

Quantum Associative Memory in HEP Track Pattern Recognition

Illya Shapoval^{1,*} and *Paolo Calafiura*^{1,**}

¹Lawrence Berkeley National Laboratory

Abstract. We have entered the Noisy Intermediate-Scale Quantum Era. A plethora of quantum processor prototypes allow evaluation of potential of the Quantum Computing paradigm in applications to pressing computational problems of the future. Growing data input rates and detector resolution foreseen in High-Energy LHC (2030s) experiments expose the often high time and/or space complexity of classical algorithms. Quantum algorithms can potentially become the lower-complexity alternatives in such cases. In this work we discuss the potential of Quantum Associative Memory (QuAM) in the context of LHC data triggering. We examine the practical limits of storage capacity, as well as store and recall errorless efficiency, from the viewpoints of the state-of-the-art IBM quantum processors and LHC real-time charged track pattern recognition requirements. We present a software prototype implementation of the QuAM protocols and analyze the topological limitations for porting the simplest QuAM instances to the public IBM 5Q and 14Q cloud-based superconducting chips.

1 Introduction

High Energy Physics (HEP) is a prime example of data intensive science. Over the next decade rapid evolution of accelerator technologies and particle detectors will increase by one order of magnitude the amount and the complexity of data coming from facilities such as Large Hadron Collider (LHC), creating new challenges for the online event selection (Trigger) systems in HEP experiments. To cope with increasing data input rates, sophisticated event selection techniques are being employed at both hardware and software levels. However, current approaches in particular to charged particle pattern recognition, scale poorly with data complexity. Under reasonable technology and cost evolution models, the physics output of the next generation of HEP experiments will be limited by their pattern recognition strategy.

The data input rates foreseen in High-Luminosity LHC (HL-LHC) and beyond impose new challenging requirements on the charged trigger systems. Data rates in HEP experiments (at the LHC and elsewhere) will continue to increase. As the corresponding algorithmic complexity of many crucial HEP data processing problems is often polynomial or worse, it is of a substantial interest to investigate alternative, non-classical approaches and algorithms capable of more efficient and scalable track recognition. To cope with the new challenges, LHC experiments have launched a series of trigger upgrade projects. For example the ATLAS

*e-mail: ishapoval@lbl.gov

**e-mail: pcalafiura@lbl.gov

experiment at CERN LHC introduced a new system of electronics, Fast Tracker (FTK) [1]. The system is aimed at real time track reconstruction at a 100 kHz Level-1 trigger rate. To meet the time budget requirements, FTK employed Associative Memory (AM) [2, 3]. The latter allowed to address the problem of track pattern recognition – the most computationally hard part of track finding – in a massively parallel way. The approach is based on nearly simultaneous and constant-time comparison of coarse-grained hits being readout from the tracker stations to those of the MC generated track patterns pre-loaded into AM. In Run 2 and Run 3, the AM pattern bank will have to store $\sim 10^9$ track patterns of 8-integer length. The bank pattern of this size requires $8 \cdot 10^3$ AM chips (AMchip06), $\sim 32kW$ of supporting power and associated cooling. It is foreseen that 8-16 times more patterns will be required in HL-LHC (2026). There are two evolutionary, linear solutions to this problem, 1) scale-up the number of AM chips of current generation, considered as cost and power inefficient; 2) more plausibly, upgrade the AMchip06 design to increase its storage capacity.

In this paper, we consider Quantum Associative Memory (QuAM) [4–8] – a quantum variant of AM based on quantum storage medium and two quantum algorithms for information storage and recall. We compare theoretical QuAM errorless performance expectations to the requirements of the current ATLAS track pattern recognition problem. We also present a software prototype for QuAM circuit generators and point out the limitations for porting QuAM to the state-of-the-art IBM quantum processor units (QPUs).

2 Quantum storage

2.1 Assembling quantum memory

Let $\xi \subseteq \{0, 1\}^n$ represents a set of N reference binary patterns of length n . QuAM is based on establishing an injection $\xi \mapsto |\xi\rangle$, where $|\xi\rangle$ denotes an orthonormal basis of the Hilbert space of a quantum mechanical system composed of n 2-level qubits. Memorizing ξ can then be done by assembling a quantum superposition:

$$|\Xi\rangle = \sum_{i=1}^N \alpha_i |\xi^i\rangle, \quad \alpha_i \in \mathbb{C} \wedge N \leq 2^n \wedge \sum_{i=1}^N |\alpha_i|^2 = 1$$

Note that the special case of $\xi = \{0, 1\}^n$ and $N = 2^n$ is trivial and describes complete quantum memory. The only practical value of this case and, more generally, of the case of N approaching 2^n , is in their setup simplicity, which can be useful for the purposes of verification and benchmarking on the Noisy Intermediate-Scale Quantum Era (NISQ) era QPUs.

Ventura and Martinez [4–6], and Trugenberger [8], proposed two alternative ways for unitary assembling of the equal-weighted partial superposition. Simple analysis revealed that the quantum storage algorithm outlined by Trugenberger features shallower circuit and lower topological complexity (see Section 4 for more details), though at the cost of one extra qubit required for auxiliary operations. From the standpoint of limitations of the state-of-the-art QPUs, these properties make this algorithm preferable for implementation.

The Trugenberger’s quantum algorithm for storing requires three memory registers spanning $2(n+1)$ qubits to operate [8]: n qubits - for temporary pattern storage, n - for permanent pattern storage and 2 qubits for storage and recall operations control. Figure 1 outlines the core iteration for storing a 2-bit pattern in this approach.

A complete quantum circuit for encoding a pattern set must have the iterations interleaved with additional quantum gates for read-in of classical bits into the temporary register. An example of such a complete circuit, along with key implementation details, will be shown in Section 5.

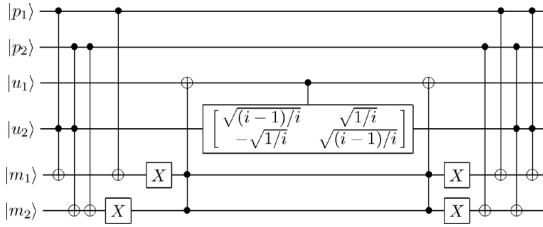


Figure 1. The quantum circuit implementing an iteration encoding a 2-bit pattern in the Trugenberger’s approach. Qubits p are used as temporary storage register, qubits u - as control register and m - as memory register. The controlled gate, acting on the u -register and parameterized with a pattern index $i \in \{1, \dots, N\}$, spawns a new term in the quantum memory superposition for the pattern being stored.

2.2 Exponential quantum capacity

The cardinality of an orthonormal basis of the Hilbert space admits, in computational complexity sense, *optimal* pattern storage capacity for patterns of bit-length equal to the number of qubits in the system. Equivalently, a quantum storage medium provides exponential scaling of its pattern capacity as a function of pattern length. Fair comparison of quantum and classical memory capacities requires accounting for auxiliary qubits that are necessary for quantum operations. However, the asymptotic effect of this additional requirement in storage algorithms of [4–6] and [8] is bound by a constant (in Section 4 we elaborate on other practical consequences of this). Figure 2 compares capacity scaling of classical and quantum associative memories, with the latter being considered in the Trugenberger’s approach.

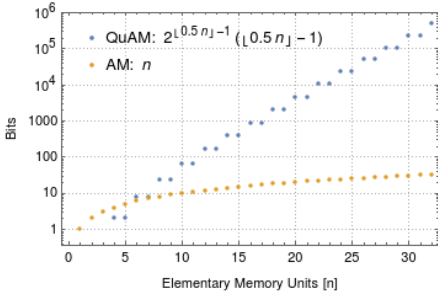


Figure 2. Capacity of quantum versus classical associative memories as a function of the number of elementary physical storage units (conventional memory cells in the classical case, and 2-level qubits in the quantum one).

In the case of the ATLAS fast track pattern recognition, the binary pattern length is defined as a sum over the Inner Tracker logical layers of interest w_i , where w is length of binary representation of a hit identifier within each layer. In LHC Run 2 and Run 3, 8 logical layers are involved in the AM-based pattern recognition. Table 1 summarizes available QuAM capacity for various granularity of the track hit resolution.

Table 1: QuAM requirements and capacity as a function of width of a detector hit identifier.

Length of detector hit identifier (bits)	8	16	32
Length of binary track pattern (bits)	64	128	256
QuAM register (qubits)	130	258	514
QuAM capacity (patterns)	$\sim 10^{19}$	$\sim 10^{38}$	$\sim 10^{77}$

Qutrits (quantum trits) may help reduce the minimum requirement for the number of physical quantum units necessary for a particular pattern length. But existent hardware implementations are less developed due to more challenging qutrit control. The use of random access encodings [9] could further reduce the requirement, though involving a tradeoff on query efficiency. We left both ideas out of the scope of this study. It is important to note,

however, that any quantum information compression technique is limited by the Holevo’s bound [10] upon accessing the information.

3 Quantum recall

Leveraging the quantum advantage of exponential memory capacity requires scalable and efficient algorithms for memory querying. Two quantum algorithms are discussed in the literature. The first, used by the original proponents of QuAM - Ventura and Martinez [4–6], is a generalization of the classic Grover’s algorithm [11–13]. The algorithm offers quadratic speedup in searching an element in an unordered dataset as compared to the best known classical counterparts, and is proven to be optimal in computational complexity sense [14]. The second memory querying algorithm [8], relies on the technique of post-selection of the measurement result, and allows to avoid the measurement-induced collapse of memory upon a query. The latter comes at the cost of getting only a binary response. Without a measurement of all pattern bits, a binary response does not provide important features of associative memory such as recall of incomplete and noisy patterns. In addition, the post-selection technique offers no quantum speedup. At the same time, as the algorithm speed is of uttermost importance when working with extremely large memories, we consider the asymptotic speedup - the cornerstone of quantum computing - the guiding feature that makes the Grover’s algorithm our primary choice for this study.

Application of the Grover’s algorithm in the QuAM context requires the use of its generalized variant for handling arbitrary amplitude distributions in the initial memory state [15, 16]. The circuit for such algorithm is outlined in Figure 3.

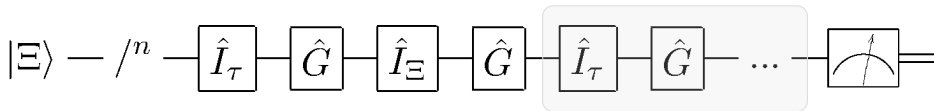


Figure 3: The quantum circuit implementing a variant of the Grover’s algorithm generalized for arbitrary (including partial) initial superposition. Only memory register (n qubits) is employed. \hat{I}_τ is the quantum oracle operator, which inverts the phase of the target state τ . Likewise, \hat{I}_Ξ - inverts phases of all terms originally present in memory. It plays the key role in mitigation the destructive interference of the *ghost* states spawned by the first Grover’s diffusion operator. \hat{G} - Grover’s diffusion operator, inverting all amplitudes about their average. \hat{I}_τ and \hat{G} comprise one Grover iteration. The boxed region denotes the Grover’s cycle containing $T_j - 2$ (introduced later in formula 1) Grover’s iterations.

In what follows, we revisit some of the theoretical aspects of the algorithm to estimate its *theoretical* efficiency in the context of ATLAS FTK pattern recognition requirements.

Let, for a given query, the quantum superposition is split as

$$|\Xi\rangle = \sum_{i=1}^m k_i(t)|\xi^i\rangle + \sum_{i=m+1}^N l_i(t)|\xi^i\rangle$$

with $k_i(t), l_i(t)$ denoting the amplitudes of matching and non-matching the pattern of interest, m – the number of matched states, and N – the total number of patterns stored in memory. An exact solution to difference equations describing the evolution of amplitudes with arbitrary initial conditions is known [15, 16]. Thus, assuming without loss of generality that $1 \leq m \leq \frac{N}{2}$, the amplitudes of the matching states evolve as

$$k_i(t) = \sum_{t_j=0}^{t-1} C(t_j) + k_i(0), \quad C(t) \equiv \frac{2}{N} \left[(N-m) \left(\bar{l}(0) \cos wt - \bar{k}(0) \sqrt{\frac{m}{N-m}} \sin wt \right) - m \left(\bar{k}(0) \cos wt + \bar{l}(0) \sqrt{\frac{N-m}{m}} \sin wt \right) \right]$$

where \bar{k}, \bar{l} are average amplitudes of matching and non-matching states, and $\cos w = 1 - \frac{2m}{N}$.

Thus, the probability $P(t) = \sum_{i=1}^m |k_i(t)|^2$ to measure a marked state (i.e., a solution) peaks at

$$T_j = NI \left(\frac{(j + 1/2)\pi - \arctan\left(\frac{\bar{k}(0)}{\bar{l}(0)} \sqrt{\frac{m}{N-m}}\right)}{\arccos\left(1 - \frac{2m}{N}\right)} \right), \quad j = 0, 1, 2, \dots \quad (1)$$

Grover’s iterations, with the nearest integer function NI defined using the rounding down rule for half-integers.

The upper bound $P_{\max} \geq P(t)$ for probability of measuring a solution is

$$P_{\max} = 1 - \sum_{i=m+1}^N |l_i(t) - \bar{l}(t)|^2 \equiv 1 - (N-m)\sigma_l^2 \quad (2)$$

where σ_l^2 is variance of non-matching amplitudes, which is a constant of motion [15, 16]. The upper bound (2) can only be reached for integer arguments of the NI function in (1) and equals 1 only in the special case of uniform initial distribution, which can never occur in practical applications of QuAM. However, for $m \ll N$, the theoretical upper bound for measuring a matching state approaches certainty.

As a demonstration, let us consider some of the pertinent properties of the classic Grover’s search in the context of the FTK requirements. The evolution of probability as the Grover’s algorithm prepares a quantum system for a measurement is visualized in Figure 4 (left) for the case of uniform initial superposition of 10^9 basis states.

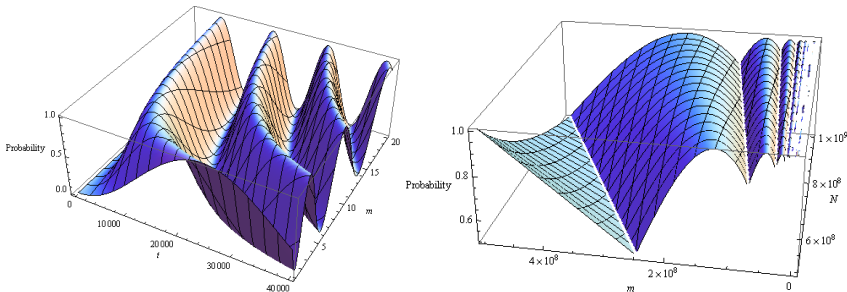


Figure 4: Theoretical estimate of (left) probability of measuring a solution as a function of the number of Grover’s iterations and matching patterns (for $N = 10^9$) and (right) the peak probability for measuring a solution as a function of the number of matching and total number of patterns stored in QuAM. Both estimates assume the special case of uniform initial superposition and errorless quantum dynamics.

For example, a query matching one pattern requires 24836 Grover’s iterations to reach the peak measurement probability of 0.999999999996. Note that the probability ramp-up is slowing down when approaching the peak. This gives an option of cutting the number of iterations down to acceptable value of the outcome probability. Another observation is that, for a given capacity, the number of iterations necessary to reach the peak probability decreases monotonically as the number of solutions increases. For example, a query matching 20 patterns requires 5553 iterations to peak at 0.9999999991 probability of measuring one of the solutions. This suggests, where applicable, another dimension for minimization of the number of Grover iterations based on wildcarded pattern matching. Note that the latter optimization, as seen in Figure 4 (right), affects the maximum achievable probability. Significance of this effect, however, is limited to the region of extremely high number of solutions.

4 Topological constraints

Limited connectivity between qubits on most state-of-the-art QPUs often constitutes the main impediment to the mapping of complex quantum algorithms onto them. This can manifest, for example, in non-efficient transpilation of 2-qubit gates leading to higher error accumulation, or in a complete topological mismatch between algorithmic and processor connectivity graphs making the mapping impossible. Connectivity problems can be addressed on the hardware side - with the advancement of the principles of operation and architectures of QPUs toward higher connectivity. They can also be mitigated by optimizing algorithms for lower connectivity requirements. Scalability of such hardware and algorithmic solutions is of uttermost importance, as quantum computing advantages are asymptotic.

In this light, it is interesting to analyze the topological complexity of the algorithms used in QuAM. It turns out that the storage algorithm suggested by Trugenberger [8] features weaker topological requirements as compared to the original one proposed by Ventura nad Martinez [4]. The topological requirements for Trugenberger’s storage are a superset of the ones for Grover’s recall. Thus, we can summarize the integral topological requirements in Figure 5, where we outlined the special cases of 2- and 3-bit patterns, as well as the general case of n -bit patterns. Importantly, the topological complexity of the algorithms does not depend on the number of patterns being stored, but rather on the length of a pattern.

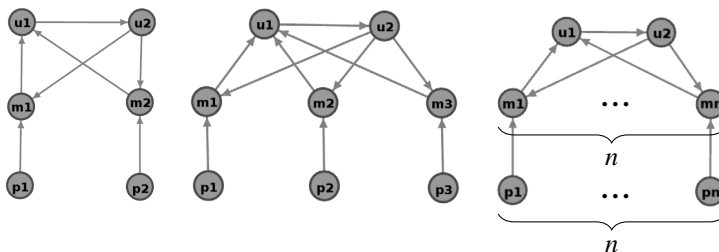


Figure 5: Integral (storage and recall) topological requirements for (from left to right) 2-bit, 3-bit and n -bit patterns. Thus, the connectivity of p - and m -register qubits is constant (1 and 3, correspondingly) as a function of pattern length, while is linear ($n + 1$) for u -register qubits.

The public IBM Q Experience QPUs we have looked at in this study include, at the time the paper is written, the 5-qubit IBM Q 5 Yorktown/Tenerife and the 14-qubit IBM Q 14 Melbourne [17] devices. By the number of qubits, only the latter could allow to run the simplest QuAM circuits (patterns of up to 5-bit length). Unfortunately, it does not satisfy the topological requirements even for the most trivial case of a 2-bit pattern. In contrast, the latter

case should be topologically compliant with the 20-qubit IBM Q 20 Tokyo available for IBM Q Network clients and will be tested in the near future.

5 Implementation on IBM Q Experience

We have implemented a software prototype that includes the QuAM storage and recall circuit generators. The prototype is based on the Trugenberger’s algorithm for storage and the generalized Grover’s algorithm for recall outlined in previous sections and is developed in the QISKit framework [18] of the IBM Q Experience Project [19].

It turns out that most gates employed by both algorithms are either directly implemented, or, like the Toffoli gate, have known exact decompositions over the elementary gate set implemented on IBM Qs. The only minor exception to this is the gate that spawns a superposing term for each new pattern (see the core gate of the storage circuit in Figure 1). The corollary of the Z-Y theorem for decomposition of a controlled- U operation as $U = e^{i\alpha}(AB)^{-1}\sigma_x B\sigma_x A$ [20] with a choice of $A = U_3(\pi, \theta, 0)$ and $B = H$, as well as a substitution of the pattern index $i = \sin^{-2} \frac{\theta}{2}$, make its decomposition straightforward.

Figure 6 shows an example of the end-to-end 2-bit quantum circuits produced by our prototype.

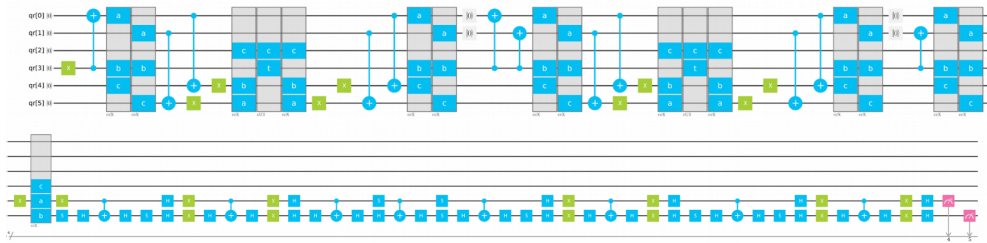


Figure 6: The end-to-end QuAM circuits produced by the Qiskit-based circuit generator prototype for (top): saving three 2-bit patterns and (bottom): retrieving one 2-bit pattern.

6 Conclusion

Leveraging the power of the quantum computing paradigm in HEP, and elsewhere, is in its infancy. The objective of this study was to initiate a discussion within the HEP community about the feasibility of applications of QuAM for charged track pattern recognition to the next generation HEP experiments.

In this work, we analyzed the topological limitations of the two QuAM initialization variants and pointed out that, with limited QPU connectivity, implementation of the Trugenberger’s algorithm is more feasible on the state-of-the-art IBM QPUs. We evaluated some of the pertinent properties of the generalized Grover’s search, extended by Ventura and Martinez, in the context of current and future HEP data processing requirements. We have also prototyped the Trugenberger’s initialization and the Grover’s algorithm generalized for arbitrary (including partial) initial superpositions in the Qiskit framework, yielding recall probabilities that matched the theoretical estimates up to the machine epsilons. The prototype will allow us to run the simplest instances of QuAM on the IBM Q 20 chip, as well as to simulate the instance of 15-bit pattern QuAM on IBM Q 32 QASM Simulator.

Many important questions, that can ultimately affect the viability of QuAM, are beyond the scope of this paper and will be addressed in follow-up studies. Our next steps will be to scale-up the simulations of QuAM to higher-order patterns and to evaluate its timing and

efficiency on state-of-the-art QPUs. We will try to address some QuAM limitations, including linear memory initialization, efficiency of probabilistic state cloning in the Grover-based setting, as well as implement the alternative querying mechanism based on the post-selection of the measurement result.

Acknowledgement

This work was funded by DOE HEP Center for Computational Excellency. Our prototype was tested on the IBM Q Experience cloud service.

References

- [1] M. Shochet, L. Tompkins, V. Cavaliere, G. P. A. Annovi, G. Volpi, Tech. Rep. CERN-LHCC-2013-007. ATLAS-TDR-021, CERN-ATLAS (2013), ATLAS Fast Tracker Technical Design Report, <https://cds.cern.ch/record/1552953>
- [2] S. Amendolia, S. Galeotti, F. Morsani, D. Passuello, L. Ristori, N. Turini, Nuclear Instruments and Methods in Physics Research Section A: Accelerators Spectrometers, Detectors and Associated Equipment **315**, 446 (1992)
- [3] S. Amendolia, S. Galeotti, F. Morsani, D. Passuello, L. Ristori, G. Sciacca, N. Turini, IEEE Transactions on Nuclear Science **39**, 795 (1992)
- [4] D. Ventura, T. Martinez, *Quantum associative memory with exponential capacity*, in *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36227)* (IEEE, 1998)
- [5] D. Ventura, T. Martinez, in *Artificial Neural Nets and Genetic Algorithms* (Springer Vienna, 1999), pp. 22–27, https://doi.org/10.1007/978-3-7091-6384-9_5
- [6] D. Ventura, Information Sciences **124**, 273 (2000)
- [7] A. Ezhov, A. Nifanova, D. Ventura, Information Sciences **128**, 271 (2000)
- [8] C.A. Trugenberger, Phys. Rev. Lett. **87**, 067901 (2001)
- [9] A. Ambainis, A. Nayak, A. Ta-Shma, U. Vazirani, *Dense Quantum Coding and a Lower Bound for 1-way Quantum Automata*, in *Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing* (ACM, 1999), STOC '99, pp. 376–383
- [10] A. Holevo, Problems Inform. Transmission **9**, 177 (1973)
- [11] L.K. Grover, *A Fast Quantum Mechanical Algorithm for Database Search*, in *Proceedings of the 28th ACM STOC* (1996), pp. 212–219, ISBN 0-89791-785-5
- [12] L.K. Grover, Science **281**, 792 (1998)
- [13] L.K. Grover, Physical Review Letters **79**, 325 (1997)
- [14] C. Zalka, Phys. Rev. A **60**, 2746 (1999)
- [15] D. Biron, O. Biham, E. Biham, M. Grassl, D.A. Lidar, in *Quantum Computing and Quantum Communications* (Springer Berlin Heidelberg, 1999), pp. 140–147
- [16] E. Biham, O. Biham, D. Biron, M. Grassl, D.A. Lidar, Phys. Rev. A **60**, 2742 (1999)
- [17] IBM, *IBM Q Experience Devices*, [accessed on 2018-12-01], <https://www.research.ibm.com/ibm-q/technology/devices/>
- [18] Qiskit Community, *Open Source Quantum Information Software Kit [software]*, [accessed on 2018-10-26], <https://www.qiskit.org/>
- [19] IBM, *IBM Q Experience Project*, [accessed on 2018-10-26], <https://www.research.ibm.com/ibm-q/>
- [20] M.A. Nielsen, I.L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10th edn. (Cambridge University Press, 2011)