

Classification of Hybrid Quantum-Classical Computing

Frank Phillipson^{1,2}[0000-0003-4580-7521], Niels Neumann¹[0000-0003-2159-8251],
and Robert Wezeman¹[0000-0002-0810-2840]

¹ TNO, Dpt. Applied Cryptography & Quantum Algorithms, The Hague, The Netherlands

² Maastricht University, School of Business and Economics, Maastricht, The Netherlands

{frank.phillipson,niels.neumann,robert.wezeman}@tno.nl

Abstract. As quantum computers mature, the applicability in practice becomes more important. Quantum computers will often be used in a hybrid setting, where classical computers still play an important role in operating and using the quantum computer. However, the term hybrid is diffuse and multi-interpretable. In this work we define two classes of hybrid quantum-classical computing: vertical and horizontal hybrid quantum-classical computing. The first is application-agnostic and concerns using and operating quantum computers. The second is application-specific and concerns running an algorithm. For both, we give a further subdivision in different types of hybrid quantum-classical computing and we introduce terms for them.

Keywords: hybrid quantum computing · classification · hybrid quantum algorithm · workflow

1 Introduction

Quantum computing is the technique of using quantum mechanical phenomena such as superposition, entanglement and interference for doing computational operations. The type of devices which are capable of doing such quantum operations are still being actively developed and named quantum computers. We distinguish between two paradigms of quantum computing devices: gate-based quantum computers and quantum annealers. A gate-based quantum computing system uses basic quantum circuit operations on qubits, similar to the classical operations on regular bits, that can be put together in any sequence to form algorithms. A quantum annealer brings a collection of qubits into an equal superposition and then applies a problem specific magnetic field. The qubits will interact under this magnetic field and move towards the state with the lowest energy, which encodes the solution of an optimisation problem.

Theory predicts that quantum computers will solve specific problems much faster than classical computers. Where classical computers have been under development for decades and are therefore quite mature, quantum computers are

still in the early stages of development. They are not yet capable of solving real world problems, due to the low number of qubits and their unstable nature causing noise, errors and loss of information. This current state of quantum computers is called the noisy intermediate-scale quantum (NISQ) era introduced by Preskill [42], where quantum computers have 50-200 qubits and their noise places serious limitations on their capabilities. Only recently IBM passed the 100-qubit barrier for the first time with its 127-qubit Eagle processor and they plan a 1,000-qubit chip, the Condor, in 2023 [26].

Other aspects than number of qubits affect the capabilities of quantum computers in practice, such as parallelism of operations and the topology layout of the qubits. Researchers are investigating innovative ways to solve valuable problems using already available NISQ systems and to achieve quantum advantage by demonstrating a significant performance advantage over today's classical computers. The latter was only quite recently claimed for the first time for an artificially created problem [2]. As quantum computers mature in the coming years, their computational power will increase and they can be applied in more settings and actually provide help in specific practical areas.

For the first practical applications in the near future, quantum computers will only execute a small part of a larger total workflow where a classical computer executes the other steps. See for example the workflow of retrieving data, training a classification algorithm and evaluating the obtained classifier [55], as shown in Fig. 1. In this example only a small part of the workflow is performed on a quantum computer. Already in 2005 such a combination was described in literature and was named hybrid quantum computing that “combine both classical and quantum computing architectures in order to leverage the benefits of both” [31].

Murray Thom, the Vice President of Software and Cloud Services at D-Wave, compared this with jets and “normal” vehicles: “Consider that while jet airplanes transformed the way we travel long distances, we still need vehicles that take us to our front door.” Thus, “quantum applications will always and only be hybrid” [37].

This term, hybrid quantum-classical computing, was and is used in various contexts, each time used for (slightly) different settings. This is confusing for many researchers and practitioners in quantum computing, both on the application side as on the hardware side. The interest in quantum computing causes an enormous amount of research output³ and, resulting, survey papers. One example of this kind is [14], where a survey of NISQ era hybrid quantum-classical machine learning research on hybrid quantum-classical systems is given, without explaining what is meant by hybrid quantum-classical exactly. This is totally left to the reader. Our goal is to provide a classification framework where authors can refer to, to make the scope of their contribution more clear to the reader. For this, we connect quantum research with classical computer science and workflow terminology. So, many of the proposed terminology in this paper exists already in

³ Google scholar already gives 2090 results for the search on ‘hybrid quantum-classical computing’ for the period January-October 2022.

computer science, think of compilation between computer languages, and workflow research, think of decomposition and activities. Our contribution is bringing them together in a clear framework for hybrid quantum-classical computing.

In this contribution we aim to describe clearly the various contexts hybrid quantum-classical computing can have in literature and to name these different approaches clearly and appropriately. In Section 2 we give an overview of various forms of hybrid quantum-classical computing that can be found in literature. Next, in Section 3, we distinguish a number of different types of hybrid quantum-classical computing from this overview and provide examples for each type. We end this paper with some conclusions and ideas for further research.

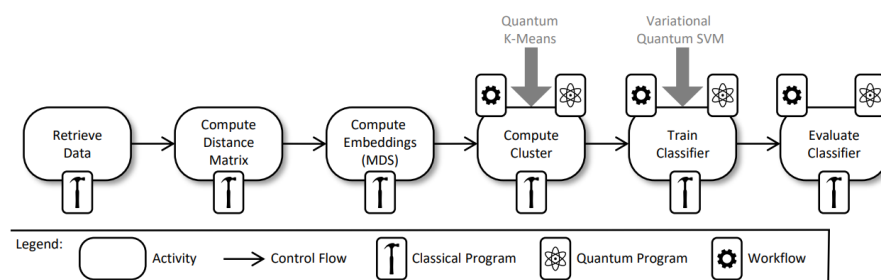


Fig. 1. An example workflow of a hybrid quantum-classical application where quantum computers perform only a small part of the computations [55]

2 Literature

We consider a global situation where we have a collection of computational tasks in which both the quantum computer and the classical computer are used. As such, hybrid forms of computing that allow for both discrete and continuous variables [34] and hybrid quantum-classical models of molecules in chemical and biological studies [22] are out of scope. We do not try to give an exhaustive overview of all research done on this topic. Our goal here is to give an overview, based on some examples, of the various meanings of the term hybrid quantum-classical computing in literature. This overview will be the basis of the proposed classification later on in this work.

Lanzagorta and Uhlmann presented one of the first hybrid algorithms that used both classical and quantum computers [31]. Later, research appeared on computing schemes and architectures to optimise the interactions between the different type of computers when executing hybrid quantum algorithms. A first example presents a candidate framework to analyse hybrid computations by fully integrating the quantum and classical resources and processes used for measurement-based quantum computing, where the feed-forward of classical

measurement results is an integral part of the quantum design [27, 25]. A second example proposes a quantum co-processor to accelerate a specific subroutine of a larger task. This is most often seen as the main reason for hybrid algorithms, for example in [41, 5, 33, 36]. The work by Li et al. [33] results in a system-level software infrastructure for hybrid quantum-classical computing. Endo [18, 19] indicates that for early quantum applications, a large portion of the computational burden is performed on a classical computer and hence fully coherent deep quantum circuits are not required. As the quantum computer takes on more computational load, noise of the quantum computer will result in more errors, which will have more impact on the total calculation. This in turn requires qubits of higher quality and error mitigation routines to suppress noise.

An important type of hybrid computing appeared with the introduction of Variational Quantum Algorithms (VQA). VQAs use a classical optimiser to train a parameterised quantum circuit and provide a framework to tackle a wide array of tasks, as shown in the extensive overview by Cerezo et al. [9]. Examples include finding ground states and excited states of molecules, optimisation, solving linear systems of equations and machine learning. The first VQA, the variational quantum eigensolver (VQE) algorithm of [39], appeared in 2014. Some papers see this group of algorithms as “a novel class of hybrid quantum-classical algorithms”, without explicitly diving in other groups of algorithms [7]. Also, [11] sees this class of algorithms as a specific example of hybrid quantum-classical computing in a noisy environment. In [54] the main reason given for hybrid quantum-classical computing is the size of the problems in combination with the available hardware. They distinguishes two types of hybrid computing: 1) Pre- or post-processing of a quantum computation on a classical computer. Examples are algorithms by Shor [50] and Simon [51] that use classical post-processing. 2) Algorithms that perform multiple iterations of quantum and classical computations. Thereby, the output of the quantum computation is improved in each iteration until the result reaches the required accuracy. An example they give using this approach is the quantum approximate optimisation algorithm (QAOA) [20].

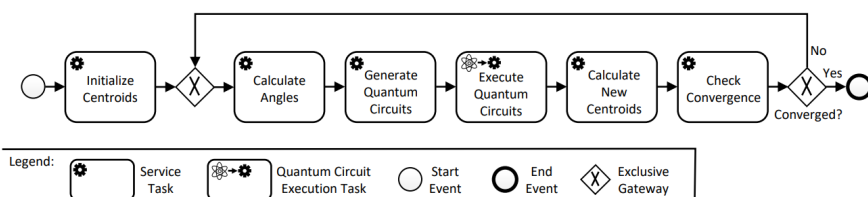


Fig. 2. The sub-workflow of the “Compute Cluster activity” shown in Fig. 1 [55]

In the works [7, 8], the term ‘quassical’ computing is coined and motivated by “Classical computing and quantum computing have obvious complementary

strengths, so instead of opposing them it might be better to combine them into a new type of computing.” They give two reasons for the combination: First, most quantum computing algorithms “require some preliminary classical pre-processing to shape the problem into one the quantum computer can recognise and then to receive the data returned by the [quantum computer] and shape it into the answer the engineer needs.” Second, “all the quantum computers we have heard of are designed as cyber-physical systems, quantum mechanical systems controlled by digital controllers”, meaning they are quassical in a trivial sense. In this light, you can also think of the classical steps needed to transform a quantum circuit to an execution as performed for example by an openQL framework [29]. They expect that this combination will stay, also when the quantum computer is in full maturity. The first remark is also mentioned by [16], who indicate that “while hybrid algorithms and platforms may just be the best first step, it is reasonable to assume that quantum applications will always be hybrid”, for example, by the need of a pre-processing step which prepares data for a quantum algorithm or a post-processing step which handles data coming from a quantum algorithm.

Another view on hybrid algorithms is given by the idea that a problem or circuit that is too big to be executed on (noisy) quantum processors of intermediate size, is partitioned automatically into smaller parts that are evaluated separately. Suchara et al. [53] suggest such an approach for gate-based quantum computers: “We advocate using a hybrid quantum-classical architecture where larger quantum circuits are broken into smaller sub-circuits that are evaluated separately, either using a quantum processor or a quantum simulator running on a classical supercomputer.” The D-Wave hybrid solvers should also be seen in this light. They offer the functionality to partition the problem into smaller pieces that fit the current chip size and are solved sequentially. The outcomes of the subroutines form the resulting (probably sub-optimal) solution. They also provide hybrid approaches, where multiple branches of a process solve the problem, some of them with a classical solver, others using the QPU, and then return the first or best solution [3]. Here, the quantum and classical computers compete in parallel to find a solution to the problem. In [10] the D-Wave Kerberos solution is used, a hybrid built-in sampler, that combines Tabu search, simulated annealing, and D-Wave sub-problem sampling on problem variables that have high-energy impact.

3 Types of Hybrid Computing

To make a clear distinction between all the different views on hybrid quantum-classical computing, we use the workflow approach presented in [55]. They propose workflows to specify the (partial) order of a collection of activities needed to execute a hybrid quantum-classical application and combine this with topologies to reveal the overall structure of hybrid quantum applications. An activity in such a workflow can be further expanded in sub-workflows. Typically, the activities are represented as nodes in a directed graph with the control flow de-

dependencies the directed edges of the graph. Figure 1 shows an example, where the workflow of a hybrid quantum-classical application in quantum machine learning is shown. The presented quantum application performs clustering on a set of input data, and based on the clustering results, trains a classifier for the classification of future data. The “gear” icon indicates sub-workflows. An example of the “Compute Cluster” sub-workflow is shown in Fig. 2. They consider two dimensions of hybrid computing, the vertical provisioning engine and the horizontal workflow-engine, from a software architecture point of view. This insight and the workflow technology are the basis of our proposed classification framework. However, we elaborate further on this and expand it into sub-categories.

We distinguish two main categories of hybrid computing:

1. Vertical hybrid quantum computing: All controlling activities required to control and operate a quantum circuit on a quantum computer, as was the case in classical computing providing compilation and controlling in the stack. An example of a quantum stack can be found in [43]. These steps are application-agnostic.
2. Horizontal hybrid quantum computing: All operational activities required to use a quantum computer and a classical computer to perform an algorithm. These steps are application-dependent. Here we use the classical workflow terminology as proposed in [17].

We can subdivide both categories further, as shown in Table 1, and explained further in the next sections. Note that some use-cases might show signs of more than one type of hybrid quantum-classical computing. It is also important to stress that these main categories are not mutually exclusive. The vertical category is mainly about computing and computing stack, the horizontal category is mainly about algorithms.

Table 1. Overview of the main categories and their sub-categories

Vertical hybrid	Horizontal hybrid
Decomposition hybrid	Processing hybrid
Implementation hybrid	Micro hybrid split
Controlling hybrid	Macro hybrid split
	Parallel hybrid
	Breakdown hybrid

3.1 Vertical Hybrid Quantum Computing

The types of vertical hybrid quantum computing contain the classical steps that have to be taken to let the quantum computer run the quantum routine. A

general overview, starting from a single activity in a sub-workflow is depicted in Fig. 3. The specific steps are described in the next sub-sections. These steps are in some way similar to the layers in the OpenQL framework [29], however, we distinguish more between steps that are topology and technology-agnostic and steps that are not.

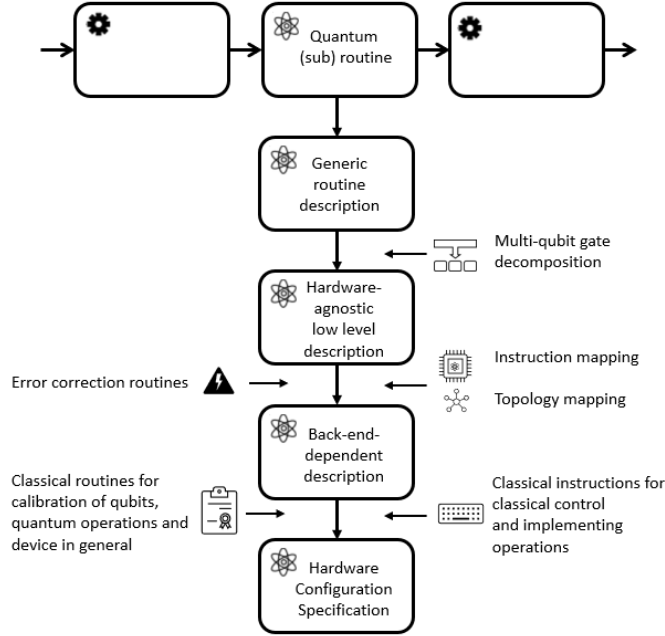


Fig. 3. Schematic view on relation between horizontal and vertical dependencies

Decomposition Hybrid The workflows of decomposition hybrid consider a higher level algorithm description, which is then decomposed in classical instructions and quantum instructions, see Fig. 4. A high-level quantum routine is broken down into low-level hardware-agnostic quantum instructions. The higher level algorithm description can be any commonly used classical language, enhanced with classical routines, or a dedicated quantum routine. For gate-based devices, this hardware-agnostic instruction set can for instance include single qubit rotations and some two-qubit gates such as the CNOT-gate and controlled-phase-gates. Libraries can help decomposing quantum instructions to low level hardware-agnostic instructions [6].

Examples of decomposition hybrid include decomposing algorithmic instruction to classical instructions and quantum instructions and mapping both to low-level instructions. This includes decomposing high-level instructions not suited

for the hardware to lower-level instructions with a more direct mapping to hardware. Classical routines can help with this decomposition [30]. The vertical hybrid can be separated into the following parts, which are depicted in Fig. 4.

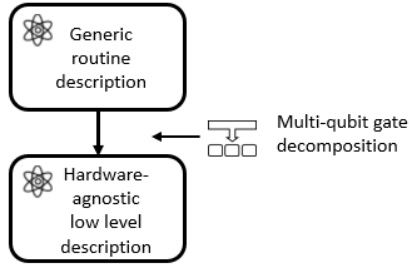


Fig. 4. Schematic view on *decomposition hybrid*: a high-level quantum routine is broken down into low-level hardware-agnostic quantum instructions

Implementation Hybrid The workflows of implementation hybrid consider all steps to map and implement the operations on a quantum computer. This workflow specifically aims to map the hardware-agnostic low-level instructions to hardware-specific instructions, see Fig. 5. This includes both the classical and quantum instructions.

Examples of implementation hybrid include mapping instructions to hardware. This includes assigning operations to qubits and taking into account the topology of the hardware backend. This workflow also outputs a time-schedule for which hardware instructions should be applied to which qubits. If necessary, this workflow also adds error correcting routines together with the classical feedback loop of these routines.

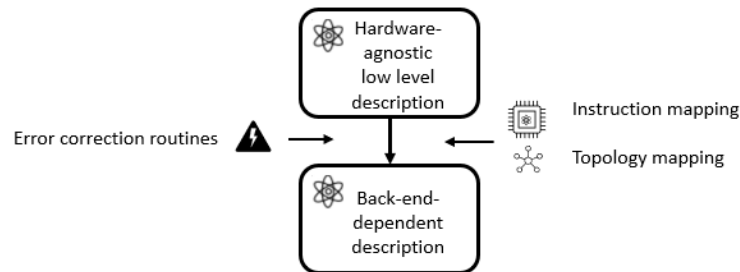


Fig. 5. Schematic view on *implementation hybrid*: low-level quantum instructions are mapped to a specific hardware-backend. This includes creating a time-schedule, assigning operations to qubits and, if necessary, apply error correcting routines

Controlling Hybrid The workflows of controlling hybrid consider all steps to operate and control a quantum computer, see Fig. 6. Due to the intricate nature of quantum computers, their operations might behave differently over time than intended. Therefore, continuous effort is needed to ensure that quantum computers behave as expected.

Examples of controlling hybrid include calibration routines of the qubits and of elementary operations on the qubits.

All of the steps in vertical hybrid quantum computing can include optimisation steps. Sometimes, optimised approximation methods yield better performing implementation than exact full implementations, for instance see [47].

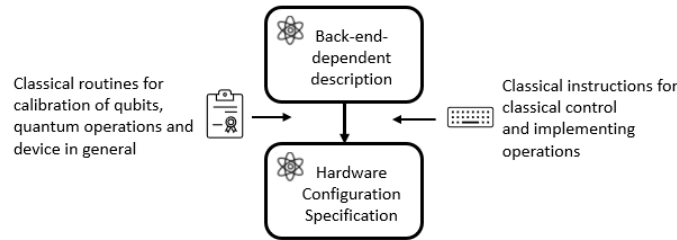


Fig. 6. Schematic view on *controlling hybrid*: this considers all steps to operate and control a quantum computer, including the actual mapping of hardware instructions to qubits and calibrating the quantum computer

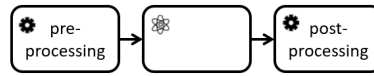


Fig. 7. Schematic *processing hybrid* workflow: a single quantum routine with additional classical pre- and post-processing

3.2 Horizontal Hybrid Quantum Computing

The types of horizontal hybrid quantum computing distinguish between the variety of orderings quantum and conventional computing steps within an algorithm.

Processing Hybrid The workflows of processing hybrid have a single quantum block, combined with classical pre-processing and classical post-processing [7, 16]. A schematic representation is shown in Fig. 7.

Examples of processing hybrid algorithms are the algorithms by Shor and Simon [53]. Another example is the distance based classifier [49, 56], where data

standardisation and normalisation are the pre-processing steps and translating the measurements to the desired kernel classifier the post-processing step.

Micro Hybrid Split The workflows of a micro hybrid split consider a single activity of a larger workflow. The workflow shown in Fig. 2 is a micro hybrid split of Fig. 1. Within the single activity, some operations are quantum and others are classical, possibly in an iterative fashion. A schematic representation is shown in Fig. 8.

Examples of micro hybrid splits are variational algorithms [7]. Measurement-based quantum computing can be seen as a special member of this class, as future measurements and operations depend on previous measurements and classical operations.

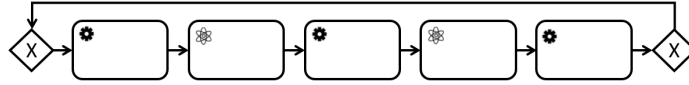


Fig. 8. Schematic *micro hybrid split* workflow: a single activity has both classical and quantum operations

Macro Hybrid Split The workflows of a macro hybrid split consider different tasks that belong to different activities within a larger algorithm. The difference with micro hybrid split workflows is small and depends mainly on the granularity with which the workflow is observed: A micro hybrid split in one workflow can be a small part of a larger macro hybrid split workflow. The activities in a macro hybrid split can also iterate and each task can be hybrid in itself. A schematic representation is shown in Fig. 9. A possible relation between macro and micro hybrid splits is shown in Fig. 10

Examples of macro hybrid splits are the hybrid quantum machine learning approach in the domain of humanities [55] and the workflow shown in Fig. 1.

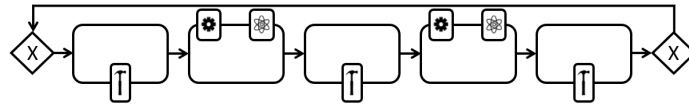


Fig. 9. Schematic *macro hybrid split* workflow: Each block is a specific task that can have both quantum and classical operations. A single block can be further subdivided in a micro hybrid split workflow

Parallel Hybrid The workflows of parallel hybrid have multiple independent branches to solve a specific problem. Each branch tries to solve the problem

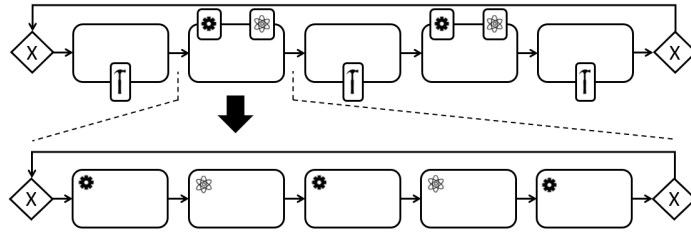


Fig. 10. Schematic relationship between *macro hybrid split* and *micro hybrid split* workflow: a block in the *macro hybrid split* can be specified as a *micro hybrid split*

independently and the first (or best) solution found is returned. Each branch can use different solvers. A schematic representation is shown in Fig. 11.

Examples include the configuration of the D-Wave-hybrid framework, where samples are parsed to four parallel solvers. One branch can for instance be a classical tabu search that either returns with certainty an answer, or is interrupted by another finished branch [3, 10].

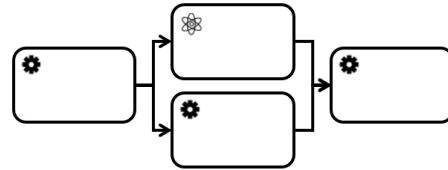


Fig. 11. Schematic *parallel hybrid* workflow: a task is processed by multiple independent branches. The answer is returned based on some criteria, for instance, coming from the branch that finishes first

Breakdown Hybrid The workflows of breakdown hybrid consider multiple small parts of a larger problem. The considered problem is too large to solve directly and is hence broken down in multiple smaller parts. Each smaller part is run on a quantum computer sequentially and the final answer is reconstructed from the partial answers. A schematic representation is shown in Fig. 12.

Examples are the gate-based approach in [53], where they advocate using a hybrid quantum-classical architecture where larger quantum circuits are broken into smaller sub-circuits that are evaluated separately and specific options within D-Wave’s hybrid solvers [3], where the problem divided into several parts that are solved using classical or quantum annealing approaches. Note that these classes are not disjoint. Here, one of the breakdown parts might be run on a classical computer in parallel with one or more quantum tasks. This would make it a combination of breakdown hybrid and parallel hybrid.

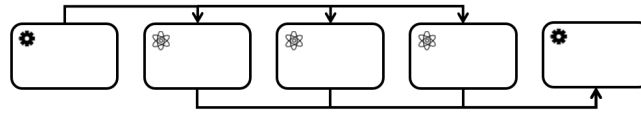


Fig. 12. Schematic *breakdown hybrid* workflow: a large problem is decomposed in smaller problems, each of which is run on a quantum computer, the final answer is reconstructed from the partial answers

4 Application

As indicated in the introduction, the number of papers that are categorised under hybrid quantum-classical computing is enormous. We will not give an exhaustive overview of all papers and their classification. As an example, we selected a few papers from 2022 that have this terms in their title or key words to illustrate the classification.

In Table 2, this classification is shown. In this table, but also in reality, the majority of papers are within the micro and macro hybrid split classes. We could not find any papers within the processing hybrid class that use the terminology hybrid quantum-classical. Papers in this class mostly use the term quantum algorithm.

5 Conclusions

It is expected that quantum computing will always need some form of classical computing to enable the calculations and the execution on the hardware platforms. This is often named hybrid quantum-classical computing. The term hybrid is however diffuse and multi-interpretable. We showed in this paper that in literature this term covers many concepts. Based on this literature and concepts from workflow approach and classical computer science, we distinguished between horizontal and vertical hybrid quantum computing and defined and named various specific types within these classes. This can help researchers and practitioners in quantum computing to make clear what they mean when using the general term ‘hybrid quantum-classical computing’ and can help in developing more concise tools within the quantum computing stack. We do not assume to be complete in our overview and categorisation. We encourage scientists and practitioners to complement this framework as part of future research on this topic.

References

1. Arthur, D., et al.: A hybrid quantum-classical neural network architecture for binary classification. arXiv:2201.01820 (2022)
2. Arute, F., Arya, K., et al.: Quantum supremacy using a programmable superconducting processor. *Nature* **574**(7779), 505–510 (2019)

Table 2. Example classification of recent papers

Title	Classification
NetQASM-a low-level instruction set architecture for hybrid quantum-classical programs in a quantum internet [13]	Decomposition
QuantumPath: A quantum software development platform [23]	Decomposition
An LLVM-based C++ compiler toolchain for variational hybrid quantum-classical algorithms and quantum accelerator [28]	Implementation
Diversity of hybrid quantum system [24]	Controlling
Quantify-scheduler: An open-source hybrid compiler for operating quantum computers in the NISQ era [12]	Controlling
The optimization landscape of hybrid quantum-classical algorithms: from quantum control to NISQ applications [21]	Micro split
A hybrid quantum-classical CFD methodology with benchmark HHL solutions [32]	Micro split
Considerations for evaluating thermodynamic properties with hybrid quantum-classical computing work flow [52]	Micro split
Hyperparameter optimization of hybrid quantum neural networks for car classification [45]	Macro split
Hybrid quantum-classical algorithms for approximate graph coloring [4]	Macro split
A hybrid quantum-classical neural network architecture for binary classification [1]	Macro split
Hybrid quantum-classical algorithm for computing imaginary-time correlation functions [46]	Macro split
Hybrid quantum-classical algorithm for hydrodynamics [58]	Macro split
Hybrid quantum-classical search algorithms [44]	Macro split
A quantum algorithm of k-means toward practical use [38]	Macro split
A hybrid quantum-classical algorithm for robust fitting. [15]	Macro split
Graph- $ Q \gg C$, a graph-based quantum/classical algorithm for efficient electronic structure on hybrid quantum/classical hardware systems: Improved quantum circuit depth performance [57]	Parallel
Optimization of robot trajectory planning with nature-inspired and hybrid quantum algorithms [48]	Parallel
A quantum approach for tactical capacity management of distributed electricity generation [40]	Parallel
Hybrid quantum-classical unit commitment [35]	Breakdown

3. Booth, M., Reinhardt, S.P., Roy, A.: Partitioning optimization problems for hybrid classical. quantum execution. Technical Report pp. 01–09 (2017)
4. Bravyi, S., Kliesch, A., Koenig, R., Tang, E.: Hybrid quantum-classical algorithms for approximate graph coloring. *Quantum* **6**, 678 (2022)
5. Bravyi, S., Smith, G., Smolin, J.A.: Trading classical and quantum computational resources. *Physical Review X* **6**(2), 021043 (2016)
6. Van den Brink, R., Phillipson, F., Neumann, N.M.: Vision on next level quantum software tooling. *Computation Tools* (2019)
7. Calude, C.S., Calude, E., Dinneen, M.J.: Guest column: Adiabatic quantum computing challenges. *ACM SIGACT News* **46**(1), 40–61 (2015)
8. Calude, C.S., et al.: Quassical computing. *International Journal of Unconventional Computing* **14**(1) (2018)
9. Cerezo, M., Arrasmith, A., et al.: Variational quantum algorithms. *Nature Reviews Physics* **3**(9), 625–644 (2021)
10. Chiscop, I., Nauta, J., Veerman, B., Phillipson, F.: A hybrid solution method for the multi-service location set covering problem. In: *Computational Science*. pp. 531–545. Springer (2020)
11. Córcoles, A.D., Kandala, A., et al.: Challenges and opportunities of near-term quantum computing systems. arXiv:1910.02894 (2019)
12. Crielaard, D., De Jong, D., et al.: Quantify-scheduler: An open-source hybrid compiler for operating quantum computers in the NISQ era. *Bulletin of the American Physical Society* (2022)
13. Dahlberg, A., van der Vecht, B., et al.: NetQASM—a low-level instruction set architecture for hybrid quantum-classical programs in a quantum internet. *Quantum Science and Technology* (2022)
14. De Luca, G.: A survey of NISQ era hybrid quantum-classical machine learning research. *Journal of Artificial Intelligence and Technology* **2**(1), 9–15 (2022)
15. Doan, A.D., Sasdelli, M., et al.: A hybrid quantum-classical algorithm for robust fitting. In: *Computer Vision and Pattern Recognition*. pp. 417–427 (2022)
16. Edwards, M.: Towards Practical Hybrid Quantum/Classical Computing. Master’s thesis, University of Waterloo (2020)
17. Ellis, C.A.: Workflow technology. *Computer Supported Cooperative Work, Trends in Software Series* **7**, 29–54 (1999)
18. Endo, S.: Hybrid quantum-classical algorithms and error mitigation. Ph.D. thesis, University of Oxford (2019)
19. Endo, S., Cai, Z., et al.: Hybrid quantum-classical algorithms and quantum error mitigation. *Journal of the Physical Society of Japan* **90**(3), 032001 (Mar 2021)
20. Farhi, E., Goldstone, J., Gutmann, S.: A quantum approximate optimization algorithm. arXiv:1411.4028 (2014)
21. Ge, X., Wu, R.B., Rabitz, H.: The optimization landscape of hybrid quantum-classical algorithms: from quantum control to NISQ applications. arXiv:2201.07448 (2022)
22. Henelius, P., Fishman, R.S.: Hybrid quantum-classical monte carlo study of a molecule-based magnet. *Physical Review B* **78**(21), 214405 (2008)
23. Hevia, J.L., Peterssen, G., Piattini, M.: QuantumPath: A quantum software development platform. *Software: Practice and Experience* **52**(6), 1517–1530 (2022)
24. Hirayama, Y.: Diversity of hybrid quantum systems. In: *Quantum Hybrid Electronics and Materials*, pp. 1–14. Springer (2022)
25. Horsman, C., Munro, W.J.: Hybrid hypercomputing: towards a unification of quantum and classical computation. arXiv:0908.2181 (2009)

26. IBM: IBM's roadmap for scaling quantum technology (2020), <https://research.ibm.com/blog/ibm-quantum-roadmap>
27. Jozsa, R.: An introduction to measurement based quantum computation. arXiv:quant-ph/0508124 (2005)
28. Khalate, P., Wu, X.C., et al.: An LLVM-based C++ compiler toolchain for variational hybrid quantum-classical algorithms and quantum accelerators. arXiv:2202.11142 (2022)
29. Khammassi, N., Ashraf, I., et al.: Openql: A portable quantum programming framework for quantum accelerators. *ACM Journal on Emerging Technologies in Computing Systems (JETC)* **18**(1), 1–24 (2021)
30. Kitaev, A.Y.: Quantum computations: algorithms and error correction. *Russian Mathematical Surveys* **52**(6), 1191–1249 (Dec 1997)
31. Lanzagorta, M., Uhlmann, J.K.: Hybrid quantum-classical computing with applications to computer graphics. In: *ACM SIGGRAPH 2005 Courses*, pp. 2–es. ACM (2005)
32. Lapworth, L.: A hybrid quantum-classical CFD methodology with benchmark HHL solutions. arXiv:2206.00419 (2022)
33. Li, J., Yang, X., Peng, X., Sun, C.P.: Hybrid quantum-classical approach to quantum optimal control. *Physical review letters* **118**(15), 150503 (2017)
34. Lloyd, S.: Hybrid quantum computing. In: *Quantum information with continuous variables*, pp. 37–45. Springer (2003)
35. Mahroo, R., Kargarian, A.: Hybrid quantum-classical unit commitment. In: *Texas Power and Energy Conference (TPEC)*. pp. 1–5. IEEE (2022)
36. McCaskey, A.J., Lyakh, D.I., Dumitrescu, E.F., Powers, S.S., Humble, T.S.: XACC: a system-level software infrastructure for heterogeneous quantum-classical computing. *Quantum Science and Technology* **5**(2), 024002 (2020)
37. Murray, T.: Three truths and the advent of hybrid quantum computing. <https://medium.com/d-wave/three-truths-and-the-advent-of-hybrid-quantum-computing-1941ba46ff8c> (June 2019)
38. Ohno, H.: A quantum algorithm of k-means toward practical use. *Quantum Information Processing* **21**(4), 1–24 (2022)
39. Peruzzo, A., McClean, J., et al.: A variational eigenvalue solver on a photonic quantum processor. *Nature communications* **5**(1), 1–7 (2014)
40. Phillipson, F., Chiscop, I.: A quantum approach for tactical capacity management of distributed electricity generation. In: *Innovations for Community Services*. pp. 323–333. Springer (2022)
41. Possignolo, R.T., Margi, C.B.: A quantum-classical hybrid architecture for security algorithms acceleration. In: *Trust, Security and Privacy in Computing and Communications*. pp. 1032–1037. IEEE (2012)
42. Preskill, J.: Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018)
43. Riesebo, L., Fu, X., et al.: Quantum accelerated computer architectures. In: *Circuits and Systems (ISCAS)*. pp. 1–4. IEEE (2019)
44. Rosmanis, A.: Hybrid quantum-classical search algorithms. arXiv:2202.11443 (2022)
45. Sagingalieva, A., Kurkin, A., et al.: Hyperparameter optimization of hybrid quantum neural networks for car classification. arXiv:2205.04878 (2022)
46. Sakurai, R., et al.: Hybrid quantum-classical algorithm for computing imaginary-time correlation functions. *Physical Review Research* **4**(2), 023219 (2022)
47. Schalkers, M.A., Möller, M.: Learning based hardware-centric quantum circuit generation. In: *Innovations for Community Services*. pp. 308–322. Springer (2022)

48. Schuetz, M.J., Brubaker, J.K., et al.: Optimization of robot trajectory planning with nature-inspired and hybrid quantum algorithms. arXiv:2206.03651 (2022)
49. Schuld, M., Fingerhuth, M., Petruccione, F.: Implementing a distance-based classifier with a quantum interference circuit. *EPL (Europhysics Letters)* **119**(6), 60002 (Sep 2017)
50. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal of Computing* **26**(5), 1484–1509 (1997)
51. Simon, D.R.: On the power of quantum computation. *SIAM Journal on Computing* **26**(5), 1474–1483 (Oct 1997)
52. Stober, S.T., Harwood, S.M., et al.: Considerations for evaluating thermodynamic properties with hybrid quantum-classical computing work flows. *Physical Review A* **105**(1), 012425 (2022)
53. Suchara, M., Alexeev, Y., et al.: Hybrid quantum-classical computing architectures. In: *Post-Moore Era Supercomputing* (2018)
54. Weder, B., Barzen, J., Leymann, F., Salm, M., Vietz, D.: The quantum software lifecycle. In: *Architectures and Paradigms for Engineering Quantum Software*. pp. 2–9 (2020)
55. Weder, B., Barzen, J., Leymann, F., Zimmermann, M.: Hybrid quantum applications need two orchestrations in superposition: A software architecture perspective. In: *Web Services*. pp. 1–13. IEEE (2021)
56. Wezeman, R., Neumann, N., Phillipson, F.: Distance-based classifier on the quantum inspire. *Digitale Welt* **4**(1), 85–91 (2020)
57. Zhang, J.H., Iyengar, S.S.: Graph- $|Q \gg C|$, a graph-based quantum/classical algorithm for efficient electronic structure on hybrid quantum/classical hardware systems: Improved quantum circuit depth performance. *Journal of Chemical Theory and Computation* **18**(5), 2885–2899 (2022)
58. Zylberman, J., Di Molfetta, G., et al.: Hybrid quantum-classical algorithm for hydrodynamics. arXiv:2202.00918 (2022)