# QuantMark: A Benchmarking API for VQE Algorithms

Meijer - van de Griend, Arianne

2022-03-15

Meijer - van de Griend , A & Nurminen , J K 2022 , ' QuantMark: A Benchmarking API for VQE Algorithms ' , IEEE Transactions on Quantum Engineering , vol. 3 . https://doi.org/10.1109/TQE.2022.3159327

http://hdl.handle.net/10138/352910
https://doi.org/10.1109/TQE.2022.3159327

IEEE Transactions on Quantum Engineering

Received October 2, 2021; revised March 1, 2022; accepted March 11, 2022; date of publication March 15, 2022; date of current version April 21, 2022.

*Digital Object Identifier 10.1109/TQE.2022.3159327*

# QuantMark: A Benchmarking API for VQE Algorithms

**ADRIANA MEIJER - VAN DE GRIEND (Member, IEEE), AND JUKKA K. NURMINEN (Member, IEEE)**

Department of Computer Science, University of Helsinki, 00100 Helsinki, Finland

Corresponding author: Adriana Meijer - van de Griend (e-mail:arianne.vandegriend@helsinki.fi).

**ABSTRACT** Thanks to the rise of quantum computers, many variations of the variational quantum eigensolver (VQE) have been proposed in recent times. This is a promising development for real quantum algorithms, as the VQE is a promising algorithm that runs on current quantum hardware. However, the popular method of comparing your algorithm versus a classical baseline in a small basis set is not meaningful in the big picture. Moreover, many papers use a different molecular representation or a different quantum computer to test their algorithms such that the used baselines are different between different papers. Thus, it is almost impossible to compare the different algorithms to each other. As a solution, we have built a benchmarking framework to standardize the VQE performance metrics, such that they can be analyzed more easily. Using our framework, any researcher working on the VQE can easily test their own algorithms against previous ones on the leaderboard without the need to reproduce previous work themselves.

**INDEX TERMS** Algorithms, benchmarking and performance characterization, noisy intermediate-scale quantum algorithms and devices, quantum computing, variational quantum eigensolver (VQE).

## I. INTRODUCTION

Since the dawn of physically realized quantum computers, quantum chemistry calculations have been considered "low-hanging fruit" for quantum computers. However, it has recently become more clear that achieving quantum advantage in quantum chemistry is still a distant goal [1], [2]. This means that current research into quantum chemistry algorithms for quantum computers needs to be replicable and comparable such that new research can be efficiently built on previous work.

In this article, we focus on the variational quantum eigensolver (VQE) algorithms [3], a type of hybrid quantum–classical algorithm where a parameterized quantum circuit is measured on a quantum computer such that the measured values should correspond to the energy of a molecule. Then, the parameters are optimized classically such that the measured energy is minimized, according to the variational principle. A recent overview of VQE algorithms and a detailed explanation of their differences were presented by Fedorov *et al.* [4].

The accuracy of the minimized energy with respect to the true ground state energy of the molecule depends on how the molecule is represented by the qubits and the quantum circuit. Particularly, the quality of the basis set and the choice of qubit encoding is important, as well as the choice of the structure of the quantum circuit, be it chemically inspired (e.g., UCCSD ansatz [3]) or not (e.g., hardware-efficient ansatz [5]). Moreover, the noise of the quantum computer and the ability of the classical optimizer to explore the search space and find the global minimum are crucial factors governing the performance of the VQE algorithm.

Clearly, VQE algorithms have many components that are interchangeable. As such, results from different VQE algorithms have different choices of components in papers. For example, ADAPT-VQE [6] was tested using the Jordan–Wigner (JW) encoding and optimized with Broyden-Fletcher-Goldfarb-Shannon (BFGS), but the original hardware-efficient ansatz was tested using Bravyi–Kitaev (BK) binary tree encoding [7] and optimized with simultaneous perturbation stochastic approximation (SPSA) [8]. This makes it very difficult to say at first glance whether any performance difference is due to the change in the algorithm or the change in the experiment. Thus, it is very difficult to compare results across different papers.

However, comparing the differences between different VQE algorithms is particularly important for identifying and verifying the effect of those differences on the barren plateau problem [9]–[13].

Unfortunately, replicating results from an existing paper is even more difficult and extremely time-consuming because the experimental differences between papers are not always

clear from the text, and the source code is not always available.

Additionally, the existing research has mainly compared the VQE algorithms against classical baselines using the same basis set. Although this shows how close the quantum algorithm can get to its theoretical optimum (precision), it does not show how well the algorithm approximates reality (accuracy) because the classical optimizer is asked to find an approximation of the real problem.

As an analogy, consider an algorithm that attempts to estimate $\pi$, but the algorithm can only get at best two decimals right by design. If it were to find $\pi = 3.146$, it would have high precision (it found the two decimals it could find), but a poor accuracy (it did not find $\pi$). Similarly, when comparing a VQE algorithm to the full configuration interaction (FCI) of the molecule in a small basis set, you are ignoring the inaccuracies that are caused by the choice of the basis set. Thus, you can incorrectly conclude that the algorithm works well, as pointed out by Elfving *et al.* [1].

Hence, the field of VQE research needs a tool that: 1) can show not only the precision but also the accuracy of an algorithm; 2) makes it easy to compare different algorithms to each other; and 3) makes it easy to replicate results, such that we can create new insights into which the algorithm is performing better, and why that is the case, and make predictions about how the algorithms would perform in the future.

We created **QuantMark**[1] as a beginning to increase the value of VQE research. QuantMark has a web interface with multiple ranked tables (leaderboards) for different VQE performance metrics to make it possible to compare different VQE algorithms more easily. QuantMark has a Python application programming interface (API) that is integrated with the Tequila library [14] for a uniform VQE representation. A detailed description of QuantMark can be found in Section II. QuantMark attempts to address the above challenges. However, they do not have clear-cut solutions, and we discuss the pros and cons of our choices in Section III. QuantMark is still under active development, and the field of VQE is still young; therefore, in Section IV, we briefly give an overview of what questions still need to be answered on the road to quantum advantage.

## II. DESCRIPTION

The aim of QuantMark is to make it easier to test and compare VQE algorithms. Particularly, we want our users to know the accuracy of their algorithm, compare it to other algorithms, and replicate results from others. In this section, we describe how QuantMark achieves these three things.

As of the time of writing, QuantMark assumes that the VQE simulates the stretching of the bond distances between the two hydrogen atoms in an $H_2$ molecule. We specify a "ground truth" classical baseline to calculate the accuracy of the VQEs. The method we used (see Section II-A) is based on basis sets and is, therefore, also an approximation, but finding the exact energy curve for arbitrary molecules with classical means is infeasible. More research is needed on the topic, as discussed in Sections III and IV. Nevertheless, we use this baseline to construct three leaderboards (see Section II-C). Because new performance criteria might still be added, QuantMark collects enough data about the VQE to calculate new metrics. Since these are enough data to replicate the results found on the leaderboard, QuantMark exposes these data such that users can replicate results themselves and even adjust parts of them for continued research (see Section II-D).

To make all of these features possible, QuantMark also contains basic data management features, such as deleting your data and toggling public availability. Users can provide a link to their source code and a link to a paper if the users were to publish the results. This not only improves awareness of existing research but also allows users to compare results from different papers, even if they were not compared in the papers themselves.

### A. BUILT-IN CLASSICAL BASELINE

A key feature of QuantMark is that it has a consistent classical baseline for VQE algorithms to compare against. In recent research, different papers have been using different basis sets (e.g., STO-3G in [6] and 6-31G in [15]) and different qubit encodings (e.g., JW in [6] and BK [5]). These differences not only influence the number of qubits and the size of the circuit needed to simulate the molecule on a quantum computer but can also bound the best energy that can be found by the VQE algorithm. Most VQE algorithms are compared against this bound of the precision of the molecular representation (i.e., the FCI calculated with the same basis set), not against a realistic ground truth. This gives a skewed view of the performance of the VQE, as discussed in [1]. Additionally, it makes a comparison of VQE algorithms in different basis sets very difficult, since they are compared against different baselines.

In an attempt to eliminate the problem of comparing against different baselines, QuantMark employs precalculated classical FCI baselines to compare against. In contrast to baselines in literature, QuantMark uses not only the "STO-3G" and "6-31G" basis sets but also a large basis set ("def2_QZVPPD") for the classical baselines. Using such a large basis set allows us to compare the VQE algorithm to a baseline that is beyond the best energy that the VQE can find. This allows a fair comparison between VQE algorithms that use different smaller basis sets. Moreover, the difference between this classical baseline and the VQE performance (accuracy) indicates the amount of work that still needs to be done in improving the representation of the molecule or the nature of the VQE algorithm, whereas the VQE performance versus the FCI in the same basis set only (precision) indicates how much the VQE can improve with better classical optimizers or less noise on the quantum computer.

[1] QuantMark is available on Github under the MIT open-source license: https://github.com/QuantMarkFramework/LibMark/

IEEE Transactions on Quantum Engineering

In QuantMark, users can get access to the precalculated energies that are stored in the database as well as their corresponding bond distances with a simple API call:

```
import libmark as lm
import numpy as np
import tequila as tq

sto_3g_fci = lm.api.get_fci("sto-3g")
distances = lm.api.get_distances()
```

Using the stored energies as opposed to calculating them yourself saves time and energy.

A downside to this approach for classical baselines is that even the large basis set that QuantMark uses right now is an approximation of the real energy and, therefore, not a perfect "ground truth" baseline. However, with the state of the VQE research at the time of writing, the use of a better baseline than the one used right now would be excessive.

In addition to precision and accuracy, resource metrics, such as the number of qubits, gates, and parameters, can also be found in QuantMark.

### B. UPLOADING YOUR VQE RESULTS

When a user has created a VQE in Tequila, they can fetch the interatomic distances at which QuantMark has precalculated the classical baselines (as discussed in Section II-A). Users can create the hydrogen molecule object and then run their algorithm as desired.

When running the VQE, users create a tracker to which the results, molecule, Hamiltonian, and quantum circuit are given. When done, users can push the results to QuantMark in bulk using their personal OAuth [16] tokens[2] to identify themselves. See the script below for a simple example script that would run on the user's local device.

After pushing the results to QuantMark, the data are processed and made privately available on the QuantMark website. Users can view their results in comparison with publicly available results on the data management page. The different comparison tables and figures, as discussed in Section II-C, will be automatically generated by QuantMark. As such, QuantMark calculates the precision and accuracy of the uploaded VQE results.

The data management page also contains basic data management functionality such as the ability to make results publicly available on the leaderboards as well as adding a reference to the paper describing the method and a link to the source code that created the results.

### C. LEADERBOARDS

An important feature of QuantMark is the ability to see the current state-of-the-art accuracy of different VQE algorithms

```
token = 'YOUR PERSONAL TOKEN HERE'
optimizer = "Nelder-Mead"
qresult = lm.tracker.get_tracker(optimizer,
                                 token)

for R in lm.api.get_distances():
    geometry = f'H 0.0 0.0 0.0\nH 0.0 0.0 {R}'
    molecule = tq.chemistry.Molecule(geometry,
                    "sto-3g", "Jordan-Wigner")
    H = molecule.make_hamiltonian()
    U = molecule.make_uccsd_ansatz(
                        trotter_steps=1)
    E = tq.ExpectationValue(H=H, U=U)
    vars = {k:0.0 for k in
                    U.extract_variables()}
    result = tq.minimize(objective=E,
        method=optimizer, initial_values=vars)
    qresult.add_run(result, molecule, H, U)

res = qresult.push()
```

at first glance. Since the benchmark problem in QuantMark is energy simulation, the goal of the VQE is to find the energy curve for the hydrogen molecule at different interatomic distances. This energy curve has interesting properties depending on its application: 1) the minimum energy found; 2) the equilibrium bond distance; 3) the shape of the curve itself. Therefore, the QuantMark leaderboard is split into three different leaderboard tables.

In the first leaderboard table, VQEs are simply ranked by the minimum energy they found. By the variational principle, a VQE should never find smaller energy than the "ground truth" minimum; therefore, QuantMark can order the VQEs by the lowest energy found.

In the second leaderboard table, VQEs are ranked based on how close the found equilibrium bond distance is to the equilibrium bond distance of the "ground truth." A VQE might find a very low energy, but if its minimum is at the wrong bond distance, it cannot be used for applications where the equilibrium bond distance is important.

In the last leaderboard table, the VQEs are ranked based on how well they approximate the curve found by the classical "ground truth" baseline. As a metric, we use the variance in error with respect to the ground truth over all bond distances. A VQE might not find the lowest energy, but approximating the shape of the curve can still be useful for applications where the curve can be calibrated in some way.

Moreover, each of the three tables in the leaderboard has the functionality to select rows for a detailed comparison. Users can also double-click on the row to inspect the details of an experiment.

Additionally, it is possible to make a selection of VQEs to plot their respective energy curves in a single figure [see Fig. 1(a)]. Moreover, if the VQEs in the selection are tested with the same basis set, we can plot the CNOT gate depth of the used quantum circuit against the accuracy of the VQE (as

[2]Using your QuantMark account that can be linked to your ORCID, Google, and/or Facebook account.

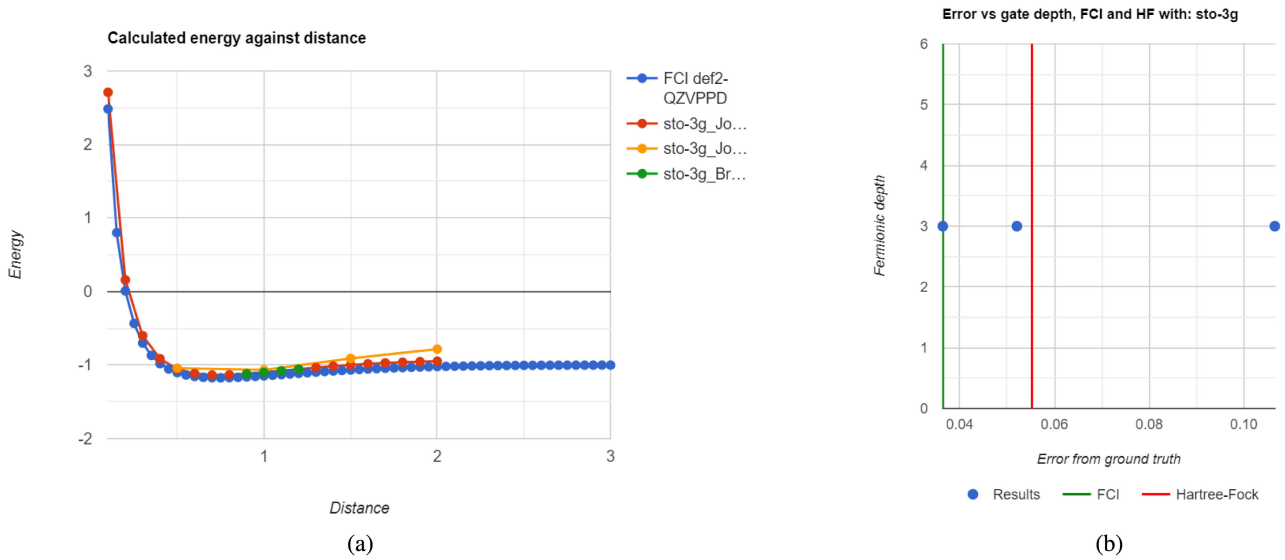


**FIG. 1. Screenshots from different comparison plots on the QuantMark website. (a) Energy curves found by different VQEs. (b) CNOT depth versus accuracy.**

was used in [17]) at equilibrium bond distance. This gives a clear overview of the tradeoff of accuracy versus circuit depth for different algorithms. An example plot is shown in Fig. 1(b), in which the $x$-axis is the accuracy with respect to our predefined ground truth. The green (left) and red (right) vertical lines, respectively, show the energy found by the FCI and Hartree–Fock methods in the same basis set as the VQE. By the design of the VQE, its performance is bound by the green line. In contrast, the red line shows the performance of a classical algorithm that only takes a single Slater determinant into account and is, therefore, a less computationally expensive approximation. Thus, the performance of a reasonably good VQE should lay between those two lines. Combining these insights with the depth of the circuit gives users new and interesting insights into the performance of their algorithm with respect to other methods.

### D. DOWNLOADING AND RECONSTRUCTING VQES

One of the most versatile features of QuantMark is the ability to download the experimental data of publicly available results. We strive to collect enough data about the VQE algorithm to be able to replicate the results such that they can easily be verified by third-party researchers. This is simply good science, but the lack of experimental details in papers or the lack of source code has made it too time-consuming to faithfully replicate results. Fortunately, since one only needs to know the Hamiltonian, the quantum circuit, and the classical optimizer (including hyperparameters), QuantMark collects enough data for replication. Thus, it stands to reason that users should be able to download it and immediately run the VQE when it was published in QuantMark.

The QuantMark Python API exposes the downloading of the raw stored data from the database. Moreover, there is a simple API call that allows users to rerun an experiment from the leaderboard on their own device. The script below gives an example of how this can be done. Note that the experiment will be executed locally on the user's device.

```
id = "ID HERE"
token = "TOKEN HERE"
data = lm.api.get_data(id, token)
old_exp = lm.api.get_experiment(id, token)
replication = old_exp.run_experiment()
```

With this, they can either verify the results on the leaderboard or make adjustments to the previous research to make it better. For example, suppose I see a VQE on the leaderboard that performs reasonably well on a quantum simulator. I have access to a real quantum computer and I wonder how well it would perform. With QuantMark, I can simply download the data and adjust the reconstructed VQE to work with the quantum computer that I have access to. This is much simpler and faster than asking the authors for the source code or trying to reimplement the algorithm from the paper.

Moreover, the trained VQE might be repurposed for other simulations or optimization problems. For example, the trained parameters could be used for a "warm start" of another parameterized circuit, similar to transfer learning in neural networks [18]. Or, using perturbation theory, the original parameterized circuit can be slightly adjusted to account for a new problem.

Thus, QuantMark will help with the cycle of improvement and collaboration.

## III. DISCUSSION AND CONCLUSION

In this article, we have shown how QuantMark can help with the evaluation and benchmarking of new VQE algorithms, as well as the replication of existing ones. In Table I, we give an

IEEE Transactions on Quantum Engineering

**TABLE I Overview of Challenges Addressed in QuantMark**

(a) **Challenge 1:** Evaluating VQE performance

| Implemented solution | pros | cons |
|---|---|---|
| Precision: Compare against FCI in VQE basis set | Well defined | Does not measure the real objective |
| Accuracy: Compare against the best classical method | The true performance of the VQE | Not scalable and ill-defined |

(b) **Challenge 2:** Choice of performance metrics

| Implemented solution | pros | cons |
|---|---|---|
| Minimum energy found | Does not require a classical baseline | Found energy might not correspond with the equilibrium bond distance |
| Closest equilibrium distance | Shows equilibrium bond distance found | Requires a classical baseline |
| Best shape of the energy curve | Shows generalizability of the VQE to non-equilibrium states | Smallest variance might not quantify the metric well |

(c) **Challenge 3:** Comparing different VQE algorithms

| Implemented solution | pros | cons |
|---|---|---|
| Same molecule; compare accuracy | Basis set independent | Differences might be basis set/encoding dependent |
| Same basis set; compare precision | Differences are VQE dependent | Differences might not be generalizable |
| Same basis set; compare circuit | Shows differences in resource cost | Differences might be encoding dependent |

(d) **Challenge 4:** Replicating VQE results

| Implemented solution | pros | cons |
|---|---|---|
| VQE reconstruction | Does not require source code | Original algorithm still unknown |
| Source code link | Source code availability | It is optional |

overview of the four challenges that QuantMark attempts to solve, how a solution is implemented, and the pros and cons of that choice of implementation.

The challenge of evaluating a VQEs performance lies between the choice of precision and accuracy. Precision is what we want to measure, but the true energies might not be known.

Similarly, the choice of performance metrics might influence how future VQEs are tuned for the leaderboard. We needed metrics that can be generalized for different tasks, such that we know what choice of VQE is good for which task. Hence, we created three different leaderboards for three different aspects of the VQEs results: the minimum energy found, the closest equilibrium bond distance found, and the best shape of the full energy curve.

Comparing different VQE algorithms is also challenging. When using the same molecule, basis set, and qubit encoding, the differences are clearly due to the VQE itself. However, some choices of ansatz might not be suitable for the same qubit encoding, and then, the different VQEs cannot be compared. Alternatively, when using the same molecule, basis set, and qubit encoding, you can never make claims about qubit cost versus performance. Thus, if two VQE experiments use the same molecule, we can compare their precision. If the experiments also use the same basis set, we can compare their accuracies and their quantum circuits.

Finally, before the creation of QuantMark, it was very difficult to replicate an existing VQE paper as a starting point for continued research because the source code was not available and hyperparameters are not always present in the paper. Now, if a VQE result is available on QuantMark, you can download its data and find the hyperparameters there and rerun the VQE on your own device. However, the original code is still unknown. Nevertheless, it might be easily found if the creator of the VQE added a link to the source code.

## IV. FUTURE WORK

QuantMark is still under active development. Therefore, the user interface, Python API, and data management functionality will still be improved and extended. However, these are only small changes in the scope of the research that still needs to be done for benchmarking VQEs and the future of QuantMark.

Current VQE experiments are toy problems to show that the concept works, which has been established now. Next, we need to figure out how well this VQE works on larger problems.

First, we need to investigate the scalability of the VQE with the support of QuantMark. By using the gathered performance data, we can try to make predictions about the scalability of the VQE in terms of quantum resources and trainable parameters. How many qubits and quantum gates are needed? Gonthier *et al.* [2] showed that quantum advantage in quantum chemistry requires hundreds of fault-tolerant qubits and months of runtime. However, when we have hundreds of fault-tolerant qubits, would we be able to classically optimize the thousands of parameters in the VQE?

Then, we need to create a more realistic leaderboard. The current QuantMark leaderboard reflects the toy model problems that VQEs are used for. Ideally, we want quantum chemistry problems that are small enough to be calculated on a near-term quantum computer, but large enough to be classically challenging. We need to develop interesting benchmarking problems, such as the ones proposed in [1], and create a leaderboard for those.

Additionally, we need to carefully consider the classical baselines for larger problems. Fortunately, because QuantMark shares its classical baselines, it is worthwhile to spend a lot of computational power on FCI baselines. However, for large enough problems, it will not be feasible to compute the full FCI at all. In that case, we need to think carefully about which approximate classical computational methods will make good baselines. For example, complete active space configuration interaction or complete active space self-consistent field.

Moreover, the biggest challenge in creating a benchmark of good candidates of quantum advantage is that we will not have a classical baseline. In which case, how can we know that our optimizer found the right answer? How do we make sure that what we created is actually solving the problem? And if we were to use all the computational power in the world to find a classical baseline, how can we make sure that the algorithm that can find that baseline can also be used for a larger analogous problem for which there is no classical baseline? These are the questions on which the future of VQE hinges.

## ACKNOWLEDGMENT

QuantMark was implemented by Leo Becker, Alejandro Bravo, Jami Heljomaa, Tapani Honkanen, Akseli Nurmi, Petri Riihikallio, Henri Sundquist, Akira Taguchi, Matias Tamsi, Lauri Vuorenkoski, and Sonja Vuorinen. The authors would like to thank Andrew Tranter for his valuable feedback.

## REFERENCES

[1] V. E. Elfving *et al.*, "How will quantum computers provide an industrially relevant computational advantage in quantum chemistry?," 2020, *arXiv:2009.12472*, doi: 10.48550/arXiv.2009.12472.

[2] J. F. Gonthier, M. D. Radin, C. Buda, E. J. Doskocil, C. M. Abuan, and J. Romero, "Identifying challenges towards practical quantum advantage through resource estimation: The measurement roadblock in the variational quantum eigensolver," 2020, *arXiv:2012.04001*, doi: 10.48550/arXiv.2012.04001.

[3] A. Peruzzo *et al.*, "A variational Eigenvalue Solver on a photonic quantum processor," *Nature Commun.*, vol. 5, 2014, Art. no. 4213, doi: 10.1038/ncomms5213.

[4] D. A. Fedorov, B. Peng, N. Govind, and Y. Alexeev, "VQE method: A short survey and recent developments," *Mater. Theory*, vol. 6, no. 1, 2022, Art. no. 2, doi: 10.1186/s41313-021-00032-6.

[5] A. Kandala *et al.*, "Hardware-efficient variational quantum EigenSolver for small molecules and quantum magnets," *Nature*, vol. 549, no. 7671, pp. 242–246, 2017, doi: 10.1038/nature23879.

[6] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, "An adaptive variational algorithm for exact molecular simulations on a quantum computer," *Nature Commun.*, vol. 10, no. 1, 2019, Art. no. 3007, doi: 10.1038/s41467-019-10988-2.

[7] S. B. Bravyi and A. Y. Kitaev, "Fermionic quantum computation," *Ann. Phys.*, vol. 298, no. 1, pp. 210–226, 2002, doi: 10.1006/aphy.2002.6254.

[8] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Trans. Autom. Control*, vol. 37, no. 3, pp. 332–341, Mar. 1992, doi: 10.1109/9.119632.

[9] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nature Commun.*, vol. 9, no. 1, 2018, Art. no. 4812, doi: 10.1038/s41467-018-07090-4.

[10] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, "Cost-function-dependent barren plateaus in shallow quantum neural networks," 2020, *arXiv:2001.00550*, doi: 10.1038/s41467-021-21728-w.

[11] S. Wang *et al.*, "Noise-induced barren plateaus in variational quantum algorithms," *Nature Commun.*, vol. 12, no. 1, 2021, Art. no. 6961, doi: 10.1038/s41467-021-27045-6.

[12] C. O. Marrero, M. Kieferová, and N. Wiebe, "Entanglement-induced barren plateaus," *PRX Quantum*, vol. 2, no. 4, 2021, Art. no. 040316, doi: 10.1103/PRXQuantum.2.040316.

[13] M. Larocca, P. Czarnik, K. Sharma, G. Muraleedharan, P. J. Coles, and M. Cerezo, "Diagnosing barren plateaus with tools from quantum optimal control," 2021, *arXiv:2105.14377*, doi: 10.48550/arXiv.2105.14377.

[14] J. S. Kottmann *et al.*, "Tequila: A platform for rapid development of quantum algorithms," *Quantum Sci. Technol.*, vol. 6, no. 2, 2021, Art. no. 024009, doi: 10.1088/2058-9565/abe567.

[15] Z.-J. Zhang, T. H. Kyaw, J. S. Kottmann, M. Degroote, and A. Aspuru-Guzik, "Mutual information-assisted adaptive variational quantum Eigensolver," *Quantum Sci. Technol.*, vol. 6, no. 3, 2021, Art. no. 035001, doi: 10.1088/2058-9565/abdca4.

[16] D. Hardt *et al.*, *The OAuth 2.0 Authorization Framework*. Standard draft-ietf-oauth-v2-31, 2012.

[17] A. G. Rattew, S. Hu, M. Pistoia, R. Chen, and S. Wood, "A domain-agnostic, noise-resistant evolutionary variational quantum eigensolver for hardware-efficient optimization in the hilbert space," 2019, *arXiv:1910.09694*, doi: 10.48550/arXiv.1910.09694.

[18] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010, doi: 10.1109/TKDE.2009.191.

**Adriana Meijer - van de Griend** (Member, IEEE) was born in Alblasserdam, The Netherlands, in 1993. She received the B.Sc. degree in computer science and the B.Sc. degree in artificial intelligence from the University of Groningen, Groningen, The Netherlands, in 2015 and 2019, respectively. Also in 2019, she received both the M.Sc. degree in computer science and the M.Sc. degree (*cum laude*) in artificial intelligence from Radboud University, Nijmegen, The Netherlands. She is currently working toward the Ph.D. degree in computer science with the University of Helsinki, Helsinki, Finland.

In 2019, she was a Teaching Assistant with Radboud University. From 2019 to 2020, she was an intern as a Research Scientist with Cambridge Quantum Computing, Cambridge, U.K. In 2020, she was a Project Researcher with the University of Turku, Turku, Finland. Her research interests include noisy intermediate-scale quantum algorithms, quantum circuit optimization, quantum compiling, and, more generally, quantum advantage.

**Jukka K. Nurminen** (Member, IEEE) received the M.Sc. and Ph.D. degrees in systems and operations research from the Helsinki University of Technology (now Aalto University), Espoo, Finland, in 1986 and 2003, respectively.

In Spring 2019, he joined the University of Helsinki, as a Professor of Computer Science. He has worked extensively on software research in industry with Nokia Research Center, Helsinki, in academia with Aalto University, and in applied research with VTT, Espoo. His main research interests include tools and techniques for developing data-intensive software systems, including testing of artificial intelligence solutions as well as software development for quantum computers.