problem instances from established libraries used in classical benchmarking, as well as random graphs we constructed for this work. For the former classes of graphs, where appropriate, we have implemented *variable reduction* schemes whereby smaller problem instances are created from the large original instances [HV02, DCD+06, LMS12]. The purpose of these problem reductions is to ensure that the qubits counts are well-spaced as discussed in Section 1.2, while retaining some of the structure of the original problem instance.

2.1.1 MaxCut

The prototypical classical problem to which NISQ algorithms (especially QAOA) are applied is the MaxCut problem, for which the cost function is defined as

$$H_C = \sum_{(j,k)\in\mathcal{E}} \frac{1}{2}\left(1 - Z_j Z_k\right), \tag{2}$$

where $\mathcal{E}$ is the set of edges in the graph $\mathcal{G}$. MaxCut is a commonly studied graph partitioning problem, which in turn has applications in many domains. In addition, MaxCut (the optimization version) is a classic NP-hard problem.

In HamLib we include a set of trivially 2-colorable graphs, which may be useful for sanity checks—*i.e.* if a quantum algorithm cannot find the optimum for these trivial graphs then it may indicate a pathology in the algorithm. These are the complete bipartite graphs (with partition sizes a and b), star graphs, and circulant graphs.

We constructed two general types of random graphs. First are random X-regular graphs for regularities 3,4,5,6. Additionally, we include Erdos-Renyi graphs GNP(n,p), where p is the probability that a given edge is present. Though previous work in quantum algorithms has studied the case of GNP$(n, \frac{d}{n-1})$ for fixed parameter d [FGG20, BM21], we have chosen not to use this graph class because nearly all such graphs are disconnected [ER+60]. We instead implement GNP$(n, k\frac{\ln n}{n})$ for

fixed parameter k, which for $k > 1.0$ is virtually guaranteed to be a connected graph [ER+60]. For these constructed graphs we include variable counts of: 4 to 10 (step 1); 10 to 48 (step 2); 50 to 200 (step 10); 300 to 1000 (step 100). These instances are used in Max-k-Cut and QMaxCut as well.

Finally, we derive graphs from Biq Mac [Wie07], a library that is widely used in classical benchmarking, and CI-QuBe [TG21], a collection of combinatorial instances for quantum benchmarking. Biq Mac contains two classes of instances: the Ising instances are lattice graphs taken from statistical physics and the 'rudy' instances are generated from the rudy graph generator. From CI-QuBe we used the Karloff [Kar96] and ratio912 data sets. Karloff shows that there is a sequence of graphs whose Goemans-Williamson (GW) approximation ratio approaches 0.878; these are the graphs at the beginning of such a sequence using the construction Karloff proposes. The ratio912 graphs are graphs in the MQLIB instance library [DGS18] that have a 0.912 approximation ratio with respect to GW. For these instances, the optimal cut is always 2/3 of the total number of edges. These GW approximations provide some insight into the hardness of each dataset.

The goal of the problem reductions is to preserve much of the structure of the original Biq Mac and CI-QuBe graphs, ensure the resulting reduced graphs are connected, and ensure that the instances are well-spaced in terms of qubit counts. To this end, we use two variable reduction methods to generate instances with arbitrary node counts. The first is random restriction [HV02, JN08, LMS12] where we randomly partition a sample of vertices (i.e. we reduce the graph by removing already partitioned vertices). The second is a random walk [DCD+06] method that creates a randomly connected subgraph. We initially attempt to use random restriction, but in cases where the random restriction is unable to create a sufficient number of connected instances, we use the random walk method.

The following table summarizes the graphs used for the HamLib instances of MaxCut.

Trivial graphs	2-colorable graphs: Complete bipartite, star, circulant with offsets {1,2}.
Random	X-Reg with $X \in \{3,4,5,6\}$, and GNP$(n, k\frac{\ln n}{n})$[link] with $k \in \{2,3,4,5\}$.
Biq Mac lib	Variable reduction method: random restriction and random walk. (Ising and Rudy)
CI-QuBe [TG21]	Variable reduction method: random restriction and random walk. (Karloff and ratio912)
(Auxiliary)	Graph instances

2.1.2 Max-k-SAT

In addition to being relevant to industrial optimization, satisfiability problems form the basis for much of theoretical computer science. Specifically, 3-SAT is often used for complexity theoretic proofs [AB09], partly because any NP-Hard problem can be cast as a 3-SAT problem.

Constructing a Hamiltonian to represent a 3-SAT problem involves summing 3-variable terms such as

$$x_i \vee x_j \vee x_k = I - \frac{1}{8}(I + Z_k)(I + Z_j)(I + Z_k) \tag{3}$$

if negations are not present. When negations are included (as they are in HamLib), the expression is

$$(\neg)^{s_i} x_i \vee (\neg)^{s_j} x_j \vee (\neg)^{s_k} x_k = I - \frac{1}{8}[I + (-1)^{s_i} Z_i][I + (-1)^{s_j} Z_j][I + (-1)^{s_k} Z_k] \tag{4}$$

where $s_j = 1$ if x_j is negated in the clause, else 0.

From classical computer science, it is known that the hardness of solving satisfiability problems increases with the clause ratio $r = m/n$, where m is the number of clauses and n is the number of variables. Analytical and theoretical results exist which estimate or place bounds on the ratio r at which a problem becomes intractable [ANP05]. In quantum computation as well, researchers have studied hardness scaling in random SAT instances [APMB20, ZSZ22].

Here, we choose clause ratios ($r \in \{2,3,4,5\}$) that cross the (both numerically and analytically estimated) thresholds for hardness in both classical [ANP05] and quantum computational studies [APMB20, ZSZ22]. Implementing variable (qubit) counts n from 4 to 1000, for each (n, r) we create 10 random Hamiltonian instances.

In addition to generating random instances, we also pull a sample collection of datasets from SATLib [HS00]. We use the following datasets: aim, lran, flat30, flat100, uf20-20, uf100-430, uf200-860, uf250-1065, uuf100-430, uuf200-860, and uuf250-1065. SATLib is a representative library of Max-k-SAT instances that contains many satisfiable and unsatisfiable instances. For variable reduction, we first convert the clauses into a graph where each vertex represents a clause, and an edge exists between two vertices if the associated clauses share a variable. Then we apply random restriction [JN08] to the variables and give random assignments to a random sampling of variables. As we remove variables and clauses, we update the graph representation accordingly. If the graph is disconnected, we compute the connected components and then greedily connect them by adding additional clauses.

Note we include the raw pre-encoded SAT instances as lists of 1-indexed integers, while the qubit operators are 0-indexed. We implement the following classes of instances for Max-k-SAT.

Random instances	Clause ratios $r \in \{2, 3, 4, 5\}$
SATLib	Limiting to 2- and 3-SAT. Random restriction (10 random instances per problem size) to reduce problem size.
(Auxiliary)	SAT instances

2.1.3 QMaxCut

Here we consider a "quantum version" of MaxCut. Though this problem is structurally similar to its classical counterparts, its "quantumness" leads to rich and surprising differences in computational complexity with respect to its classical counterpart [Par23].

Quantum MaxCut (QMaxCut) is a quantum optimization problem (i.e., a local Hamiltonian problem) that can be understood as a generalization of the MaxCut problem. Given a (possibly weighted) graph $\mathcal{G}$ with vertices $i \in \mathcal{V}$, edges $(i, j) \in \mathcal{E}$, and non-negative weights $w_{ij} \in \mathcal{W}$, the problem can be defined as finding the *maximum* eigenvalue of the following Hamiltonian:

$$H = \sum_{(i,j)\in\mathcal{E}} w_{ij} H_{ij}, \tag{5}$$

where $H_{ij} = \frac{1}{4}(I - X_iX_j - Y_iY_j - Z_iZ_j)$. All instances in HamLib use $w_{ij} \in \{0, 1\}$.

In general, QMaxCut is QMA-Hard [PM15], where QMA is the quantum analog of the NP complexity class in classical computing [Pap03]. There are several approximation algorithms for this problem, with the best approximation ratio for generic instances being 0.562 [Lee22].

This problem is quantum in the sense that instead of finding the maximum eigenvalue of a real-coefficient degree-2 polynomial P over commutative variables $I, Z_1, \ldots, Z_n$, the problem is to find the maximum eigenvalue of a real-coefficient degree-2 polynomial Q over non-commutative variables $I, X_1, Y_1, Z_1, \ldots, X_n, Y_n, Z_n$. This polynomial can be expressed as a Hermitian matrix $Q \in \mathbb{C}^{2^n \times 2^n}$ instead of a diagonal matrix $P \in \mathbb{C}^{2^n \times 2^n}$.

QMaxCut resembles MaxCut in that each term H_{ij} is a scaled version of the corresponding MaxCut term of H_C with additional Pauli X and Y terms. In particular, the diagonal of H above is a scaled version of the MaxCut Hamiltonian, H_C. These structural similarities have enabled generalizations of approximation algorithms and hardness results for MaxCut to the QMaxCut setting [HNP+23]. In this sense, QMaxCut has served as a testbed for developing algorithmic ideas to approximate quantum local Hamiltonians, just as MaxCut serves as a canonical classical constraint satisfaction problem.

Finding a maximum energy state for QMaxCut is an equivalent problem to finding a minimum energy state of a quantum Heisenberg model, described below; although this leads to subtle differences from the point of view of approximate solving [GP19]. This is one instance of what is known as quantum 2-local Hamiltonians, for which finding their minimum or maximum eigenstates is relevant for quantum many-body physics.

While MaxCut has been a focal point for quantum algorithms such as QAOA, it is not clear whether quantum algorithms can offer approximation advantages for classical problems. For example, it is NP-hard to approximate MaxCut better than the celebrated Goemans-Williamson algorithm [GW95] under the Unique Games Conjecture [KKMO07], and NP-hardness is likely a barrier for polynomial-time quantum algorithms as well. Thus it may be fruitful to consider inherently *quantum* problems when seeking quantum approximation advantages. Yet the types of local Hamiltonian problems typically

studied in physics have not generally been amenable to approximation algorithms in the same way that classical problems have been. We include QMaxCut in HamLib because it may serve as a canonical quantum problem in the study of quantum approximation advantages. Approximation algorithms for QMaxCut employ techniques for approximating MaxCut as well as quantum approaches such as variational algorithms [GP19, AGM20, PT21, PT22, Kin22, Lee22]. HamLib's problem instances of QMaxCut use the same graph instances as MaxCut. The relevant table of instances is in Section 2.1.1.

2.2 Discrete-variable ($d > 2$) optimization problems

Here we consider classical cost functions that are defined over discrete variables taking more than 2 values. As in the case of binary optimization problems, we create both random instances and instances derived from widely-used libraries.

Unlike MaxCut and Max-k-SAT, there is a rich set of choices for how these non-binary variables are encoded into qubits [SSH22]. We implement the unary (i.e. one-hot), Gray, and standard binary encodings for the discrete combinatorial problems of this section. A notable omission is the domain wall encoding, which has been shown to improve computational efficiency in some cases [Cha19]; we did not include it because this encoding is not yet included in mat2qubit [Saw22]. Note that different encodings require different numbers of qubits per variable, which in turn can lead to space-depth tradeoffs when choosing encodings [SMK+20, SSH22].

2.2.1 Max-k-Cut

The Max-k-Cut problem is a generalization of MaxCut. For a graph $G = (V, E)$ with weight function $w : E \to \mathbb{R}$ for the edge set E, the Max-k-Cut problem amounts to finding a partition of V into k disjoint sets $\{C_1, \ldots, C_k\}$ that maximizes the sum of the edges between disjoint sets, i.e., the cost function is

$$\sum_{\substack{i,j\in[k]\\ i>j}} \sum_{\substack{u\in C_i\\ v\in C_j}} w(\{u, v\}). \tag{6}$$

In this work, we consider only unweighted graphs such that the weight function becomes the indicator function $\mathbf{1}_E$, i.e., 1 if $\{u, v\} \in E$ and 0 otherwise. This unweighted version of Max-k-Cutis equivalent to finding the maximum k-colorable subgraph. Max-k-Cutis APX-complete [FJ97] and is equivalent to MaxCut for $k = 2$.

We implement the same random graphs created for the MaxCut section, as well as a set of instances derived from COLOR02 [col], a graph library used for benchmarking, including classical studies of Max-k-Cut [GHHP13, LMM06]. In the latter case, we use a random depth-first search [DCD+06] to achieve arbitrary-sized, connected subgraphs of the original problem. For all problems in the Max-k-Cut dataset, we use three encodings: unary (i.e. one-hot), Gray, and standard binary. We summarize the Max-k-Cut instances in the following table. We use $k \in \{3, 4, 5\}$.

Trivial graphs	Complete k-partite with $k \in \{3, 4, 5\}$, circulant with offsets $\{3, 4, 5, 6\}$, windmill with $k \in \{3, 4, 5, 6\}$ cliques.
Random	Random X-Regular for $X \in \{3, 4, 5, 6\}$, GNP$(n, d/n - 1)$ for $d \in \{2, 3, 4, 5, 6\}$.
COLOR02	Variable counts reduced via random depth-first search (10 random restrictions per graph) up to 200 nodes.
(Auxiliary)	Graph instances

2.2.2 Traveling Salesperson Problem

The traveling salesperson problem [BDVD+98, Lap92] (TSP) is a widely studied problem in combinatorics, and relates to routing problems encountered in industry. Our dataset includes both cost Hamiltonians and permutation penalties for the traveling salesperson problem, whose cost function is defined as

$$\sum_{a=0}^{M-1} d\left(\tau(a), \tau(a+1 \mod M)\right). \tag{7}$$

where τ is a permutation over a set of M cities. We encode this problem as a permutation of integers (as opposed to a binary problem with constraints). One disjoint set of qubits is used per

city, *i.e.* if there are N_c cities, then $N_c \times \mathcal{D}(N_c)$ qubits are required, where $\mathcal{D}(K)$ is the number of qubits required to encode a discrete variable of cardinality K. This way of representing TSP is amenable to encoding-dependent penalties that have previously been called *pair permutation penalties* [SSH22].When implementing this problem, one may alternatively use penalty-free methods that constrain the state [HWO+19], including graph-derived partial mixers [SSH22].

Notably, for the decision version of TSP, there is a phase transition [GW96] with respect to parameter $\alpha = l/\sqrt{nA}$, where n is the number of cities, A is the area, and the question is whether a path length of $\leq l$ exists. When $\alpha < \alpha^*$ there is unlikely to be any path shorter than l. This phase transition has been observed to occur approximately at $\alpha^* = 0.78$. We use this relation to guide our construction of random instances. For a given n, we select random positions in a square of area $A = l^2/(\alpha^*)^2 n$. Roughly speaking, this implies that there is approximately a 50% chance that a path of length l exists; we arbitrarily choose $l = 100$.

For the 77 problem instances in TSPlib [Rei91] containing fewer than 1000 cities, we reduce the number of cities one at a time, by removing the city with the lowest weight, *i.e.* the city that has the shortest distance to its two closest neighbors. The justification is that the two shortest distances are likely to be connected in the optimal route; hence the rest of the path is unlikely to radically change. This is just a heuristic and we acknowledge there may be topologies for which removing cities in this way will qualitatively change the solution to the problem. However, we note that this variable reduction procedure is conceptually similar to what is done in previously used greedy algorithms [GCY+11, Lap92].

We implement the following problem instances of TSP:

Random	Random points chosen in a square of area $A = 100^2/0.78^2 n$.
TSPLib	All symmetric instances from TSPLib with $N_{cities} < 1000$. Problem size reduction is implemented by iteratively removing the node with the smallest weight.
(Auxiliary)	Distance matrices for all problem instances.

2.3 Condensed matter physics models

Here we describe the condensed matter models included in the dataset. We implement spin, fermion, and boson models, covering 1D, 2D, and 3D lattices. We expect the fermionic and bosonic models to be the more useful contributions of this section as they involve more effort to prepare.

We use both periodic boundary conditions (PBC) and non-PBC for all lattices. For users who would like to add local disorder, we have included some qubit-encoded occupation number operators for each individual site; these may be used to add local disorder.

Lattices

We include 1D grids, 2D grids, 2D ribbons, and 3D grids. The 2D instances include square, triagonal, and hexagonal (honeycomb) arrangements. Qubit counts of 2 through 1000 are used, with spacing increasingly sparse as qubit count increases. Importantly, we implement a "snaking" pattern for all grids, which leads to more efficient fermionic representations [VC05]. We include the mapping from node ID to grid position as well, which may optionally be obtained using the `read_gridpositions_hdf5()` function found in the Appendix.

1D	PBC and non-PBC.
2D	Square, triangle, hexagonal. PBC and non-PBC.
2D ribbon	Width 2 through 5 for square lattice. PBC and non-PBC.
3D	Cubic lattices. PBC and non-PBC.
(Auxiliary)	Grid positions (mapping of node IDs to positions).

2.3.1 Transverse-field Ising model

The transverse-field Ising model (TFIM) is the simplest of the condensed matter models that we include. Notably, the one-dimensional version of TFIM is classically tractable even in the presence of

disorder. The Hamiltonian is defined as

$$H = \sum_i h_i X_i + \sum_{\langle i,j \rangle} Z_i Z_j, \tag{8}$$

where the sum is over each edge $\langle i, j \rangle$ in the lattice. Quantum critical points have been found to exist at $h \approx 3$ in two-dimensional models [Kallin et al 2013], as well as at $h \approx 5.16$ in three-dimensional TFIM [Tepaske and Luitz 2021]. These values guide our choices for HamLib—in order to include Hamiltonians on both sides of these critical points we implement values $h \in \{0, 0.1, 0.5, 1, 2, 3, 4, 5, 6\}$ [BD02].

2.3.2 Heisenberg model

We implement the following Hamiltonian for the quantum Heiseberg model,

$$H_{\text{Heis}} = \sum_{i=1}^{N} \left(\vec{\sigma}_j \cdot \vec{\sigma}_{j+1} + h_j Z \right). \tag{9}$$

where $\vec{\sigma}_i = (X_i, Y_i, Z_i)$. The Hamiltonian is also known as the Heisenberg XXX model with external magnetic field and can be solved by the Bethe ansatz in the one-dimensional case without disorder [Fra17, GJS19]. Based on values used in previous work, HamLib includes $h \in \{0, 0.1, 0.5, 1, 2, 3, 5\}$ [CMN+18, Akı13]. While HamLib only includes the Heisenberg XXX model, anisotropic variants such as the Heisenberg XXZ and XYZ models [WR10] can easily be generated from the lattice data.

2.3.3 Fermi-Hubbard model

The Fermi-Hubbard Hamiltonian [Hub63] captures the behavior of fermions on lattice sites. It is defined as

$$H_{FH} = -t \sum_{\langle i,j \rangle, \sigma} \left(c^\dagger_{i,\sigma} c_{j,\sigma} + c^\dagger_{j,\sigma} c_{i,\sigma} \right) + U \sum_i n_{i\uparrow} n_{i\downarrow}, \tag{10}$$

where $\langle i, j \rangle$ labels adjacent lattice sites i and j , σ labels the fermion spin, c and $c^\dagger$ are the fermionic annihilation and creation operators, respectively, and $n_{j\sigma} = c^\dagger_{j\sigma} c_{j\sigma}$ is the number operator associated with spin σ and site j. The first term contains the noninteracting portion of the Hamiltonian and describes fermions hopping between adjacent sites with tunneling amplitude t. The second term describes the on-site interaction between fermions with strength U. The Fermi-Hubbard model can be solved analytically for $U = 0$ and $t = 0$. However, in the general case, an analytical solution is only known in 1D [LW03]. In 2D, the Fermi-Hubbard model has been investigated extensively through simulations, with findings suggesting that the 2D Fermi-Hubbard Hamiltonian in the intermediate coupling regime ($U/t = 4, 6, 8$) near half-filling is especially difficult to solve [LAB+15].

HamLib includes numerous instances of the Fermi-Hubbard Hamiltonian for different lattice structures and dimensions. We implement three fermion-to-qubit mappings for each instance: Jordan-Wigner, parity, Bravyi-Kitaev [CRO+19]. We include files for the pre-encoded fermionic encodings as well. Based on previous studies on 1D, 2D, and 3D versions of the Fermi-Hubbard model [LAB+15], the parameters we use are the same across all grid classes. In particular, we take $t = 1$ and $U \in \{0, 2, 4, 6, 8, 12\}$. We include pre-encoded fermionic Hamiltonians in OpenFermion format.

2.3.4 Bose-Hubbard model

The Bose-Hubbard (BH) model is defined as

$$H_{BH} = -t \sum_i \left(b^\dagger_{i+1} b_i + h.c. \right) + \frac{U}{2} \sum_i n_i (n_i - 1) \tag{11}$$

where $b^\dagger_i$ (b_i) are creation (annihilation) operators, the number operator $n_i \equiv b^\dagger_i b_i$, t is the tunneling strength (set to $t = 1$ in this work), and U is the site energy. Equation (11) often includes an additional term proportional to the chemical potential μ, which effectively sets the particle count. We omit this chemical potential term in our dataset, with the assumption that the user will set an arbitrary number of particles at the beginning of the simulation.

The standard version of this model exhibits two phases, a Mott insulator and a superfluid [FWGF89, FM94]. More complex versions of the model lead to more exotic phases including density wave [PP05] and supersolid phases [BSZK95]. There has been substantial previous work on qubit-based simulations of bosonic degrees of freedom [Som05, MSAH18a, MSAH18b, KS19, SMK+20, SGH20, TAM+21, LSY21], though to our knowledge no qubit-based experimental demonstrations to date.

For our dataset, we chose dimensional-dependent values for U, where parameters were based on the phase diagrams reported in reference [FM94]. The transition between Mott insulator and superfluid is dependent on the filling $\langle n_0 \rangle$, the average number of particles per site. The user may effectively set $\langle n_0 \rangle$ during simulation (*e.g.* by choosing a particular product state at the beginning of the simulation). For most values of $\langle n_0 \rangle$ (which may be set for example by choosing a Fock state at the beginning of the simulation), this range for U will yield some Hamiltonians in the Mott insulator regime and some in the superfluid regime. For 1D, $U/t \in [2, 10, 20, 30, 40]$. For 2D, $U/t \in [10, 30, 50, 70, 100]$. For 3D, $U/t \in [20, 40, 60, 90, 120]$.

We implement 3 mappings for each instance: unary (one-hot), standard binary, and Gray, where we implement bosonic mode truncations of both 4 and 8 ($d = 4, 8$). We included files for the pre-encoded bosonic encodings as well, in mat2qubit format.

2.4 Chemistry

2.4.1 Electronic structure

The molecular electronic structure Hamiltonian may be expressed as

$$H = \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s \tag{12}$$

where $a^\dagger$ and a are the usual fermionic creation an annihilation operators. HamLib included molecules from a range of chemical classes; further, we ensured that the molecules include a range of difficulty. Unless otherwise specified, the nuclear coordinates used are those experimental geometries found in the online database of the National Institute of Standards and Technology (NIST), `https://cccbdb.nist.gov/geometries.asp`.

An important consideration was the choice of active space, as one goal of HamLib is to have problems with evenly-spaced qubit counts. Hence for each molecule, we include Hamiltonians of *several qubit counts*, in order to allow the user to choose a level of complexity. We chose the smallest qubit count by calculating the natural orbital occupation and keeping all orbitals for which the occupation is greater than 0.1. We use the def2SVP basis set for molecules containing transition metals, STO-6G for elemental hydrogen, and ccPVDZ for all others. We ensured that the coupled cluster CCSD(T) energy converges with this condition. Then, to create several instances with more qubits, we then added orbitals to this set either one orbital at a time or one set of degenerate orbitals at a time, to create the 5 smallest qubit counts in addition to the large Hamiltonian that uses the full basis set.

Further, because the orbitals are ordered by energy, a user may straightforwardly create their own Hamiltonian of arbitrary qubit count by removing all qubits with index above some number, and choose their own active space. Our Hamiltonian sizes range up to 160 qubits, with the largest instances being hydrogen chains.

The following molecules were picked based on different benchmark studies that have been performed in the chemistry literature in recent years. There are many high-accuracy classical methods for simulating such systems, and a recent study on benzene [EAD+20] highlighted many of the most modern methods that can be used for the same molecule. Studies like these are seldom done for a wide range of molecules, and thus we rely on many different papers that span a range of different molecules to look at. Our dataset includes main group diatomics, hydrogen chains, and transition metal containing diatomics.

For the transition metal molecules, we selected 10 systems that range from "easy" to "hard." For these molecules, a recent work took a selected CI approach to use for a benchmark set of systems, and then compared different coupled cluster techniques against these results [HTL+19]. We observed that there is no obvious standard ordering of the hardness of these systems that is consistent with all the different coupled cluster techniques. For purposes of this work, we use CCSD(T) as the method in which to assess the difficulty of the systems, as it is one of the most widely used rigorous methods

in quantum chemistry for treating correlated systems. By this definition, these molecules in order of difficulty are VH (1.0), ScO (1.4), ScC (3.9), TiH (4.9), CrO (8.1), MnN (9.6), FeC (10.7), CuC (12.3), CoH , (22.7), NiO (38.2), where in parenthesis is the CCSD(T) error in units of kJ/mol. We include the famously difficult chromium dimer (Cr_2) as well.

We now discuss Hamiltonians composed only of hydrogen. If quantum computers are going to be used for computing properties of molecules and materials, they must focus on the most challenging Hamiltonians for which classical algorithms struggle. In electronic structure, these are systems with a large number of open-shell (singly occupied) orbitals, which leads to strong electron correlation. A difficulty when benchmarking quantum algorithms is that such strong correlation often requires very large qubit counts, because in typical molecules only a small subset of the electrons are strongly correlated. This creates a large size gap between the systems typically used for benchmarking quantum algorithms (such as diatomics or small organic molecules, which contain a small number of electrons that become strongly entangled only when bonds are stretched) and the larger transition metal systems which remain hard to solve via state-of-the-art classical algorithms and could therefore be potential candidates for practical quantum advantage [RWS+17, LLD+19].

The hydrogen systems bridge this gap and constitute an ideal benchmark for quantum algorithms, because they are strongly correlated even when the basis set is relatively small. They contain the smallest possible molecular building block (the hydrogen atom), thus reducing qubit counts, but their symmetric structure causes strong electron correlation. By arranging the atoms in different geometries and dimensions, these systems can be used for testing the ability of different wave function parameterizations to capture different entanglement regimes. Hydrogen model systems have long served as a benchmark for electronic structure algorithms on classical computers [JP80, MCC+17, SHE20], and are currently being used to test quantum or hybrid quantum-classical algorithms [GEBM19, SHE20, KMZC+22, BMDTW22].

For all four structural motifs (linear chains, rings, sheets and pyramids), we included H_n systems with $n = 10, 12, 14, 16$, using the STO-6G basis set. These were studied in Stair *et al.* [SE20], which benchmarked classical electronic structure methods including selected configuration interaction (selected CI), singular value decomposition full CI, and density matrix renormalization group (DMRG). This allows a direct comparison of any quantum algorithms with the classical benchmark data in said paper. For $n \leq 40$ we include systems with bond lengths ranging from 0.5 Å to 2.0 Å in steps of 0.1 Å; for $n > 40$ we include only the bond length 1 Å. For the linear H_n structures, we include the atom counts n = 2,4,6,8,10,12,14,16,18,20, 24,28,32,36,40, 50,60,70,80.

For the systems with over 16 atoms, exact classical calculations are difficult or unfeasible and we therefore only provide the restricted Hartree-Fock (RHF) energy. The larger Hamiltonians may be of use for estimating the resources required for Hamiltonian simulation algorithms [LBG+21].

We used 3 mappings for each problem instance: Jordan-Wigner, parity mapping, and Bravyi-Kitaev. For each encoding, we include a basis state with the correct particle count, intended for use as an initial state. We also include the pre-encoded fermionic Hamiltonians. We implement the following molecules for electronic structure.

Hydrogen (H_n)	1D chains, 1D rings, 2D sheets, 3D pyramidal clusters
Main group diatomics	H_2, N_2, O_2, F_2, B_2, C_2, Be_2; LiH, BeH, BH, CH, NH, OH, HF, O_2, Li_2, NaLi, Na_2
Trans. metals	VH, ScO, ScC, TiH, CrO, MnN, FeC, CuC (low spin), CoH, NiO; Cr_2
Bond breaking	F_2 (single bond), O_2 (double bond), N_2 (triple bond)

2.4.2 Vibrational structure

Molecular vibrational structure [WDC80] is a problem that is often intractable on classical computers, especially when strong anharmonicity and resonanaces (e.g. Fermi resonances) are present [JMS12, MPR21]. Accurate calculations of the vibrational structure of molecules are crucial in identifying unknown molecules in fundamental chemical physics experiments and in astrochemical settings. Confirmed assignments in vibrational spectroscopy also provide direct probes of potential energy surfaces of molecules and fundamental probes of reactivity. Further, the precise locations of vibrational energy levels, which are difficult to compute classically, can directly impact the kinetics of chemical reactions.

In the harmonic basis, the vibrational structure Hamiltonian is

$$H = \frac{1}{2}\sum_{i}^{M} \omega_i (q_i^2 + p_i^2) + \sum_{\{ijk\}} h_{ijk} q_i q_j q_k + \sum_{\{ijkl\}} h_{ijkl} q_i q_j q_k q_l + \cdots , \tag{13}$$

where q_i are canonical position operators, q_i are momentum operators, ω_i are the harmonic frequencies, and $h_{ij\ldots}$ are higher-order coupling constants.

In some respects, vibrational structure might often be easier to implement on a quantum computer than electronic structure because the former does not require particle conservation and some properties of vibrational Hamiltonians may be more favorable [SPT21]. Recent efforts in developing quantum algorithms for calculating vibrational structure have demonstrated how some vibrational classical algorithms (e.g., vibrational coupled cluster) can be adapted to quantum hardware [MMS+19, SH19, OBRT20].

For this benchmarking set, our focus is on diversifying the set of Hamiltonians that exist for benchmarking new methods in vibrational structure. We include a limited set of diatomic molecules, but our primary focus is on triatomic and larger systems. Triatomic systems are the smallest systems that can have resonances that complicate their vibrational structures. Expanding to tetra-atomic and larger systems allows one to study interesting and nontrivial differences in physical behavior between related molecules. For instance, BH_3, BF_3, and BHF_2 each have a similar trigonal planar geometry, but their vibrational Hamiltonians are substantially different in practice due to the different mass distributions across the geometry [SA70].

We include a mix of molecule sizes and a mix of vibrational truncation levels, all of which are given in the key names of the HDF5 files. We include up to fourth-order terms excluding those with more than two unique indices (as such terms are often negligible), which is a common approximation in the quantum chemistry literature. This truncated expansion is common in vibrational structure calculations and with a good choice of the coordinate system is often well-converged. The force fields were generated with the CFOUR program package [MCH+20], at the highly accurate CCSD(T)/ANO1 (tetra-atomics and smaller without Cl or S), CCSD(T)/cc-PVTZ (tetratomics and smaller with S or Cl), or Hartree-Fock/ANO0 (larger molecules) level of theory.

These Hamiltonians are stored in wavenumber units (cm^{-1}) commonly used by spectroscopists. We also include qubit encodings of dipole operators $\mu_{\{x,y,z\}}$, which are required for example in the calculation of transition probabilities [RC19, JSP+21, INE+22, SH22] when determining infrared spectra. Finally, in Appendix B we report statistics for how closely-space the energy levels are in these Hamiltonians, as resonances (i.e. closely-space energy transitions) tend to be hard to simulate classically.

We include 3 encodings for each instance: unary (one-hot), standard binary, and Gray, as well as the pre-encoded "bosonic" Hamiltonians in mat2qubit [Saw22] string format. The following molecules are included.

Diatomics	BeH, BH, CH, CO, F_2
Triatomics	C_2H, C_2O, CH_2, COH, H_2O, H_2S, H_3^+, HCO, HNC, HNO, NH_2, NOH, O_3, SF_2, SO_2
Tetratomics	BF_3, BH_3, BHF_2, CH_3, CH_3^+, ClCOH, FCCF, H_2CC (vinylidene), H_2CO, H_2O_2, HCCH
Larger Molecules	Allene (C_3H_4), Cyclopropene (C_3H_4), Ethylene Oxide (C_2H_4O), Propargyl cyanide (HC_3H_2CN)
(Auxiliary)	Qubit-encoded dipole operators $\hat{\mu}_x$, $\hat{\mu}_y$, $\hat{\mu}_z$; Cartesian coordinates of molecule and harmonic normal modes; VPT2 Calculated transition energies

3 Benchmarking discussion

As noted previously, HamLib is not itself a benchmarking suite. Rather, in conjunction with a set of computational tasks, it may be used to define proper benchmarks. Such computational tasks may include "full" algorithms such as eigenvalue finding or quantum dynamics, but also narrower subroutines such as estimation of expectation values or compilation routines [PWS+16, CFG+22, LSKA, LJV+21, CBG21, TGO+22, YAGJ+21, MPJ+19, WRV+23].

In this section, we discuss the various types of benchmarks that would benefit from a diverse library of Hamiltonians. At its heart, the purpose of benchmarking is to make *comparisons*—answering questions such as "Which of these computers completes the task more quickly?" or "Will this new algorithm yield more accurate results than previous state-of-the-art?" We distinguish between three categories. First, benchmarks that allow one to compare quantum algorithms (independent of hardware choice); second, those that compare different quantum hardware platforms including error-prone devices; and third, those that compare compilation and top-of-stack tasks. Though these three can be considered conceptually independent to some extent, many benchmarking tasks will fall under multiple categories.

Any benchmark involves the comparison of computational resources, accuracy, or both. In the former case, one studies the computational time (related to quantum circuit depth and circuit repetitions) and/or space requirements (number of qubits). Often, there are time-space trade-offs, implying that the best algorithmic choice is hardware-dependent. In the case of studying accuracy, one instead compares which computational approach leads to lower errors. Below, we discuss several possible tasks for which one might define benchmarks using HamLib. Many related quantum benchmarks have been previously proposed in the literature, as discussed in Section 1.2.

Finally, we note that the above tasks may be used either for comparing quantum algorithms against each other ("quantum-vs-quantum"), or for comparing quantum algorithms against classical algorithms ("quantum-vs-classical"). Both types of comparisons are valuable and our intention is for HamLib to be used in both.

3.1 Benchmarking quantum algorithms

Here we discuss benchmarks designed to study quantum algorithms, independently of the quantum hardware used. The primary early use cases we envision for HamLib are eigenvalue finding and quantum dynamics. As mentioned above, these may be analyzed both in terms of solution quality (e.g. accuracy) and in terms of computational resources.

Eigenvalue finding is a common task, both in quantum problems (for which ground or excited states are of interest) and classical optimization (where an extremal eigenvalue corresponds to the optimal value). For extremal eigenvalues (i.e. ground states), several quantum algorithms have been developed. Perhaps most well-studied for near-term hardware are the variational quantum eigensolver (VQE) [MRBAG16] and the quantum approximate optimization algorithm (QAOA) [FGG14]. Both of these involve a plethora of algorithmic choices. Other quantum algorithms for this task include adiabatic quantum optimization (or adiabatic state preparation) [FGGS00], quantum imaginary-time evolution (QITE) [MST+20], feedback-based quantum algorithms [MRGS22], and quantum subspace methods [PM19, KMZC+22]. Finding excited (non-extremal) eigenvalues requires either distinct quantum algorithms or modifications of ground state algorithms. Some notable approaches to excited state finding include the folded spectrum method [WZ94, MRBAG16], quantum variational deflation [HWB19], and quantum equations of motion [OKC+20]; the unmodified versions of QITE and quantum subspace methods naturally provide estimates to excited states.

Notably, each algorithmic component may be benchmarked separately. For VQE and QAOA specifically, there are arguably three main components to consider. First, one may implement different quantum circuit ansatzae, also known as parameterized quantum circuits [CRO+19, GEBM19, XK20, ZKK+21, GZB+20, LOCC22]. Second, one may compare different classical optimizers [MRBAG16, GS17, SMM22]. Third, one may compare methods for calculating expectation values $\langle\psi|H|\psi\rangle$, which is often a very time-consuming subroutine [VYI20, GAD+20, CvSW+21, HKP20, KMZC+22, HMR+21].

Simulating quantum dynamics—i.e. implementing the exponential $e^{-iH\Delta t}$ of a Hamiltonian—is a task distinct from eigenvalue finding and for which there are many algorithmic choices to be made. The most space-efficient approach is to use product formulas such as Suzuki-Trotter decompositions, for which one may consider different orders as well as more complex approaches such as randomized product formulas [Llo96, COS19, CST+21]. If one is primarily concerned with asymptotic scaling, as will often be the case when considering long-term hardware, there are several more advanced Hamiltonian simulation methods that require ancilla qubits [LC17, LSTT23, MOTT23]. Notably, exponentiation is a key subroutine of other algorithms, including the above-mentioned quantum subspace methods and some quantum machine learning protocols.

For the algorithms discussed above, the choice of *encoding* is very important and may be studied as a standalone consideration. Fermionic problems, bosonic and vibrational problems, and discrete com-

binatorial problems all involve rich choices of encoding [CRO+19, SRL12, SMK+20, SGH20, SSH22]. Indeed, simply changing the encoding while leaving the rest of the quantum algorithm unchanged is a valuable study in itself.

We briefly touch on quantum machine learning (QML), which is conceptually distinct from eigenvalue finding and quantum dynamics. Some important machine learning tasks include classification, regression, clustering, and dimensionality reduction. Though such tasks were originally proposed in the context of classical data such as images or user data, some recent work has gone into the "quantum quantum" version of machine learning, whereby a quantum algorithm is used to analyze quantum data [HBM+21]. In principle, some subsets of HamLib may be used in benchmarking for such tasks, as has already been proposed in QML for chemistry [SLS+22].

3.2 Benchmarking quantum hardware

In this section, we discuss procedures designed to benchmark specific quantum devices, especially near-term noisy devices [Pre18]. Our hope is that HamLib spawns a set of benchmarks meant to complement the randomized benchmarking protocols that are often used in the industry [KLR+08]. Such hardware-to-hardware comparisons are one major motivation behind benchmarking in classical computing, for example, when evaluating the performance of a new generation of processor or supercomputer (see Section 1.3).

The most obvious benchmarks in this category involve running eigenvalue finding and Hamiltonian simulation on noisy hardware, to determine which quantum device yields the most accurate result. Though protocols such as quantum volume [CBS+19, MGE12, KLR+08, EWP+19] are valid, transferable, and objective metrics, it is unclear how such metrics relate to the accuracy of a practical workload. It could be that, when applied to scientifically relevant problems, some algorithms are less (or more) sensitive to noise than expected. For instance, it is not implausible that in some instances doubling the decoherence rate will lead to a relatively small reduction in accuracy for some problems. If there is such robustness to noise in a given device, then it is useful to understand its source and to know for which applications it occurs. It is worth mentioning that studying energy consumption of the quantum device will be an important future direction as well [FC07, BDPF+21].

Another class of benchmarks related to hardware involves the choice of encoding a logical qubit into physical qubits. This may involve a choice as simple as rotating each individual qubit basis, to mitigate e.g. amplitude damping noise. Or the consideration may be more complex, such as choosing between different decoherence-free subspaces [LBW03]. Again, it is plausible that the best choice of qubit encoding is problem-dependent, which could be studied using the HamLib library.

Finally, error mitigation protocols form an important area of study. There have been several proposed methods for mitigating errors on NISQ hardware [EBL18, CLK+21, HMO+21]. Benchmarking such procedures for the different problem classes of HamLib may allow one to understand how and whether to apply such techniques in real applications.

3.3 Benchmarking compilation and the full stack

Here we discuss benchmarking of the quantum computing "stack," especially compilation routines [JPK+14, KWP+22, KIMK22]. We place such efforts in a qualitatively separate category because in principle they do *not* require real simulations nor real hardware. In other words, this section discusses benchmarks that do not involve interacting with the actual Hilbert space (whether via quantum hardware or via classical simulators).

Notably, though noisy hardware is likely to be the first testing environment for HamLib, we also intend for HamLib to be used with fully error-corrected hardware in mind. Among other differences, the gate set in error-corrected hardware tends to be different than that used in NISQ computers. Especially important is resource estimation for error-corrected quantum computing. Useful studies of this type have been performed for chemistry, where estimates for required T gate counts have been determined for classically intractable molecules [RWS+17, LLD+19, vBLH+21]. In such cases, because one is not performing the actual quantum computation, one must use informed estimates for quantities such as how many operations will be required to prepare a trial state for QPE and how much overlap this trial state will have with the ground state. Such studies, performed using HamLib, may help researchers begin to understand how required resources scale with problem size, when solving various problems on error-corrected quantum computers.

Mapping problems to qubit representations is often a time-consuming procedure. For instance, fermionic problems must implement the Jordan-Wigner (or related) mapping in order to preserve commutation properties, and the classical steps required may be substantial when one has (not uncommonly) $O(N^4)$ terms to consider. As mentioned above, analogous mappings are required for vibrational and bosonic problems, as well as classical problems. There is likely much room for efficiency improvement in generating qubit representations, for example via pre-processing, memoization, and parallelization.

Methods for quantum circuit construction and optimization may be benchmarked as well [KM13, ASD14, NRS+18, WVMN19, SDC+20, PPJ+21, XLP+22, PSI+23, SIS+23, Dev22]. Both the speed of the compilation and its quality must be considered, where circuit depth might be one quality criterion. As an example, there are a plethora of choices for compiling circuits for Hamiltonian exponentiation, each of which has different accuracy, space, and depth trade-offs [DKPVDW20, SSJM21, KIMK22, PBW22, KLS23]. Further, there are multiple ways to decompose a given unitary [BBC+95], including via randomized compilation [WE16], each of which might be considered by the compiler. Gate fusions and cancellations may be handled with different approaches that may need to be compared. It will be interesting to see the extent to which classical compilation, a mature field of many decades, will bring techniques to bear on quantum circuit compilation.

Finally, we note that there are many aspects of running the quantum stack that are intimately related to a particular hardware design. When two-qubit gates are available only between physically adjacent qubits (which is the case for most hardware types), gate scheduling protocols [MTC+06, GP18] must be benchmarked and the choice of qubit topology (i.e. connectivity) intimately affects performance [HJG+20]. As noted above, multiple choices for decompositions into native gates may be available as well, some of which may lead to more optimal circuits than others. And further into the future, certain aspects of the quantum-classical interface as it relates to quantum error correction ought to be optimized, including the analysis of syndrome measurements [Rof19]. As hardware-software codesign [LWS+21, WSM22] becomes more prevalent in quantum computer development, the HamLib dataset may play a role in guiding many hardware design choices.

4 Concluding remarks

We have introduced a wide-ranging dataset of qubit Hamiltonians, for a variety of quantum problems including condensed matter models, electronic structure, and vibrational structure, as well as classical problems including Max-k-SAT, Max-k-Cut, and the traveling salesperson problem. We have left actual benchmarking studies to future work, using this first work to only introduce and characterize the dataset.

A primary purpose of this work was to provide the community with a set of problem instances that can be used directly and immediately, without the need to learn and implement the tedious multi-step processes required. Preparing electronic and vibrational structure instances for real-world molecules, for instance, involves significant domain knowledge and many software steps. Further, our inclusion of problems with real-world parameters and attributes (electronic structure, vibrational structure, TSP for European cities) is meant to encourage the community to continue its focus on problems instances that more closely relate to industrial impact.

HamLib may be used for several distinct types of tests: (a) simulating small problems on current noisy quantum computers, (b) classically emulating quantum computers, or (c) studying processes for which the quantum circuit itself does not need to be simulated, *e.g.* for compilation procedures up to hundreds or thousands of qubits. These Hamiltonians may be implemented in conjunction with a range of algorithm classes for which new algorithmic approaches need to be tested or hyperparameters tuned: VQE, QAOA, adiabatic quantum state preparation, Krylov-subspace methods, and various long-term Hamiltonian simulation methods.

Quality classical datasets have long been an indispensable component of benchmarking for the design of supercomputers, processors, linear algebra methods, and AI algorithms. We hope that this dataset fills a similar role for the quantum computing community.

Acknowledgements

We thank RoaRQ consortium for useful discussions on dataset coverage, Shrihan Agarwal for useful discussions on TSP, Timothy Proctor for manuscript feedback, and Shavindra Premaratne and Neil Mehta for testing portions of the dataset. We acknowledge support from Sandia National Laboratories' Laboratory Directed Research and Development Program under the Truman Fellowship. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. This material is based upon work supported by the U.S. Department of Energy (DOE), Office of Science, National Quantum Information Science Research Centers, Quantum Systems Accelerator. Additional support is acknowledged from DOE, Office of Science, Office of Advanced Scientific Computing Research, Accelerated Research in Quantum Computing, Fundamental Algorithmic Research in Quantum Computing. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231. The authors would like to acknowledge the use of the University of Oxford Advanced Research Computing (ARC) facility in carrying out this work (http://dx.doi.org/10.5281/zenodo.22558). NMT is grateful for support from NASA Ames Research Center. We acknowledge funding from the NASA ARMD Transformational Tools and Technology (TTT) Project. DMD acknowledges financial support by the EPSRC Hub in Quantum Computing and Simulation (EP/T001062/1). Part of this work is funded by U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Co-Design Center for Quantum Advantage under Contract No. DE-SC0012704. D.B.N. was supported by NASA Academic Mission Services, Contract No. NNA16BD14C.

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[AGM20] Anurag Anshu, David Gosset, and Karen Morenz. Beyond Product State Approximations for a Quantum Analogue of Max Cut. In Steven T. Flammia, editor, *15th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2020)*, volume 158 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 7:1–7:15, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[Akı13] Ümit Akıncı. Critical behavior of the 3d anisotropic quantum heisenberg model in a trimodal random field distribution. *Journal of magnetism and magnetic materials*, 333:156–161, 2013.

[ANP05] Dimitris Achlioptas, Assaf Naor, and Yuval Peres. Rigorous location of phase transitions in hard optimization problems. *Nature*, 435(7043):759–764, 2005.

[APMB20] Vishwanathan Akshay, Hariphan Philathong, Mauro ES Morales, and Jacob D Biamonte. Reachability deficits in quantum approximate optimization. *Physical review letters*, 124(9):090504, 2020.

[ASD14] Nabila Abdessaied, Mathias Soeken, and Rolf Drechsler. Quantum circuit optimization by hadamard gate reduction. In *Reversible Computation: 6th International Conference, RC 2014, Kyoto, Japan, July 10-11, 2014. Proceedings 6*, pages 149–162. Springer, 2014.

[BBC+95] Adriano Barenco, Charles H Bennett, Richard Cleve, David P DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical review A*, 52(5):3457, 1995.

[BD02] Henk WJ Blöte and Youjin Deng. Cluster monte carlo simulation of the transverse ising model. *Physical Review E*, 66(6):066110, 2002.

[BDPF+21] Casey Berger, Agustin Di Paolo, Tracey Forrest, Stuart Hadfield, Nicolas Sawaya, Michal Stechly, and Karl Thibault. Quantum technologies for climate change: Preliminary assessment. *arXiv preprint arXiv:2107.05362*, 2021.

[BDVD+98] Rainer E Burkard, Vladimir G Deineko, Rene Van Dal, Jack AA van der Veen, and Gerhard J Woeginger. Well-solvable special cases of the traveling salesman problem: a survey. *SIAM review*, 40(3):496–546, 1998.

[BIS+18] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, Shahnawaz Ahmed, Vishnu Ajith, M Sohaib Alam, Guillermo Alonso-Linaje, B AkashNarayanan, Ali Asadi, et al. Pennylane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968*, 2018.

[BM21] Sami Boulebnane and Ashley Montanaro. Predicting parameters for the quantum approximate optimization algorithm for max-cut from the infinite-size limit. *arXiv preprint arXiv:2110.10685*, 2021.

[BMDTW22] Hugh G. A. Burton, Daniel Marti-Dafcik, David P. Tew, and David J. Wales. Exact electronic states with shallow quantum circuits through global optimisation. *arXiv preprint arXiv:2207.00085*, 2022.

[BSZK95] G. G. Batrouni, R. T. Scalettar, G. T. Zimanyi, and A. P. Kampf. Supersolids in the Bose-Hubbard hamiltonian. *Phys. Rev. Lett.*, 74:2527–2530, Mar 1995.

[BWF+00] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, TN Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[CBG21] Arjan Cornelissen, Johannes Bausch, and András Gilyén. Scalable benchmarks for gate-based quantum computers. *arXiv preprint arXiv:2104.10698*, 2021.

[CBS+19] 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.

[CFG+22] Kean Chen, Wang Fang, Ji Guan, Xin Hong, Mingyu Huang, Junyi Liu, Qisheng Wang, and Mingsheng Ying. VeriQBench: A benchmark for multiple types of quantum circuits. *arXiv preprint arXiv:2206.10880*, 2022.

[Cha19] Nicholas Chancellor. Domain wall encoding of discrete variables for quantum annealing and qaoa. *Quantum Science and Technology*, 4(4):045004, 2019.

[CLK+21] Ningping Cao, Junan Lin, David Kribs, Yiu-Tung Poon, Bei Zeng, and Raymond Laflamme. NISQ: Error correction, mitigation, and noise simulation. *arXiv preprint arXiv:2111.02345*, 2021.

[CMN+18] Andrew M Childs, Dmitri Maslov, Yunseong Nam, Neil J Ross, and Yuan Su. Toward the first quantum simulation with quantum speedup. *Proceedings of the National Academy of Sciences*, 115(38):9456–9461, 2018.

[col] COLOR02: Graph Coloring and its Generalizations. https://mat.tepper.cmu.edu/COLOR02/. Accessed: 2023-04-14.

[COS19] Andrew M Childs, Aaron Ostrander, and Yuan Su. Faster quantum simulation by randomization. *Quantum*, 3:182, 2019.

[CRO+19] Yudong Cao, Jonathan Romero, Jonathan P Olson, Matthias Degroote, Peter D Johnson, Mária Kieferová, Ian D Kivlichan, Tim Menke, Borja Peropadre, Nicolas PD Sawaya, et al. Quantum chemistry in the age of quantum computing. *Chemical reviews*, 119(19):10856–10915, 2019.

[CST+21] Andrew M Childs, Yuan Su, Minh C Tran, Nathan Wiebe, and Shuchen Zhu. Theory of Trotter error with commutator scaling. *Physical Review X*, 11(1):011020, 2021.

[CVH+22] M Cerezo, Guillaume Verdon, Hsin-Yuan Huang, Lukasz Cincio, and Patrick J Coles. Challenges and opportunities in quantum machine learning. *Nature Computational Science*, 2(9):567–576, 2022.

[CvSW+21] Ophelia Crawford, Barnaby van Straaten, Daochen Wang, Thomas Parks, Earl Campbell, and Stephen Brierley. Efficient quantum measurement of Pauli operators in the presence of finite sampling error. *Quantum*, 5:385, January 2021.

[DCD+06] Pierre Dupont, Jérôme Callut, Grégoire Dooms, Jean-Noël Monette, Yves Deville, and BP Sainte. Relevant subgraph extraction from random walks in a graph. *Universite Catholique de Louvain, UCL/INGI, Number RR*, 7, 2006.

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[DDSG+20] Pierre-Luc Dallaire-Demers, Michal Stechly, Jerome F Gonthier, Ntwali Toussaint Bashige, Jonathan Romero, and Yudong Cao. An application benchmark for fermionic quantum simulations. *arXiv preprint arXiv:2003.01862*, 2020.

[Dev22] Cirq Developers. Cirq, December 2022. See full list of authors on Github: https://github .com/quantumlib/Cirq/graphs/contributors.

[DGS18] Iain Dunning, Swati Gupta, and John Silberholz. What works best when? a systematic evaluation of heuristics for Max-Cut and QUBO. *INFORMS Journal on Computing*, 30(3):608–624, 2018.

[Dix91] Kaivalya M Dixit. The SPEC benchmarks. *Parallel computing*, 17(10-11):1195–1209, 1991.

[DKPVDW20] Ross Duncan, Aleks Kissinger, Simon Perdrix, and John Van De Wetering. Graph-theoretic simplification of quantum circuits with the ZX-calculus. *Quantum*, 4:279, 2020.

[DL21] Yulong Dong and Lin Lin. Random circuit block-encoded matrix and a proposal of quantum linpack benchmark. *Physical Review A*, 103(6):062412, 2021.

[DLH13] Jack Dongarra, Piotr Luszczek, and M Heroux. HPCG technical specification. *Sandia National Laboratories, Sandia Report SAND2013-8752*, 2013.

[DLP03] Jack J Dongarra, Piotr Luszczek, and Antoine Petitet. The linpack benchmark: past, present and future. *Concurrency and Computation: practice and experience*, 15(9):803–820, 2003.

[EAD+20] Janus J Eriksen, Tyler A Anderson, J Emiliano Deustua, Khaldoon Ghanem, Diptarka Hait, Mark R Hoffmann, Seunghoon Lee, Daniel S Levine, Ilias Magoulas, Jun Shen, et al. The ground state electronic energy of benzene. *arXiv preprint arXiv:2008.02678*, 2020.

[EBL18] Suguru Endo, Simon C Benjamin, and Ying Li. Practical quantum error mitigation for near-future applications. *Physical Review X*, 8(3):031027, 2018.

[EHW+20] Jens Eisert, Dominik Hangleiter, Nathan Walk, Ingo Roth, Damian Markham, Rhea Parekh, Ulysse Chabaud, and Elham Kashefi. Quantum certification and benchmarking. *Nature Reviews Physics*, 2(7):382–390, 2020.

[ER+60] Paul Erdős, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

[EWP+19] 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.

[FC07] Wu-chun Feng and Kirk Cameron. The green500 list: Encouraging sustainable supercomputing. *Computer*, 40(12):50–55, 2007.

[FGG14] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[FGG20] Edward Farhi, David Gamarnik, and Sam Gutmann. The quantum approximate optimization algorithm needs to see the whole graph: A typical case. *arXiv preprint arXiv:2004.09002*, 2020.

[FGGS00] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*, 2000.

[FJ97] A. Frieze and M. Jerrum. Improved approximation algorithms for MAXk-CUT and MAX BISECTION. *Algorithmica*, 18(1):67–81, May 1997.

[FM94] JK Freericks and H Monien. Phase diagram of the Bose-Hubbard model. *EPL (Europhysics Letters)*, 26(7):545, 1994.

[Fra17] Fabio Franchini. *An Introduction to Integrable Techniques for One-Dimensional Quantum Systems*. Springer International Publishing, 2017.

[FRH+22] Jernej Rudi Finžgar, Philipp Ross, Leonhard Hölscher, Johannes Klepsch, and Andre Luckow. Quark: A framework for quantum computing application benchmarking. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 226–237. IEEE, 2022.

[FWGF89] Matthew P. A. Fisher, Peter B. Weichman, G. Grinstein, and Daniel S. Fisher. Boson localization and the superfluid-insulator transition. *Phys. Rev. B*, 40:546–570, Jul 1989.

[GAD+20] Pranav Gokhale, Olivia Angiuli, Yongshan Ding, Kaiwen Gui, Teague Tomesh, Martin Suchara, Margaret Martonosi, and Frederic T. Chong. $O(N^3)$ measurement cost for variational quantum eigensolver on molecular hamiltonians. *IEEE Transactions on Quantum Engineering*, 1:1–24, 2020.

[GCY+11] Xiutang Geng, Zhihua Chen, Wei Yang, Deqian Shi, and Kai Zhao. Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search. *Applied Soft Computing*, 11(4):3680–3689, 2011.

[GEBM19] Harper R. Grimsley, Sophia E. Economou, Edwin Barnes, and Nicholas J. Mayhall. An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nature Communications*, 10, 2019.

[GHHP13] Philippe Galinier, Jean-Philippe Hamiez, Jin-Kao Hao, and Daniel Porumbel. *Recent Advances in Graph Vertex Coloring*, pages 505–528. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[GJS19] Etienne Granet, Jesper Lykke Jacobsen, and Hubert Saleur. Analytical results on the heisenberg spin chain in a magnetic field. *Journal of Physics A: Mathematical and Theoretical*, 52(25):255302, may 2019.

[GP18] Gian Giacomo Guerreschi and Jongsoo Park. Two-step approach to scheduling quantum circuits. *Quantum Science and Technology*, 3(4):045003, 2018.

[GP19] Sevag Gharibian and Ojas Parekh. Almost Optimal Classical Approximation Algorithms for a Quantum Generalization of Max-Cut. In Dimitris Achlioptas and László A. Végh, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*, volume 145 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31:1–31:17, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[GS17] Gian Giacomo Guerreschi and Mikhail Smelyanskiy. Practical optimization for hybrid quantum-classical algorithms. *arXiv preprint arXiv:1701.01450*, 2017.

[GW95] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

[GW96] Ian P Gent and Toby Walsh. The TSP phase transition. *Artificial Intelligence*, 88(1-2):349–358, 1996.

[GZB+20] Bryan T Gard, Linghua Zhu, George S Barron, Nicholas J Mayhall, Sophia E Economou, and Edwin Barnes. Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm. *npj Quantum Information*, 6(1):1–9, 2020.

[HBM+21] Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R McClean. Power of data in quantum machine learning. *Nature communications*, 12(1):2631, 2021.

[HJG+20] Adam Holmes, Sonika Johri, Gian Giacomo Guerreschi, James S Clarke, and Anne Y Matsuura. Impact of qubit connectivity on quantum algorithm performance. *Quantum Science and Technology*, 5(2):025009, 2020.

[HKP20] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, June 2020.

[HMO+21] William J Huggins, Sam McArdle, Thomas E O'Brien, Joonho Lee, Nicholas C Rubin, Sergio Boixo, K Birgitta Whaley, Ryan Babbush, and Jarrod R McClean. Virtual distillation for quantum error mitigation. *Physical Review X*, 11(4):041036, 2021.

[HMR+21] William J. Huggins, Jarrod R. McClean, Nicholas C. Rubin, Zhang Jiang, Nathan Wiebe, K. Birgitta Whaley, and Ryan Babbush. Efficient and noise resilient measurements for quantum chemistry on near-term quantum computers. *npj Quantum Information*, 7(1), February 2021.

[HNP+23] Yeongwoo Hwang, Joe Neeman, Ojas Parekh, Kevin Thompson, and John Wright. Unique games hardness of Quantum Max-Cut, and a conjectured vector-valued Borell's inequality. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1319–1384. SIAM, 2023.

[HS00] Holger H Hoos and Thomas Stützle. SATLIB: An online resource for research on sat. *Sat*, 2000:283–292, 2000.

[HSS08] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

[HTL+19] Diptarka Hait, Norman M Tubman, Daniel S Levine, K Birgitta Whaley, and Martin Head-Gordon. What levels of coupled cluster theory are appropriate for transition metal systems? a study using near-exact quantum chemical values for 3d transition metal binary compounds. *Journal of chemical theory and computation*, 15(10):5370–5385, 2019.

[Hub63] John Hubbard. Electron correlations in narrow energy bands. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 276(1365):238–257, 1963.

[HV02] Johan Håstad and S. Venkatesh. On the advantage over a random assignment. In *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*, STOC '02, page 43–52, New York, NY, USA, 2002. Association for Computing Machinery.

[HWB19] Oscar Higgott, Daochen Wang, and Stephen Brierley. Variational quantum computation of excited states. *Quantum*, 3:156, 2019.

[HWO+19] Stuart Hadfield, Zhihui Wang, Bryan O'gorman, Eleanor G Rieffel, Davide Venturelli, and Rupak Biswas. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms*, 12(2):34, 2019.

[HWPC05] Klemens Hammerer, Michael M Wolf, Eugene Simon Polzik, and J Ignacio Cirac. Quantum benchmark for storage and transmission of coherent states. *Physical review letters*, 94(15):150503, 2005.

[INE+22] Yohei Ibe, Yuya O Nakagawa, Nathan Earnest, Takahiro Yamamoto, Kosuke Mitarai, Qi Gao, and Takao Kobayashi. Calculating transition amplitudes by variational quantum deflation. *Physical Review Research*, 4(1):013173, 2022.

[JMS12] Piotr Jankowski, ARW McKellar, and Krzysztof Szalewicz. Theory untangles the high-resolution infrared spectrum of the ortho-H2-CO van der Waals complex. *Science*, 336(6085):1147–1150, 2012.

[JN08] Matti Järvisalo and Ilkka Niemelä. The effect of structural branching on the efficiency of clause learning sat solving: An experimental study. *Journal of Algorithms*, 63(1-3):90–113, 2008.

[JOH+13] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, et al. Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. *APL materials*, 1(1):011002, 2013.

[JP80] K. Jankowski and J. Paldus. Applicability of coupled-pair theories to quasidegenerate electronic states: A model study. *International Journal of Quantum Chemistry*, 18:1243–1269, 11 1980.

[JPK+14] Ali JavadiAbhari, Shruti Patil, Daniel Kudrow, Jeff Heckey, Alexey Lvov, Frederic T Chong, and Margaret Martonosi. ScaffCC: A framework for compilation and analysis of quantum computing programs. In *Proceedings of the 11th ACM Conference on Computing Frontiers*, pages 1–10, 2014.

[JSP+21] Hamza Jnane, Nicolas PD Sawaya, Borja Peropadre, Alan Aspuru-Guzik, Raul Garcia-Patron, and Joonsuk Huh. Analog quantum simulation of non-condon effects in molecular spectroscopy. *ACS Photonics*, 8(7):2007–2016, 2021.

[Kar96] Howard Karloff. How good is the Goemans-Williamson MAX CUT algorithm? In *Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing*, pages 427–434, 1996.

[KIMK22] Y Kharkov, A Ivanova, E Mikhantiev, and A Kotelnikov. Arline benchmarks: Automated benchmarking platform for quantum compilers. *arXiv preprint arXiv:2202.14025*, 2022.

[Kin22] Robbie King. An improved approximation algorithm for quantum max-cut. *arXiv preprint arXiv:2209.02589*, 2022.

[KKMO07] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM Journal on Computing*, 37(1):319–357, 2007.

[KLMT00] Emanuel Knill, R Laflamme, R Martinez, and C-H Tseng. An algorithmic benchmark for quantum information processing. *Nature*, 404(6776):368–370, 2000.

[KLR+08] Emanuel Knill, Dietrich Leibfried, Rolf Reichle, Joe Britton, R Brad Blakestad, John D Jost, Chris Langer, Roee Ozeri, Signe Seidelin, and David J Wineland. Randomized benchmarking of quantum gates. *Physical Review A*, 77(1):012307, 2008.

[KLS23] Andrey Boris Khesin, Jonathan Z Lu, and Peter W Shor. Graphical quantum clifford-encoder compilers from the ZX calculus. *arXiv preprint arXiv:2301.02356*, 2023.

[KM13] Vadym Kliuchnikov and Dmitri Maslov. Optimization of Clifford circuits. *Physical Review A*, 88(5):052307, 2013.

[KMZC+22] 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. Real-time evolution for ultracompact hamiltonian eigenstates on quantum hardware. *PRX Quantum*, 3:020323, 5 2022.

[KS19] Natalie Klco and Martin J Savage. Digitization of scalar fields for quantum computing. *Physical Review A*, 99(5):052335, 2019.

[KWP+22] Pradnya Khalate, Xin-Chuan Wu, Shavindra Premaratne, Justin Hogaboam, Adam Holmes, Albert Schmitz, Gian Giacomo Guerreschi, Xiang Zou, and AY Matsuura. An LLVM-based C++ compiler toolchain for variational hybrid quantum-classical algorithms and quantum accelerators. *arXiv preprint arXiv:2202.11142*, 2022.

[LAB+15] James PF LeBlanc, Andrey E Antipov, Federico Becca, Ireneusz W Bulik, Garnet Kin-Lic Chan, Chia-Min Chung, Youjin Deng, Michel Ferrero, Thomas M Henderson, Carlos A Jiménez-Hoyos, et al. Solutions of the two-dimensional Hubbard model: Benchmarks and results from a wide range of numerical algorithms. *Physical Review X*, 5(4):041041, 2015.

[Lap92] Gilbert Laporte. The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(2):231–247, 1992.

[LBG+21] Joonho Lee, Dominic W. Berry, Craig Gidney, William J. Huggins, Jarrod R. McClean, Nathan Wiebe, and Ryan Babbush. Even more efficient quantum computations of chemistry through tensor hypercontraction. *PRX Quantum*, 2:030305, Jul 2021.

[LBW03] Daniel A Lidar and K Birgitta Whaley. Decoherence-free subspaces and subsystems. In *Irreversible quantum dynamics*, pages 83–120. Springer, 2003.

[LC17] Guang Hao Low and Isaac L Chuang. Optimal hamiltonian simulation by quantum signal processing. *Physical review letters*, 118(1):010501, 2017.

[LCM+23] Thomas Lubinski, Carleton Coffrin, Catherine McGeoch, Pratik Sathe, Joshua Apanavicius, and David E Bernal Neira. Optimization applications as quantum performance benchmarks. *arXiv preprint arXiv:2302.02278*, 2023.

[Lee22] Eunou Lee. Optimizing quantum circuit parameters via SDP. *arXiv preprint arXiv:2209.00789*, 2022.

[LJV+21] Thomas Lubinski, Sonika Johri, Paul Varosy, Jeremiah Coleman, Luning Zhao, Jason Necaise, Charles H. Baldwin, Karl Mayer, and Timothy Proctor. Application-oriented performance benchmarks for quantum computing, 2021.

[LLD+19] Zhendong Li, Junhao Li, Nikesh S Dattani, CJ Umrigar, and Garnet Kin Chan. The electronic complexity of the ground-state of the FeMo cofactor of nitrogenase as relevant to quantum simulations. *The Journal of chemical physics*, 150(2), 2019.

[Llo96] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 1996.

[LMM06] C. Lucet, F. Mendes, and A. Moukrim. An exact method for graph coloring. *Computers and Operations Research*, 33(8):2189–2207, 2006.

[LMS12] Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Kernelization-preprocessing with a guarantee. *The Multivariate Algorithmic Revolution and Beyond*, 7370:129–161, 2012.

[LOCC22] Lorenzo Leone, Salvatore FE Oliviero, Lukasz Cincio, and M Cerezo. On the practical usefulness of the hardware efficient ansatz. *arXiv preprint arXiv:2211.01477*, 2022.

[LSKA] Ang Li, Samuel Stein, Sriram Krishnamoorthy, and James Ang. QASMBench: A low-level quantum benchmark suite for NISQ evaluation and simulation. *ACM Transactions on Quantum Computing*.

[LSTT23] Guang Hao Low, Yuan Su, Yu Tong, and Minh C Tran. *PRX Quantum*, 4:020323, 2023.

[LSY21] Junyu Liu, Jinzhao Sun, and Xiao Yuan. Towards a variational Jordan-Lee-Preskill quantum algorithm. *arXiv preprint arXiv:2109.05547*, 2021.

[LW03] Elliott H Lieb and FY Wu. The one-dimensional Hubbard model: a reminiscence. *Physica A: statistical mechanics and its applications*, 321(1-2):1–27, 2003.

[LWS+21] Gushu Li, Anbang Wu, Yunong Shi, Ali Javadi-Abhari, Yufei Ding, and Yuan Xie. On the co-design of quantum software and hardware. In *Proceedings of the Eight Annual ACM International Conference on Nanoscale Computing and Communication*, pages 1–7, 2021.

[MAAM21] Koen Mesman, Zaid Al-Ars, and Matthias Möller. QPack: Quantum approximate optimization algorithms as universal benchmark for quantum computers, 2021.

[MB18] Mohammad Mohammadi and Timur Bazhirov. Comparative benchmarking of cloud computing vendors with high performance linpack. In *Proceedings of the 2nd International Conference on High Performance Compilation, Computing and Communications*, pages 1–5, 2018.

[MCC+17] Mario Motta, David M. Ceperley, Garnet Kin Lic Chan, John A. Gomez, Emanuel Gull, S. Guo, Carlos A. Jiménez-Hoyos, Tran Nguyen Lan, Jia Li, Fengjie Ma, Andrew J. Millis, Nikolay V. Prokof'ev, Ushnish Ray, Gustavo E. Scuseria, Sandro Sorella, Edwin M. Stoudenmire, Qiming Sun, Igor S. Tupitsyn, Steven R. White, Dominika Zgid, and Shiwei Zhang. Towards the solution of the many-electron problem in real materials: Equation of state of the hydrogen chain with state-of-the-art many-body methods. *Physical Review X*, 7:031059, 9 2017.

[MCH+20] Devin A Matthews, Lan Cheng, Michael E Harding, Filippo Lipparini, Stella Stopkowicz, Thomas-C Jagau, Péter G Szalay, Jürgen Gauss, and John F Stanton. Coupled-cluster techniques for computational chemistry: The CFOUR program package. *The Journal of Chemical Physics*, 152(21):214108, 2020.

[MGE12] Easwar Magesan, Jay M Gambetta, and Joseph Emerson. Characterizing quantum gates via randomized benchmarking. *Physical Review A*, 85(4):042311, 2012.

[MMB+10] Richard C Murphy, Kamesh Madduri, David A Bader, Aydın Buluç, Jack J Dongarra, Simon D Hammond, Hartmut Kaiser, Manjunath Kudlur, David A Padua, and Samuel W Williams. Introducing the graph 500 benchmark. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, pages 1–11. ACM, 2010.

[MMS+19] Sam McArdle, Alexander Mayorov, Xiao Shan, Simon Benjamin, and Xiao Yuan. Digital quantum simulation of molecular vibrations. *Chemical science*, 10(22):5725–5735, 2019.

[MOTT23] Alexander Miessen, Pauline J Ollitrault, Francesco Tacchino, and Ivano Tavernelli. Quantum algorithms for quantum dynamics. *Nature Computational Science*, 3(1):25–37, 2023.

[MPJ+19] 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.

[MPR21] Tina Mathea, Taras Petrenko, and Guntram Rauhut. VCI calculations based on canonical and localized normal coordinates for non-abelian molecules: accurate assignment of the vibrational overtones of allene. *The Journal of Physical Chemistry A*, 125(4):990–998, 2021.

[MRBAG16] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, 2016.

[MRC+20] Peter Mattson, Vijay Janapa Reddi, Christine Cheng, Cody Coleman, Greg Diamos, David Kanter, Paulius Micikevicius, David Patterson, Guenther Schmuelling, Hanlin Tang, et al. MLPerf: An industry standard benchmark suite for machine learning performance. *IEEE Micro*, 40(2):8–16, 2020.

[MRGS22] Alicia B. Magann, Kenneth M. Rudinger, Matthew D. Grace, and Mohan Sarovar. Feedback-based quantum optimization. *Phys. Rev. Lett.*, 129:250502, Dec 2022.

[MRS+20] Jarrod R McClean, Nicholas C Rubin, Kevin J Sung, Ian D Kivlichan, Xavier Bonet-Monroig, Yudong Cao, Chengyu Dai, E Schuyler Fried, Craig Gidney, Brendan Gimby, Pranav Gokhale, Thomas Häner, Tarini Hardikar, Vojtěch Havlíček, Oscar Higgott, Cupjin Huang, Josh Izaac, Zhang Jiang, Xinle Liu, Sam McArdle, Matthew Neeley, Thomas O'Brien, Bryan O'Gorman, Isil Ozfidan, Maxwell D Radin, Jhonathan Romero, Nicolas P D Sawaya, Bruno Senjean, Kanav Setia, Sukin Sim, Damian S Steiger, Mark Steudtner, Qiming Sun, Wei Sun, Daochen Wang, Fang Zhang, and Ryan Babbush. OpenFermion: the electronic structure package for quantum computers. *Quantum Science and Technology*, 5(3):034014, jun 2020.

[MSAH18a] Alexandru Macridin, Panagiotis Spentzouris, James Amundson, and Roni Harnik. Digital quantum computation of fermion-boson interacting systems. *Physical Review A*, 98(4):042312, 2018.

[MSAH18b] Alexandru Macridin, Panagiotis Spentzouris, James Amundson, and Roni Harnik. Electron-phonon systems on a universal quantum computer. *Physical review letters*, 121(11):110504, 2018.

[MST+20] Mario Motta, Chong Sun, Adrian TK Tan, Matthew J O'Rourke, Erika Ye, Austin J Minnich, Fernando GSL Brandão, and Garnet Kin-Lic Chan. Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution. *Nature Physics*, 16(2):205–210, 2020.

[MTC+06] Tzvetan S Metodi, Darshan D Thaker, Andrew W Cross, Frederic T Chong, and Isaac L Chuang. Scheduling physical operations in a quantum information processor. In *Quantum Information and Computation IV*, volume 6244, pages 210–221. SPIE, 2006.

[NMP+23] Akimoto Nakayama, Kosuke Mitarai, Leonardo Placidi, Takanori Sugimoto, and Keisuke Fujii. Vqe-generated quantum circuit dataset for machine learning. *arXiv preprint arXiv:2302.09751*, 2023.

[NRS+18] Yunseong Nam, Neil J Ross, Yuan Su, Andrew M Childs, and Dmitri Maslov. Automated optimization of large quantum circuits with continuous parameters. *npj Quantum Information*, 4(1):23, 2018.

[OBRT20] Pauline J Ollitrault, Alberto Baiardi, Markus Reiher, and Ivano Tavernelli. Hardware efficient quantum algorithms for vibrational structure calculations. *Chemical science*, 11(26):6842–6855, 2020.

[OKC+20] Pauline J Ollitrault, Abhinav Kandala, Chun-Fu Chen, Panagiotis Kl Barkoutsos, Antonio Mezzacapo, Marco Pistoia, Sarah Sheldon, Stefan Woerner, Jay M Gambetta, and Ivano Tavernelli. Quantum equation of motion for computing molecular excitation energies on a noisy quantum processor. *Physical Review Research*, 2(4):043140, 2020.

[Pap03] Christos H Papadimitriou. Computational complexity. In *Encyclopedia of computer science*, pages 260–265. 2003.

[Par23] Ojas Parekh. Synergies between operations research and quantum information science. *INFORMS J. on Computing*, 35(2):266–273, 2023.

[PBW22] Tom Peham, Lukas Burgholzer, and Robert Wille. Equivalence checking of quantum circuits with the ZX-calculus. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 12(3):662–675, 2022.

[PM15] Stephen Piddock and Ashley Montanaro. The complexity of antiferromagnetic interactions and 2D lattices. *arXiv preprint arXiv:1506.04014*, 2015.

[PM19] Robert M Parrish and Peter L McMahon. Quantum filter diagonalization: Quantum eigendecomposition without full quantum phase estimation. *arXiv preprint arXiv:1909.08925*, 2019.

[PP05] Ramesh V Pai and Rahul Pandit. Superfluid, mott-insulator, and mass-density-wave phases in the one-dimensional extended Bose-Hubbard model. *Physical Review B*, 71(10):104508, 2005.

[PPJ+21] Jessica Pointing, Oded Padon, Zhihao Jia, Henry Ma, Auguste Hirth, Jens Palsberg, and Alex Aiken. Quanto: Optimizing quantum circuits with automatic generation of circuit identities. *arXiv preprint arXiv:2111.11387*, 2021.

[Pre18] John Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, 2018.

[PRY+22] 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.

[PSI+23] Jennifer Paykin, Albert T Schmitz, Mohannad Ibrahim, Xin-Chuan Wu, and AY Matsuura. PCOAST: a Pauli-based quantum circuit optimization framework. *arXiv preprint arXiv:2305.10966*, 2023.

[PT21] Ojas Parekh and Kevin Thompson. Application of the Level-2 Quantum Lasserre Hierarchy in Quantum Approximation Algorithms. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 102:1–102:20, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[PT22] Ojas Parekh and Kevin Thompson. An optimal product-state approximation for 2-local quantum hamiltonians with positive terms. *arXiv preprint arXiv:2206.08342*, 2022.

[PWS+16] Ojas Parekh, Jeremy Wendt, Luke Shulenburger, Andrew Landahl, Jonathan Moussa, and John Aidun. Benchmarking adiabatic quantum optimization for complex network analysis. *arXiv preprint arXiv:1604.00319*, 2016.

[PYF21] Elija Perrier, Akram Youssry, and Chris Ferrie. QDataset: Quantum datasets for machine learning, 2021.

[RC19] Alessandro Roggero and Joseph Carlson. Dynamic linear response quantum algorithm. *Phys. Rev. C*, 100:034610, Sep 2019.

[RDRVL14] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

[Rei91] Gerhard Reinelt. TSPLIB—a traveling salesman problem library. *ORSA journal on computing*, 3(4):376–384, 1991.

[RK21] Salonik Resch and Ulya R Karpuzcu. Benchmarking quantum computers and the impact of quantum noise. *ACM Computing Surveys (CSUR)*, 54(7):1–35, 2021.

[Rof19] Joschka Roffe. Quantum error correction: an introductory guide. *Contemporary Physics*, 60(3):226–245, 2019.

[RWS+17] Markus Reiher, Nathan Wiebe, Krysta M Svore, Dave Wecker, and Matthias Troyer. Elucidating reaction mechanisms on quantum computers. *Proceedings of the national academy of sciences*, 114(29):7555–7560, 2017.

[SA70] Maurice E Schwartz and Leland Cullen Allen. Ab initio studies of the electronic structures of bh3 bh2f, bhf2 and bf3. *Journal of the American Chemical Society*, 92(6):1466–1471, 1970.

[Saw22] Nicolas PD Sawaya. mat2qubit: A lightweight pythonic package for qubit encodings of vibrational, bosonic, graph coloring, routing, scheduling, and general matrix problems, 2022.

[SDC+20] Seyon Sivarajah, Silas Dilkes, Alexander Cowtan, Will Simmons, Alec Edgington, and Ross Duncan. t|ket>: a retargetable compiler for NISQ devices. *Quantum Science and Technology*, 6(1):014003, 2020.

[SE20] Nicholas H. Stair and Francesco A. Evangelista. Exploring Hilbert space on a budget: Novel benchmark set and performance metric for testing electronic structure methods in the regime of strong correlation. *Journal of Chemical Physics*, 153:104108, 9 2020.

[SGH20] Nicolas PD Sawaya, Gian Giacomo Guerreschi, and Adam Holmes. On connectivity-dependent resource requirements for digital quantum simulation of d-level particles. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 180–190. IEEE, 2020.

[SH19] Nicolas PD Sawaya and Joonsuk Huh. Quantum algorithm for calculating molecular vibronic spectra. *The journal of physical chemistry letters*, 10(13):3586–3591, 2019.

[SH22] Nicolas PD Sawaya and Joonsuk Huh. Improved resource-tunable near-term quantum algorithms for transition probabilities, with applications in physics and variational quantum linear algebra. *arXiv preprint arXiv:2206.14213*, 2022.

[SHE20] Nicholas H. Stair, Renke Huang, and Francesco A. Evangelista. A multireference quantum Krylov algorithm for strongly correlated electrons. *Journal of Chemical Theory and Computation*, 16:2236–2245, 4 2020.

[SIS+23] Albert T Schmitz, Mohannad Ibrahim, Nicolas PD Sawaya, Gian Giacomo Guerreschi, Jennifer Paykin, Xin-Chuan Wu, and AY Matsuura. Optimization at the interface of unitary and non-unitary quantum operations in PCOAST. *arXiv preprint arXiv:2305.09843*, 2023.

[SLS+22] Manas Sajjan, Junxu Li, Raja Selvarajan, Shree Hari Sureshbabu, Sumit Suresh Kale, Rishabh Gupta, Vinit Singh, and Sabre Kais. Quantum machine learning for chemistry and physics. *Chemical Society Reviews*, 2022.

[SMK+20] Nicolas PD Sawaya, Tim Menke, Thi Ha Kyaw, Sonika Johri, Alán Aspuru-Guzik, and Gian Giacomo Guerreschi. Resource-efficient digital quantum simulation of d-level systems for photonic, vibrational, and spin-s hamiltonians. *npj Quantum Information*, 6(1):1–13, 2020.

[SMM22] Harshdeep Singh, Sabyashachi Mishra, and Sonjoy Majumder. Benchmarking of different optimizers in the variational quantum algorithms for applications in quantum chemistry. *arXiv preprint arXiv:2208.10285*, 2022.

[Som05] Rolando D Somma. Quantum computation, complexity, and many-body physics. *arXiv preprint quant-ph/0512209*, 2005.

[SPT21] Nicolas PD Sawaya, Francesco Paesani, and Daniel P Tabor. Near-and long-term quantum algorithmic approaches for vibrational spectroscopy. *Physical Review A*, 104(6):062419, 2021.

[SRL12] Jacob T Seeley, Martin J Richard, and Peter J Love. The Bravyi-Kitaev transformation for quantum computation of electronic structure. *The Journal of chemical physics*, 137(22):224109, 2012.

[SSH22] Nicolas PD Sawaya, Albert T Schmitz, and Stuart Hadfield. Encoding trade-offs and design toolkits in quantum algorithms for discrete optimization: coloring, routing, scheduling, and other problems. *arXiv preprint arXiv:2203.14432*, 2022.

[SSJM21] Albert T Schmitz, Nicolas PD Sawaya, Sonika Johri, and AY Matsuura. Graph optimization perspective for low-depth trotter-suzuki decomposition. *arXiv preprint arXiv:2103.08602*, 2021.

[TAM+21] Yu Tong, Victor V Albert, Jarrod R McClean, John Preskill, and Yuan Su. Provably accurate simulation of gauge theories and bosonic systems. *arXiv preprint arXiv:2110.06942*, 2021.

[TG21] Reuben Tate and Swati Gupta. CI-QuBe. *GitHub repository*, 2021.

[TGO+22] 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.

[vBLH+21] Vera von Burg, Guang Hao Low, Thomas Häner, Damian S Steiger, Markus Reiher, Martin Roetteler, and Matthias Troyer. Quantum computing enhanced computational catalysis. *Physical Review Research*, 3(3):033055, 2021.

[VC05] Frank Verstraete and J Ignacio Cirac. Mapping local hamiltonians of fermions to local hamiltonians of spins. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09012, 2005.

[VYI20] Vladyslav Verteletskyi, Tzu-Ching Yen, and Artur F. Izmaylov. Measurement optimization in the variational quantum eigensolver using a minimum clique cover. *The Journal of Chemical Physics*, 152(12), 03 2020. 124114.

[WBAG11] James D Whitfield, Jacob Biamonte, and Alán Aspuru-Guzik. Simulation of electronic structure hamiltonians using quantum computers. *Molecular Physics*, 109(5):735–750, 2011.

[WDC80] Edgar Bright Wilson, John Courtney Decius, and Paul C Cross. *Molecular vibrations: the theory of infrared and Raman vibrational spectra*. Courier Corporation, 1980.

[WE16] Joel J Wallman and Joseph Emerson. Noise tailoring for scalable quantum computation via randomized compiling. *Physical Review A*, 94(5):052325, 2016.

[Wie07] Angelika Wiegele. Biq mac library—a collection of Max-Cut and quadratic 0-1 programming instances of medium size. *Preprint*, 51, 2007.

[WR10] T. Werlang and Gustavo Rigolin. Thermal and magnetic quantum discord in heisenberg models. *Phys. Rev. A*, 81:044101, Apr 2010.

[WRV+23] Dian Wu, Riccardo Rossi, Filippo Vicentini, Nikita Astrakhantsev, Federico Becca, Xiaodong Cao, Juan Carrasquilla, Francesco Ferrari, Antoine Georges, Mohamed Hibat-Allah, et al. Variational benchmarks for quantum many-body problems. *arXiv preprint arXiv:2302.04919*, 2023.

[WSM22] Karen Wintersperger, Hila Safi, and Wolfgang Mauerer. QPU-system co-design for quantum HPC accelerators. In *Architecture of Computing Systems: 35th International Conference, ARCS 2022, Heilbronn, Germany, September 13–15, 2022, Proceedings*, pages 100–114. Springer, 2022.

[WVMN19] Robert Wille, Rod Van Meter, and Yehuda Naveh. Ibm's qiskit tool chain: Working with and developing for real quantum computers. In *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1234–1240. IEEE, 2019.

[WZ94] Lin-Wang Wang and Alex Zunger. Solving Schrödinger's equation around a desired energy: Application to silicon quantum dots. *The Journal of Chemical Physics*, 100(3):2394–2397, 1994.

[XK20] Rongxin Xia and Sabre Kais. Qubit coupled cluster singles and doubles variational quantum eigensolver ansatz for electronic structure calculations. *Quantum Science and Technology*, 6(1):015001, 2020.

[XLP+22] Mingkuan Xu, Zikun Li, Oded Padon, Sina Lin, Jessica Pointing, Auguste Hirth, Henry Ma, Jens Palsberg, Alex Aiken, Umut A Acar, et al. Quartz: superoptimization of quantum circuits. In *Proceedings of the 43rd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, pages 625–640, 2022.

[YADM+19] Kübra Yeter-Aydeniz, Eugene F Dumitrescu, Alex J McCaskey, Ryan S Bennink, Raphael C Pooser, and George Siopsis. Scalar quantum field theories as a benchmark for near-term quantum computers. *Physical Review A*, 99(3):032306, 2019.

[YAGJ+21] 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.

[ZKE+19] AA Zhukov, Evgeniy O Kiktenko, AA Elistratov, WV Pogosov, and Yu E Lozovik. Quantum communication protocols as a benchmark for programmable quantum computers. *Quantum Information Processing*, 18(1):1–23, 2019.

[ZKK+21] Zi-Jian Zhang, Thi Ha Kyaw, Jakob S Kottmann, Matthias Degroote, and Alán Aspuru-Guzik. Mutual information-assisted adaptive variational quantum eigensolver. *Quantum Science and Technology*, 6(3):035001, 2021.

[ZSZ22] Bingzhi Zhang, Akira Sone, and Quntao Zhuang. Quantum computational phase transition in combinatorial problems. *npj Quantum Information*, 8(1):1–11, 2022.

A HamLib code snippets

We provide a collection of Python functions that can be used to interact with the HamLib dataset. The code is depends on domain-specific Python packages, including NetworkX [HSS08], OpenFermion [MRS+20], and mat2qubit [Saw22], and some other widely used libraries. We specifically assume that the following imports have been made before using the functions in Sections A.1 and A.2:

```
import networkx as nx
import mat2qubit as m2q
import openfermion as of
import h5py
import numpy as np
```

A.1 Loading HDF5 file structure

We recommend using the function `print_hdf5_structure` to inspect the path tree structure of the datasets stored in the HamLib HDF5 files and `get_hdf5_keys` to return a list of keys to all datasets stored in a HamLib HDF5 file. This list of keys can be used to extract data from the HDF5 file using the functions provided in Section A.2.

Both `print_hdf5_structure` and `get_hdf5_keys` make use of the decorator function `parse_through_hdf5`.

```
def parse_through_hdf5(func):
    """Decorator function that iterates through an HDF5 file and performs
    the action specified by `func` on the internal and leaf nodes in the HDF5 file."""
    def wrapper(obj, path='/', key=None):
        if type(obj) in [h5py._hl.group.Group, h5py._hl.files.File]:
            for ky in obj.keys():
                func(obj, path, key=ky, leaf=False)
                wrapper(obj=obj[ky], path=path + ky + '/', key=ky)
        elif type(obj) == h5py._hl.dataset.Dataset:
            func(obj, path, key=None, leaf=True)
    return wrapper
```

```
def print_hdf5_structure(fname_hdf5: str):
    """Print the path structure of the HDF5 file.

    Args
    ----
        fname_hdf5 (str): full path where HDF5 file is stored
    """

    @parse_through_hdf5
    def action(obj, path='/', key=None, leaf=False):
        if key is not None:
            print((path.count('/')-1)*'\t', '-', key, ':', path + key + '/')
        if leaf:
            print((path.count('/')-1)*'\t', '[^^DATASET^^]')

    with h5py.File(fname_hdf5, 'r') as f:
        action(f['/'])
```

```
def get_hdf5_keys(fname_hdf5: str):
    """Get a list of keys to all datasets stored in the HDF5 file.

    Args
    ----
        fname_hdf5 (str): full path where HDF5 file is stored
    """
    all_keys = []

    @parse_through_hdf5
    def action(obj, path='/', key=None, leaf=False):
        if leaf is True:
            all_keys.append(path)

    with h5py.File(fname_hdf5, 'r') as f:
        action(f['/'])
```

```
    return all_keys
```

A.2 Loading HDF5 data

The five functions listed below can be used to load data stored at a specific key in the HDF5 file back into an appropriate Python objects, which respectively are a NetworkX graph, a dictionary of grid positions, OpenFermion operators, mat2qubit operators and a list of clauses.

```
def read_graph_hdf5(fname_hdf5: str, key: str):
    """Read networkx graphs from HDF5 file at specified key. Returns a single networkx
     graph.
    """
    with h5py.File(fname_hdf5, 'r') as f:
        G = nx.Graph(list(np.array(f[key])))

    return G
```

```
def read_gridpositions_hdf5(fname_hdf5: str, key: str):
    """Read grid positions, stored as attribute of each networkx graph from HDF5 file
    at specified key. Returns grid positions of nodes associated with a single graph.
    """
    with h5py.File(fname_hdf5, 'r') as f:
        dataset = f[key]
        gridpositions_dict = dict(dataset.attrs.items())

    return gridpositions_dict
```

```
def read_openfermion_hdf5(fname_hdf5: str, key: str, optype=of.QubitOperator):
    """Read any openfermion operator object from HDF5 file at specified key.
    'optype' is the op class, can be of.QubitOperator or of.FermionOperator.
    """
    with h5py.File(fname_hdf5, 'r', libver='latest') as f:
        op = optype(f[key][()].decode("utf-8"))

    return op
```

```
def read_mat2qubit_hdf5(fname_hdf5: str, key: str):
    """Returns mat2qubit's qSymbOp operator from HDF5 file at specified key."""
    with h5py.File(fname_hdf5, 'r') as f:
        op = m2q.qSymbOp(f[key][()].decode("utf-8"))

    return op
```

```
def read_clause_list_hdf5(fname_hdf5: str, key: str):
    """Read clause list from HDF5 file at specified key.Returns clause list in DIMACS
    format."""
    clause_list = []
    with h5py.File(fname_hdf5, 'r') as f:
        for clause in list(np.array(f[key])):
            clause_list.append([v for v in clause])

    return clause_list
```

A.3 Loading and unzipping to memory

For certain workflows, especially ones targeting some of the smaller subdatasets of HamLib, we expect that it might be useful to load the data directly from the webportal into a Python environment. This has an additional benefit of code portability.

Alternatively one may download the zip files directly. We recommend manually downloading the larger (>100 MB) files of HamLib, as the uncompressed data is often 10x the size of the compressed data. Additionally, we note that for some users a manual download may be the only option—during testing, some users were unable to access HamLib via the commands below, due to virtual private network (VPN) or related issues.

The code snippet below provides an example of how this can be achieved. We note that this approach loads the full zip file into memory and thus might be slow for some of the larger datasets.

```
import zipfile
import requests
from io import BytesIO

url = 'https://portal.nersc.gov/cfs/m888/dcamps/hamlib/test/graph/test_graph.zip'

r = requests.get(url, stream=True)
z = zipfile.ZipFile(BytesIO(r.content))

hdf5_filename = z.namelist()[0]
G = read_graph_hdf5(z.open(hdf5_filename, 'r'), "test_graph")
```

B Resonances in vibrational Hamiltonians

For molecular vibrational Hamiltonians, the table shows how many pairs of transitions are within the specified energy threshold. Because near-degenerate transitions (i.e. resonances) are difficult to treat with classical algorithms, the number of such transitions may be used as a rough proxy for classical hardness.

Molecule	25 cm^{-1}	10 cm^{-1}	5 cm^{-1}	2 cm^{-1}	1 cm^{-1}
BH_3	148	74	40	40	39
H_2CO	27	9	7	3	2
BHF_2	20	3	0	0	0
H_2CC	11	2	1	1	1
H_2O_2	31	21	10	1	0
BF_3	106	68	44	44	44
CH_3	88	58	38	34	34
CH_3^+	117	57	45	36	34
ClCOH	20	8	3	1	1
FCCF	460	252	184	102	87
HCCH	150	111	72	56	48
C_2H	25	21	21	17	11
C_2O	27	19	13	13	11
CH_2	0	0	0	0	0
COH	1	0	0	0	0
H_2O	3	3	2	0	0
H_2S	4	1	0	0	0
H_3^+	7	7	7	7	7
HCO	1	0	0	0	0
HNC	21	17	11	11	11
HNO	1	1	0	0	0
NH_2	2	0	0	0	0
O_3	0	0	0	0	0
SF_2	9	0	0	0	0
SO_2	0	0	0	0	0
NOH	0	0	0	0	0
BeH	0	0	0	0	0
BH	0	0	0	0	0
CH	0	0	0	0	0
CO	0	0	0	0	0
F_2	0	0	0	0	0
HF	0	0	0	0	0
Allene	3243	1707	1224	951	791
Cyclopropene	2827	1030	512	181	90
Ethylene oxide	4090	1902	883	349	174
Propargyl Cyanide	6611	2760	1482	564	294