EPJ Web of Conferences **295**, 12002 (2024) https://doi.org/10.1051/epjconf/202429512002
CHEP 2023

Co-Design of Quantum Hardware and Algorithms in Nuclear and High Energy Physics

Maja Franz[1,*], *Pía* Zurita[2,**], *Markus* Diefenthaler[3,***], and *Wolfgang* Mauerer[1,4,****]

[1]Technical University of Applied Sciences Regensburg, Regensburg, Germany
[2]Complutense University of Madrid, Madrid, Spain
[3]Thomas Jefferson National Accelerator Facility, Newport News, VA USA
[4]Siemens AG, Corporate Research, Munich, Germany

Abstract. Quantum computing (QC) has emerged as a promising technology, and is believed to have the potential to advance nuclear and high energy physics (NHEP) by harnessing quantum mechanical phenomena to accelerate computations. In this paper, we give a brief overview of the current state of quantum computing by highlighting challenges it poses and opportunities it offers to the NHEP community. Noisy intermediate-scale quantum (NISQ) computers, while limited by imperfections and small scale, may hold promise for near-term quantum advantages when coupled with co-designed quantum algorithms and special-purpose quantum processing units (QPUs). We explore various applications in NHEP, including quantum simulation, event classification, and real-time experiment control, emphasising the potential of variational quantum circuits and related techniques. To identify current interests of the community, we perform an analysis of recent literature in NHEP related to QC.

1 Introduction

Quantum computing (QC) is a promising early-stage technology that offers novel approaches to simulation and analysis in nuclear and high energy physics (NHEP). By basing computations directly on quantum mechanical phenomena, speed-ups and other advantages for many computationally hard tasks are potentially achievable. For instance, as already noted by Feynman in the 20th century, QC opens up the possibility to mimic the dynamical or statistical characteristics of a quantum system in a controlled manner, allowing for the simulation of protons, neutrons or other particles, which is currently inefficient or imprecise using classical computers [1]. Additionally, specific machine learning problems are known to be more efficiently solvable by quantum kernel methods than by using classical approaches [2]. As the theoretical underpinning and the practical realisation of QC are still subject to considerable scientific debate, this raises the question of applicability in NHEP.

In this contribution, we describe the current state of affairs in QC: currently available noisy intermediate-scale quantum (NISQ) computers suffer from a very limited number of quantum bits, and are subject to considerable imperfections, which narrows their practical computational capabilities. To overcome these limitations, we propose that the *co-design* [3–

*maja.franz@othr.de **marzurit@ucm.es ***mdiefent@jlab.org ****wolfgang.mauerer@othr.de

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

6] of quantum algorithms and hardware is one route towards practical utility: by identifying and developing quantum algorithms and designing special-purpose quantum processing units (QPUs) for specific problems, near-term quantum advantages are likely throughout a variety of domains. We aim to progress towards a joint understanding of requirements, limitations and possibilities throughout the communities of physicists, computer scientists and engineers.

To this end, we identify possible classes of applications in NHEP, ranging from quantum process simulation [7] over event classification directly at the quantum level [8] to optimal real-time control of experiments [9]. These types of algorithms are particularly suited for quantum algorithms that involve variational quantum circuits, but might also benefit from special-purpose techniques like (Gaussian) Boson Sampling. We outline challenges and opportunities in the cross-domain cooperation between QC and NHEP, and show routes towards co-designed systems and algorithms. In particular, we target an interdisciplinary exchange of ideas by establishing a joint understanding of requirements, limitations and possibilities.

The remainder of this article is structured as follows: In Section 2, we give an overview of quantum algorithms, which are mentioned and utilised throughout the literature in NHEP. Subsequently, in Section 3, we summarise common quantum hardware designs, of which the properties can be characterised by so-called co-design parameters from the algorithmic point-of-view. The hardware metrics pose a baseline for designing special purpose QPUs for certain algorithms. We will delve into the implications of the co-design parameters for the specific case of the quantum approximate optimisation algorithm (QAOA) in Section 4. Finally, we conclude in Section 5.

2 Quantum Algorithms

The literature describes various efforts to utilise QC for problems in NHEP. This section discusses challenges and opportunities of the primarily mentioned algorithmic paradigms.

"Low Level" Algorithms

A large share of the QC literature employs a *quantum circuit* [10] as a universal computational model for defining quantum algorithms. Inspired by classical computing, quantum circuits consist of sequences of instructions, in QC that is a series of unitary quantum logic gates, to represent the evolution of a quantum state.

It has been shown that certain problems, building on the circuit model, can theoretically be solved more efficiently using quantum algorithms than classical algorithms. Two well-known examples are Grover's algorithm [11] and Shor's algorithm [12], which respectively provide a quadratic speed-up for the exploration of unstructured search spaces and efficiently solve prime factorisation and discrete logarithm problems. However, these algorithms require error-corrected qubits [6], which are not available on current NISQ devices. Additionally, to utilise these algorithms with problem sizes, which are intractable by classical algorithms, requires about 10 to 100 times more qubits than currently available. Therefore, these "low-level" algorithms are only of limited use today.

Unorthodox Approaches

Additionally to the conventional circuit model, there exist alternative methods for utilising quantum mechanical phenomena for computation, as for instance, adiabatic quantum computing (AQC) [13]. It has been shown that AQC is equivalent to the circuit model, which implies that AQC is also a universal method for QC [14]. In this approach, the solution to a computational problem is encoded by the ground state of a Hamiltonian. To determine this

resulting problem Hamiltonian, an initial Hamiltonian with an easily attainable ground state is prepared, which then proceeds to the final Hamiltonian. In an ideal setting, the adiabatic theorem guarantees that the system remains in the ground state and thus yields the solution if the system evolves slowly enough.

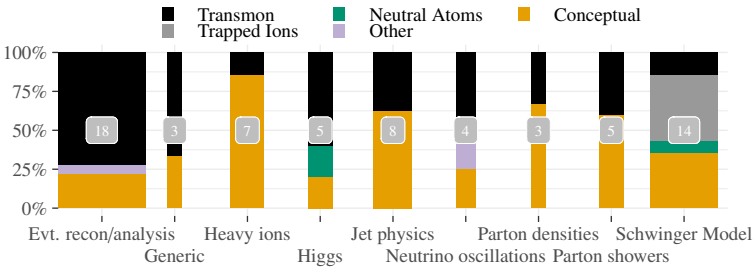

Figure 1. Recent research published at the intersection of NHEP and QC.

Similarly, in quantum annealing (QA), a system also evolves towards the ground state of the problem Hamiltonian. However, it does so without necessarily relying on the adiabatic theorem, but also on stochastic dynamics [15], which makes it easier to perform QA in non-ideal settings, which may be exposed to physical noise or temperatures above zero. Therefore, QA poses a heuristic, which is often proposed as a compromise between ideal universal AQC and realisation in hardware, albeit a drawback is the loss of computational universality in terms of unconditional efficient simulation, which means that unlike AQC or the circuit model, QA can not compute any quantum algorithm, such as Shor's algorithm [12]. Determining the runtime of QA for a given problem is possible based on the physical properties of the underlying Hamiltonian, albeit it is a computationally hard problem itself [13]. Therefore it is difficult to make assumptions on the efficiency of QA for general problems.

QA can be used to solve certain types of problems, which can be formulated as unconstrained binary optimisation problems (QUBOs). Using well-known techniques from computer science, all problems in **NP** (the class of problems that can be efficiently solved (in polynomial runtime) on non-deterministic Turing machines and related models of computation, yet is widely believed to escape efficient solubility on deterministic and probabilistic machines) can be cast in QUBO form [16]–in other words, they can be always solved by finding the optimum solution of a multivariate binary quadratic polynomial instead of executing an explicit algorithm. It is not expected that QA can solve any **NP** problems efficiently (*i.e.*, in polynomial time) [17]. Nevertheless, in practice, QA has the potential to outperform classical approaches, such as simulated annealing [3, 18]. **NP** problems occur in a variety of fields, including NHEP; finding suitable mappings to QUBOs and solving them with the means of QA is therefore of particular interest in current QC research.

Another non-universal model on QC may be defined upon the problem of (Gaussian) Boson Sampling [19]. Boson Sampling [17] is a problem known to be not efficiently solvable on classical computers. Given the variant of Gaussian Boson Sampling (GBS) [19], which is easier to implement experimentally, as it relies on Gaussian states of light instead of photon number (Fock) states of the original proposal, which are harder to generate, first prototypes of Boson Sampling can be implemented on photonic hardware [20]. While it has been shown that GBS does bring computational advantage [21], it is yet unclear which problems could benefit from this special-purpose technique. While reductions of specific graph problems to Boson sampling are known [22], their practical utility remains unclear.

Quantum Simulation

A promising research direction in NHEP is concerned with the simulation of physical systems using QC. In general, quantum simulators are controllable quantum systems, which are used to mimic the properties of another, typically less controllable, quantum system [1].

Quantum simulation (QS) can be roughly classified into digital QS, analogue QS, or a more recent combination of the two, known as digital-analogy QS [1]. In the purely digital approach, the evolution associated with a Hamiltonian is decomposed into single- and two-qubit unitary gates. Although this approach is universal through approximating any quantum operation to arbitrary accuracy, it is generally assumed that this is only true for error-corrected qubits, which are hard to manufacture in the current NISQ era. The analogue approach aims to closely mimic the characteristics of the simulated quantum model for the whole continuous evolution by defining Hamiltonians similar to the simulated system. Analogue QS allows for simulations with a better scalability than digital QS, but they are limited to being realised by models that can be constructed in the laboratory [23]. In the digital-analogue QS, analogue blocks provide a scalable structure by reducing the number of gates and hence the experimental error, while the inclusion of digital steps amplifies the variety of possible operations. Digital-analogue QS may be the best suited in the non-error corrected regime [1].

NISQ Algorithms

Most algorithms outlined above necessitate more and less flawed qubits than currently available on NISQ devices. To avoid these limitations, the class of variational hybrid quantum-classical algorithms allows a certain degree of control over the number of qubits and the circuit depth (*i.e.*, the number of sequential gates), and they are therefore assumed to be particularly suited for NISQ devices. Similar to a classical neural network [24], VQCs have been proven to be universal approximators [25], that is, they are capable of reproducing any continuous function, given a sufficient number of parameters. This characteristic makes the VQC a promising choice (and a drop-in replacement) for a variety of optimisation and machine learning algorithms, for which function approximators are a fundamental concept.

While there are several indications of quantum advantage in variational algorithms [2, 26, 27], further research on implementing these algorithms on certain types of NISQ hardware is required to enable practical applications in the near-term.

3 Quantum Hardware Designs

Various vendors are currently researching and developing multiple technologies to implement QPUs, including, for example, trapped ions [28], neutral atoms [29, 30], superconducting transmons [31], or photonic systems [32]. As shown in Figure 1, a large share of the NHEP literature focuses on conceptual quantum algorithms and does not consider the influence of hardware implementations. Nevertheless, many experiments in NHEP also use superconducting transmon systems, which are currently the most prevalent hardware implementation due to remarkable recent advancements from major players such as IBM [31]. Interest in alternative technologies seems to be in its infancy. However, with the efforts of various other vendors (*e.g.* [28, 29, 32]), first commercially available trapped-ion, neutral atom or photonic systems become accessible for QC research.

Every physical realisation of QC is subject to noise and imperfections that deviate from the theoretically desired perfect system. Therefore, it is uncertain which fundamental physical concepts will underpin future quantum computers. Furthermore, different hardware implementations exhibit a variety of characteristics that cannot be easily translated into established measures of quality, performance, or scalability for quantum algorithms. Hence, in

	Transmons	Trapped Ions	Neutral Atoms
#Qubits	50-127	11-25	~100
Single qubit gate time	ns	μs	μs
Two qubit gate time	ns	μs	~ns
T1 time	$> 100\ \mu$s	10-100 s	~s
T2 time	~100 μs	0.2-1 s	~ms - s
Single qubit gate error	~0.1 %	0.01-1 %	0.1-0.4 %
Two qubit gate error	0.1-2.5 %	0.04-2.7 %	0.5-4.5 %
Temperature requirements	~0 K	Room temperature[2]	Room temperature[2]
Qubit coupling density	< 10 %	100 %	10-20 %

Table 1. Key characteristics of three different hardware approaches to quantum computing.

order to determine a suitable type of hardware, an analysis of hardware metrics is required for each algorithm and each application, individually. These metrics can be summarised as *co-design parameters* that need to be considered when developing quantum algorithms [6]. Table 1 lists a selection of co-design parameters of three recent hardware implementations[1].

Hardware-software quantum co-design entails determining co-design parameters, which are both, suitable for a particular algorithm, but also technically feasible. The design of dedicated QPUs based on the baseline hardware implementation is therefore highly interdisciplinary, involving physicists who specify applications in NHEP, computer scientists, who identify and adapt the corresponding quantum algorithm to a hardware platform, and engineers, who design and build the QPU, tailored to the needs of the algorithm.

As QC has great potential for simulation in NHEP [34], among the co-design parameters, the number of qubits is particularly important for the simulation of large quantum systems. Another promising direction could be the analysis of events [35] or the real-time control of experiments [9] by quantum machine learning or optimisation techniques. Since both, quantum machine learning and quantum optimisation, can be based on completely different algorithmic paradigms, for instance QA or VQCs, it is difficult to specify general "good" co-design parameters for the numerous algorithmic variants available. Therefore, in the next section, we will investigate appropriate hardware metrics for one specific quantum optimisation algorithm, as an example for future special-purpose hardware-software co-design.

4 Example: co-design for QAOA

The quantum approximate optimisation algorithm (QAOA) [36] is a variational hybrid quantum algorithm for solving combinatorial optimisation problems on NISQ hardware. It identifies the global minimum of a cost function over a set of discrete variables.

4.1 Quantum Approximate optimisation Algorithm (QAOA)

In QAOA, a quantum circuit is executed, which consists of $p \in \mathbb{N}$ layers of unitary operators. The unitary operators are determined by a set of $2p$ parameters $\vec{\beta}, \vec{\gamma} \in \mathbb{R}^p$. Similar to quantum annealing, with QAOA the minimum of an objective function, in QUBO form can be identified. The extent of speedups that QAOA can achieve when compared to classical approaches

[1]For transmon systems, we considered quantum devices from IBM, where we obtained the metrics from the quantum framework Qiskit [33], which provides snapshots of IBM quantum systems [31]. The values for trapped-ion systems were extracted for quantum devices from IonQ [28] and for neutral atom systems from Ref. [30].

[2]Conditions apply; see Ref. [30] for the fine print.

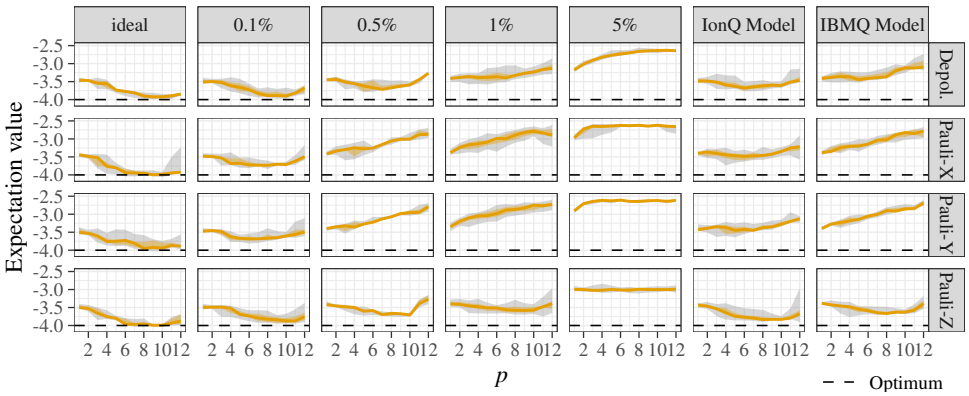

Figure 2. Expectation values for QAOA with different number of layers (p), noise types (rows) and gate error rates (columns). The orange line represents the average of ten runs, the light grey areas correspond to the minima and maxima, the light orange areas correspond to the first and third quartiles.

is not yet fully understood. Nevertheless, a classical algorithm that can efficiently sample the output distribution of QAOA, even for p = 1, is impossible given reasonable complexity-theoretic assumptions [26]. This represents a possible indication of quantum advantage, yet additional theoretical or experimental progress is necessary for practical applications.

The quantum operations are embedded into a classical optimisation loop; after applying the unitary operators to an initial state, the expectation value of the circuit, which represents the problem Hamiltonian and encodes the objective function, is measured. Using a classical optimiser, circuit parameters are changed to minimise the expectation value [4].

4.2 Influence of Noise on optimisation

The optimisation process can be influenced by noise arising from various sources in NISQ devices, such as decoherence (a loss of quantum information over time) or imprecise calculations and measurements (see also Table 1). Here, we illustrate the effect of gate errors on QAOA, and use bit/phase-flip and depolarising errors as described in detail in Ref. [10]. A bit-flip error can be modelled by applying a Pauli-X before a payload operation with a certain probability that specifies the error rate. Likewise, bit-phase-flip errors and phase-flip errors can be created by substituting the Pauli-X gate with the Pauli-Y or Pauli-Z gate, respectively, in the approach. The depolarising procedure randomly applies Pauli-X, Y, or Z operators with a given error rate.

Inspired by the gate error rates of current NISQ devices (see Table 1), we ran QAOA with multiple one- and two-qubit error rates from 0% (ideal case) to 5%. We also constructed noise models based on error rates of two commercially available QPUs, the *IonQ Forte* [28], a QC based on trapped ions with a single-qubit error rate of 0.2% and a two-qubit error rate of 0.4%, and the *IBMQ Kolkata* [31], a system based on superconducting transmons with a single-qubit error rate of 0.032% and a two-qubit error rate of 1.091%[2].

We solve a small instance of *MaxCut*, a seminal problem in **NP**, using QAOA on a simulator. To determine the influence of imperfections, we performed ten runs with different seeds for each type of noise and error rate on a *Quantum Learning Machine*.

The results are shown in Figure 2. In the ideal case, without gate errors, the expectation value converges to the optimum as the number of layers, p, increases. In the noisy cases,

[2]Error rates are taken from Qiskit [33] for *IBMQ Kolkata* and *IonQ Forte*.

as p increases, more gates and thus more sources of gate errors are introduced into the system. Therefore, the expectation values even diverge from the optimum as the gate error rate increases. This demonstrated that gate errors are particularly detrimental to QAOA performance, albeit acceptable results can be achieved at low error rates (0.1%), or in the presence of errors limited to Pauli-Z.

Therefore, a co-designed special-purpose QPU designed for this particular case of QAOA should be tailored to include error correction codes to avoid Pauli-X and Pauli-Y errors. This may also be a relevant criterion for the choice of an underlying hardware implementation. For instance, trapped-ion quantum systems are particularly resilient to Pauli-X errors [37, 38] and are therefore a potentially suitable candidate. Additionally, the introduction of gates that are not critical to the computation should be avoided. One aspect of avoiding unnecessary gates is to reduce the amount of swap gates required to create two-qubit operations between physical qubits that do not share a connection. As shown in Refs. [3, 5], even a small increase in the qubit coupling density can drastically reduce the number of swap gates required.

4.3 Latency and Integration

In addition to noise, several other factors affect QAOA performance. As discussed in detail in Ref. [4], the total execution time for a hybrid algorithm, such as QAOA, comprises several contributions, for instance, circuit execution time, the classical parameter optimisation time, the number of samples required to obtain accurate statistics, or the communication time between the classical computer and the QPU involved. Wintersperger et al. [4] compared three different QPU deployment scenarios: (1) having the QPU in the cloud, which is a popular access model for QC vendors, (2) connecting the QPU to the classical CPU via a local bus, and (3) an on-chip QPU-CPU integration. Although the on-chip scenario is associated with the shortest communication time between QPU and CPU, the total execution time improvements over the local bus scenario can be neglected. However, the local bus scenario could achieve an approximate execution time reduction of 24.3% over the cloud scenario. Integrating further co-design methods, such as reducing the overall circuit execution time through decreasing the number of swap-gates, could further reduce the execution time [4].

Consequently, co-designing a QPU for QAOA and other NISQ-era algorithms is non-trivial and subject to multi-fold factors, yet may pave a way towards practical utility.

5 Conclusion

The potential of QC in the field of NHEP is both, promising and challenging. Physical simulation, analysis and optimisation in large-scale systems or real-time control of experiments hold great potential, yet imperfections in NISQ eradicate any computational advantages. We propose that a potential solution lies in hardware-software co-design, while interdisciplinary collaboration is essential to deepen the understanding of optimisation goals and criteria.

References

[1] L. Lamata et al., Advances in Physics: X **3**, 1457981 (2018)
[2] Y. Liu, S. Arunachalam, K. Temme, Nature Physics **17**, 1013 (2021)
[3] M. Schönberger, S. Scherzinger, W. Mauerer, *Ready to Leap (by Co-Design)? Join Order Optimisation on Quantum Hardware*, in *Proc. ACM SIGMOD/PODS* (2023)
[4] K. Wintersperger, H. Safi, W. Mauerer, *QPU-System Co-Design for Quantum HPC Accelerators*, in *Proc. 35th Int. Conf. on the Architecture of Computing Systems* (Gesellschaft für Informatik, 2022), pp. 100–114, ISBN 978-3-031-21867-5

[5] H. Safi, K. Wintersperger, W. Mauerer, *Influence of HW-SW-Co-Design on Quantum Computing Scalability*, in *Proceedings of the IEEE Quantum Software Week* (2023)
[6] F. Greiwe, T. Krüger, W. Mauerer, *Effects of Imperfections on Quantum Algorithms: A Software Engineering Perspective*, in *Proc. IEEE Quantum Software Week* (2023)
[7] W.A. de Jong et al., Phys. Rev. D **104**, L051501 (2021)
[8] K. Terashi, M. Kaneda, T. Kishimoto, M. Saito, R. Sawada, J. Tanaka, Computing and Software for Big Science **5**, 1 (2021)
[9] M. Schenk et al. (2022), arXiv:2209.11044
[10] M.A. Nielsen, I.L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2010), ISBN 978-1-107-00217-3
[11] L.K. Grover, *A fast quantum mechanical algorithm for database search*, in *Proc. 20th annual ACM symposium on Theory of computing* (1996), pp. 212–219
[12] P.W. Shor, SIAM Review **41**, 303 (1999)
[13] T. Albash, D.A. Lidar, Rev. Mod. Phys. **90**, 015002 (2018)
[14] D. Aharonov et al., SIAM Review **50**, 755 (2008)
[15] W. Vinci, D.A. Lidar, npj Quantum Information **3**, 38 (2017)
[16] A. Lucas, Frontiers in Physics **2** (2014)
[17] S. Aaronson, A. Arkhipov, Theory of Computing **9**, 143 (2013)
[18] T. Krüger, W. Mauerer, *QA-Based Software Components: An Experimental Case Study with SAT Solving*, in *Proc. IEEE/ACM 42nd ICSEW* (2020), pp. 445–450
[19] C.S. Hamilton et al., Phys. Rev. Lett. **119**, 170501 (2017)
[20] A. Deshpande et al., Science Advances **8**, eabi7894 (2022)
[21] L.S. Madsen et al., Nature **606**, 75 (2022)
[22] J.M. Arrazola, T.R. Bromley, Phys. Rev. Lett. **121**, 030503 (2018)
[23] A.J. Daley et al., Nature **607**, 667 (2022)
[24] K. Hornik, M. Stinchcombe, H. White, Neural Networks **2**, 359 (1989)
[25] A. Pérez-Salinas et al., Quantum **4**, 226 (2020)
[26] E. Farhi, A.W. Harrow (2019), arXiv:1602.07674
[27] M. Franz et al., *Uncovering instabilities in variational-quantum deep q-networks* (2022), Journal of the Franklin Institute
[28] IonQ, *IonQ* (2023), https://ionq.com
[29] PASQAL, *PASQAL* (2023), https://www.pasqal.com
[30] K. Wintersperger et al., EPJ Quantum Technology **10**, 32 (2023)
[31] IBM, *IBM Quantum* (2023), https://quantum-computing.ibm.com
[32] Xanadu, *Xanadu* (2023), https://www.xanadu.ai
[33] Qiskit contributors, *Qiskit: An open-source framework for quantum computing* (2023)
[34] T.S. Humble et al. (2022), arXiv:2203.07091
[35] T. Schwägerl et al. (2023), arXiv:2303.13249
[36] E. Farhi, J. Goldstone, S. Gutmann (2014), arXiv:1411.4028
[37] S. Ebadi et al., Nature **595**, 227 (2021)
[38] K.N. Schymik et al., Phys. Rev. A **102**, 063107 (2020)

Acknowledgements MF and WM acknowledge support by the German Federal Ministry of Education and Research (BMBF), funding program quantum technologiesfrom basic research to market, grant numbers 13NI6092 and 13N15647, and from the High-Tech Agenda Bavaria.