Generating Low-Depth Quantum Variational Circuits for the Hydroxyl Cation to Attain High Fidelity on NISQ Computers

Quantum Computing for Drug Discovery Challenge at ICCAD 2023

JASON HAN, Rice University, Houston, USA
JASON LUDMIR, Rice University, Houston, USA
HAMZA SHILI, Rice University, Houston, USA
TIRTHAK PATEL, Rice University, Houston, USA

Abstract — Quantum computing and machine learning have facilitated the rapid advancement of drug discovery by enabling us to simulate and understand complex molecular interactions. Our work focuses on calculating the ground state energy of the hydroxyl cation (·OH), a pivotal molecule in drug interactions, through open-source protocols that incorporate quantum computing, machine learning, and compiler optimization techniques to accelerate therapeutic developments using current near-term intermediate-scale quantum (NISQ) computers. Our circuit-synthesis-based approach helps us achieve an average ground state energy estimation accuracy of 97.74% and an average circuit duration of 7,065.6 across ten required seeds and three required noise models while ensuring that the total number of shots remains below 1,800,000.

1 INTRODUCTION

Quantum Computing for Drug Discovery. In the NISQ (Noisy Intermediate-Scale Quantum) era, the convergence of quantum computing and machine learning presents an unprecedented opportunity for revolutionizing drug discovery [2]. Quantum computers, despite their current limitations, offer the potential to efficiently simulate complex molecular structures and interactions, dramatically accelerating the exploration of drug candidates [1]. These quantum simulations can guide researchers in identifying promising compounds, predicting their properties, and optimizing their designs with unparalleled precision [10]. This can help reduce the time and cost associated with drug development but also holds the potential to unlock novel therapeutic solutions for a wide range of diseases, marking a transformative leap in the field of pharmaceutical research [8].

The Problem. The problem is to implement an open-source protocol that can automatically calculate the ground state energy of the hydroxyl cation (OH) for a specified hamiltonian. Given its significant role in various drug interactions and physiological processes, a comprehensive understanding of the hydroxyl cation is of significant importance [15].

Our Solution. To address this problem, our goal is to design a shallow-depth, few-gates circuit that would get minimally impacted by noise, but also achieve high accuracy with as few shots as possible on NISQ computers. To this end, we take a three-step approach: (1) generate the ansatz using the Unitary Selective Coupled-Cluster Method (UCCSD) [6] for high energy estimation accuracy using few shots; (2) train the ansatz using the Variational Quantum Eigensolver (VQE) algorithm; and (3) minimize the depth of the final trained ansatz using BQSKit [4] approximate synthesis, so that the circuit performs well on real noisy quantum computers.

The main contributions of this work are as follows.

- This work proposes the use of approximate synthesis to reduce the depth of an exactly accurate UCCSD circuit from a depth of 10,388 gates to a depth of only one one-qubit gate in the best case, considerably reducing the duration and improving the accuracy of the circuit.
- Our approach achieves an average ground state energy estimation accuracy of 97.74% and an average circuit duration of 7,065.6 across ten required seeds and three required noise models

while ensuring that the total number of shots remains below 1,800,000. This is because we are able to maintain low one-qubit (≈20 per circuit on average) and two-qubit (≈5 per circuit on average) physical gate counts.

- We have open-sourced our code for analysis and reproducibility: *https://github.com/positivetechnologylab/ptl-iccad-contest-2023*.

Code Organization. The code is organized as follows:

- `FinalCircuits/` - Directory containing three QASM files for the three noise models, prefixed with 'FakeCairo', 'FakeKolkata', and 'FakeMontreal', for the Cairo, Kolkata, and Montreal noise models, respectively. The accuracies and seeds are also provided in the name of the files, for reproducibility.
- `Hamiltonian/` - Directory containing a file `Hamiltonian.txt` that is used as the Hamiltonian of the input hydroxyl radical. This directory is identical to the provided Hamiltonian directory.
- `NoiseModel/` - Directory containing the noise models used by the code for evaluating the accuracy of the quantum circuits. This directory is identical to the provided Noise Model directory.
- `README.md` - Repository readme with setup and execution instructions.
- `iccad_prod_workflow_doc.ipynb` - Tutorial notebook for an end-to-end workflow for our circuit synthesis and optimization workflow.
- `iccad_prod_workflow_doc.py` - Python file containing functions used in our workflow.
- `main.py` - Python file to be run from the command line to execute our workflow programmatically.
- `requirements.txt` - Requirements to be installed before running the Python scripts. These include the Qiskit, Qiskit Nature, Qiskit Aer, and BQSKit Python libraries.
- `PTL_ICCAD_Design.pdf` - The design document describing our solution to the contest problem in detail (this document).

Execution Instructions. To run the code, follow these steps:

(1) Clone the repository to your local machine:

```
git clone https://github.com/positivetechnologylab/
ptl-iccad-contest-2023.git
```

(2) Navigate to the repository directory:

```
cd ptl-iccad-contest-2023
```

(3) Install the required dependencies:

```
pip install -r requirements.txt
```

(4) Run the main script:

```
python main.py <noisemodel_name> <seed> [shots]
```

where <noisemodel_name> is one of 'fakecairo', 'fakekolkata', or 'fakemontreal' for the three backends respectively, <seed> is an integer seed used for the workflow, and [shots] is an optional argument (default is 2852) used for VQE and circuit evaluation in the workflow.

(5) To interpret the results, there is a line in the output with the following format: *Accuracy Score: __%*. This is the accuracy score for that particular run.

The GitHub repository contains additional instructions for running the provided Python notebook tutorial and developer instructions.

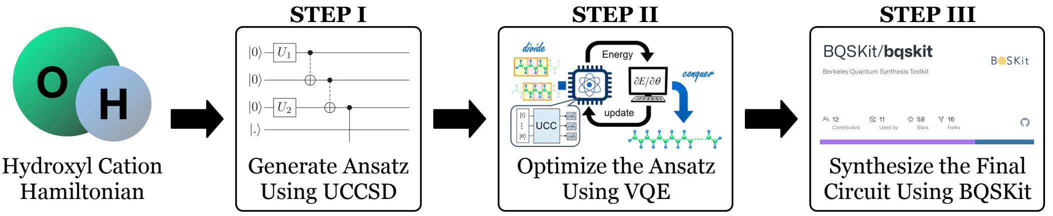

Fig. 1. We take a three-step approach toward optimizing the ansatz for the given problem: (1) generate the ansatz using UCCSD for high energy estimation accuracy using few shots, (2) train the ansatz using the VQE algorithm, and (3) minimize the depth of the final trained ansatz using BQSKit approximate synthesis, so that the circuit performs well on real noisy quantum computers.

Report Organization. The rest of this report is organized as follows. Sec. 2 describes our approach to solving the problem. Then, Sec. 3 describes our implementation, experimental methodology, and metrics. Sec. 4 describes our results and analyses their impact. Lastly, we conclude in Sec. 5.

2 OUR APPROACH

Overview. As shown in Fig. 1, our approach can be broken into three main steps: (1) generating the ansatz using UCCSD for high energy estimation accuracy using few shots, (2) training the ansatz using the VQE algorithm, and (3) minimizing the depth of the final trained ansatz using BQSKit approximate synthesis, so that the circuit performs well on real noisy quantum computers.

Why UCCSD? We select the UCCSD (Unitary Selective Coupled-Cluster Method) ansatz because it has high accuracy stemming from coupled-cluster theory, one of the most accurate classical methods for many-body simulations [6]. Therefore, in an ideal, noiseless simulation, the UCCSD ansatz would give us a highly accurate circuit for computing the ground state energy of the hydroxyl cation. However, the UCCSD ansatz tends to be deep due to its composition of many-body operations which scale unfavorably with the size of the quantum system [17]. Thus, our approach reduces the depth while maintaining the accuracy of the UCCSD ansatz using fine-grained approximation synthesis. We train the parameters of this circuit using the VQE algorithm [5].

Leveraging Fine-Grained Approximate Synthesis. We use BQSKit (the Berkeley Quantum Synthesis Toolkit) to perform global circuit optimization on the trained UCCSD circuit [4]. BQSKit employs a suite of state-of-the-art algorithms to optimize quantum circuits, including QSearch for finding an optimal circuit for a given unitary operation, QFAST (Quantum Fast Approximate Synthesis Tool) for decomposing unitary operations into smaller unitaries, and QGo for partitioning the circuit and optimizing each partition [3] [16] [14]. These circuit synthesis tools have proven to generate circuits of orders of magnitude fewer gates [11].

As shown in Fig. 2, the deep UCCSD circuit is partitioned into manageable sub-circuits, then synthesized into smaller circuits, and finally combined to construct the final circuit. Synthesis is the procedure to generate an equivalent shallow-depth circuit for a deep circuit using layer-by-layer construction by minimizing the distance between the unitary matrices of the two circuits (original deep circuit and constructed shallow circuit). However, due to exponential scaling, synthesis cannot be done when the circuit size is large in terms of the number of qubits. Therefore, the circuit has to be paritioned into manageable sub-circuits and then synthesis can be performed on each sub-circuit. *Note that no additional shots are required by BQSKit during the synthesis procedure because no quantum computer runs are required.*

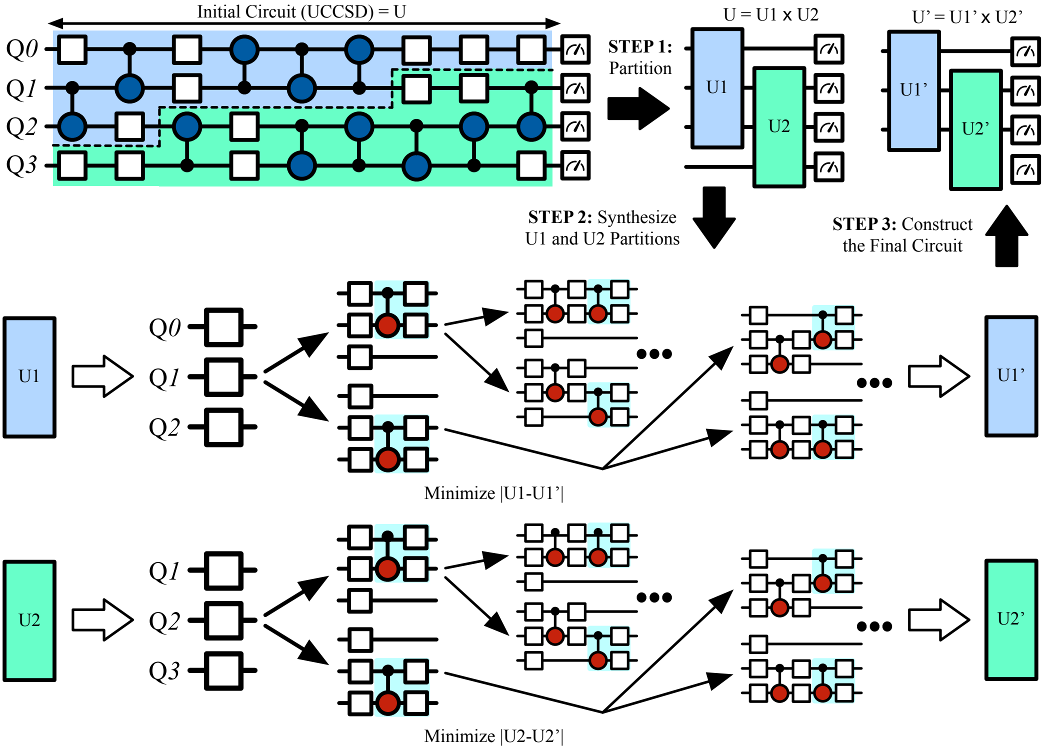

Fig. 2. We take a scalable approach to perform a fine-grained approximation of the initial, deep UCCSD circuit using BQSKit [4]. First, the deep circuit (U) is partitioned into manageable-sized circuits [14]. Then each partition is synthesized into shallow circuits using layer-by-layer construction until the distance between the original and the constructed partitions' unitaries becomes sufficiently small [3] [13] [16]. Finally, the new shallow circuit (U') is constructed by combining the synthesized partitions [14].

As a note, we have previously contributed to building the BQSKit library by implementing the techniques used in this work – algorithm for quick partitioning, proof for error upper bounds on the impact of approximate synthesis, and technique for the efficient synthesizing of circuits using coarse-grained approximations (see Quest *[12]).*

3 METHODOLOGY

Quantum Experimental Setup. We used the IBM Qiskit version 0.44.2, Qiskit Aer version 0.12.2, Qiskit Nature version 0.6.2, and BQSKit version 1.1.0 Python libraries for our implementation. To construct the UCCSD ansatz for the hydroxyl cation, we first use the PySCFDriver class in Qiskit Nature to obtain a model of the hydroxyl cation. We then construct the UCCSD ansatz using the UCCSD class in Qiskit, passing in the number of spatial orbitals and particles for the hydoxyl cation, as well as using the Jordan-Wigner Mapper [9] and Hartree-Fock Method [7] for the initial state provided by Qiskit, which are well-known methods for quantum simulations.

We then perform VQE on the result UCCSD ansatz using the Estimator class with the statevector simulator from Qiskit Aer, using 2852 shots per Pauli string. We initialize the parameters of the UCCSD ansatz to 0 for simplicity. We then transpile the parameterized UCCSD ansatz (with the

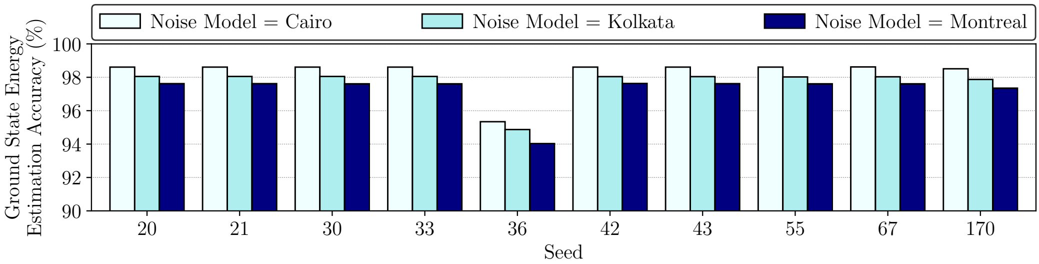

Fig. 3. We plot the ground state energy estimation accuracy results to demonstrate the trend across the three noise models: Cairo consistently produces the highest accuracies and Montreal produces the lowest accuracies, with Kolkata always falling in between. Note the y-axis starts at 90%.

optimization level set to 0) on the QASM simulator from Qiskit to obtain a circuit with one- and two-qubit gates. Next, we used the BQSKit compiler with the default parameters and the optimization level set to 3. We use an optimization level of 3 because it uses a larger number of iterations to optimize the circuit parameters within a given layer.

The result of the BQSKit compilation yields our machine-independent compiled circuit. To evaluate the correctness of our circuit on the FakeMontreal backend provided by Qiskit, we transpile our BQSKit-compiled circuit onto the FakeMontreal backend, mapping the first 12 logical qubits to the first 12 physical qubits on the FakeMontreal backend. We then read in the provided Hamiltonian and parse the file into a SparsePauliOp observable in Qiskit, padding the observables with 15 identity operators for the 15 other qubits on the FakeMontreal backend, as we only use the first 12 qubits for measuring the value of the Hamiltonian. We then use the Estimator class from Qiskit Aer with the provided noise models to obtain the expectation of the Hamiltonian (i.e., the ground state energy). We then compare this value with the reference value to obtain our accuracy score in Table 1. Note that the circuit duration and gate counts are with respect to the final circuit compiled on the FakeMontreal backend. All computations were run on a 16-core Google Cloud VM with 128GB of memory.

Classical Experimental Setup. The classical experimental setup is identical to the classical setup in the provided example code, as we believe that the provided classical simulation accurately calculates the ground state energy for the hydroxyl cation. Namely, we use the Qiskit, Qiskit Nature, and Qiskit Aer Python libraries. We use the PySCFDriver from Qiskit Nature to obtain a model of the hydroxyl radical, and we then use the GroundStateEigensolver from Qiskit Nature, using the well-known Jordan Wigner Mapper [9] and the NumpyMinimumEigensolver from Qiskit to compute the minimum energy in the system. We add the computed energy and the nuclear repulsion energy to obtain the ground state energy, a practice consistent with Hartree-Fock theory.

Metrics Analyzed. The first metric we analyze is the **ground state energy estimation accuracy**. This metric measures the accuracy of the estimated energy from the ideal energy when running the circuit on a noisy quantum computer. It is defined as: $\left(1 - \left|\frac{E_{estimated} - E_{ideal}}{E_{ideal}}\right|\right) \times 100\%$. We also report the **circuit duration** and **number of shots** required, which combined measure the circuit's run time, i.e., performance. We also report the **number of SX + X gates** and the **number of CX gates** in the generated circuit, which both affect the above metrics. Finally, we show the **compilation time**, which measures the total time it took to train and optimize the circuit.

Table 1. Flagship results for all combinations of the ten seeds and three noise models provided including ground state energy estimation accuracy, total number of shots across 631 Pauli strings, and circuit duration.

Circuit ID	**Seed**	**Noise Model**	**Energy Estimation Accuracy (%)**	**Circuit Duration (dt)**	**Number of Shots (#)**
1	20	Cairo	98.61	320	1,799,612
2	20	Kolkata	98.05		
3	20	Montreal	97.62		
4	21	Cairo	98.61	320	
5	21	Kolkata	98.05		
6	21	Montreal	97.62		
7	30	Cairo	98.61	320	
8	30	Kolkata	98.05		
9	30	Montreal	97.61		
10	33	Cairo	98.61	320	
11	33	Kolkata	98.05		
12	33	Montreal	97.61		
13	36	Cairo	95.34	67,776	
14	36	Kolkata	94.87		
15	36	Montreal	94.03		
16	42	Cairo	98.61	320	
17	42	Kolkata	98.04		
18	42	Montreal	97.63		
19	43	Cairo	98.61	320	
20	43	Kolkata	98.04		
21	43	Montreal	97.62		
22	55	Cairo	98.61	320	
23	55	Kolkata	98.02		
24	55	Montreal	97.61		
25	67	Cairo	98.62	320	
26	67	Kolkata	98.03		
27	67	Montreal	97.61		
28	170	Cairo	98.51	320	
29	170	Kolkata	97.87		
30	170	Montreal	97.35		
Average			97.74	7,065.6	1,799,612

4 RESULTS AND ANALYSIS

Our approach helps us achieve an average ground state energy estimation accuracy of 97.74% and an average circuit duration of 7,065.6 dt across ten required seeds and three required noise models while ensuring that the total number of shots remains below 1,800,000. The energy estimation accuracy across seeds and noise models is illustrated in Fig. 3. We observe that all accuracies are above 94% (average accuracy is 97.74%) and that the three noise models follow the same trend across the 10 provided seeds (with Cairo having the highest accuracy, followed by Kolkata, then by Montreal), reflecting the consistency of our results. We note that the choice of noise model does not affect the accuracy more than two percentage points, indicating the robustness of our method. The accuracy slightly varies with choice of seed: we see that seed 20 has

Table 2. Additional results for all combinations of the ten seeds and three noise models provided including the number of one-qubit SX + X (We do not show RZ gates as they are virtual and have no noise impact) gates, the number of two-qubit CX gates, and the end-to-end compilation time.

Circuit ID	Seed	Noise Model	Number of SX + X Gates (#)	Number of CX Gates (#)	Compilation Time (s)
1	20	Cairo	16	0	6066.69
2	20	Kolkata			
3	20	Montreal			
4	21	Cairo	16	0	6694.66
5	21	Kolkata			
6	21	Montreal			
7	30	Cairo	16	0	6491.12
8	30	Kolkata			
9	30	Montreal			
10	33	Cairo	16	0	6693.49
11	33	Kolkata			
12	33	Montreal			
13	36	Cairo	52	46	16348.94
14	36	Kolkata			
15	36	Montreal			
16	42	Cairo	16	0	7055.191
17	42	Kolkata			
18	42	Montreal			
19	43	Cairo	16	0	8733.90
20	43	Kolkata			
21	43	Montreal			
22	55	Cairo	16	0	8559.09
23	55	Kolkata			
24	55	Montreal			
25	67	Cairo	16	0	5771.07
26	67	Kolkata			
27	67	Montreal			
28	170	Cairo	18	0	9780.00
29	170	Kolkata			
30	170	Montreal			
Average			19.8	4.6	7,363.51

an average accuracy of around 98%, but seed 36 has an average accuracy of around 94%. This is because the choice of seed both affects the optimal parameters yielded from VQE and the decisions that the BQSKit compiler makes. However, we note that, for nine out of the ten seeds, we produce a circuit with duration 320 and with average accuracy around 98% for the provided noise models, illustrating the consistency of our method. From Fig. 3, we also see that all seeds except for seed 36 and 55 have the same distribution of accuracies: this is because, for these seeds, the compiled circuit had the same structure and duration, as seen in Table 1. For the duration of the circuits, although the average is 7,065.6 dt, nine out of the ten circuits have a depth of 320 dt. All results were run with 2852 shots with 631 Pauli strings, yielding a total of 1,799,612 shots.

Our approach achieves an average number of 19.8 SX + X gates and 4.6 CX gates for transpiled circuits on the Montreal backend, with an average end-to-end compilation time of 2.05 hours.[1] Building off of our results above, we analyze the characteristics of the compiled circuits on the Montreal backend. We note that the number of gates is varies based on the seed: as seed in Table 2, seed 36 has 52 SX + X gates and 46 CX gates, seed 170 has 18 SX + X gates and 0 CX gates, and all other seeds have the 16 SX + X gates and 0 CX gates. For the seeds that yield circuits with the same number of gates, the circuits have identical topology, the only difference being parameters in the RZ gates. A majority of the circuits across the seeds have 16 SX gates and 0 CX gates, yielding a low-noise circuit due to its shallow depth and lack of two qubit gates. We note that the noise models have no affect on the compiled circuit because, when compiling the circuit, we have no knowledge of the underlying noise model.

5 CONCLUDING REMARKS

In this work, we proposed a circuit-synthesis-based approach to calculate the ground state energy of the hydroxyl cation (·OH), a pivotal molecule in drug interactions. Our approach achieves an average ground state energy estimation accuracy of 97.74% and an average circuit duration of 7,065.6 across ten required seeds and three required noise models while ensuring that the total number of shots remains below 1,800,000. *Our open-source algorithm demonstrates how carefully combining quantum computing, machine learning, and compiler optimization techniques can accelerate drug developments using near-term intermediate-scale quantum (NISQ) computers.*

ACKNOWLEDGEMENTS

We are thankful to the organizers of the *Quantum Computing for Drug Discovery Challenge at ICCAD 2023* for their helpful and timely feedback, especially Zhiding Liang. This work was also supported by Rice University, Center for Research Computing (CRC) at Rice University, and Google Colab Enterprise. We used ChatGPT, an AI language model developed by OpenAI, for partial assistance in writing, and all such texts were verified and edited for correctness. IBM Quantum was also used for this work. The views expressed are those of the authors and do not reflect the official policy or position of IBM or the IBM Quantum team.

REFERENCES

[1] 2021. Quantum computing's potential for drug discovery: Early stage industry dynamics. *Drug Discovery Today* 26, 7 (2021), 1680–1688. https://www.sciencedirect.com/science/article/pii/S1359644621002750

[2] Y. Cao, J. Romero, and A. Aspuru-Guzik. 2018. Potential of quantum computing for drug discovery. *IBM Journal of Research and Development* 62, 6 (2018), 6:1–6:20. https://doi.org/10.1147/JRD.2018.2888987

[3] Marc G. Davis, Ethan Smith, Ana Tudor, Koushik Sen, Irfan Siddiqi, and Costin Iancu. 2020. Towards Optimal Topology Aware Quantum Circuit Synthesis. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*. 223–234. https://doi.org/10.1109/QCE49297.2020.00036

[4] BQSKit Developers. 2023. Berkeley Quantum Synthesis Toolkit (BQSKIT). https://bqskit.lbl.gov/.

[5] Dmitry A. Fedorov, Bo Peng, Niranjan Govind, and Yuri Alexeev. 2022. VQE method: a short survey and recent developments. *Materials Theory* 6, 1 (2022), 2. https://doi.org/10.1186/s41313-021-00032-6

[6] Harper R. Grimsley, Daniel Claudino, Sophia E. Economou, Edwin Barnes, and Nicholas J. Mayhall. 2019. Is the Trotterized UCCSD Ansatz Chemically Well-Defined? *Journal of Chemical Theory and Computation* 16 (2019), 1–6. Issue 1. https://doi.org/10.1021/acs.jctc.9b01083

[7] D. R. Hartree. 1928. The Wave Mechanics of an Atom with a Non-Coulomb Central Field. Part I. Theory and Methods. *Mathematical Proceedings of the Cambridge Philosophical Society* 24 (1928), 89–110. https://doi.org/10.1017/S0305004100011919

[1] All computations were run on a 16-core Google Cloud VM with 128GB of memory, but the machine was under different loads, which contributed significantly to the variation in compilation time.

[8] Parichehr Hassanzadeh. 2020. Towards the quantum-enabled technologies for development of drugs or delivery systems. *Journal of Controlled Release* 324 (2020), 260–279. https://doi.org/10.1016/j.jconrel.2020.04.050

[9] P. Jordan and E. Wigner. 1928. Über das Paulische Äquivalenzverbot. *Z. Physik* 47 (1928), 631–651. https://doi.org/10.1007/BF01331938

[10] Weitang Li, Zigeng Huang, Changsu Cao, Yifei Huang, Zhigang Shuai, Xiaoming Sun, Jinzhao Sun, Xiao Yuan, and Dingshun Lv. 2022. Toward practical quantum embedding simulation of realistic chemical systems on near-term quantum computers. *Chemical Science* (2022). https://pubs.rsc.org/en/content/articlehtml/2022/sc/d2sc01492k Accessed: 2023-10-08.

[11] Lindsay Bassman Oftelie, Roel Van Beeumen, Ed Younis, Ethan Smith, Costin Iancu, and Wibe A. de Jong. 2022. Constant-depth circuits for dynamic simulations of materials on quantum computers. *Materials Theory* 6, 1 (mar 2022). https://doi.org/10.1186/s41313-022-00043-x

[12] Tirthak Patel, Ed Younis, Costin Iancu, Wibe de Jong, and Devesh Tiwari. 2022. QUEST: Systematically Approximating Quantum Circuits for Higher Output Fidelity. In *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*. 514–528.

[13] Ethan Smith, Marc Grau Davis, Jeffrey Larson, Ed Younis, Lindsay Bassman Oftelie, Wim Lavrijsen, and Costin Iancu. 2023. LEAP: Scaling Numerical Optimization Based Synthesis Using an Incremental Approach. *ACM Trans. Quantum Comput.* 4, 1 (February 2023), 1–23. https://doi.org/10.1145/3548693

[14] Ethan Smith and Mitiq Team. 2020. QGo: A Quantum Global Optimizer for Quantum Circuit Compilation. *arXiv preprint arXiv:2012.09835* (2020). https://arxiv.org/abs/2012.09835

[15] V Wineman-Fisher, Y Al-Hamdani, I Addou, A Tkatchenko, and S Varma. 2019. Ion-Hydroxyl Interactions: From High-Level Quantum Benchmarks to Transferable Polarizable Force Fields. *J Chem Theory Comput* 15, 4 (2019), 2444–2453. https://doi.org/10.1021/acs.jctc.8b01198 Epub 2019 Mar 13.

[16] Eyad Younis, Koushik Sen, Kathy Yelick, and Costin Iancu. 2021. QFAST: Conflating Search and Numerical Optimization for Scalable Quantum Circuit Synthesis. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 232–243. https://doi.org/10.1109/QCE52317.2021.00041

[17] Yu Zhang, Lukasz Cincio, Christian F. A. Negre, Piotr Czarnik, Patrick J. Coles, Petr M. Anisimov, Susan M. Mniszewski, Sergei Tretiak, and Pavel A. Dub. 2022. Variational Quantum Eigensolver with Reduced Circuit Complexity. *npj Quantum Information* 8 (2022), 96. https://doi.org/10.1038/s41534-022-00599-z