



State of practice: Evaluating GPU performance of state vector and tensor network methods

Marzio Vallero^{ID}*, Paolo Rech, Flavio Vella

University of Trento, Trento, Italy

ARTICLE INFO

Dataset link: <https://github.com/MarzioVallero/FGCS-state-of-practice>

Keywords:

Quantum
Tensor network
HPC

ABSTRACT

The frontier of quantum computing (QC) simulation on classical hardware is quickly reaching the hard scalability limits for computational feasibility. Nonetheless, there is still a need to simulate large quantum systems classically, as the Noisy Intermediate Scale Quantum (NISQ) devices are yet to be considered fault tolerant and performant enough in terms of operations per second. Each of the two main exact simulation techniques, state vector and tensor network simulators, boasts specific limitations.

This article investigates the limits of current state-of-the-art simulation techniques on a test bench made of eight widely used quantum subroutines, each in different configurations, with a special emphasis on performance. We perform both single process and distributed scalability experiments on a supercomputer. We correlate the performance measures from such experiments with the metrics that characterise the benchmark circuits, identifying the main reasons behind the observed performance trends. Specifically, we perform distributed sliced tensor contractions, and we analyse the impact of pathfinding quality on contraction time, correlating both results with topological circuit characteristics. From our observations, given the structure of a quantum circuit and the number of qubits, we highlight how to select the best simulation strategy, demonstrating how preventive circuit analysis can guide and improve simulation performance by more than an order of magnitude.

1. Introduction

Do we really need to build quantum computers?

The answer is blatantly affirmative, however knowing where the hard boundary for the classical simulation of quantum systems lies is most definitely *not* a straightforward task.

The quantum computing paradigm, since its theoretical inception by one of the forefathers of modern physics R. Feynman in 1982 [1], served to extend the classical definition of computation in order to better describe the quantum properties of nature, under the hypothesis that *an experiment*, which is a purposefully built physical system, can be said to be *performing computations* under specific conditions [2]. Years later, many quantum algorithms would be proposed, able to exploit the characteristics of this new paradigm to achieve unprecedented speedups in unstructured search problems [3] or to efficiently solve hard mathematical problems, such as prime factorisation of large numbers [4], and other more recent applications in the fields of optimisation [5], machine learning [6] and chemistry [7].

In the last decade, progress in the various technologies used for building quantum devices has reached commercial applications, such as

the cloud services proposed by IBM, D-Wave and Google, among others. These systems, however, are yet to be considered mature, as most still fail in preserving coherent quantum states, suffer cross-talk among their constituent elements and employ imperfect logical operators. Furthermore, superconduction-based devices, which make up a wide margin of the operational quantum devices as of the writing of this article, have been recently proven to be exceedingly susceptible to external radiation events [8–11], once again hindering their applicability.

Meanwhile, the top performing supercomputers have surpassed the exascale number of floating operations per second [12]. Naturally, such an achievement pushes up the threshold for the complexity of quantum systems that can be classically simulated over a reasonable time span [13]. This opens up the possibility for some quantum algorithms to be efficiently converted into quantum-inspired classical algorithms. Moreover, the testing and validation of newly proposed quantum algorithms, and the accuracy measurement of real quantum devices' outputs must be done through simulation, to double check results and ensure correct operation. With the increasingly higher optimisation

* Corresponding author.

E-mail addresses: marzio.vallero@unitn.it (M. Vallero), paolo.rech@unitn.it (P. Rech), flavio.vella@unitn.it (F. Vella).

of specialised software libraries over commodity hardware accelerators, such as graphic processing units (GPUs), the rapid simulation of *small* quantum algorithms is becoming more and more easy to accomplish [14–17]. Such practices are, of course, still limited in memory and time by the exponential requirements of quantum simulation, but better exploitation of current classical computation resources may lead to efficient simulation of small quantum subroutines without needing to compensate for hardware-level noise and external radiation events, possibly reducing or annulling the queries made to cloud-based quantum computers for low qubit sized problems. There are various classical simulation techniques for quantum algorithms other than state vector simulation and tensor network contractions, such as stabilisers theory, p-block simulation [18,19], or branching-based techniques [20]. Notably, however, stabiliser theory is limited to the application of a non-universal set of Clifford gates, whilst p-block simulation leverages approximations in the representation that lead to very limited entanglement. Being the only two approaches able to find an exact solution, the analysis we propose focuses on state vector simulation and tensor network contraction techniques. Recent papers have demonstrated that GPU-based quantum simulation approaches can bridge the gap between the capabilities of current quantum computers and classical supercomputers [21], although hardware specific parallel quantum simulation methods had been introduced long time before in both the single-GPU [22,23] and distributed setting [24]. Plenty of review articles have addressed and measured the effectiveness of classical approaches for quantum circuit simulation. Some of those works tested were limited to state vector simulation approaches only [25–27], comparing the overall computational efficiency for a small set of quantum circuits on GPU accelerators, whilst others proposed hybrid CPU–GPU systems for heterogeneous quantum simulations that could outperform GPU-only implementations [28] alongside communication optimisations for this hybrid setting [29]. Other works have instead only considered the performance of tensor network contractions, focusing on problem specific optimisations to improve GPU-based simulation performance [30], or on general approaches for tensor network simulation [31]. However, there is little investigation on the correlations between quantum circuit structure and time-to-solution, comparing both state-vector simulation and tensor network contraction methods on a large set of quantum circuits on modern GPU hardware.

The purpose of this article is to understand where the limit for efficient quantum simulation on classical hardware lies, emphasising the computational aspects, such as distributed performance, scalability, time and memory footprints of quantum algorithms, with the objective to find quantum circuit features that correlate to simulation performance. The questions we seek an answer for are:

- What is the performance of state of the art quantum simulation methods?
- Which topological features of a quantum circuit correlate to simulation performance, and which simulation approach is more suitable?
- Are there limitations to distributed quantum simulation, and can we predict them?

We show that, by profiling quantum circuits with the approach presented in this paper, the simulation time can reach a speedup of *up to one order of magnitude*, especially for large quantum circuits, on a single GPU. Furthermore, we report the results from distributed tensor contraction simulations, highlighting speedups of more than 364× with respect to single GPU performance, and the we trace the impact of pathfinding quality on the contraction performance, obtaining speedups of up to 4.79× through tuning. The proposed circuit metrics to performance correlation is achieved by characterising a purposefully selected suite of well known quantum circuit subroutines according to objective metrics, and checking how those scale with respect to the size of the quantum circuit. All the circuits we consider are parameterisable

over the number of qubits in the system, and some of them feature additional customisation parameters, such as layer repetition. Moreover, they have been selected as to have practical applicability in terms of exact simulation. These same circuits have been simulated on CINECA's Leonardo supercomputer, the 9th supercomputer in the Top500 list, using both state-vector and tensor network contraction methods through NVIDIA's *cuQuantum* library [14], highlighting which one boasts the better performance for each workload. Our work proposes a practical methodology to pick the most efficient simulation strategy according to a given set of static characteristics of the circuit.

The rest of the paper is organised as follows: Section 2 provides a short summary of quantum computing as a whole, and it is followed by Section 3, which introduces how classical algorithms for quantum simulation work. Section 4 gives a definition of the metrics and of the quantum circuits considered for this study. Section 5 characterises the quantum circuits according to the aforementioned metrics, then presents performance results with respect to execution times and peak memory occupancy, scaling of distributed tensor network contractions and impact of pathfinding resources on tensor network contraction times. Lastly, Section 6 concludes the paper by expanding on the hereby presented work by opening new paths for investigation in future works.

2. Background

Nature, on an atomic and subatomic scale, is inherently quantum. When tasked with modelling and simulating the properties of such atomic-scale phenomena, it is reasonable to do so with objects that are able to express the same quantum properties that are to be investigated. This implies that the binary computation paradigm has to evolve towards a representation which encompasses such additional characteristics.

2.1. Quantum computing

Quantum computing is an expansion of binary computing able to tackle any problem that the latter approach can tackle, whilst at the same time providing more efficient solutions to problems that are deemed intractable in the classical domain. This is achieved by exploiting *ad hoc* resources, namely *superposition* and *entanglement*.

Quantum computers make use of *qubits*, the quantum counterpart of classical bits, to encode information. Each qubit is described by two complex probability amplitudes, as follows

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad \alpha, \beta \in \mathbb{C} \quad (1)$$

Each amplitude, when squared, represents the probability for the qubit to collapse on its corresponding classical state when observed. It must hold for the sum of the two squared probabilities to equal unity, i.e. $\alpha^2 + \beta^2 = 1$. Whenever the two amplitudes assume non-integer values, the qubit is said to be in *superposition* between the two basis states. The property of *entanglement* refers to the fact that two or more qubits can share a non-classical correlation, such that when one of the entangled qubits is observed, it lets us infer information regarding the state of the others without measuring them. This second property sprouted issues with locality of quantum mechanics in the well-known EPR paradox [32], later solved by J.S. Bell [33], which defined the minimum set of these namesake entangled quantum states. These two qubit states cannot be described as the product of two independent qubits, such as

$$|\Phi^+\rangle = \alpha|00\rangle + \beta|11\rangle \quad (2)$$

Entanglement is believed to be the fundamental resource responsible for quantum speedup, although superposition plays an important role as well, since circuits with low entanglement have been proven to be trivial to simulate [34].

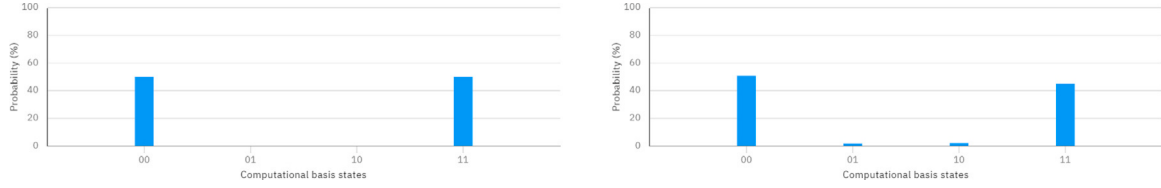


Fig. 1. Comparison between the theoretical output distribution of the Bell circuit (left) and the experimental output distribution obtained from a real quantum device, an IBM Falcon r4T processor, over 1024 measurement shots (right).

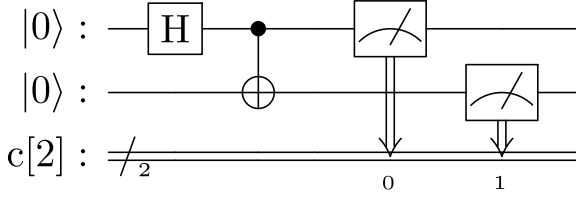


Fig. 2. Quantum circuit encoding the $|\Phi^+\rangle$ Bell state and measure operators.

2.2. Quantum circuits

Algorithms in the QC field are expressed via quantum circuits, a graphical notation derived from Penrose's notation [35]. They are read from left to right, following the flow of information. Operations on one or multiple qubits are applied via *quantum gates*, which are represented by $2^N \times 2^N$ unitary matrices, with N being the number of qubits acted upon by the gate. The most basic single qubit operators include the Pauli X, Y, Z gates and the basis-swap Hadamard gate, represented by the following matrices

$$\begin{aligned} X &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, & Y &= \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \\ Z &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, & H &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \end{aligned} \quad (3)$$

Multiple qubit gates are generally employed to generate entanglement, such as the controlled-not (CNOT) gate, that applies an X gate to a target qubit if the control qubit is in the $|1\rangle$ state. The matrix representation of the CNOT gate is

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (4)$$

Similarly to the binary paradigm, it is possible to define universal quantum gate sets that can be used to encode any operator as the composition of these fundamental ones. This is at the basis of the construction of universal quantum computers.

Fig. 2 shows the quantum circuit that encodes the $|\Phi^+\rangle$ Bell state of Eq. (1): the two qubits are initialised in state $|0\rangle$, then the first qubit is put in the equiprobable superposition state $|+\rangle$ and is used as the control of a CNOT gate. The second qubit is now entangled the first.

The execution of a quantum circuit on a real quantum device yields a single *classical* output bit string. By repeatedly executing and measuring the same circuit, it is possible to sample an output distribution, which converges to the element-wise square of the state vector made of the probability amplitudes of each possible quantum state in the system. Experimental results and expected theoretical results oftentimes differ, due to intrinsic errors in the quantum device, cross talk among qubits and the number of samples made. Such an example has been provided in Fig. 1. Classical simulation of quantum systems is mainly used to compute the expected theoretical output distribution of a quantum circuit, so as to later compare it to the output of a real quantum device.

3. Quantum circuit simulation

Quantum simulation can refer to two concepts: either the usage of real quantum devices to simulate other quantum systems, or the usage of classical machines to compute the theoretical output of a quantum algorithm. For the sake of clarity, this paper will always refer to the latter when talking about *quantum circuit simulation*. The objective of simulation is not to thwart the development of real quantum devices, but rather to validate the outputs of such machines against their theoretical expected outputs. Moreover, given the still relatively scarce availability of real quantum devices and the limitations of current device technology, such as low coherence times, quantum simulators provide a means for validating new and possibly deeper quantum algorithms. There are various approaches to simulate a quantum circuit, the two main ones that provide exact results are: *state vector* simulation [36] and *tensor network* contraction [37], which are detailed in the following subsections.

We aim at understanding how the inherent structure of a quantum circuit can affect the execution time, so as to preemptively identify which simulation strategy works best for which kind of circuit, by making use of *ad hoc* metrics. These metrics provide a description of the overall structure of the quantum circuit, highlighting critical areas for the improvement of modern simulators. It will be possible to infer that any other quantum circuit, reflecting the characteristics provided in this paper, will scale similarly in terms of simulation. The main current limitation of state vector simulators is the inherent exponential memory blowout linked to the system size, that has been tentatively compensated through state vector compression [38], however the distributed application of vector-matrix multiplications still scales exponentially on the system size. Tensor networks have already shown promising results, with useful applications in the field of verification of real quantum computer's outputs [39–41], however due to the limited exploitation of the internal structures of the circuit-derived graph representation, the contraction path used is not granted to be optimal.

3.1. State vector simulation

Quantum states are represented by a wave function, which can be encoded into a state vector. Given any quantum system of N qubits, its corresponding state vector will contain 2^N complex probability amplitudes, one for each possible output bit string. Quantum gates are applied by splitting the state vector into smaller vectors of size equal to that of the gate to be applied, then each sub-vector is multiplied with the gate matrix and the resulting sub-vectors are reassembled in the evolved state vector. The splitting operation is performed according to the qubits over which the operator is applied. This can be intuitively understood by considering the ordered set of output bit strings: the probability amplitudes corresponding to a given sequence of qubits, which depends on the qubit indices the operator acts onto, are grouped together. An example of this process for both single and two qubit gates is depicted in Fig. 3.

The state vector simulation's complexity scales linearly in time with respect to the number of gates [36]. However, the memory footprint of the state vector and the number of vector-matrix multiplications performed increase exponentially with the number of qubits present

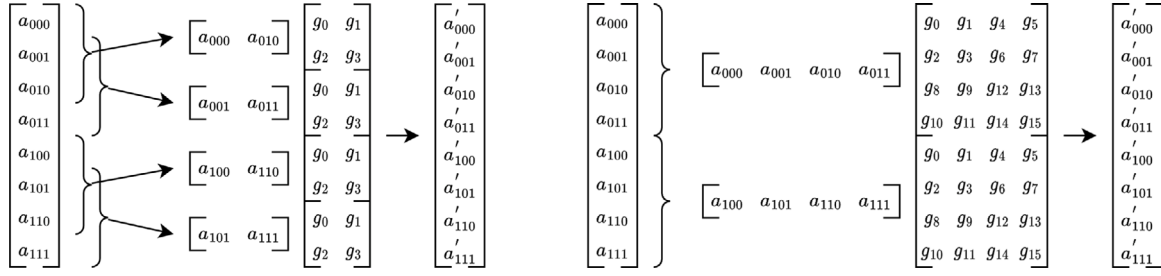


Fig. 3. Splitting of the state vector and application of the vector-matrix multiplication in a 3 qubit system. On the left, a single qubit gate is applied to qubit 1, so the amplitude pairs are grouped by following a 0-1 repeated scheme for the amplitude's index. On the right, a double qubit gate is applied on qubits 0 and 1, so the amplitude pairs are grouped following a 00-01-10-11 scheme. All amplitude indices are written in little endian and ordered top to bottom.

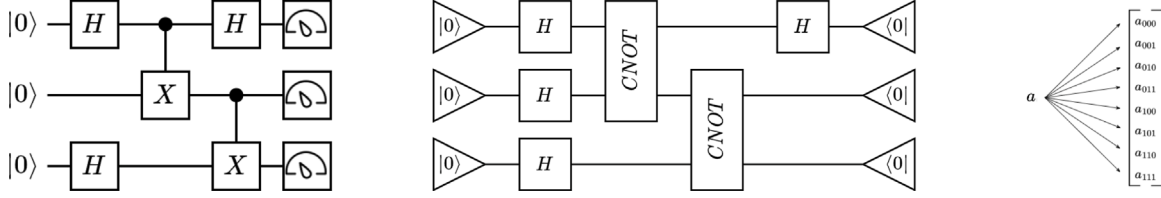


Fig. 4. The process of converting a quantum circuit into a tensor network to extract a probability amplitude. On the left, an example of a quantum circuit. In the centre, the circuit gets converted into a tensor network representation, where single and double qubit gates become order-2 and order-3 tensors, respectively. On the right, after the tensor network contraction, we get the probability amplitude of a specific bit string: by repeating this process over all output bit strings, one may reconstruct the whole state vector.

in the system to be simulated, so this approach is not scalable indefinitely. To put that into perspective, it is possible to roughly estimate the number of atoms in the observable universe to be $10^{82} \approx 2^{270}$ [42]: this means that, if we were to store a single amplitude value inside each of them to represent a state vector, we could only represent state vectors of systems with up to 270 qubits. Well known quantum algorithms need significantly more logical qubits [43], and this is without considering the cost in terms of classical computation time, which may add up to reach unfathomable time scales [44]. Overall, the state vector approach is generally convenient when simulating small quantum systems, as it produces a full description of the output wavefunction. The task of state vector simulation inherently exposes chances for parallelisation, both for circuit specific approaches [45,46], and general circuit-agnostic approaches [14,47]. Despite the hindrances induced by communication and memory requirements, state vector simulation can still efficiently simulate quantum systems of limited size better than other competing methods [26,48].

3.2. Tensor network simulation

Quantum gates and quantum basis states are represented by tensors. The graphical representation of a quantum circuit can be read as a directed acyclic graph (DAG), where the vertices are represented by quantum gates or basis states and the edges are represented by the qubit wires. The input tensors are the basis states, encoded as follows

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (5)$$

All other gates use their standard matrix representation. The contraction of an edge corresponds to the multi-dimensional generalisation of the dot product between two tensors over a shared index. The measurement operators at the end of the quantum circuit are substituted by open indices. Whenever a full network contraction operation is performed, the output open indices are closed with the conjugate tensors of the basis states that encode a specific bit string [49]. Doing so, followed by the contraction of the newly closed indices, produces the probability amplitude of the chosen bit string. Fig. 4 provides graphical insight over this process.

It is possible to use tensor networks to reconstruct the whole state vector, by closing and contracting the output indices over different bit

strings, however doing so incurs in the same limits of the state vector simulator for storing the final vector.

Observation I

Tensor network contractions can reconstruct the full state vector, and such process can be trivially parallelised over different bitstrings.

The memory occupancy of the tensor network grows linearly with respect to the number of quantum gates and qubits in the system. This approach moves the complexity of simulation to that of finding an optimal contraction path for the tensor network, which is known to be an NP-hard problem [50]. Efficient heuristics specialised for quantum circuit-derived tensor networks have been proposed [51,52], however there is no *catch all* solution for this kind of problem. The pathfinding algorithm used in this work [37], despite representing the state-of-the-art for this class of problems, only strives to optimise having the lowest possible amount of floating-point operations across the whole contraction process, which does not prevent the formation of large intermediate tensors, something that inevitably reduces contraction performance. Besides, as it will be discussed in Section 5.4, the optimiser may easily be locked in a local minimum in some problems, whereas other problems feature smoother landscapes in terms of pathfinding complexity. If the contraction path is not optimal, it may lead to increased computation time, possibly making it less efficient than state vector simulation altogether. There is abundant evidence supporting the fact that tensor contraction can simulate quantum circuits and systems that lie outside of the reach of state vector methods [39,53]. To the interested reader, we suggest some resources for tensor network theory by J. Biamonte [54,55].

4. Benchmarks and metrics

To assess the performance of current state of the art simulators and to select the most efficient one, it is necessary to use a set of metrics and quantum circuits which are relevant and well established in the quantum computing field. We rely on two quantum circuit benchmarking suites, which are widely recognised in the literature: SupermarQ [56] and QASMBench [57]. Both of these suites provide their own sets of quantum circuits, that, however, have been specifically selected for

testing the hardware performance of real quantum devices. For this reason, some of these circuits boast little to no practical use in the context of noiseless exact simulation, such as the error correcting code circuits in SupermarQ, or the Greenberger–Horne–Zeilinger.

Observation II

Not all quantum circuits generally used for benchmarking are computationally representative in a classical simulation environment.

Furthermore, both suites introduce a list of metrics that characterise the topological nature of static quantum circuits. These metrics provide a measure of topological properties of the graph derived from the quantum circuit representation, letting us correlate such properties with the runtime performance statistics.

4.1. SupermarQ

In the SupermarQ [56] suite, six metrics are introduced, however we will only consider the ones that have topological significance, referring to all elements which may alter the circuit-derived graph, that is the relative presence and the disposition of two-qubit or higher size quantum gates. All metrics range in $[0, 1]$, where higher is closer to 1.

4.1.1. Program communication

This metric measures the amount of interconnections present in a quantum program, computed as the ratio of the average degree of interaction of the quantum circuit in graph form with that of a maximally connected graph with a number of nodes equal to the number of qubits in the circuit. The term $d(q_i)$ is the degree of the i th qubit.

$$PC = \frac{\sum_i^N d(q_i)}{N(N-1)} \quad (6)$$

4.1.2. Critical depth

The critical depth represents the ratio between the longest chain of two-qubit operators and the total number of two qubit gates in the circuit. It gives a measure on whether the program's output heavily relies on distributed entanglement or not. n_{e_d} is the total number of two qubit gates on the circuit's critical depth path, while n_e is the number of two qubit gates in the circuit.

$$CD = n_{e_d} / n_e \quad (7)$$

4.1.3. Entanglement ratio

This measure is the ratio of the number of entanglement operators, n_e , with the total number of gates in the quantum circuit, n_g .

$$E = n_e / n_g \quad (8)$$

4.1.4. Parallelism

It is a measure of the number of concurrent operations made in the same time step, intuitively understood as the degree of *compression* of the quantum circuit. The number of gates n_g is compared with the depth d of the program, then such value is normalised with respect to the number of qubits n .

$$P = \left(\frac{n_g}{d} - 1 \right) \frac{1}{n-1} \quad (9)$$

4.2. QASMBench

The metrics introduced in the QASMBench [57] suite are more tied to the architectural implementation of physical quantum devices. Follows the definition of the only topologically significant metric.

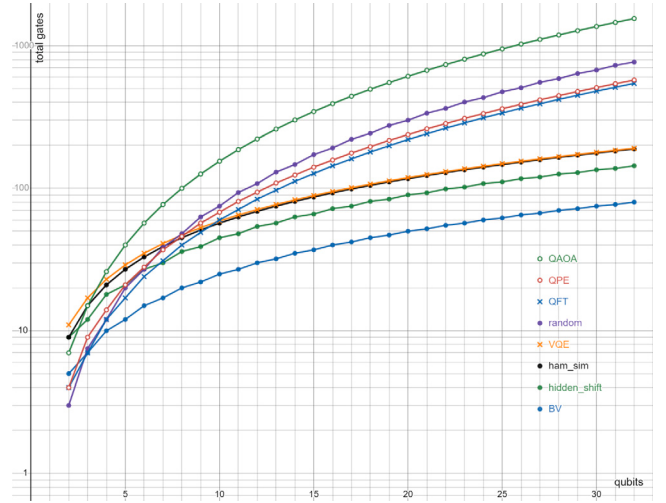


Fig. 5. The scaling of the number of total gates of the circuits considered in this paper, from size 2 to size 32. The Y-axis is in logarithmic scale. The controlling variables are set as $P = 1$, $k = 0.5$, $M = \lfloor kN \rfloor$, $L = 1$, $T = 1$.

4.2.1. Entanglement ratio

This metric defines the spread of entanglement among the qubits in the quantum circuit. It checks whether there are a few qubits which feature most of connections towards the others, or if all qubits are sharing the same amount of entangling connections. In a quantum program with N qubits, the number of two-qubit gates acting on the i -th qubit is $n_{g_2}(q_i)$, while the average number of two-qubit gates per qubit is $\bar{n}_{g_2}$.

$$EV = \frac{\log(\sum_{i=0}^N (n_{g_2}(q_i) - \bar{n}_{g_2})^2 + 1)}{N} \quad (10)$$

4.3. Benchmark circuits

In order to provide a broad, extensive and scalable evaluation, we consider a specific set of circuits, selected to encompass some of the applications for quantum computing that do not leverage the presence of quantum noise. As such, their results are significant in terms of exact theoretical simulation. The list of circuits, with information regarding usage, scaling of the number of gates and references, is detailed in Table 1. All circuits considered can be freely expanded over any problem size, making them easily adaptable to benchmark future hardware and simulation platforms. For the sake of the hereby presented analysis, we tested all circuit qubit sizes in the range $[2 - 32]$.

4.3.1. QAOA

The *Quantum Approximate Optimisation Algorithm* is a variational circuit that use all-to-all connectivity to encode classical problems, such as the max-cut problem. The algorithm version used in this paper is the vanilla one [5] with parameter $P = 1$, although other versions exist, such as the one with ZZ-Swap gates [62].

4.3.2. Random

The *Random* quantum circuit, notably dubbed *Quantum Supremacy circuit* by the GoogleAI group that introduced it [44], is composed of multiple repeated layers of random gates picked from the set $\mathcal{G} = \{H, X, RZ, RX, RY, CNOT, CZ, SWAP\}$. The number of layers has been set equal to the number of qubits in the system. Given the random nature of the circuit, it is possible to define a lower and an upper bound for the number of gates that can be found in the circuit. This circuit has been purposefully built to avoid any internal structure, so as to be as complex as possible to simulate. Despite achieving such a goal, the applicability for this subroutine to real world problems remains questionable at best.

Table 1

The quantum circuits used as benchmarks for the evaluations of this paper.

Circuit name	Description	Total gates	Total multi-qubit gates	Ref.
QAOA	Quantum approximate optimisation algorithm	$\frac{3}{2}PN(N-1) + 2N$	$PN(N-1)$	[5]
Random	Google quantum supremacy circuit	$(1-k)(N(\lfloor N/2 \rfloor + N\%2)) + kN^2$	$kN(\lfloor N/2 \rfloor)$	[44]
QPE	Quantum phase estimation	$\frac{N(N-1)}{2} + 2N - 1 + \lfloor \frac{(N-1)}{2} \rfloor$	$\frac{(N^2-N)}{2} + N - 2 + \lfloor \frac{(N-1)}{2} \rfloor$	[58]
QFT	Quantum Fourier transform	$\frac{1}{2}N(N+1) + \lfloor N/2 \rfloor$	$\frac{1}{2}(N^2 - N) + \lfloor N/2 \rfloor$	[59]
VQE	Variational quantum Eigensolver	$L(5N-1) + N$	$L(N-1)$	[6]
Hamiltonian sim.	One-dimensional Hamiltonian time evolution	$3T(2N-1)$	$T(N-1)$	[56]
Hidden shift	Find the shift s such that $g(x) = f(x+s)$	$3N + 2M + \lfloor N/2 \rfloor$	$\lfloor N/2 \rfloor$	[60]
Bernstein–Vazirani	Hidden bit string extraction	$2N + M$	M	[61]

4.3.3. QPE

The *Quantum Phase Estimation* subroutine is one of the foundational steps in Shor's algorithm [4,58], able to solve the order finding problem of a modulo function.

4.3.4. QFT

The *Quantum Fourier Transform* [59] is one of the most widely known quantum subroutines, which uses phase encoding to efficiently perform the Fourier transform. This quantum circuit is a good candidate for simulation, since it is built by the recursive application of the same operator, possibly giving rise to exploitable internal structures.

4.3.5. VQE

The *Variational Quantum Eigensolver* is an hybrid quantum–classical algorithm that uses iterative optimisation to find the ground state of a molecule encoded in a quantum register. The circuit is built by repeating L times a given structure. For the sake of simplicity, the version used in this paper assumes $L = 1$. Its applications focus mainly on, but are not limited to, the simulation of the bond energies of chemical compounds.

4.3.6. Hamiltonian simulation

This circuit encompasses a general approach for the encoding and simulation of the time evolution of a given Hamiltonian in a quantum computer. The circuit is characterised by the repeated application of a quantum subroutine over $T = \text{total_time}/\text{time_step}$ iterations. The benchmark we consider computes the magnetic interactions of a monodimensional chain of spins.

4.3.7. Hidden shift

The *Hidden Shift* quantum circuit is able to find the value s of a function $g(x) = f(x+s)$ by performing a single query to the function, leveraging the superposition of all the possible inputs. The term k represents the percentage of bits equal to 1 in the binary representation of the shift value.

4.3.8. Bernstein–Vazirani

This quantum algorithm can solve the problem of finding out the bit string that satisfies a given function by performing a single query, whilst the classic algorithm would require at most N queries, where N is the total number of possible bit strings. Once again, the term k is the percentage of bits set to 1 in the binary representation of the solution.

5. Results

In order to find an answer to the research questions defined in Section 1, we will start by characterising the quantum circuits according to the metrics introduced in Section 4, then we will perform various simulations, with the objective to relate the metrics with execution time, memory occupancy, and in the specific case of tensor networks, distributed sliced contraction performance and pathfinding efficacy. The library used for running the simulations is the NVIDIA cuQuantum library (version v24.03) [14], adapted to the specific needs of our analysis. This gives us access to three different GPU accelerated simulation backends:

- *qsim-cusv*: a state vector simulator that uses the *cuStateVec* backend.
- *qsim-cuda*: a state vector simulator that uses the *cupy* backend.
- *cutn*: a tensor network simulator that uses the *cuTensorNet* backend for contraction.

All experiments have been run on CINECA's Leonardo supercomputer. Apart from the distributed experiments, all other experiments have been run on a single node, using 8 cores of an Intel Xeon Platinum 8358 CPU, 128 GB of RAM and one NVIDIA Ampere A100 64 GB GPU. Simulations have all been limited to a problem size of 32 qubits, as that is the largest statevector that can be represented in a single available GPU, although larger tensor networks could indeed be simulated. Given the high computational cost of state vector and tensor network methods, CPU-based simulation algorithms will not be considered for this analysis, since it would not provide any meaningful comparison in terms of performance.

5.1. Quantum circuit properties

Following the results reported in Fig. 6, we will analyse each metric independently. The metrics for the Random, Bernstein–Vazirani and Hidden Shift benchmarks have been averaged over 100 circuit samples, to compensate for the fact that these circuits do not have a constant topology.

The *program communication* keeps a constant value of 1 for the QAOA, QPE and QFT circuits, suggesting that those three algorithms feature at least a two-qubit operation with each of the other qubits in the quantum register. This means that the resulting topology of the circuit will be that of a fully connected graph. All other circuits, on the other hand, quickly drop towards values proximal to 0.1 as soon as the number of qubits in the system increases, meaning that most of the qubits do not interact directly. This can be explained by circuit structures where a small number of qubits interact with all of the others, or by circuit structures where all qubits interact only with their closest neighbours. The main outlier is the Random circuit, which stabilises at a value of about 0.5, meaning that, on average, each qubit interacts with at least half of the total number of qubits in the circuit.

The *critical depth* starts at value 1 for most circuits at low qubit sizes and rapidly drops towards the range [0.08, 0.22] for the QAOA, QPE, QFT, Random and Hidden Shift benchmarks. This, together with the *program communication* score, means that the highly entangled structure of the first three circuits is not due to a chain of two qubit operators. The Hidden shift and the Random circuits, having both low *program communication* and *critical depth* scores, imply that the derived graph structure is sparsely connected, with a few “central” qubits sporting most of the two qubit gates towards all other qubits. The remainder of the quantum circuits in the test suite maintain a constant value at 1.0. This, together with the *program communication* metric implies that the graph structure of the VQE, Hamiltonian simulation and Bernstein–Vazirani circuits can be reduced to that of a single chain of nodes.

The *entanglement ratio* attains its maximum value in the QPE and QFT benchmarks, as those circuits are mainly composed of two-qubit gates. The QAOA and Random circuits are composed from 40% to 60%

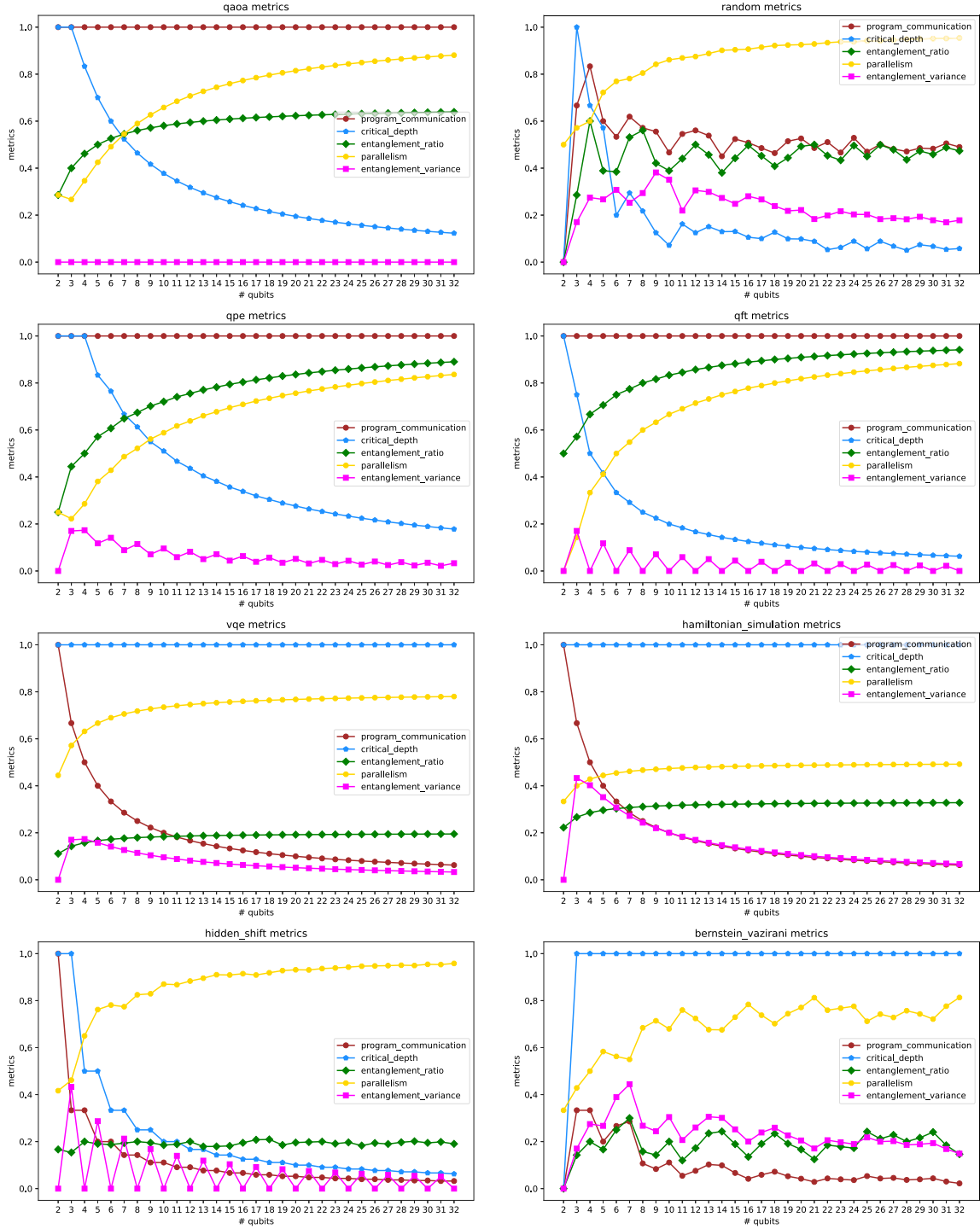


Fig. 6. Metrics computed for all the benchmark circuits. Given that the topological structure of the Random, Hidden Shift and Bernstein–Vazirani circuits depends on an initialisation seed, the metrics for this circuit have been averaged over 100 different problem instances.

by multiple qubit gates, with the former saturating at 60% as the size of the system increases to 32 qubits, whilst the latter, given its non-deterministic structure, boasts an average of about 50%. The other circuits, the Hamiltonian simulation, VQE, Hidden shift and Bernstein–Vazirani, are mainly made of single qubit gates, which can get easily processed during tensor network contraction [37], as such their *ER* scores are lower, ranging [0.15 – 0.35].

The *parallelism* metric grows with respect to the circuit size for all the algorithms considered, with initial values ranging from 0.0 for the QFT to 0.5 for the Random. Generally, however, the metric's value

saturates to different levels, with the QAOA, Random, QPE, QFT and Hidden Shift circuits passing the threshold $P > 0.8$ for systems sizes of 32 qubits, suggesting that the density of their derived topology is very high. On the other hand, the VQE and Bernstein–Vazirani circuits saturate in the range [0.6 – 0.8], suggesting a slightly more sparse topology. The Hamiltonian simulation circuit saturates at value $P \approx 0.5$, highlighting its dependence on sequential processing of quantum information and a lower topological density.

The *entanglement ratio* rapidly approaches zero for almost all circuits considered in the benchmark suite. Notably, the QAOA circuit has a

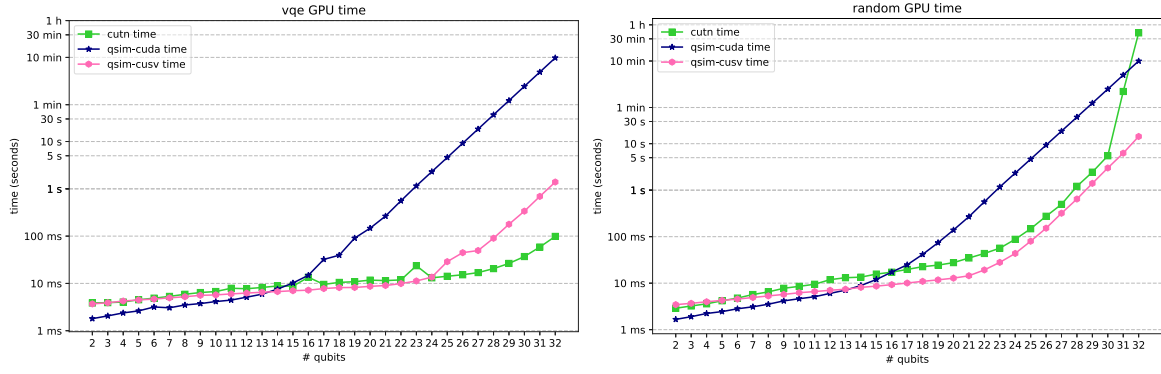


Fig. 7. The execution times for the VQE circuit (left) and the Random circuit (right). All data has been collected by doing 3 warmup runs and then averaging the results of 10 additional runs. The tensor network time is the sum of the pathfinding time, done in CPU, and the contraction time, done in GPU.

Table 2

The *nsys* profiler metrics gathered from the QFT benchmark on the *qsim-cuda* and the *qsim-cusv* simulators.

qubits	qsim-cuda			qsim-cusv		
	Memory	Kernels	Memory usage (KiB)	Memory	Kernels	Memory usage (KiB)
13	1.6%	98.4%	88.16	38.7%	61.3%	26 839.04
14	1.2%	98.8%	152.28	39.1%	60.9%	26 900.48
15	33.2%	66.8%	280.53	29.3%	70.7%	27 033.6
16	33.9%	66.1%	537.03	30.1%	69.9%	27 289.6
21	8.5%	91.5%	16 435.2	14.3%	85.7%	43 212.8
22	3.9%	96.1%	32 952.32	8.0%	92.0%	59 607.04
23	2.1%	97.9%	65 873.92	4.0%	96.0%	92 436.48
24	1.1%	98.9%	131 717.12	2.1%	97.9%	158 105.6
25	0.6%	99.4%	263 413.76	1.1%	98.9%	289 433.6

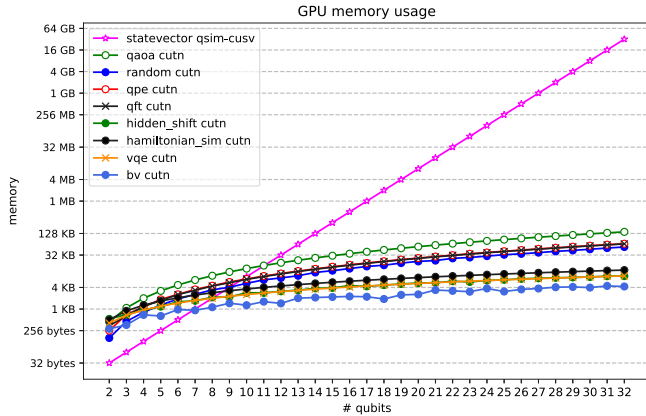


Fig. 8. Memory occupancy of a general state vector and the circuit derived tensor networks.

constant variance value of 0.0, meaning that independently from the circuit size, the number of two qubit operators is evenly split amongst all the qubit in the system. The QPE, QFT, VQE and Hidden Shift algorithms see a rapid decrease in the metric's value, approaching the range $[0.0 - 0.05]$ for circuits of 32 qubits, again hinting at the fact that most of the qubits take part in a similar amount of multi-qubit operations. The only two exceptions are the Random and the Bernstein-Vazirani circuits, which instead have a higher value of $ER \in [0.18 - 0.25]$. In the case of the Random circuit, this is due to the fact that the structure of the circuit does not follow a predefined scheme, whilst in the Bernstein-Vazirani circuit it is directly depended on the number of 1s present in the solution binary bitstring of the oracle function.

5.2. Single GPU simulation performance

Fig. 8 details the memory requirements scaling for a general state vector simulator and the tensor network representations of all the circuits considered in our benchmark.

The memory occupancy for both the *qsim-cuda* and the *qsim-cusv* simulators is the same, as they both have to store in memory the whole state vector of complex probability amplitudes. Each probability amplitude is stored as a *complex-64* single precision binary number, scaling exponentially in memory, since the size of the state vector is $N = 2^n$, with n being the number of qubits in the system. The state vector is updated when applying new quantum gates, however its size remains unaltered, regardless of how many subsequent operations are applied to it.

The tensor network representation instead stores the initial state as a sequence (1, 2) tensors, and each quantum gate as either a (2, 2) order-2 tensor, in the case of single qubit operators, or as a (2, 2, 2, 2) order-4 tensor, in the case of a controlled gate, adding two dimensions of size 2 for each additional input of the operator. As such, the size of the tensor network representation scales linearly with the number of gates in the quantum circuit and the number of qubits in the system.

In Fig. 7, we can see a side by side comparison of the execution times of the VQE and the Random circuits. The time performance data has been collected, for each circuit configuration, as the average time over 10 runs after having performed 3 warmup runs. Thus we collectively performed 3244 quantum circuit simulations for this experiment. Code and execution time data relative to these other circuits is available at [63], along with *Nsight* profiles of a subset of the considered quantum circuit benchmarks, which are summed up in Table 2. We notice how the *qsim-cuda* simulator has the overall fastest performance for circuits with size of 14 qubits or less, after which the performance of the simulator degrades significantly.

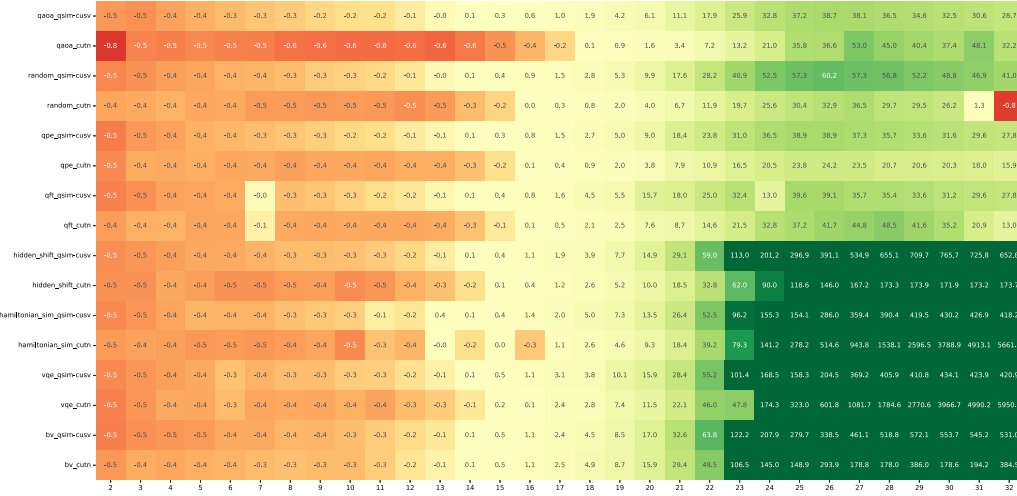


Fig. 9. Heatmap of the speedup of the *qsim-cusv* and the *cutn* simulators with respect to the *qsim-cuda* simulator.

The *qsim-cuda* simulator moves the whole statevector in memory through cache at once using synchronous mempcpy calls, while the *qsim-cusv* simulator instead splits the statevector into multiple sub-statevector, invoking multiple asynchronous mempcpy operations and applying the gate separately on each sub-statevector. This leads to overall better performance for the *qsim-cuda* simulator in quantum circuits that use less than 14 qubits, as a single synchronous mempcpy operation is required, whilst *qsim-cusv* suffers the high number of asynchronous mempcpy operations, which overpower the computational intensity of applying one quantum gate. The shared memory size of the NVIDIA A100, the device we used for simulations, is only configurable up to 164 KB: the state vector of a 14 qubits system, at *complex-64* single precision, occupies 128 KB. As soon as the statevector size exceeds that of the cache, the *qsim-cuda* simulator is forced to split the statevector into sub-statevectors (like in the *qsim-cusv* case), but keeps doing so with synchronous mempcpy operations. On the other hand, the *qsim-cusv* simulator manages to hide more of the computational complexity of applying quantum gates by overlapping asynchronous mempcpy operations. With problem sizes larger than 22 qubits, however, the computational intensity of applying a quantum gate starts to overshadow the advantage of asynchronous I/O, with a degradation of performance. This performance difference between *qsim-cuda* and *qsim-cusv* is largely due to the memory access patterns of those two simulators. The *qsim-cusv* simulator can make use of index bit swapping, which gives it an advantage in cache locality with respect to the *qsim-cuda* simulator [14].

In the case of the Hamiltonian simulation, VQE and QAOA circuits, the tensor network outperforms both state vector simulators. This is a consequence of the fact that these circuits are mainly composed of single qubit gates, corresponding to an *entanglement ratio* score lower than 0.5, and have a well distributed amount of entanglement across the system, meaning an *entanglement ratio* score lower than 0.2. This leads to having small intermediate tensors during the contraction step, that eventually get merged together into larger tensors right towards the end of the process. As soon as one of these two metrics raises up too much, the pathfinding algorithm we employed [37] struggles find an optimal contraction path. Moreover, once all the order-2 tensors have been contracted, the resulting topology resembles that of a matrix product state (MPS), which is optimal for contraction. This is an important information, as the contractions of order-2 tensors are easy to perform and do not increase the intermediate tensor order during the contraction process.

Observation III

Unstructured and unbalanced tensor networks give rise to large intermediate tensors during contraction in function of their qubit size and *program communication* metric, hindering performance.

The circuit with the worst performance for the *cutn* simulator is the Random circuit, which is the only problem where performance degrades by more than an order of magnitude, mainly due to its lack of an internal structure, something which can be clearly noticed on the right of Fig. 7. In all other benchmark circuits, the performance in terms of time is comparable to that of a state vector simulator, whilst keeping the reduced memory footprint of the tensor network representation.

In Fig. 9, we can see the relative speedup of the *qsim-cusv* and the *cutn* simulators when compared to the *qsim-cuda* simulator, whose baseline performance is mostly dependent on the system size, rather than on the number of gates to be processed. We can notice negative speedups in the left half of the heatmap, as the *qsim-cuda* backend outperforms the two other simulators. On the second half of the heatmap, we can appreciate noticeable speedups on both backends. In the QAOA, Random, QPE and QFT circuits the speedups range to up to 60× for the *cusv* backend and up to 53× for the *cutn* backend. We can notice larger speedups on the Hidden Shift, Hamiltonian Simulation, VQE and Bernstein–Vazirani circuits, which exceed the 5000× speedup value for Hamiltonian Simulation and the VQE. These last four circuits are the ones that scale more slowly in terms of the number of quantum gates, as previously seen in Fig. 5. The main outlier is the Random circuit, which has poor performance on the *cutn* backend, with a negative speedup at high qubit sizes.

Due to limitations in our software environment, we were unable to extend our analysis to distributed statevector simulation, and as such to test larger quantum circuits with this approach. However, the time to solution trend is evident for both statevector simulators. As previously discussed in the paper, and as highlighted in Fig. 8, the statevector method suffers from diminishing returns as the problem size grows. We note some large scale distributed results in the literature, which are in accordance with our single-gpu trends [27].

5.3. Distributed sliced tensor contraction performance

In order to understand how the performance of tensor network contraction scales as we increase the number of GPUs, we designed a strong scaling experiment. We implemented a distributed version of the tensor network contraction algorithm, by leveraging the cuTensorNet

library, MPI and NCCL, and ran scaling experiments for all of the circuits in the benchmark at size 32 qubits, apart from the Random circuit which was limited at size 28 qubits, by using an increasing number of GPUs and compute nodes on the Leonardo Supercomputer, with the objective to test the efficacy of tensor network slicing in improving contraction efficacy. The algorithm starts by spawning one MPI process for each available GPU, and first performs a distributed pathfinding on the whole network. The best path, selected according to the lowest FLOPs count, is broadcast to all other MPI processes through the MPI communicator. The tensor network is then sliced in a number of sub-networks equal to the number of MPI processes, in order to provide to each MPI process, and thus each GPU, a comparable amount of FLOPs to be performed. Each GPU contracts its own sub-network, and all the partial results are reduced with a sum operation through the NCCL communicator, that yields the final amplitude result.

In Fig. 10, we present the strong scaling results of this experiment, from using 1 GPU going up to 256 GPUs. The node configuration on Leonardo includes 4 GPUs per node, and each node is interconnected with an Nvidia Mellanox network, reaching up to 200 Gbit/s node to node transfer rates. Given the size of the partial results being reduced, network bandwidth does not act as a computational bottleneck, but rather the arithmetic intensity of the contraction of the *worst* sub-network. We can see from the strong scaling results that not all the quantum circuits selected in the benchmark exhibit large performance gains, particularly the Hidden Shift, VQE, Bernstein–Vazirani and Hamiltonian simulation. These circuits lack complexity in the structure of their circuit-derived tensor networks, as reported by their asymptotically decreasing program communication metric in relation to the size of the quantum circuit: this makes it so that the overall arithmetic intensity of each sub-network fails to saturate the GPU's computational capacity. The QAOA sees a noticeable improvement in terms of contraction performance, steadily lowering the contraction time from 81 ms on a single GPU to as low as 19 ms on 16 GPUs, a speedup of more than 4.2 $\times$. Likewise, the Random circuit's contraction time goes from a mean of 1.89 s when using a single GPU to about 120 ms in the case with 256 GPUs, a speedup of about 15 $\times$. The largest improvements in the contraction times can be obtained on the QFT and QPE quantum circuits, which drop more than one order of magnitude in contraction time when going from running on one GPU to 16 GPUs: we respectively measure a speedup of more than 294 $\times$ on the QFT circuit and a speedup of more than 364 $\times$ on the QPE circuit, in lieu of just increasing by 256 $\times$ the computational resources available. This is due to the fact that these two last quantum circuits feature high *program communication* and *entanglement ratio* scores, which means that most of the contraction operations will involve order-4 tensors, which quickly ramps up the size of the intermediate tensor during contraction. The higher the number of sub-tensor networks being contracted, the smaller the overall order of the final sub-tensor is going to be. Scaling the number of GPUs thus prevents the formation of these large sub-tensors, which get contracted only once in the final reduce step, lowering the overall contraction time.

Observation IV

Quantum circuits with large program communication and entanglement ratio scores benefit the most from sliced distributed contraction, reaching superlinear speedups with respect to a linear increase in computational power.

Albeit some of the benchmark circuits considered have been found to be trivially solvable even by a single GPU, we demonstrated how specific descriptive metrics of a quantum circuit, namely the program communication and entanglement ratio metrics, can let us foresee the presence or absence of a computational gain with a distributed sliced tensor network contraction.

On the topic of evaluating distributed sliced performance, it is still unclear how to measure weak scaling performance of different

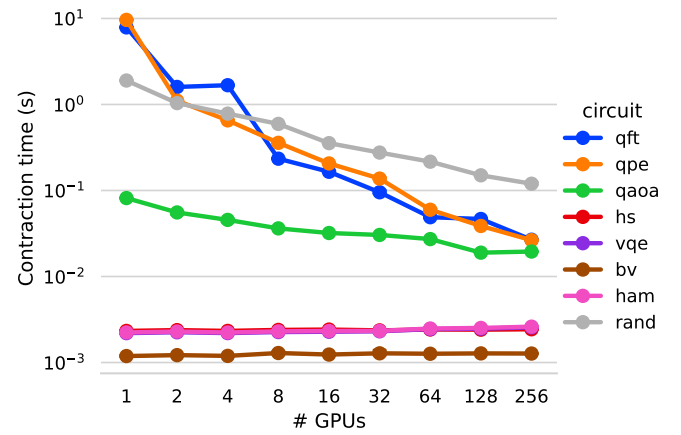


Fig. 10. Strong scaling of distributed sliced tensor network contraction performance on circuits of size 32 qubits, with the exclusion of the Random circuit at size 28 qubits. The number of GPUs (# GPUs) also corresponds the number of MPI processes. Each point represents the mean of 30 datapoints.

quantum circuits, that is measuring the performance of a problem when scaling equally the problem's complexity and the available computing resources. The problems considered in this benchmark are parameterised by the number of qubits, which does not provide direct control over the problem's contraction complexity, which mainly depends on the treewidth of the tensor network [37]. Nonetheless, in Fig. 11 we address this point by considering the QAOA, QPE and QFT circuits over an increasing problem size, ranging from 32 to 40 qubits, scaling from one to 256 GPUs. We imposed a maximum contraction time limit of 120 s. In the leftmost subplot, the QAOA circuit, which is the least computationally intense of the three, is easily handled by a single GPU up to 40 qubits. Most importantly, we observe a communication bottleneck at 256 GPUs, since the intensity of contracting the sub-tensor networks is smaller than the time required to reduce all of them to a single scalar. In the central and rightmost plots, the QPE and QFT circuits, boasting higher computational intensity, start to exceed the imposed time limit at 36 and 34 qubits, respectively. Notably, both circuits take better advantage of the large scale, solving 40 qubit circuits when using 32 and 64 GPUs, reaching a communication lower bound at 128 GPUs. Based on these results, we argue that it could be possible to develop a synthetic parameterisable quantum circuit that grows in terms of the circuit derived tensor network's treewidth to actively measure the weak scaling performance of tensor network contractions. This would most probably end up being a variation of a Random circuit, which however holds no meaning in terms of problem solution.

Observation V

The complexity of contracting quantum circuit derived tensor networks does not scale with the problem size, i.e. the number of qubits. As such, specialised synthetic benchmarks are needed to measure the weak scaling of tensor network contraction.

For the sake of this article, we can safely predict a correlation with the strong scaling capacity of the cuTensorNet library in relation to the program communication metric of a quantum circuit. For quantum circuits with program communication scores of one can efficiently leverage large multi-GPU acceleration. On the opposite case, when the program communication approaches zero, one GPU can suffice, and no advantage is gained from increasing computational resources.

5.4. Pathfinding impact on tensor contraction performance

There is a need to classify quantum circuit derived tensor networks according to their pathfinding complexity. Specifically, we are

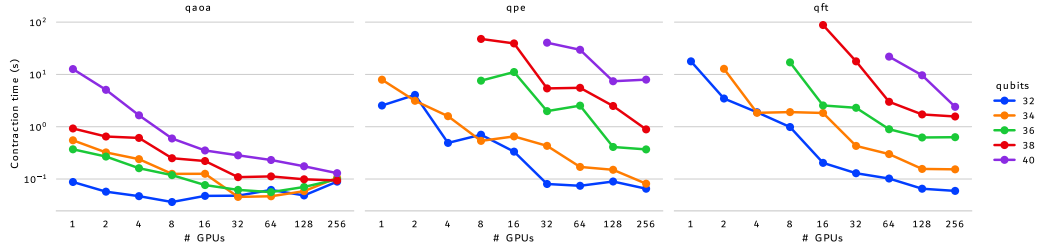


Fig. 11. Strong scaling of distributed sliced tensor network contraction performance on the QAOA, QPE and QFT circuits of sizes ranging from 32 to 40 qubits. The number of GPUs (# GPUs) also corresponds the number of MPI processes.

interested in predicting which circuits can be contracted with higher efficiency in correlation to an increase in the resources available to the pathfinding algorithm. We investigate how a variation in the resources available to the cuTensorNet pathfinding algorithm, namely the number of additional samples performed, impacts on the contraction time of the quantum circuit derived tensor networks in the benchmark. In this context, a sample size of n indicates that $n + 1$ pathfinding calls have been made. We measure the total time required by the pathfinding algorithm to sequentially compute all of the samples by enforcing a single thread. Although this search may be easily distributed, the total computation time required by the pathfinder is still the same. The size of each quantum circuit is kept fixed at 32 qubits, apart from the random circuit of size 28 qubits, due to representability reasons.

In Fig. 12, we plot the variation in the pathfinding and contraction time with respect to an increase in the number of samples, and find three categorisations for the quantum circuits used in this benchmark: pathfinding-bound problems, contraction-bound problems and unbounded problems. Pathfinding-bound problems include the VQE and Hamiltonian simulation circuits, since the complexity landscape of their possible contraction paths is mostly flat. This means that such circuits provide no advantage in terms of contraction time when assigning additional resources to the pathfinding algorithm. Contraction-bound problems include the QAOA, Hidden Shift, Bernstein-Vazirani and Random circuits, and they are characterised by having a non-trivial pathfinding complexity landscape, but trivial contraction complexity. This means that providing additional resources to the pathfinder indeed gives rise to better solutions in terms of total FLOPS in contraction, but the overall difference in terms of contraction efficiency is unnoticeable. Unbounded problems include The QFT and QPE problems, since they show an direct correlation between the amount of resources provided to the pathfinding algorithm and a reduction in the contraction time. In fact, as the number of samples increases, in both cases we can measure a speedup in terms of contraction time of $1.9\times$ on the QPE and of $4.79\times$ on the QFT by increasing the pathfinding time by about $29\times$. From these observations it follows that quantum circuits characterised by an *entanglement ratio* metric that converges to *one* map to more complex pathfinding problems that can efficiently leverage additional pathfinding resources, whilst if the same metric saturates to values lower than 0.4, the problem is bounded, either in pathfinding or contraction.

Observation VI

Quantum circuit derived tensor networks can be classified in pathfinding-bound problems, contract-bound problems, and unbounded problems. Only the latter show an anticorrelation between pathfinding time and contraction time.

Although this result might point towards diminishing returns, it must be noted that the pathfinding time was purposefully measured in a sequential manner on a single thread, but each sample is indeed independent from the other, as such using one independent thread and core per sample is going to keep the real-time pathfinding time constant. Moreover, in terms of real-world tensor contraction, contractions are performed with a batched approach, where the contraction path

is computed only once, but the actual contraction may be performed hundreds or thousands of times, thus outweighing the steep initial pathfinding cost. Given the intense computational cost of pathfinding and its reliance on performing numerous samples, an open question remains on whether this algorithm can be further accelerated using GPUs in order to increase the number of concurrent samples while still fitting a low time bound.

5.5. Lessons learnt

The simulation performance is ruled by a plethora of factors. State vector simulators are inherently limited by the problem size, given the exponential memory requirement, thus trying to simulate systems larger than ~ 50 qubits with state vectors suffers from diminishing returns [38]. Moreover, the distribution of the subvector-matrix multiplications scales exponentially over the problem size, hindering the time performance. The main advantage lies in being able to access the complete set of information encoded in the wave function. Modern state vector simulators can take advantage of GPU caching in order to limit the computation time in small scale problems, but still hit a hard memory bandwidth limit as soon as the cache's capacity is saturated. Tensor Network contraction has a lot of potential for problems which embody a structure with well balanced connectivity, as that of a structured mesh, as this generates small sized intermediate tensors during the contraction process. As such, current pathfinding algorithms are optimised for finding such structures, and struggle whenever the topology becomes more irregular, such as when most of the entanglement operations in a circuit are concentrated on a few qubits. This is because the size of the intermediate tensors immediately spikes up, slowing the dot product with the remainder of tensors. The main advantage of this approach is the fact that it scales linearly in memory, thus allowing the simulation of larger systems, so long as the circuit's structure is favourable for contraction. In fact, by scaling the number of qubits in order to saturate the memory capacity of an 80 GB NVIDIA A100 GPU, assuming that connectivity information is stored in the shared memory of the machine, one would need more than $7.5k$ logical qubits for a QAOA circuit with $P = 1$, or more than $40k$ qubits for the Random circuit, the two circuits in the benchmark with the highest gate scaling per qubit. Despite the fact that there is no guarantee of convergence towards an optimal path, tensor network contraction is the only approach to exact simulation of quantum circuits that can scale favourably to circuits with dimensions larger than 50 qubits. Previous works show that parallelising the contraction process is trivial, thus the main bottleneck remains the pathfinding algorithm, a well known NP-hard problem [64].

As we have observed, circuits which are characterised by an *entanglement ratio* score greater than 0.2 have a highly unbalanced structure, reducing the efficacy of the pathfinding algorithm for tensor network contraction. Likewise, if more than half of the circuit's gates are double qubit gates, which amounts to a *entanglement ratio* score greater than 0.5, the tensor network approach starts to scale poorly, as stated in Observation 4. This is due to the presence of large tensors early on during the contraction, slowing down the overall process. In both of the

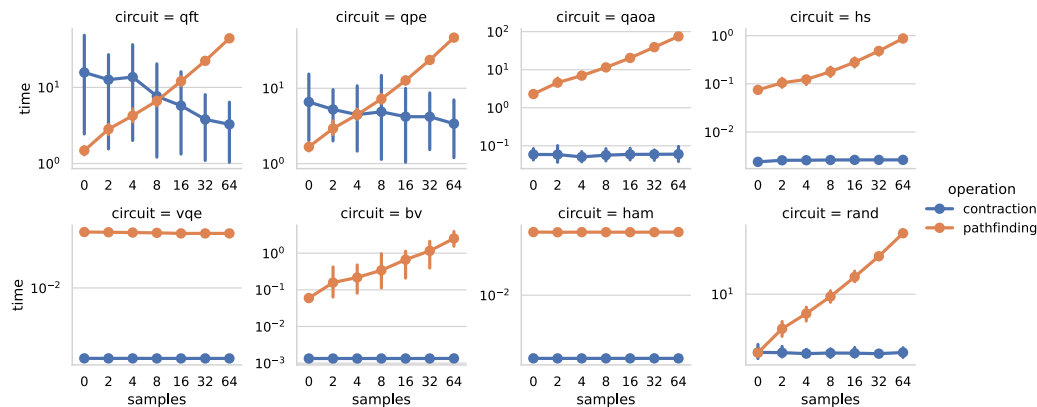


Fig. 12. Pathfinding time (orange) and contract time (blue) for tensor networks derived from circuits of size 32 qubits. The number of samples performed by the pathfinder are executed sequentially in order to extrapolate the total time required for this computation, whilst contractions are performed using a single GPU. Notice different y-axis limits for each circuit in the benchmark. Vertical bars represent the 90th percentile interval.

previous cases is thus suggested to use a state vector solver. However, if these first two conditions are not met, one must check for the *program communication* and the *critical depth* scores. If they are opposite to one another, with one being smaller than 0.2 and the other being larger than 0.9, then the best simulation method is the tensor network contraction. This is due to the fact that, if the previous conditions about the *entanglement ratio* and the *entanglement ratio* scores are met, a *program communication* score greater than 0.9 and a *critical depth* score lower than 0.2 indicate a circuit structure in which most qubits interact with each other, but there are little to no repeated interactions. On the other hand, a *program communication* score smaller than 0.15 together with a *critical depth* score larger than 0.9 indicate a circuit structure composed of many chained two-qubit interactions among the same pairs of qubits. In both cases the tensor network becomes a pseudo-regular grid, which can be efficiently contracted whilst keeping intermediate tensor sizes at bay, leading to time performance gains of up to one order of magnitude.

Program communication is especially important in determining whether a quantum circuit contraction can be efficiently contracted in a distributed sliced setting. Those circuits display superlinear speedups with respect to the available compute resources. Moreover, we showed how this is only justifiable to provide additional pathfinding resources to unbounded quantum circuit tensor contractions, as they can provide further speedups, whilst all other circuits can save on using additional resources on pathfinding.

6. Conclusions

The paper characterised the performance of a selected suite of relevant quantum circuits when simulated on state of the art simulator backends and high performance hardware. At first, the circuits have been characterised according to objective metrics that could describe their topological structure, to later correlate them to the performance of the simulators. The results point towards the fact that statevector simulators become heavily communication bound as soon as the size of the statevector exceeds that of the GPU's cache. The tensor network contraction has proven to perform better in circuits that have well distributed entanglement among the qubits, with a low overall number of two qubit gates in relation to the total number of gates.

It is reasonable to assume that by improving the memory access pattern of a state vector backend, one may improve the time performance of any benchmark to be simulated. The tensor network backend proved to be highly dependent on the efficacy of the pathfinding algorithm. The overall performance of this second backend can already outpace the state vector simulator in some of the benchmark problems, whilst

keeping a comparable, albeit slower performance in the remainder of the test runs. It is reasonable to assume that the development of further pathfinding optimisations for these harder to tackle topologies may push the performance of the tensor contraction backend to become the fastest simulation approach for quantum circuits, for example by limiting the growth of intermediate tensors during contraction. This would let us leverage the fact that the representation of tensor networks in memory enables the validation of larger quantum circuits and computers. Moreover, the promising scaling of distributed pathfinding and sliced contraction approaches over large tensor networks have the chance to further close the gap on simulating real quantum computer. We highlight the absence of a class of real-world quantum algorithms with parameterisable contraction complexity. The development of such a class of algorithms would provide means for measuring weak scaling performance of distributed tensor network contraction libraries, easing performance comparisons between quantum and classical systems. Future works could investigate the applicability of GPU accelerated pathfinding algorithms, so as to further improve the path quality and reduce the overall contraction time in unbounded problems.

CRediT authorship contribution statement

Marzio Vallero: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Paolo Rech:** Writing – review & editing, Supervision, Resources, Project administration, Investigation, Funding acquisition, Conceptualization. **Flavio Vella:** Writing – review & editing, Supervision, Resources, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Marzio Vallero reports equipment, drugs, or supplies was provided by Interuniversity Consortium Cineca. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors acknowledge financial support from ICSC (Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing), funded by the European Union – NextGenerationEU. This research has been

funded by the Italian Ministry for University and Research (MUR) through the “Departments of Excellence 2023-2027” program (L.232/-2016) awarded to the Department of Industrial Engineering. This project has been supported by Q@TN, the joint lab between the University of Trento, FBK (Fondazione Bruno Kessler), INFN (National Institute for Nuclear Physics) and CNR (National Research Council), with financial resources from PAT (Provincia Autonoma di Trento). We acknowledge the CINECA award under the ISCRA initiative, for the availability of high-performance computing resources and support. This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101175702.

Data availability

The repository containing all data and source code for this paper is available at <https://github.com/MarzioVallero/FGCS-state-of-practice>.

References

- [1] R.P. Feynman, Simulating physics with computers, *Internat. J. Theoret. Phys.* 21 (6/7) (1982) 467–488.
- [2] D. Horsman, S. Stepney, R.C. Wagner, V. Kendon, When does a physical system compute? *Proc. R. Soc. A: Math. Phys. Eng. Sci.* 470 (2169) (2014) [Online]. Available: <http://dx.doi.org/10.1098/rspa.2014.0182>.
- [3] L.K. Grover, A fast quantum mechanical algorithm for database search, 1996.
- [4] P.W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM J. Comput.* 26 (5) (1997) 1484–1509, [Online]. Available: <http://dx.doi.org/10.1137/s0097539795293172>.
- [5] E. Farhi, J. Goldstone, S. Gutmann, A quantum approximate optimization algorithm, 2014.
- [6] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P.J. Love, A. Aspuru-Guzik, J.L. O'Brien, A variational eigenvalue solver on a photonic quantum processor, *Nat. Commun.* 5 (2014) 4213.
- [7] R. Mullin, Let's talk about quantum computing in drug discovery, *C EN Glob. Enterp.* 98 (35) (2020) 20–22, [Online]. Available: <http://dx.doi.org/10.1021/cen-09835-feature2>.
- [8] A.P. Vepsäläinen, A.H. Karamlou, J.L. Orrell, A.S. Dogra, B. Loer, F. Vasconcelos, D.K. Kim, A.J. Melville, B.M. Niedzielski, J.L. Yoder, S. Gustavsson, J.A. Formaggio, B.A. VanDevender, W.D. Oliver, Impact of ionizing radiation on superconducting qubit coherence, *Nature* 584 (7822) (2020) 551–556, [Online]. Available: <http://dx.doi.org/10.1038/s41586-020-2619-8>.
- [9] C.D. Wilen, S. Abdullah, N.A. Kurinsky, C. Stanford, L. Cardani, G. D'Imperio, C. Tomei, L. Faoro, L.B. Ioffe, C.H. Liu, A. Opremcak, B.G. Christensen, J.L. DuBois, R. McDermott, Correlated charge noise and relaxation errors in superconducting qubits, *Nature* 594 (7863) (2021) 369–373, [Online]. Available: <http://dx.doi.org/10.1038/s41586-021-03557-5>.
- [10] M. McEwen, L. Faoro, K. Arya, A. Dunsworth, T. Huang, S. Kim, B. Burkett, A. Fowler, F. Arute, J.C. Bardin, A. Bengtsson, A. Bilmes, B.B. Buckley, N. Bushnell, Z. Chen, R. Collins, S. Demura, A.R. Derk, C. Erickson, M. Giustina, S.D. Harrington, S. Hong, E. Jeffrey, J. Kelly, P.V. Klimov, F. Kostritsa, P. Laptev, A. Locharla, X. Mi, K.C. Miao, S. Montazeri, J. Mutus, O. Naaman, M. Neeley, C. Neill, A. Opremcak, C. Quintana, N. Redd, P. Roushan, D. Sank, K.J. Satzinger, V. Shvarts, T. White, Z.J. Yao, P. Yeh, J. Yoo, Y. Chen, V. Smelyanskiy, J.M. Martinis, H. Neven, A. Megrant, N. Ioffe, R. Barends, Resolving catastrophic error bursts from cosmic rays in large arrays of superconducting qubits, *Nat. Phys.* 18 (1) (2021) 107–111, [Online]. Available: <http://dx.doi.org/10.1038/s41567-021-01432-8>.
- [11] G. Casagrande, M. Vallero, F. Vella, P. Rech, Understanding the contributions of terrestrial radiation sources to error rates in quantum devices, *IEEE Trans. Nucl. Sci.* 72 (4) (2025) 1324–1334.
- [12] Top500, 2023, [Online]. Available: <https://www.top500.org/news/frontier-remains-sole-exaflop-machine-and-retains-top-spot-improving-upon-its-previous-hpl-score/>.
- [13] T. Hoeffler, T. Häner, M. Troyer, Disentangling hype from practicality: On realistically achieving quantum advantage, *Commun. ACM* 66 (5) (2023) 82–87, [Online]. Available: <http://dx.doi.org/10.1145/3571725>.
- [14] S. Stanwyck, H. Bayraktar, T. Costa, Cuquantum: Accelerating quantum circuit simulation on GPUs, in: APS March Meeting Abstracts, in: APS Meeting Abstracts, vol. 2022, 2022, Q36.002.
- [15] Q.A. team, collaborators, Qsim, 2020, [Online]. Available: <http://dx.doi.org/10.5281/zenodo.4023103>.
- [16] G. Aleksandrowicz, T. Alexander, P. Barkoutsos, L. Bello, Y. Ben-Haim, D. Bucher, F.J. Cabrera-Hernández, J. Carballo-Franquis, A. Chen, C.-F. Chen, J.M. Chow, A.D. Córcoles-Gonzales, A.J. Cross, A. Cross, J. Cruz-Benito, C. Culver, S.D.L.P. González, E.D.L. Torre, D. Ding, E. Dumitrescu, I. Duran, P. Eendebak, M. Everitt, I.F. Sertage, A. Frisch, A. Fuhrer, J. Gambetta, B.G. Gago, J. Gomez-Mosquera, D. Greenberg, I. Hamamura, V. Havlicek, J. Hellmers, Łukasz Herok, H. Horii, S. Hu, T. Imamichi, T. Itoko, A. Javadi-Abhari, N. Kanazawa, A. Karazeev, K. Krulich, P. Liu, Y. Luh, Y. Maeng, M. Marques, F.J. Martín-Fernández, D.T. McClure, D. McKay, S. Meesala, A. Mezzacapo, N. Moll, D.M. Rodríguez, G. Nannicini, P. Nation, P. Ollitrault, L.J. O'Riordan, H. Paik, J. Pérez, A. Phan, M. Pistoia, V. Prutyantov, M. Reuter, J. Rice, A.R. Davila, R.H.P. Rudy, M. Ryu, N. Sathaye, C. Schnabel, E. Schoute, K. Setia, Y. Shi, A. Silva, Y. Siraichi, S. Sivarajah, J.A. Smolin, M. Soeken, H. Takahashi, I. Tavernelli, C. Taylor, P. Teylour, K. Trabing, M. Treinish, W. Turner, D. Vogt-Lee, C. Vuillot, J.A. Wildstrom, J. Wilson, E. Winston, C. Wood, S. Wood, S. Wörner, I.Y. Akhalwaya, C. Zoufal, Qiskit: An open-source framework for quantum computing, 2019, [Online]. Available: <http://dx.doi.org/10.5281/zenodo.2562111>.
- [17] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M.S. Alam, G. Alonso-Linaje, B. AkashNarayanan, A. Asadi, J.M. Arrazola, U. Azad, S. Banning, C. Blank, T.R. Bromley, B.A. Cordier, J. Ceroni, A. Delgado, O.D. Matteo, A. Dusko, T. Garg, D. Guala, A. Hayes, R. Hill, A. Ijaz, T. Isaacson, D. Ittah, S. Jahangiri, P. Jain, E. Jiang, A. Khandelwal, K. Kottmann, R.A. Lang, C. Lee, T. Loke, A. Lowe, K. McKiernan, J.J. Meyer, J.A. Montañez-Barrera, R. Moyalard, Z. Niu, L.J. O'Riordan, S. Oud, A. Panigrahi, C.-Y. Park, D. Polatajko, N. Quesada, C. Roberts, N. Sá, I. Schoch, B. Shi, S. Shu, S. Sim, A. Singh, I. Strandberg, J. Soni, A. Száva, S. Thabet, R.A. Vargas-Hernández, T. Vincent, N. Vitucci, M. Weber, D. Wierichs, R. Wiersema, M. Willmann, V. Wong, S. Zhang, N. Killoran, PennyLane: Automatic differentiation of hybrid quantum–classical computations, 2022.
- [18] G.F. Viamontes, Efficient Quantum Circuit Simulation (Ph.D. dissertation), University of Michigan, United States, USA, 2007, aAI3253425.
- [19] X. Xu, S. Benjamin, J. Sun, X. Yuan, P. Zhang, A herculean task: Classical simulation of quantum computers, 2023, [Online]. Available: <https://arxiv.org/abs/2302.08880>.
- [20] A. Ahmadzadeh, H. Sarbazi-Azad, Performance analysis and modeling for quantum computing simulation on distributed gpu platforms, *Quantum Inf. Process.* 23 (11) (2024) 373, [Online]. Available: <http://dx.doi.org/10.1007/s1128-024-04580-x>.
- [21] J. Doi, H. Horii, C. Wood, Efficient techniques to gpu accelerations of multi-shot quantum computing simulations, 2023, [Online]. Available: <https://arxiv.org/abs/2308.03399>.
- [22] E. Gutiérrez, S. Romero, M.A. Trenas, E.L. Zapata, Quantum computer simulation using the cuda programming model, *Comput. Phys. Comm.* 181 (2) (2010) 283–300, [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0010465509003117>.
- [23] A. Amariutei, S. Caraiman, Parallel quantum computer simulation on the gpu, in: 15th International Conference on System Theory, Control and Computing, 2011, pp. 1–6.
- [24] A. Avila, A. Maron, R. Reiser, M. Pilla, A. Yamin, Gpu-aware distributed quantum simulation, in: Proceedings of the 29th Annual ACM Symposium on Applied Computing, in: SAC '14, Association for Computing Machinery, New York, NY, USA, 2014, pp. 860–865, [Online]. Available: <http://dx.doi.org/10.1145/2554850.2554892>.
- [25] S. Heng, T. Kim, Y. Han, Exploiting gpu-based parallelism for quantum computer simulation: A survey, *IEIE Trans. Smart Process. Comput.* 9 (6) (2020) 468–476.
- [26] A.J. Gangapuram, A.M. Läuchli, C. Hempel, Benchmarking quantum computer simulation software packages: State vector simulators, *SciPost Phys. Core* 7 (2024) 075, [Online]. Available: <https://scipost.org/10.21468/SciPostPhysCore.7.4.075>.
- [27] J. Faj, I. Peng, J. Wahlgren, S. Markidis, Quantum computer simulations at warp speed: Assessing the impact of gpu acceleration: A case study with ibm qiskit aer, nvidia thrust and cuquantum, in: 2023 IEEE 19th International Conference on E-Science, E-Science, 2023, pp. 1–10.
- [28] J. Doi, H. Takahashi, R. Raymond, T. Imamichi, H. Horii, Quantum computing simulator on a heterogenous hpc system, in: Proceedings of the 16th ACM International Conference on Computing Frontiers, in: CF '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 85–93, [Online]. Available: <http://dx.doi.org/10.1145/3310273.3323053>.
- [29] C. Jiao, W. Zhang, L. Shen, Communication optimizations for state-vector quantum simulator on cpu+gpu clusters, in: Proceedings of the 52nd International Conference on Parallel Processing, in: ICPP '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 203–212, [Online]. Available: <http://dx.doi.org/10.1145/3605573.3605631>.
- [30] D. Lykov, A. Chen, H. Chen, K. Keipert, Z. Zhang, T. Gibbs, Y. Alexeev, Performance evaluation and acceleration of the qtensor quantum circuit simulator on gpus, in: 2021 IEEE/ACM Second International Workshop on Quantum Computing Software, QCS, 2021, pp. 27–34.

- [31] F. Pan, H. Gu, L. Kuang, B. Liu, P. Zhang, Efficient quantum circuit simulation by tensor network methods on modern gpus, *ACM Trans. Quantum Comput.* 5 (4) (2024) [Online]. Available: <http://dx.doi.org/10.1145/3696465>.
- [32] A. Einstein, B. Podolsky, N. Rosen, Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.* 47 (1935) 777–780, [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRev.47.777>.
- [33] J.S. Bell, On the einstein podolsky rosen paradox, *Phys. Phys. Fiz.* 1 (1964) 195–200, [Online]. Available: <https://link.aps.org/doi/10.1103/PhysicsPhysiqueFizika.1.195>.
- [34] G. Vidal, Efficient classical simulation of slightly entangled quantum computations, *Phys. Rev. Lett.* 91 (14) (2003) [Online]. Available: <http://dx.doi.org/10.1103/physrevlett.91.147902>.
- [35] B. Coecke, R. Duncan, Interacting quantum observables: categorical algebra and diagrammatics, *New J. Phys.* 13 (4) (2011) 043016, [Online]. Available: <http://dx.doi.org/10.1088/1367-2630/13/4/043016>.
- [36] B. Fang, M.Y. Özkaya, A. Li, Ümit V. Çatalyürek, S. Krishnamoorthy, Efficient hierarchical state vector simulation of quantum circuits via acyclic graph partitioning, 2022.
- [37] J. Gray, S. Kourtis, Hyper-optimized tensor network contraction, *Quantum* 5 (2021) 410, [Online]. Available: <http://dx.doi.org/10.22331/q-2021-03-15-410>.
- [38] X.-C. Wu, S. Di, E.M. Dasgupta, F. Cappello, H. Finkel, Y. Alexeev, F.T. Chong, Full-state quantum circuit simulation by using data compression, in: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, in: SC '19, Association for Computing Machinery, New York, NY, USA, 2019, [Online]. Available: <http://dx.doi.org/10.1145/3295500.3356155>.
- [39] G. Kalachev, P. Pantelev, M.-H. Yung, Multi-tensor contraction for xeb verification of quantum circuits, 2022, [Online]. Available: <https://arxiv.org/abs/2108.05665>.
- [40] C. Guo, Y. Zhao, H.-L. Huang, Verifying random quantum circuits with arbitrary geometry using tensor network states algorithm, *Phys. Rev. Lett.* 126 (2021) 070502, [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.126.070502>.
- [41] Y. Liu, Y. Chen, C. Guo, J. Song, X. Shi, L. Gan, W. Wu, W. Wu, H. Fu, X. Liu, D. Chen, Z. Zhao, G. Yang, J. Gao, Verifying quantum advantage experiments with multiple amplitude tensor network contraction, *Phys. Rev. Lett.* 132 (3) (2024) [Online]. Available: <http://dx.doi.org/10.1103/PhysRevLett.132.030601>.
- [42] Planck Collaboration, P.A.R. Ade, N. Aghanim, M. Arnaud, M. Ashdown, J. Aumont, C. Baccigalupi, A.J. Banday, R.B. Barreiro, J.G. Bartlett, N. Bartolo, E. Battaner, R. Battye, K. Benabed, A. Benoît, A. Benoit-Lévy, J.P. Bernard, M. Bersanelli, P. Bielewicz, J.J. Bock, A. Bonaldi, L. Bonavera, J.R. Bond, J. Borrill, F.R. Bouchet, F. Boulanger, M. Bucher, C. Burigana, R.C. Butler, E. Calabrese, J.F. Cardoso, A. Catalano, A. Challinor, A. Chamballu, R.R. Chary, H.C. Chiang, J. Chluba, P.R. Christensen, S. Church, D.L. Clements, S. Colombi, L.P.L. Colombo, C. Combet, A. Couais, B.P. Crill, A. Curto, F. Cuttaia, L. Danese, R.D. Davies, R.J. Davis, P. de Bernardis, A. de Rosa, G. de Zotti, J. Delabrouille, F.X. Désert, E. Di Valentino, C. Dickinson, J.M. Diego, K. Dolag, H. Dole, S. Donzelli, O. Doré, M. Douspis, A. Ducout, J. Dunkley, X. Dupac, G. Efstathiou, F. Elsner, T.A. Enßlin, H.K. Eriksen, M. Farhang, J. Fergusson, F. Finelli, O. Forni, M. Frailis, A.A. Fraisse, E. Franceschi, A. Frejsel, S. Galeotta, S. Galli, K. Gangui, C. Gauthier, M. Gerbino, T. Ghosh, M. Giard, Y. Giraud-Héraud, E. Giusarma, E. Gjerløw, J. González-Nuevo, K.M. Górski, S. Gratton, A. Gregorio, A. Gruppiso, J.E. Gudmundsson, J. Hamann, F.K. Hansen, D. Hanson, D.L. Harrison, G. Helou, S. Henrot-Versillé, C. Hernández-Monteagudo, D. Herranz, S.R. Hildebrandt, E. Hivon, M. Hobson, W.A. Holmes, A. Hornstrup, W. Hovest, Z. Huang, K.M. Huffenberger, G. Hurier, A.H. Jaffe, T.R. Jaffe, W.C. Jones, M. Juvela, E. Keihänen, R. Keskitalo, T.S. Kisner, R. Kneissl, J. Knoche, L. Knox, M. Kunz, H. Kurki-Suonio, G. Lagache, A. Lähteenmäki, J.M. Lamarre, A. Lasenby, M. Lattanzi, C.R. Lawrence, J.P. Leahy, R. Leonardi, J. Lesgourgues, F. Levrier, A. Lewis, M. Liguori, P.B. Lilje, M. Linden-Vornle, M. López-Cañiego, P.M. Lubin, J.F. Macías-Pérez, G. Maggio, D. Maino, N. Mandolesi, A. Mangilli, A. Marchini, M. Maris, P.G. Martin, M. Martinelli, E. Martínez-González, S. Masi, S. Matarrese, P. McGehee, P.R. Meinhold, A. Melchiorri, J.B. Melin, L. Mendes, A. Mennella, M. Migliaccio, M. Millea, S. Mitra, M.A. Miville-Deschênes, A. Moneti, L. Montier, G. Morgante, D. Mortlock, A. Moss, D. Munshi, J.A. Murphy, P. Naselsky, F. Nati, P. Natoli, C.B. Netterfield, H.U. Nørgaard-Nielsen, F. Novello, D. Novikov, I. Novikov, C.A. Oxborrow, F. Paci, L. Pagano, F. Pajot, R. Paladini, D. Paoletti, B. Partridge, F. Pasian, G. Patanchon, T.J. Pearson, O. Perdereau, L. Perotto, F. Perrotta, V. Pettorino, F. Piacentini, M. Piat, E. Pierpaoli, D. Pietrobon, S. Plaszczynski, E. Pointecouteau, G. Polenta, L. Popa, G.W. Pratt, G. Prézeau, S. Prunet, J.L. Puget, J.P. Rachen, W.T. Reach, R. Rebolo, M. Reinecke, M. Remazeilles, C. Renault, A. Renzi, I. Ristorcelli, G. Rocha, C. Rosset, M. Rossetti, G. Roudier, B. Rouillé d'Orfeuil, M. Rowan-Robinson, J.A. Rubio-Martín, B. Rusholme, N. Said, V. Salvatelli, L. Salvati, M. Sandri, D. Santos, M. Savelainen, G. Savini, D. Scott, M.D. Seiffert, P. Serra, E.P.S. Shellard, L.D. Spencer, M. Spinelli, V. Stolyarov, R. Stompor, R. Sudiwala, R. Sunyaev, D. Sutton, A.S. Suur-Uski, J.F. Sygnet, J.A. Tauber, L. Terenzi, L. Toffolatti, M. Tomasi, M. Tristram, T. Trombetti, M. Tucci, J. Tuovinen, M. Türlér, G. Umana, L. Valenziano, J. Valiviita, F. Van Tent, P. Vielva, F. Villa, L.A. Wade, B.D. Wandelt, I.K. Wehus, M. White, S.D.M. White, A. Wilkinson, D. Yvon, A. Zacchei, A. Zonca, Planck 2015 results. xiii. cosmological parameters, *Astron. Astrophys.* 594 (2016) A13.
- [43] J. Ha, J. Lee, J. Heo, Resource analysis of quantum computing with noisy qubits for shor's factoring algorithms, *Quantum Inf. Process.* 21 (2) (2022) 60, [Online]. Available: <http://dx.doi.org/10.1007/s11128-021-03398-1>.
- [44] F. Arute, K. Arya, R. Babbush, D. Bacon, J.C. Bardin, R. Barends, R. Biswas, S. Boixo, F.G.S.L. Brandao, D.A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, K. Guerin, S. Habegger, M.P. Harrigan, M.J. Hartmann, A. Ho, M. Hoffmann, T. Huang, T.S. Humble, S.V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P.V. Klimov, S. Knysh, A. Korotkov, F. Kostritsa, D. Landhuis, M. Lindmark, E. Lucero, D. Lyakh, S. Mandrà, J.R. McClean, M. McEwen, A. Megrant, X. Mi, K. Michielsen, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, C. Neill, M.Y. Niu, E. Ostby, A. Petukhov, J.C. Platt, C. Quintana, E.G. Rieffel, P. Roushan, N.C. Rubin, D. Sank, K.J. Satzinger, V. Smelyanskiy, K.J. Sung, M.D. Trevithick, A. Vainsencher, B. Villalonga, T. White, Z.J. Yao, P. Yeh, A. Zalcman, H. Neven, J.M. Martinis, Quantum supremacy using a programmable superconducting processor, *Nature* 574 (7779) (2019) 505–510, [Online]. Available: <http://dx.doi.org/10.1038/s41586-019-1666-5>.
- [45] D. Lykov, R. Shaydulin, Y. Sun, Y. Alexeev, M. Pistoia, Fast simulation of high-depth qaoa circuits, in: *Proceedings of the SC '23 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis*, in: SC-W '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 1443–1451, [Online]. Available: <http://dx.doi.org/10.1145/3624062.3624216>.
- [46] M. Morita, Y. Tomita, J. Koyama, K. Kimura, Simulator demonstration of large scale variational quantum algorithm on hpc cluster, *IEEE Access* 12 (2024) 85219–85230.
- [47] T. Jones, B. Koczor, S.C. Benjamin, Distributed simulation of statevectors and density matrices, 2023, [Online]. Available: <https://arxiv.org/abs/2311.01512>.
- [48] Y. Kimura, S. Li, H. Sato, M. Fujita, Decision diagram vs. state vector: A comparative study on quantum computing simulation efficiency, in: *2024 IEEE International Conference on Quantum Computing and Engineering, QCE, Vol. 01*, 2024, pp. 757–763.
- [49] I.L. Markov, Y. Shi, Simulating quantum computation by contracting tensor networks, *SIAM J. Comput.* 38 (3) (2008) 963–981, [Online]. Available: <http://dx.doi.org/10.1137/050644756>.
- [50] M.X. Goemans, D.P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, *J. ACM* 42 (6) (1995) 1115–1145, [Online]. Available: <http://dx.doi.org/10.1145/227683.227684>.
- [51] F. Pan, P. Zhang, Simulation of quantum circuits using the big-batch tensor network method, *Phys. Rev. Lett.* 128 (2022) 030501, [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.128.030501>.
- [52] R. Schutski, T. Khakhulin, I. Oseledets, D. Kolmakov, Simple heuristics for efficient parallel tensor contraction and quantum circuit simulation, *Phys. Rev. A* 102 (2020) 062614, [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.102.062614>.
- [53] E. Pednault, J. Gunnels, G. Nannicini, L. Horesh, T. Magerlein, E. Solomonik, R. Wisniew, Breaking the 49-qubit barrier in the simulation of quantum circuits, *Lawrence Livermore Natl. Lab. J.* 10 (2017).
- [54] J. Biamonte, V. Bergholm, Tensor networks in a nutshell, 2017.
- [55] J. Biamonte, Lectures on quantum tensor networks, 2020.
- [56] T. Tomesh, P. Gokhale, V. Omole, G.S. Ravi, K.N. Smith, J. Vizslai, X.-C. Wu, N. Hardavellas, M.R. Martonosi, F.T. Chong, Supermarq: A scalable quantum benchmark suite, 2022.
- [57] A. Li, S. Stein, S. Krishnamoorthy, J. Ang, Qasmbench: A low-level qasm benchmark suite for nisq evaluation and simulation, 2022.
- [58] A.Y. Kitaev, Quantum measurements and the abelian stabilizer problem, 1995.
- [59] D. Coppersmith, An approximate fourier transform useful in quantum factoring, 2002.
- [60] W. van Dam, S. Hallgren, L. Ip, Quantum algorithms for some hidden shift problems, 2002.
- [61] E. Bernstein, U. Vazirani, Quantum complexity theory, *SIAM J. Comput.* 26 (5) (1997) 1411–1473, [Online]. Available: <http://dx.doi.org/10.1137/S0097539796300921>.
- [62] I.D. Kivlichan, J. McClean, N. Wiebe, C. Gidney, A. Aspuru-Guzik, G.K.-L. Chan, R. Babbush, Quantum simulation of electronic structure with linear depth and connectivity, *Phys. Rev. Lett.* 120 (11) (2018) [Online]. Available: <http://dx.doi.org/10.1103/physrevlett.120.110501>.
- [63] Anonymous, Data and code repository to be disclosed after the review process, 2025.
- [64] E. Pednault, J.A. Gunnels, G. Nannicini, L. Horesh, T. Magerlein, E. Solomonik, E.W. Draeger, E.T. Holland, R. Wisniew, Pareto-efficient quantum circuit simulation using tensor contraction deferral, 2020.