



Kent Academic Repository

Pérez-Delgado, Carlos A and Vinjanampathy, Sai (2021) *Coherent Parallelization of Universal Classical Computation*. New Journal of Physics, 23 . ISSN 1367-2630.

Downloaded from

<https://kar.kent.ac.uk/92288/> The University of Kent's Academic Repository KAR

The version of record is available from

<https://doi.org/10.1088/1367-2630/ac3a17>

This document version

Publisher pdf

DOI for this version

Licence for this version

CC BY (Attribution)

Additional information

Versions of research works

Versions of Record

If this version is the version of record, it is the same as the published version available on the publisher's web site. Cite as the published version.

Author Accepted Manuscripts

If this document is identified as the Author Accepted Manuscript it is the version after peer review but before type setting, copy editing or publisher branding. Cite as Surname, Initial. (Year) 'Title of article'. To be published in *Title of Journal*, Volume and issue numbers [peer-reviewed accepted version]. Available at: DOI or URL (Accessed: date).

Enquiries

If you have questions about this document contact ResearchSupport@kent.ac.uk. Please include the URL of the record in KAR. If you believe that your, or a third party's rights have been compromised through this document please see our [Take Down policy](https://www.kent.ac.uk/guides/kar-the-kent-academic-repository#policies) (available from <https://www.kent.ac.uk/guides/kar-the-kent-academic-repository#policies>).

PAPER • OPEN ACCESS

Coherent parallelization of universal classical computation

To cite this article: Carlos A Perez-Delgado and Sai Vinjanampathy 2021 *New J. Phys.* **23** 123015

View the [article online](#) for updates and enhancements.

You may also like

- [Spectrometry of pulsed photon radiation](#)
Rolf Behrens, Hayo Zutz and Julian Busse
- [Dynamic vortex Mott transition in triangular superconducting arrays](#)
Pei Zi-Xi, Guo Wei-Gui and Qiu Xiang-Gang
- [The HolmiumHydrogen System](#)
F. C. Perkins and C. E. Lundin



PAPER

Coherent parallelization of universal classical computation

OPEN ACCESS

RECEIVED
12 May 2021REVISED
1 November 2021ACCEPTED FOR PUBLICATION
16 November 2021PUBLISHED
8 December 2021

Original content from
this work may be used
under the terms of the
[Creative Commons
Attribution 4.0 licence](#).

Any further distribution
of this work must
maintain attribution to
the author(s) and the
title of the work, journal
citation and DOI.

Carlos A Perez-Delgado^{1,*} and Sai Vinjanampathy^{2,3} ¹ School of Computing, University of Kent, Canterbury, Kent CT2 7NF, United Kingdom² Department of Physics, Indian Institute of Technology Bombay, Powai, Mumbai-400076, India³ Centre for Quantum Technologies, National University of Singapore, 3 Science Drive 2, 117543, Singapore

* Author to whom any correspondence should be addressed.

E-mail: c.perez@kent.ac.uk and sai@phy.iitb.ac.in

Keywords: quantum advantage, quantum computation, super-Heisenberg metrology, quantum batteries

Abstract

Previously, higher-order Hamiltonians (HoH) had been shown to offer an advantage in both metrology and quantum energy storage. Here, we axiomatize a model of computation that allows us to consider such Hamiltonians for the purposes of computation. From this axiomatic model, we formally prove that an HoH-based algorithm can gain up to a quadratic speed-up over classical sequential algorithms—for any possible classical computation. We show how our axiomatic model is grounded in the same physics as that used in HoH-based quantum advantage for metrology and battery charging. Thus we argue that any advance in implementing HoH-based quantum advantage in those scenarios can be co-opted for the purpose of speeding up computation.

1. Introduction

Quantum many-body interactions have been shown to be able to provide an advantage over classical and quantum processes limited to one- and two-body interactions. This advantage relies on higher order Hamiltonian (HoH) interactions and has been studied both theoretically and experimentally in the context of quantum metrology [1–4], and quantum energy storage [5–8]. It is a matter of ongoing debate whether higher order Hamiltonians can, in fact, be harnessed *effectively* as to provide quantum advantage. The argument against revolves around whether HoH can occur naturally, or whether they can be engineered without having to pay a hefty resource cost. On the other hand, systems that behave (at least to some order of approximation) as having a HoH *have been* shown experimentally. A notable example is that of the well-known *Dicke Hamiltonian* [9–11]. Based on this Dicke Hamiltonian, Ferraro *et al* built an experimental proposal that results in HoH-based quantum advantage for battery charging [7].

It is not our purpose here to argue either position, or even to take a (strong) side. Rather, in this manuscript we consider the implications of higher order Hamiltonians for computation. We demonstrate a method called *coherent parallelization* (CP) that can speed-up universal classical computation by up to a factor that is quadratic in the size of the input. CP is very similar to traditional parallelization on a classical computer. Firstly, it does not change the *total gate complexity* of a problem. Instead—similar to standard parallelization—CP reduces the amortized time-cost [12] of each individual gate. Secondly, it is a generally applicable method that can be used on *all* problems/algorithms. Just like standard parallelization, again, not all algorithms benefit equally from CP: the advantage CP provides depends on the algorithm's level of parallelizability. In contrast to classical parallelization which can provide up to a linear advantage over traditional sequential algorithms, CP can provide up to a quadratic advantage.

Moving forward, we proceed with the assumption that engineering (with feasible costs) HoH-based systems *is possible at least in principle*, and shall focus on discussing what the implications of such an assumption are for *computation*.

When considering HoH interactions for computation, it is important to consider two central questions. The first question is whether a measure of algorithmic complexity remains meaningful within the context of HoH interactions. The second related question is whether such HoH interactions are physically realizable. To address the first question, we present a model of computation that properly limits the type of

multi-qubit gates our computation is allowed to use. Intuitively our model is to quantum computers what a parallel processor (e.g. a GPU) is to classical computers. An (ideal, theoretical) parallel processor can operate on an arbitrary number of bits at the same time in parallel. However, the gates it can apply in one time-step are limited to parallel versions of basic gates. For example, a parallel processor acting on $2n$ bits *cannot* perform a distinct and arbitrary operation on each of the $2n$ bits. It can however perform a bit-wise XOR operation between the first n bits and the second n bits. Similar restrictions apply in our model.

With regards to implementability, our computational model allows us to limit our attention to physical Hamiltonians that are the most likely to be implementable. We can limit our model to only allow the same type of HoHs as considered in other quantum technologies such as quantum sensors [1–4] and quantum batteries. Doing this, allows us to implement a sped-up universal computation using *only* the same type of HoH interactions used for metrology and/or battery charging. For example, as we discuss later, we can co-opt the same *Dicke Hamiltonian* proposed for speeding up battery charging, to speed computation instead. Following this, we axiomatize the HoH interactions in our computational model and show that the computational speedup achieved depends on the algorithm under consideration and the nature of the HoH.

In the following section we give an overview of the established literature on HoH advantage as it has been applied to the fields of quantum metrology, and quantum energy-storage. In the subsequent two sections we extend these results to the realm of computation. We conclude with a discussion on the relevance and implications of our results.

2. Prior results on higher order Hamiltonians

We begin with a comparison of the effect of dynamical correlations generated by an initially entangled state vs HoH interactions. It is well established that entanglement in the input state allows a measurement to beat the so-called shot noise limit in quantum metrology [13, 14]. This shot noise limit states that the variance of an unbiased estimator scales as $\langle N \rangle^{-1}$, where $\langle N \rangle$ is the average number of photons in the system. Entangled states with linear Hamiltonians were shown to beat this limit and improve the variance to scale no better than the standard quantum limit $\langle N \rangle^{-1}$ under various physical conditions [15]. The quantum advantage derived from HoH is distinctly different [1–4] from the aforementioned standard quantum limit, since it can potentially do better than this limit [16]. Furthermore, such an advantage was recently applied and extended to the task of charging quantum batteries [5, 6] demonstrating a quadratic advantage over classical charging protocols. We review this method briefly.

There is an upper bound on the speed by which quantum evolution can occur known as the quantum speed limit (QSL). For a closed, finite-dimensional system we define the semi-norm of the Hamiltonian as

$$p(H) = |\langle H - h_{\min} I \rangle| = h_{\max} - h_{\min}. \quad (1)$$

QSL states that the time τ to transform any quantum state ρ_0 to another state ρ_f under a physical map is no smaller than τ_{QSL} , given by

$$\tau \geq \tau_{\text{qsl}} := \mathcal{L}(\rho_0, \rho_f) \max \left(\frac{1}{E}, \frac{1}{\Delta E} \right). \quad (2)$$

Here $\mathcal{L}(\rho_0, \rho_f)$ is the Bures angle between the two states, $E = \tau^{-1} \int_0^\tau dt \langle H(t) - h_{\min} \rangle$ is related to the average energy [17] and $\Delta E = \tau^{-1} \int_0^\tau dt \Delta H(t)$ is related to the average standard deviation of energy [18]. Here h_{\min} is the instantaneous ground state of the time-dependent Hamiltonian and $\Delta H(t)$ is the instantaneous standard deviation of the energy. Hence QSL gives a direct trade-off between the semi-norm $p(H)$ and the minimum time it requires to perform a particular evolution using that Hamiltonian. This should match with canonical intuition: the greater the energy in the system, the faster it can evolve. To exemplify this tradeoff explicitly, consider the transition between the eigenstates of a non-degenerate qubit. This can be implemented quantum mechanically by $U = \sigma_x$. The unitary operator in turn can be implemented by setting the Hamiltonian of the qubit system to $H = \sigma_x$ for a time $t = \pi/2$. If instead we use the Hamiltonian $H = 2\sigma_x$, then we reduce the time needed to complete the evolution of an NOT gate in half. In general the time t scales in inverse proportion to the semi-norm $p(H)$.

Hence $p(H)$ is a measure the effectiveness of an algorithm and we will exploit this in the next section. Given this, the comparison of two battery charging protocols must necessarily be done by holding the semi-norm constant. Any relative advantage that still survives by holding the energetic resources constant can then be reasonably ascribed to dynamical correlations.

The traditional way of charging two batteries in parallel is to apply the unitary operator $\sigma_x = e^{-i\pi/2\sigma_x}$ to each individual qubit. Collectively the system's Hamiltonian is set to $H_{\parallel} = \sigma_x \otimes I + I \otimes \sigma_x$, and evolved for time $t = \pi/2$. Besides this parallel protocol, another way to implement this joint operation is in a coherent fashion. Instead of the Hamiltonian used above we could employ $H_{\#} = \sigma_x \otimes \sigma_x$. Note that in

both cases the system needs to be evolved under the appropriate Hamiltonian for time $t = \pi/2$. However, as previously discussed, $p(H)$ is a resource for state transformations (and hence computation) since the quantum speed limit t_{QSL} bound depends on this quantity. Hence to fairly compare the parallel and the coherent strategy it is necessary to fix the resources, namely $p(H)$. Since $p(H_{\parallel}) = 2$ whereas $p(H_{\#}) = 1$, we can scale $H'_{\#} = 2H_{\#}$, and stay within the same norm limit τ_{QSL} as the parallel implementation H_{\parallel} . Therefore, $H'_{\#}$ implements both NOT gates in half the time that H_{\parallel} requires.

Moreover, this argument *scales*. Charging N different quantum batteries, using a coherent approach $H_{\#}(N) = \bigotimes_N \sigma_x$ is N times faster than using a parallel Hamiltonian $H_{\parallel}(N) = \sum_N \sigma_x^{(N)}$. Given that the *naive* parallel approach is N times faster than a sequential approach, this makes the quantum coherent approach N^2 times faster than classical, sequential, battery charging [5, 6].

Before moving on to our generalization of this approach to universal computation, it is worth discussing its merits. First, it is essential to note that the demand above to hold semi-norm constant is not merely some mathematical contrivance. Energy, it has been known for some time, is *the most* fundamental of resources. A bound on energy puts a bound on space and matter, which in turn puts a bound on any other resource one may consider (including, say, computational gates). From an experimental perspective greater energetic requirements imply a host of other issues including engineering larger couplings [19–22], increased bandwidth and amplitude requirement from controls and heating of the experimental sample [23]. Hence placing a bound on the operator semi-norm is not just mathematically consistent, but it is also physically essential to assess quantum advantage.

Second, as already noted, we wish to only allow our computational model to use the same HoHs as those used in improved metrology [1–3], and improved battery-charging [6].

In order to move our discussion to universal classical computation, consider that charging a quantum battery system precisely coincides with implementing an NOT gate on the subspace spanned by the least and top energy eigenstates. We will formalize and exploit the intuition that charging N batteries amounts to applying N NOT gates in the next section.

3. Uniform Hamiltonian model of computation

We begin this section by introducing our model of computation. There exist many models of computation, both classical and quantum. While all models share the same computational power (in terms of *what* they can compute), each model has different computational cost functions. These costs depend on the fundamental *axioms* of each model. An example of this is the difference between *sequential* and *parallel* models of computation. A sequential model of computation, such as the circuit model or the random access machine (RAM) [24], quantifies the cost of an algorithm solely as a function of the number of logical gates (itself as a function of the input size) the algorithm needs to perform to solve a problem. A parallel model of computation, like the PRAM [25], instead assumes multiple gates can be performed at the same time, and quantifies the total *amortized* [12] time cost of implementing all the gates of a given algorithm.

Different computational models do not just provide different axiomatic frameworks by which to analyze algorithms, however. Many of these models have a direct correspondence to a set of physical devices that behave in much the same way as described by the model. Following the example above, sequential computational models describe how many CPUs work, whereas a PRAM better describes GPUs and many modern CPUs.

In the same vein, our main focus here will be the axioms of our computational model, and their logical consequences. We will, however, argue that our model's axioms are fundamentally and soundly based on existing physics. Furthermore, we will show the connection between our axiomatic model and existing physical systems, both natural and engineered, that can potentially implement the computational advantages we describe herein.

The first key axiom of our model is the use of *action* (energy times time) as the fundamental yardstick by which to measure the efficiency of algorithms. By this we mean energy as measured in *joules* and time as measured in *seconds*. This has several advantages. The first and most obvious one is that this choice is quite clearly *not arbitrary*. Time and energy are fairly universally considered throughout all branches of physics as fundamental. The efficiency of processes throughout physics are gauged in terms of their energy and time consumptions.

Another important advantage is that, through careful definitions, we can recover algorithmic complexity in terms of gates within our energy/time-based model, and hence regain *all* the results therein. Furthermore, the model can switch between sequential and parallel computation by simply tuning a single parameter (the upper bound τ on the semi-norm as defined below). Finally, this model will allow us to consider and discuss computation using Hamiltonians as building blocks. This in turn will allow us to consider the advantages of using higher-order Hamiltonians as discussed in the previous section.

Our model of computation consists of an input/output system, a battery system, and a control system. The input/output system is simply where the input is stored at the beginning of the computation, and where the output is deposited at the end. We consider it to be a system somewhat external to the computation, having its own (arbitrary) Hamiltonian. The computation itself is driven by the control system. This system has its own Hamiltonian—whose study is central in our analysis. It draws energy from the computer's battery system, to drive its computation. Bounding the amount of energy the computer is allowed to draw at any single point allows us to obtain a meaningful sense of algorithmic complexity.

In the context of computation, if H is the Hamiltonian driving the evolution of a quantum register during the implementation of a quantum gate, increasing the operator semi-norm $p(H)$ can arbitrarily speed up literally any computation [41]. Hence, in order for time complexity of algorithms to remain meaningful within our model we must set a limit $p(H(t)) \leq \tau$, where $\tau(n)$ is either constant, or is allowed to scale at most linearly in the size of the input n . Setting $\tau(n)$ to a constant will make our computer scale similarly to strictly sequential classical and quantum computers. On the other hand, letting $\tau(n)$ to scale at most linearly in the size of the input n , allows our idealized computer to model classical and quantum parallel machines.

Next, we discuss the second central axiom of our model. In order for algorithms and their complexity to be meaningful within our model it is not sufficient to just have an energy bound on the internal Hamiltonian of the computation. We also require these Hamiltonians to be *uniform*. What this means, intuitively, is that the Hamiltonian is not allowed to be completely arbitrary (within its energy bounds), but must rather look uniform in space. This is similar to how parallel/concurrent algorithms and models of computation [such as (quantum) cellular automata [26]] work: an operator can be applied repeatedly throughout the computer's memory-state, but it must be *the same* operator throughout. For example, a parallel quantum processor can apply an NOT gate to an unbounded size set of (quantum) bits in its internal memory, but it *cannot* apply different arbitrary gates to an arbitrary set (and number) of bits.

Formally, in our model we constrain the internal computational Hamiltonian $H(t)$, at any time t' , to be of the form:

$$H(t') = \sum_i^m \bigotimes_j^r \mathbf{H}^{d(i,j)}(t'), \quad (3)$$

where $\mathbf{H}(t')$ is a single, or k -qubit Hamiltonian, $d(i, j)$ is either zero or one, $r = \lfloor n/k \rfloor$, where n is the total number of qubits in the system. What the above definition means is that at any point our computer can apply, say, an NOT gate to any number of qubits. By allowing $\mathbf{H}(t')$ to be k -qubit Hamiltonian means our computer can also, for example, apply a series of CNOT or Toffoli gates to arbitrary sets of qubits. The constraint, however, also means that our computer cannot apply an arbitrary gate, to an arbitrary qubit. The uniformity constraint within our model mirrors precisely the same constraint in other parallel computational models, both classical [25] and quantum [26].

A complete formal presentation of our computational model is given in the [appendix](#).

4. Coherent parallelization of classical algorithms

In the previous section we discussed how, in order for time complexity of algorithms to remain meaningful within our model, we must set a limit $p(H(t)) \leq \tau$, where $\tau(n)$ is a constant that can scale at most linearly in the size of the input n . Within this semi-norm constraint, we will discuss how we can make use of quantum correlations generated by non-linear Hamiltonians to speed up computation. We first discuss an example of CP applied to the NOT gate, followed by the formal theorem.

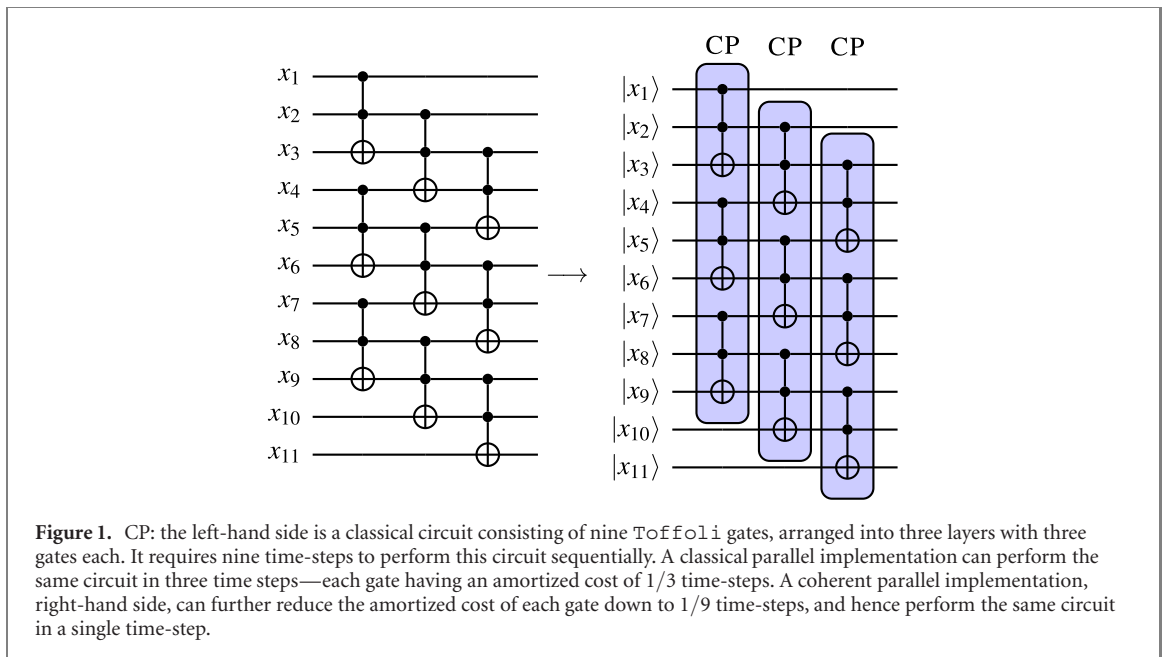
The traditional way of implementing two instances of the NOT gate in parallel is to apply the unitary operator $\sigma_x = e^{-i\pi/2\sigma_x}$ to each individual qubit. Collectively the system's Hamiltonian is set to $H_{\parallel} = \sigma_x \otimes I + I \otimes \sigma_x$, and evolved for time $t = \pi/2$. Using the results from quantum batteries [6] discussed in the previous section, we know that holding the semi-norm constant, this process can be 'coherently parallelized' by the Hamiltonian $2\sigma_x \otimes \sigma_x$ which executes this transformation in half the time (figure 1).

In the following two results we generalize and extend this result to *all* operators that are both unitary and Hermitian (see the [appendix](#) for proofs).

Lemma 1. *Let H be a Hermitian, unitary operator. Then, for any positive integer m :*

$$p(H_{\parallel}(m)) = mp(H) \quad (4)$$

$$p(H_{\#}(m)) = p(H). \quad (5)$$



Here $H_{\parallel}(m)$ ($H_{\#}(m)$) refers to the standard (coherent) parallel implementation of m copies of the unitary, Hermitian operator H . The following theorem follows directly from the previous lemma.

Theorem 1 (Coherent parallelization). *Let H be any Hermitian unitary gate acting on a d -dimensional system or qudit. Implementing m gates in parallel using a standard parallel computation implementation H_{\parallel} is m times slower than using a CP approach $H_{\#}$.*

An important observation about lemma 1 (and thus the statement of theorem 1) is that it is *not* merely an upper bound on the speed-up, but rather a strict equality, and thus acts as both an upper and a lower bound. In other words, the lemma gives an upper bound for how fast a classical, or non-coherent, parallel process can be, and then states that a coherent process can be faster by a factor of m .

The Toffoli gate is both a unitary and Hermitian operator. Moreover, it is universal for (reversible) classical computation. Hence, any classical reversible computation on n bits can be implemented as a circuit consisting of d layers, each consisting of $1 \leq m \leq n$ Toffoli gates. At each layer (time-step) one may choose to implement the Toffoli gates sequentially, in parallel, or coherently together using the method we described above. Using this latter method allows us to reduce the amortized time cost of implementing each individual Toffoli gate to $1/m$ times the amortized (classical) parallel amortized cost of implementation, or to $1/m^2$ the standard sequential time cost of implementation.

This advantage is maximized in the case of highly parallelizable reversible circuits (where $m \approx n$ at every depth). In such cases the above method has the effect of reducing the time required to run the algorithm by two polynomial orders. For example, an $O(n^3)$ algorithm can be performed with $O(n)$ resources, an $O(n^2 \log n)$ algorithm with $O(\log n)$ resources etc.) We arrive at the following result:

Corollary 1. *Let $\{C_n\}_n$ be a uniform family of reversible circuits. The same computation can be performed using CP in time $T_{\#}(n) = O(\mathcal{C}(n)/\Delta^2)$ where $\Delta = \mathcal{C}/\mathcal{D}$.*

Here \mathcal{C} and \mathcal{D} refer to the circuit-cost complexity and circuit-depth complexity respectively of the algorithm in question.

Any *irreversible* classical circuit over the gate set $\{\text{NAND}\}$ can be transformed into a reversible one over the gate set $\{\text{Toffoli}\}$ using one of many techniques [27–29]. Hence, the above technique can be applied to *any* classical algorithm. Bennett’s method to convert irreversible computation to reversible computation, which consists of replacing all NAND gates with Toffoli ones (and introducing ancillary bits) neither increases the computational time complexity, nor does it change the computational depth complexity [42]. With this, we state our final theorem:

Theorem 2 (Coherent parallelization of classical circuits). *Let $\{C_n\}_n$ be a uniform family of classical circuits over the universal gate set $\{\text{NAND}\}$. The same computation can be performed using CP in time $T_{\#}(n) = O(\mathcal{C}(n)/\Delta^2)$ where $\Delta = \mathcal{C}(n)/\mathcal{D}(n)$.*

Here n is the size of the input, $\mathcal{C}(n)$ is the circuit gate cost (number of gates) of the classical circuit, and $\mathcal{D}(n)$ is the depth of the circuit. In short, CP can be used to reduce the time cost of any classical algorithm to below its depth complexity.

Finally, let us briefly consider the question of how an implementation of this computational speed-up may proceed in practice. One possibility is to use a *Dicke model* Hamiltonian [9–11]. This is a model of a scenario where m atoms are trapped within a fraction of their photon emission wavelength. This ensures that the photons emitted are indistinguishable from one another, and the atoms all couple collectively and coherently to the field. The Dicke Hamiltonian with m atoms is thus

$$H = \omega_c a^\dagger a + \omega_z \sum_{j=1}^m \sigma_j^z + \frac{2\lambda}{\sqrt{m}} (a + a^\dagger) \sum_{j=1}^m \sigma_j^x, \quad (6)$$

where $a(a^\dagger)$ are the annihilation (creation) operators of the photon, and σ_j^x, σ_j^z are the standard Pauli operators acting on the two energy levels that couple to the field of i 'th atom. This collective coupling gives rise to what is known as *Dicke super-radiance* where the energy density of the photon emission grows as m^2 , whereas the energy density of the decay of m atoms independently coupled to a field only grows as m . Many physical implementations of the Dicke model exist. One recent experiment by Zhang *et al* [30], for example, collectively drives 5×10^6 atoms in this fashion.

Ferrero *et al* [7] describes a proposal for implementing quantum battery-charging using the Dicke model. They show that a set of m quantum batteries can be charged coherently faster using this Hamiltonian, as compared to one where the atoms are coupled separately to a field.

We can take a similar approach. At first glance, the Dicke Hamiltonian has the effect of coherently applying an NOT gate to each atom in parallel. However, this is only the case if we consider each atom to only have two energy levels. This, of course, does not need to be the case. If we consider each atom to have (at least) *eight* energy levels (formally a higher spin Tavis–Cummings model [31]), then each atom can store *three* qubits of information. If we assign the logical three-qubit states $|110\rangle$ and $|111\rangle$ to the two energy states of each atom that couple to the field. We are free to assign the six remaining logical states to energy levels as we see fit. Then the above Dicke Hamiltonian, *without any further modification*, coherently implements m Toffoli gates on $3m$ qubits, and gives the same speed-up as the quantum battery-charging speed-up described by Ferrero *et al* [7].

Time is the usual resource by which the cost of computation (both classical and quantum) is measured. As discussed earlier, however, this yardstick only makes physical sense when *energy* is kept constant. While power is usually not considered in computational resource counting discussions, power scales directly proportional with time, when energy is constant. Hence the quadratic power requirement [7] can be understood as a restatement of this speedup and in principle should be implementable via a simple harmonic mode as noted above.

5. Discussion

In this paper we set out to achieve two different goals. First, was to introduce a model of computation that sets aside measures of complexity based on arbitrary universal gate-sets, and replace that with fundamental notions of time and energy. In this model we can regain all the standard notions and results in computational complexity, while also being able to push the envelope and study computation at a level deeper than unitary gates/circuits: at the Hamiltonian level. By studying computation at the Hamiltonian level, and extending our study to consider higher-order Hamiltonians we achieve our second goal. We present a method that can be used to speed-up the implementation of Toffoli gates. Because Toffoli gates are universal for classical computation, CP can be used to speed-up all classical computation.

CP is not an algorithm. It is a method to exploit quantum correlations [43] to reduce the amortized time-cost of individual gates. The closest analogue in classical computing is *parallel computation* [32]. Parallel computation does not reduce the gate complexity of any problem. Neither does CP. They both reduce the amortized time-cost of individual gates—the former by spreading the workload among several classical processors, the second by exploiting quantum coherence. The PRAM model of computation [25] was introduced in order to study the theoretical speed-up of parallelization, and compare parallel algorithm implementations to their standard sequential counterparts. Similarly, we have introduced a quantum model of computation in definition 2, to be able to formally describe and prove the speed-up of CP, compared to non-coherent (classical) algorithm implementations. While the theoretical maximum benefit of parallelization is seldom achieved, many practical, very useful, real-world implementations of parallelization exist, from GPUs to cluster computing. Below we will mention various quantum physical systems that could one day serve the same role for CP. Before doing so, we explore the benefit of CP.

While CP is itself not an algorithm, it is worth comparing it to the only other currently known way to exploit quantum effects for the purposes of accelerating computation: quantum algorithms.

Grover's algorithm is the most similar, in its end-effect, to CP. Grover's algorithm can be used to solve any problem within the class NP, however, it only gives an advantage for problems that do not (currently) have an algorithm that solves the problem more efficiently than brute-force search. Once a problem has an algorithm that solves it at least quadratically more efficiently than brute-force search, Grover's algorithm ceases to provide any advantage. On the other hand, CP can accelerate any existing classical algorithm using quantum coherence. Hence, CP can provide a quantum advantage on *any* possible computational problem, no matter how efficient current classical algorithms are.

A direct, important, consequence of our result then relates to provable quantum advantage. Previously, outside of oracle/black-box scenarios, the only provable computational advantage of quantum computation devices over classical was for a very narrow set of problems [33]. Because CP can improve the run-time of *any* classical algorithm, we now have provable quantum advantage for *all* computational problems.

The proportion of this advantage grows the more parallelizable the problem is. The maximum advantage is achieved for problems that can be efficiently solved using parallel computation using low (logarithmic) depth circuits—i.e. problems in the class NC. Any such problem can be solved with CP in logarithmic time. Hence CP is particularly well-suited for speeding up ubiquitous mathematical tasks such as matrix multiplication and speed up physically important tasks such as *Monte Carlo* simulations, genetic algorithms and many particle-physics simulations. Other computations that are particularly well-suited for CP that are worth mentioning due to their real-world applications include machine-learning tasks like hyperparameter grid search and cryptographic tasks such as proof-of-work in crypto-currencies and blockchain technologies.

Let us consider sorting as a concrete example of the advantage obtainable using CP. Sorting n integers can be done sequentially in $O(n \log n)$ time. This time is optimal for general integers. A fully classical parallel approach can do the same sorting in $O(\log n)$ time optimally. Using the same parallel algorithm, but using CP to speed up the implementation of gates allows us to sort these same n integers in $O(\frac{1}{n} \log n) \in O(1)$ time.

Next, we comment on the accounting of resources associated with implementing CP. From a computational complexity perspective, our method acts similarly to classical parallelization. Both methods reduce the total time-cost of solving a problem, without reducing the computational complexity, by reducing the *amortized* cost of implementing each individual gate.

From a physical perspective, our use of higher-order Hamiltonians, and our resource-cost analysis is completely in-line with the use of higher-order Hamiltonians in other physical settings such as metrology and battery charging. In the case of metrology, for instance, super-Heisenberg metrology is a speed-up over other forms of quantum metrology achievable through the use of higher-order Hamiltonians.

The topic of physical implementations of HoHs is a rich, ongoing, research area, with many inroads, including proposed implementations and several experimental setups [16, 34, 35]. While that is not the topic of this paper, it is worth discussing challenges in that field, as they relate to, and can impact, possible future CP physical implementations.

One common concern with all experiments involving delivering high power in a relatively short timescale is environmental heating. We note here that this depends entirely on the physical machine description of the quantum hardware. For instance, nuclear magnetic resonance experiments often have to contend with heating due to the presence of other chemical species around the target qubit that can also absorb the delivered power [36, 37]. This is a lesser concern for engineered quantum systems and linear optics [38]. The implementation of efficient HoH hardware will surely be an important task at the intersection of materials research and quantum engineering in the near future.

Much progress has been made toward addressing these and other potential issues and creating viable control of HoHs. Since super-Heisenberg metrology has been theoretically shown to be possible only using higher-order Hamiltonians [1–3], implementation of HoHs for this and other applications has been discussed before. There are various proposals for implementing these speedups using widely different experimental setups, such as scattering in Bose condensates [16], Duffing nonlinearity in nano-mechanical resonators [34], two-pass effective non-linearity with an atomic ensemble [35], Kerr-like nonlinearities [36], and nonlinear quantum atom-light interfaces [37]. Finally, and most importantly, this speed-up has now been experimentally demonstrated [4]. In summary, the literature strongly seems to suggest that it is indeed possible to speed-up a quantum process, *without changing its circuit complexity*, by using higher-order Hamiltonians. CP simply extends this effect from metrology to computational speed-up.

In the previous section we discussed how to give an *in-principle* experimental demonstration of CP. In fact, we argue that any existing Tavis–Cummings implementation where each atom has at least eight energy levels is already an in-principle demonstration of CP. We add two points here: first, is the acknowledgment

that this is far from a useful practical set-up. Centrally, the question of how to load the qubits into, and out of, the atoms in the Dicke cavity remains an important engineering question. The second point is that this is only *one* possible implementation avenue. Many systems with the desired collective quantum behavior have been studied beyond the ones already mentioned above—both natural [9] and engineered [38].

From a theoretical perspective, CP provides, for the first time, a clear tradeoff between the time required to perform a computation and quantum correlations. It opens a new approach of studying computation complexity that goes beyond circuit complexity to also consider quantum correlation complexity. From a more practical perspective, the theoretical maximum advantage of CP may or may not be achieved, at least in the short-term. While CP provides an asymptotic speedup when $m = n$, it is possible that the advantage, in real-world applications of CP, be limited to the case where m is a constant dependent on the size of the quantum processor [44]. In this case, the CP speedup scales as m^2 , where m is the number of qubits (atoms) that can be driven collectively together. Using the Zhang *et al* experimental implementation of the Dicke model above [30] as an example, this would allow CP to provide a speed-up by a constant factor of 25×10^{12} . While this is not an asymptotic speed-up, it is still one worth pursuing.

Acknowledgments

The authors would like to thank Rosario Fazio, Felix Binder, Yingkai Ouyang for discussions and comments on early versions of this manuscript. CP-D would like to acknowledge funding through the EPSRC Quantum Communications Hub (EP/T001011/1). SV acknowledges support from an IITB-IRCC Grant No. 16IRCCSG019, by the National Research Foundation, Prime Minister's Office, Singapore under its Competitive Research Programme (CRP Award No. NRF-CRP14-2014-02), a DST-SERB Early Career Research Award (ECR/2018/000957) and DST-QUEST Grant No. DST/ICPS/QuST/Theme-4/2019.

Appendix.

We begin with a formal definition of the function $p(\cdot)$ that we have used in the main body of the paper.

Definition 1. Let H be any Hermitian operator. Then

$$p(H) = |\langle H - h_{\min} I \rangle| = h_{\max} - h_{\min}, \quad (7)$$

where h_{\max} , h_{\min} are the maximum and minimum eigenvalues of H respectively.

Next is a discussion of the model of computation we are presenting for the first time in this paper.

Model of computation—we start this section with a formal definition of our computational model:

Definition 2 (Computing machine). A computing machine (CM) consists of a closed physical system with three subsystems B, C, S : the *battery*, *control*, and *input/output* systems respectively.

Battery: consists of an unbounded countable number of two-dimensional subsystems each with Hamiltonian $H_B = \sigma_z$.

Input/output: consists of a countably infinite dimensional system with Hamiltonian H_0 that can be arbitrarily chosen. All but a finite subsystem S of dimension $N = 2^n$ is set to the ground state of H_0 at the beginning of this computation. The state $\rho_0(\rho_f)$ of the subsystem S at the start (end) of the computation is called the *input* (*output*).

Control: exchanges energy with the battery subsystem to power the application of a Hamiltonian $H(t)$ for a time T to the input/output system during computation. This can be done with standard energy conserving unitary operators. This Hamiltonian is such that

$$p(H(t)) \leq \tau(n), \quad \forall t, \quad (8)$$

where n is the size of the input and $\tau(n)$ is a constant that at most scales linearly in n . Furthermore, $H(t)$, at any time t' , must be of the form:

$$H(t') = \sum_i^m \bigotimes_j^r \mathbf{H}^{d(i,j)}(t'), \quad (9)$$

where $\mathbf{H}(t')$ is a single, or k -qubit Hamiltonian, $d(i, j)$ is either zero or one, $r = n/k$, where n is the total number of qubits in the system.

The purpose of our model of computation is to act as the most general abstraction of natural process that can perform computation, without ignoring any of the necessary physical properties of such a process.

The purpose of the battery subsystem is to account for the energy required to perform the computation. In order to be able to compare meaningfully different computations, a standard battery Hamiltonian is chosen for every possible computer and computation performed.

The input/output subsystem is how the computer communicates with the external world, and meaningfully performs computation. The subsystem is initialized to the input state before computation. At the end of the computation the subsystem should then hold the output state. An arbitrary Hamiltonian for this system is allowed in order to be able to model—and quantify the energetic resources in—computation on different information carriers. These information carriers can be anything from ions in a trap or potential well, to nuclei, to anyons, depending on the actual implementation of the quantum computer and they all different real-world Hamiltonians.

While we leave the possibility open in our model to any possible input/output system, we will be particularly interested in (and restrict further discussion to) systems with a homogenous repeating structure, e.g. n spin $-1/2$ particles each with Hamiltonian σ_z and pairwise Ising interaction.

Finally, the sole purpose of the control subsystem is to provide a *locus* for the computation itself. It mediates between the battery and the input/output system, and performs the computation itself by drawing power from the former, and applying an external Hamiltonian to the latter. This Hamiltonian $H(t)$ is time dependent, and arbitrarily chosen based on the computation to be performed. As mentioned in the main text, in order to maintain the meaningfulness of algorithmic time complexity within our model we impose to restrictions on H . First, $p(H(t))$ must be bound from above. Our bound $\tau(n)$ is dependent on n to allow for parallel computation (performing multiple gates on different qubits at the same time). If we further limit τ to be a constant independent of n we can define a *sequential* computing machine. In this paper we focus on the more general (parallel) model. Second, equation (9) enforces that our Hamiltonian be *uniform* in space. Mathematically, this restriction follows similar constraints on circuits and parallel processor models, and for the same reasons. Without this restriction it is possible to ‘hide’ the difficulty of a computational problem within the definition of the algorithm (be it circuit or Hamiltonian.) Physically, this restricts our model to the type of Hamiltonians that are considered physical realizable (e.g. the Dicke Hamiltonian).

There are many ways (computational models) to describe classical computations. Here we use the well understood standard circuit model. We understand a uniform family of reversible circuits $\{C_n\}_n$ to consist of circuits C_n consisting of only **Toffoli** gates, each acting on n bits of input. Let $\mathcal{D}(C_n)$ be the circuit depth of C_n . For every $0 \leq i \leq N = 2^n$ let $C_n(i)$ be the result of running the circuit C_n on the binary representation of i as input.

Furthermore when discussing a particular algorithm described as a family of circuits, we will use \mathcal{C} , and \mathcal{D} to refer to its circuit-cost complexity and circuit-depth complexity respectively.

Coherent parallelization—we now focus on our method for increasing the efficiency/speed of arbitrary classical computations.

Consider a quantum system with m identical sub-systems (qubits, qudits or the tensor product thereof). Let $H[i]$, $1 \leq i \leq m$ for any Hermitian and/or unitary operator H to mean H applied to the i ’th subsystem. Formally:

$$H[i] = \left(\bigotimes_{i-1} I \right) \otimes H \otimes \left(\bigotimes_{n-i-1} I \right). \quad (10)$$

For any Hermitian, unitary operator H we define

$$H_{\parallel}(m) = \sum_{i=1}^m H[i] \quad (11)$$

$$H_{\#}(m) = \bigotimes_m H. \quad (12)$$

We then have the following result.

Lemma 1. *Let H be a Hermitian, unitary operator. Then, for any positive integer m :*

$$p(H_{\parallel}(m)) = mp(H) \quad (13)$$

$$p(H_{\#}(m)) = p(H). \quad (14)$$

Proof. Since H is both Hermitian and unitary, its only possible eigenvalues are ± 1 . So either $p(H) = 2$, or $p(H) = 0$. If $p(H) = 0$, then the lemma follows trivially. Therefore, lets assume $p(H) = 2$. Let $|+\rangle(|-\rangle)$ be

+1(−1) valued eigenket respectively of H . Then

$$\left(\sum_{i=1}^m H[i]\right) \left(\bigotimes_m |\pm\rangle\right) = \pm m \left(\bigotimes_m |\pm\rangle\right) \tag{15}$$

and

$$\left(\bigotimes_m H\right) \left(\bigotimes_m |\pm\rangle\right) = \pm 1 \left(\bigotimes_m |\pm\rangle\right). \tag{16}$$

To complete the proof we must show that $(\bigotimes_n |\pm\rangle)$ are the maximum and minimum valued (respectively) eigenkets of both $(\bigotimes_n H)$ and $(\sum_{i=1}^n H[i])$.

We show that the largest eigenvalue of $\bigotimes_m H$ is 1. We proceed by contradiction. Assume there exists a vector $|\omega\rangle$ such that

$$\left(\bigotimes_n H\right) |\omega\rangle = \omega |\omega\rangle, \tag{17}$$

where $\omega \in \mathbb{R}$ and $\omega > 1$, and that this is the largest valued eigenket of $\bigotimes_m H$. Given that $|\omega\rangle$ is an eigenket of $\bigotimes_n H$ it must be that it may be written as

$$|\omega\rangle = \bigotimes_{i=1}^n |\omega_n\rangle, \tag{18}$$

where each ket $|\omega_n\rangle$ is an eigenket of H . Furthermore, we can write

$$\omega |\omega\rangle = \left(\bigotimes_n H\right) |\omega\rangle = \bigotimes_{i=1}^n H |\omega_n\rangle = \bigotimes_{i=1}^n 1 |\omega_n\rangle, \tag{19}$$

where in the last equality we used the facts that $|\omega_n\rangle$ is an eigenket of H , and that H is both Hermitian and unitary. From here it follows that $1 < \omega = 1$, which is a contradiction. An identical argument can be used to show that -1 is the minimum eigenvalue of $H_{\#}$, and similar arguments can be used that the $\pm n$ are the maximum/minimum eigenvalues of H_{\parallel} . □

The following theorem follows directly from the previous lemma, and our computing machine definition.

Theorem 1 (Coherent parallelization). *Let H be any Hermitian unitary gate acting on a d -dimensional system or qudit. Implementing m gates in parallel using a standard parallel computation implementation H_{\parallel} is m times slower than using a CP approach $H_{\#}$.*

Proof. Without loss of generality let the bound $\tau = 1$. Then, in order to use the standard parallelization method within the bound set, one must use a normalized version of the parallel Hamiltonian $H'_{\parallel}(m) = H_{\parallel}(m)/m$. On the other hand to implement the m gates using CP one may use standard CP Hamiltonian $H_{\#}(m)$ as defined above, since it is already normalized to 1. Then $p(H_{\parallel}(m)) = p(H_{\#}(m)) = 1$, as required. However,

$$\bigotimes_m H = e^{-i\pi/2mH'_{\parallel}(m)} = e^{-i\pi/2H_{\#}(m)}, \tag{20}$$

which shows that using the Hamiltonian $H_{\#}(m)$ one can implement the desired gate $\bigotimes_m H$ a factor of m times faster than using $H_{\parallel}(m)$. □

We note here that we can generalize lemma 1 and theorem 1 to the CP implementation, and speed-up, of m distinct gates—rather than m copies of the same gate—as long as each of the m gates is both unitary and Hermitian. However, we shall see in the next result that this generalization is unnecessary for our purposes of achieving CP speed-up of universal classical computation.

Theorem 2 (Coherent parallelization of reversible circuits). *Let $\{C_n\}_n$ be a uniform family of reversible circuits, and let $A(n) = \{H(t), T\}$ be the implementation of said circuit as a computing machine algorithm and $T(n)$ is the time required to run A on an input of size of n . The same computation can be performed using CP in time $O(T(n)/\Delta)$ where $\Delta = \mathcal{C}/\mathcal{D}$.*

Proof. First we note that the average number of gates in $\{C_n\}_n$ at each depth d is given by $\Delta = \mathcal{C}/\mathcal{D}$. Hence, by theorem 1 the implementation of the gates of $\{C_n\}_n$ at depth d can be sped up on average by a factor of Δ using CP over a standard implementation. Taking the behavior at the asymptotic limit as $n \rightarrow \infty$ gives us the desired result. □

Note that in the previous theorem we are comparing a CP implementation to a standard computing machine implementation of a classical reversible circuit. However, this latter implementation is already parallel (all gates at any depth d are taken to be implemented at once). Obviously, a parallel implementation has a speed factor advantage of $\Delta = \mathcal{C}/\mathcal{D}$ over a sequential implementation. We have hence proven the following corollary.

Corollary 1. *Let $\{C_n\}_n$ be a uniform family of reversible circuits. The same computation can be performed using CP in time $T_{\#}(n) = O(\mathcal{C}(n)/\Delta^2)$ where $\Delta = \mathcal{C}/\mathcal{D}$.*

We conclude with the following result.


Theorem 3 (Coherent parallelization of classical circuits). *Let $\{C_n\}_n$ be a uniform family of classical circuits over the universal gate set $\{\text{NAND}\}$. The same computation can be performed using CP in time $T_{\#}(n) = O(\mathcal{C}(n)/\Delta^2)$ where $\Delta = \mathcal{C}(n)/\mathcal{D}(n)$.*

Proof. For this proof we first convert $\{C_n\}_n$ to a reversible family of circuits that has both the same depth- and circuit-complexity, and then simply apply corollary 1. For the first step we use a result by Bennett [27, 28] that states that any irreversible circuit family with space complexity \mathcal{S} , circuit depth complexity \mathcal{D} and circuit complexity \mathcal{C} can be perfectly simulated using a reversible circuit with space complexity $\mathcal{S} + \mathcal{C}$, circuit depth complexity \mathcal{D} and circuit complexity \mathcal{C} . \square

As noted earlier, there are many methods to convert an irreversible circuit into a reversible circuit all of which have a space/depth complexity tradeoff. For our purposes, Bennett's method is optimal as it allows us to reach the theoretical optimal time performance for CP. For many real-world applications it may be beneficial to consider newer irreversible-to-reversible transformation methods [29].

ORCID iDs

Perez-Delgado Carlos A  <https://orcid.org/0000-0003-3536-2549>

Sai Vinjanampathy  <https://orcid.org/0000-0002-5919-5442>

References

- [1] Beltrán J and Luis A 2005 *Phys. Rev. A* **72** 045801
- [2] Boixo S, Flammia S T, Caves C M and Geremia J 2007 *Phys. Rev. Lett.* **98** 090401
- [3] Roy S M and Braunstein S L 2008 *Phys. Rev. Lett.* **100** 220501
- [4] Napolitano M, Koschorreck M, Dubost B, Behbood N, Sewell R J and Mitchell M W 2011 *Nature* **471** 486
- [5] Binder F C, Vinjanampathy S, Modi K and Goold J 2015 *New J. Phys.* **17** 075015
- [6] Campaioli F, Pollock F A, Binder F C, Celeri L, Goold J, Vinjanampathy S and Modi K 2017 *Phys. Rev. Lett.* **118** 150601
- [7] Ferraro D, Campisi M, Andolina G M, Pellegrini V and Polini M 2018 *Phys. Rev. Lett.* **120** 117702
- [8] Le T P, Levensen J, Modi K, Parish M M and Pollock F A 2018 *Phys. Rev. A* **97** 022106
- [9] Gross M and Haroche S 1982 *Phys. Rep.* **93** 301
- [10] Kirton P, Roses M M, Keeling J and Dalla Torre E G 2019 *Adv. Quantum Technol.* **2** 1800043
- [11] Garraway B M 2011 *Phil. Trans. R. Soc. A* **369** 1137
- [12] Tarjan R E 1985 *SIAM J. Algebr. Discrete Methods* **6** 306
- [13] Tóth G and Apellaniz I 2014 *J. Phys. A: Math. Theor.* **47** 424006
- [14] Szczykulska M, Baumgratz T and Datta A 2016 *Adv. Phys. X* **1** 621
- [15] Motes K R, Olson J P, Rabeaux E J, Dowling J P, Olson S J and Rohde P P 2015 *Phys. Rev. Lett.* **114** 170802
- [16] Boixo S, Datta A, Davis M J, Flammia S T, Shaji A and Caves C M 2008 *Phys. Rev. Lett.* **101** 040403
- [17] Margolus N and Levitin L B 1998 *Physica D* **120** 188–95
- [18] Mandelstam L and Tamm I 1991 *Selected Papers* (Berlin: Springer) pp 115–23
- [19] Chang D, Sørensen A S, Hemmer P and Lukin M 2006 *Phys. Rev. Lett.* **97** 053002
- [20] Trügler A and Hohenester U 2008 *Phys. Rev. B* **77** 115403
- [21] Steele G A, Hüttel A K, Witkamp B, Poot M, Meerwaldt H B, Kouwenhoven L P and van der Zant H S J 2009 *Science* **325** 1103
- [22] Liu R, Zhou Z-K, Yu Y-C, Zhang T, Wang H, Liu G, Wei Y, Chen H and Wang X-H 2017 *Phys. Rev. Lett.* **118** 237401
- [23] Koch C P 2016 *J. Phys.: Condens. Matter.* **28** 213001
- [24] Cook S A and Reckhow R A 1972 *STOC '72: Proc. 4th Annual ACM Symp. on Theory of Computing* (New York: ACM) pp 73–80
- [25] Fortune S and Wyllie J 1978 *STOC '78: Proc. 10th Annual ACM Symposium on Theory of Computing* (New York: ACM) pp 114–8
- [26] Pérez-Delgado C A and Cheung D 2007 *Phys. Rev. A* **76** 032320
- [27] Bennett C H 1973 *IBM J. Res. Dev.* **17** 525
- [28] Bennett C H 1989 *SIAM J. Comput.* **18** 766
- [29] Amy M, Roetteler M and Svore K M 2017 *Int. Conf. on Computer Aided Verification* (Springer) pp 3–21
- [30] Zhang Z, Lee C H, Kumar R, Arnold K J, Masson S J, Grimsmo A L, Parkins A S and Barrett M D 2018 *Phys. Rev. A* **97** 043858
- [31] Tavis M and Cummings F W 1968 *Phys. Rev.* **170** 379
- [32] Stockmeyer L and Vishkin U 1984 *SIAM J. Comput.* **13** 409
- [33] Bravyi S, Gosset D and König R 2018 *Science* **362** 308
- [34] Woolley M J, Milburn G J and Caves C M 2008 *New J. Phys.* **10** 125018
- [35] Chase B A, Baragiola B Q, Partner H L, Black B D and Geremia J M 2009 *Phys. Rev. A* **79** 062107

- [36] Rivas A and Luis A 2010 *Phys. Rev. Lett.* **105** 010403
- [37] Napolitano M and Mitchell M W 2010 *New J. Phys.* **12** 093016
- [38] Roy T *et al* 2017 *Phys. Rev. Appl.* **7** 054025
- [39] Lloyd S 2000 *Nature* **406** 1047
- [40] Campaioli F, Pollock F A, Binder F C and Modi K 2018 *Phys. Rev. Lett.* **120** 060409
- [41] Disregarding the creation of blackholes due to the increase in energy density [39].
- [42] While it does increase the space complexity, this is irrelevant to our analysis here.
- [43] The method's usage of quantum correlations can be seen by the fact that for classical computation the system both begins and ends up in a pure state manifold given that both the initial and final states are computational product states. The entangling Hamiltonian [40] generates quantum correlations which can be calculated as the reduced entropy of any bipartition during the computation.
- [44] This is not different to the case of classical parallelization. While theoretically, a PRAM allows for an asymptotic speedup over a standard RAM, most real physical implementations, e.g. GPUs, provide *only* a scalar speedup.