The Logic in Computer Science Column

by

Yuri Gurevich

Microsoft Research
One Microsoft Way, Redmond WA 98052, USA
`gurevich@microsoft.com`

Using Quantum Computing to Learn Physics

Nathan Wiebe

Abstract

Since its inception at the beginning of the twentieth century, quantum mechanics has challenged our conceptions of how the universe ought to work; however, the equations of quantum mechanics can be too computationally difficult to solve using existing computers for even modestly large systems. Here I will show that quantum computers can sometimes be used to address such problems and that quantum computer science can assign formal complexities to learning facts about nature. Hence, computer science should not only be regarded as an applied science; it is also of central importance to the foundations of science.

1 Introduction

I first became aware that there is a deep problem at the heart of physics in a first year chemistry class. The instructor began his very first lecture by writing an equation on the board:

$$i\hbar\frac{\partial\psi(r,t)}{\partial t} = \left[\frac{-\hbar^2}{2m}\nabla^2 + V(r,t)\right]\psi(r,t), \tag{1}$$

and asked if anyone knew what this was. When no one did, he continued by saying that this was the Schrödinger equation, the fundamental equation of quantum mechanics, and that chemisty, biology and everything around us is a consequence of this equation. He then concluded (perhaps hyperbolically) by saying that the only problem is that no one knows how to solve it in general and for this reason we had to learn chemistry. For me, this lesson raised an important question: how can we be certain that the Schrödinger equation correctly describes the dynamics of these systems if solving it is computationally intractable?

This is the central question of this article, and in the course of addressing it we will not only challenge prevailing views of experimental quantum physics but also see that computer science is fundamental to the foundations of science.

The essence of the problem is that relatively simple quantum systems, such as chemicals or spin lattices, seem to be performing tasks that are well beyond the capabilities of any conceivable supercomputer. This may come as a surprise since (1) is a linear partial differential equation that can be solved in time polynomial in the dimension of the vector ψ, which represents the "quantum state" of an ensemble of particles. The problem is that the dimension of the vector ψ grows exponentially with the number of interacting particles. This means that even writing down the solution to the Schrödinger equation is exponentially expensive, let alone performing the operations needed to solve it.

Richard Feynman was perhaps the first person to clearly articulate that this computational intractability actually poses an opportunity for computer science [9]. He suspected that these seemingly intractable problems in quantum simulation could only be addressed by using a computer that possesses quantum properties such as superposition and entanglement. This conjecture motivated the field of quantum computing, which is the study of computers that operate according to quantum mechanical principles.

The fact that quantum computing has profound implications for computer science was most famously demonstrated by Shor's factoring algorithm, which runs in polynomial time [20] whereas no polynomial time algorithm currently exists for factoring. This raises the possibility that a fast scalable quantum computer might be able to easily break the cryptography protocols that make secure communication over the internet possible. Subsequently, Feyman's intuition was shown to be correct by Seth Lloyd [16], who showed that quantum computers can provide exponential advantages (over the best known algorithms) for simulating broad classes of quantum systems. Exponential advantages over existing algorithms have also been shown for certain problems related to random walks [7], matrix inversion [13, 24] and quantum chemistry [15], to name a few.

At present, no one has constructed a scalable quantum computer. The main problem is that although quantum information is very flexible it is also very fragile. Even looking at a quantum system can irreversibly damage the information that it carries. As an example, let's imagine that you want to see a stationary electron. In order to do so, you must hit it with a photon. The photon must carry a substantial amount of energy (relative to the electron's mass) to measure the position precisely and so the interaction between the photon and electron will impart substantial momentum; thereby causing the electron's momentum to become uncertain. Such effects are completely negligible for macroscopic objects, like a car, because of they have substantial mass; whereas, these effects are significant in microscopic systems. This behavior is by no means unique to photons and electrons: the laws of quantum mechanics explicitly forbid learning information about a *general* quantum state without disturbing it.

This extreme hypersensitivity means that quantum devices can lose their quantum properties on a very short time scale (typically a few milliseconds or microseconds depending on the system). Quantum error correction strategies can correct such problems [18]. The remaining challenge is that the error rates in quantum operations are still not low enough (with existing scalable architectures) for such error correction schemes to correct more error than they introduce. Nonetheless, scalable quantum computer designs already operate close to this "threshold" regime where quantum error correction can lead to arbitrarily long (and arbitrarily accurate) quantum computation [10].

What provides quantum computing with its power? Quantum computing superficially looks very much like probabilistic computing [18]. Rather than having definite bit strings, a quantum computer's data is stored as a quantum superposition of different bit strings. For example, if $\boldsymbol{v_0} = (1,0)^t$ and $\boldsymbol{v_1} = (0,1)^t$ (where $\cdot^t$ is the vector transpose) then the state $(\boldsymbol{v_0} + \boldsymbol{v_1})/\sqrt{2}$ is a valid state for a single quantum bit. This quantum bit or "qubit" can be interpretted as being simultaneously 0 and 1 with equal weight. In this case, if the state of the qubit is measured (in the computational basis) then the user will read $(0,1)^t$ with 50% probability and $(1,0)^t$ with 50% probability. Quantum computers are not just limited to having equal "superpositions" of zero and one. Arbitrary linear combinations are possible and the probability of measuring the quantum bitstring in the quantum computer to be k (in decimal) is $|\boldsymbol{v}_k|^2$ for any unit vector $\boldsymbol{v} \in \mathbb{C}^N$. Therefore, although quantum states are analog (in the sense that they allow arbitrary weighted combinations of bit strings) the measurement outcomes will always be discrete, similar to probabilistic computing.

The commonalities with probabilistic computation go only so far. Quantum states are allowed to be complex vectors (i.e. $\boldsymbol{v} \in \mathbb{C}^{2^n}$) with the restriction that $\boldsymbol{v}\boldsymbol{v}^* = 1$ (here $\cdot^*$ is the conjugate–transpose operation); furthermore, any attempt to recast the quantum state as a probability density function will result (for almost all "pure" quantum states) in a quasi–probability distribution that has negative probability density for certain (not directly observable) outcomes [23]. The state in a quantum computer is therefore fundamentally distinct from that of a probabilistic computer.

In essence, a quantum computer is a device that can approximate arbitrary rotations on an input quantum state vector and measure the result. Of course giving a quantum computer the ability to perform *any* rotation is unrealistic. A quantum computer instead is required to approximate these rotations using a discrete set of gates.

The quantum gate set that is conceptually simplest consists of only two gates: the controlled–controlled not (the Toffoli gate), which can perform any reversible Boolean circuit, and an extra quantum gate known as the Hadamard gate (H). This extra gate promotes a conventional computer to a quantum computer. The Hadamard gate which is a linear operator that has the following action on the "0" and "1" quantum bit states:

$$
\begin{aligned}
H : \begin{bmatrix}1\\0\end{bmatrix} &\mapsto \begin{bmatrix}1/\sqrt{2}\\1/\sqrt{2}\end{bmatrix},\\
H : \begin{bmatrix}0\\1\end{bmatrix} &\mapsto \begin{bmatrix}1/\sqrt{2}\\-1/\sqrt{2}\end{bmatrix}.
\end{aligned} \tag{2}
$$

In other words, this gate maps a zero–bit to an equal weight superposition of zero and one and maps a one–bit to a similar vector but with an opposite sign on one of the components. The Toffoli gate is a three–qubit linear operator that, at a logical level, acts non–trivially on only two computational basis vectors (i.e. bit string inputs): 110 ↦ 111 and 111 ↦ 110. All other inputs, such as 011, are mapped to themselves by the controlled–controlled not gate. Using the language of quantum state vectors, the action of the Toffoli gate on the two–dimensional subspace that it acts non–trivially upon is

$$
\begin{aligned}
\text{Tof} : \begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}1\\0\end{bmatrix} &\mapsto \begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix}\\
\text{Tof} : \begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix} &\mapsto \begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}1\\0\end{bmatrix}
\end{aligned} \tag{3}
$$

This finite set of gates is universal, meaning that any valid transformation of the quantum state vector can be implemented within arbitrarily small error using a finite sequence of these gates [3]. These gates are also reversible, meaning that any quantum algorithm that only uses the Hadamard and Toffoli gates and no measurement can be inverted. I will make use of this invertibility later when we discuss inferring models for quantum dynamical systems.

Measurement is different from the operations described above. The laws of quantum mechanics require that any attempt to extract information from a generic quantum state will necessarily disturb it. Measurements in quantum computing reflect this principle by forcing the system to *irreversibly* collapse to a computational basis vector (i.e. it becomes a quantum state that holds an ordinary bit string). For example, upon measurement

$$
\begin{bmatrix}\sqrt{\frac{2}{3}}\\ \\ -\sqrt{\frac{1}{3}}\end{bmatrix} \xrightarrow[\text{measurement}]{} \begin{cases}[1,0]^t & \text{with } P = 2/3\\ \\ [0,1]^t & \text{with } P = 1/3\end{cases}, \tag{4}
$$

where $[1,0]^t$ represents logical 0 and $[0,1]^t$ is the logical 1 state. There clearly is no way to invert this procedure. In this sense, the act of measurement is just like flipping a coin. Until you look at the coin, you can assign a prior probability distribution to it either being heads or tails, and upon measurement the distribution similarly "collapses" to either a heads or tails result. Quantum computation thus combines elements of reversible computing with probabilistic computing wherein the inclusion of the Hadamard gate introduces both the ability to have non–positive quantum state vectors and superposition (the ability for quantum bits to be in the state 0 and 1 simultaneously) and thereby promotes the system to a universal quantum computer.

Although quantum computing is distinct from probabilistic computing, it is unclear whether quantum computation is fundamentally more powerful. Using the language of complexity theory, it is known that the class of decision problems that a quantum computer can solve efficiently with success probability greater than $2/3$, BQP, obeys $\mathsf{BPP} \subseteq \mathsf{BQP} \subseteq \mathsf{PSPACE}$. Here BPP is the class of decision problems that can be efficiently solved with success probability greater than $2/3$ using a probabilistic Turing machine, and PSPACE is the class of problems that can be solved using polynomial space and (possibly) exponential time using a deterministic Turing machine. It is *strongly* believed that $\mathsf{BPP} \subset \mathsf{BQP}$ since exponential separations exist between the number of times that quantum algorithms and the best possible conventional algorithms have to query an oracle to solve certain problems [18, 7]; however, the precise relationship between the complexity classes remains unknown.

The apparent difficulty of simulating large quantum systems creates an interesting dilemma for quantum computing: although true quantum computers are outside of our capabilities at present, it would appear that purpose built analog devices could be used to solve problems that are truly difficult for conventional computers. Recent experiments by the NIST group have demonstrated that a two–dimensional lattice of ions with over two hundred and seventeen ions with programmable interactions can be created [6]. This device can be thought of as a sort of *analog quantum simulator* in the sense that these interactions can be chosen such that the system of ions approximate the dynamics of certain condensed physics models known as Ising models.

On the surface, it would seem that this system of ions may be performing calculations that are beyond the reach of any conceivable supercomputer. If the quantum state for such a system were "pure" (which roughly speaking means that it is an entirely quantum mixture) then the state vector would be in $\mathbb{C}^{2^{217}}$. If the quantum state vector were expressed as an array of single precision floating point numbers then the resultant array would occupy roughly 1.3×10^{61} megabytes of memory. It is inconceivable that a conventional computer could even store this vector, let alone solve the corresponding Schrödinger equation.

A conventional computer does provide us with something that the analog simulator does not: the knowledge that we can trust the computer's results. The analog simulator has virtually no guarantees attached that the dynamical model that physicists believe describes the system is actually correct to a fixed number of digits of accuracy. Finding a way to use computers to inexpensively certify that such quantum devices function properly not only remains an important challenge facing the development of new quantum technologies but it also poses a fundamental restriction on our abilities to understand the dynamics of large quantum systems, provided that Feynman's conjecture is correct.

2 Why is Certification A Problem?

In essence, the central question of this paper is that of whether we can practically test quantum mechanical models for large quantum systems. Although this is a question about physics, it can only be resolved by using the tools computer science. In particular, we will see that formal costs can be assigned to model inference and that computational complexity theory can provide profound insights about the limitations of our ability to model nature using computers.

The importance of computational complexity to the foundations of physics has only recently been observed. In particular, the apparent emergence of thermodynamics in closed quantum systems is, in some cases, a consequence of the cost of distinguishing the "true" quantum mechanical probability distribution from the thermodynamical predictions scaling exponentially with the number of particles in the ensemble [21, 19]. This is especially intriguing since it provides further evidence for the long held conjecture that there is an intimate link between the laws of thermodynamics in physics and computational complexity [17].

The problem of modeling, or certifying, quantum systems is essentially the problem of distinguishing distributions. Although the problem of distinguishing distributions has long been solved, the problem of doing so under particular operational restrictions can actually be surprisingly subtle (even in the absence of quantum effects). To see this, consider the following problem of deciding whether a device yields samples from a probability distribution $p(x)$ or from another distribution $q(x)$. The probability of correctly distinguishing which one of two possible discrete probability distributions $p(x)$ and $q(x)$ based on a single sample, using the best possible data processing technique, is given by the variational distance:

$$P_{\rm dist} = \frac{1}{2}\left(1 + \frac{1}{2}\sum_x |p(x) - q(x)|\right). \tag{5}$$

Given any non–zero bias in this measurement, the Chernoff bound shows that the success probability can be boosted to nearly 100% by repeating the experiment a logarithmically large number of times.

For typical quantum systems, $P_{\rm dist}$ is of order one [21] which means that in principle such systems are easy to distinguish. In practice, the processing method that efficiently distinguishes $p(x)$ from $q(x)$ may be impractically difficult to find in quantum mechanical problems. Thus distinguishing typical quantum systems can still be challenging.

As a motivating example, assume that you have a dice with $10,000$ sides that is promised to either be fair (i.e. $p(x) = 1/10,000$) or that the probability distribution describing the dice was itself randomly drawn from another distribution such that $\mathbb{E}_x(p(x)) = 1/10,000$ and the variance in the probabilities obeys $\mathbb{V}_x(p(x)) = 10^{-8}$. This problem is substantially different from the base problem of distinguishing two known distributions because $q(x)$ is unknown but information about the distribution that $q(x)$ is drawn from is known. A number of samples that scales at least as the fourth–root of the dimension are needed to distinguish the distributions with high probability (over samples and $p(x)$) [21, 11]. This problem is not necessarily hard: a rough lower bound on the number of samples needed is 10.

Distinguishing typical probability distributions that arise from quantum mechanics from the uniform distribution is similar to the dice problem except that in large quantum systems "the dice" may have 2^{200} sides or more. In the aforementioned example a minimum of roughly 2^{50} samples will be needed to distinguish a quantum distribution from the uniform distribution with probability greater than $2/3$ [21, 11]. Although collecting 2^{50} samples may already be prohibitively large, the number of samples needed at least doubles for every 4 qubits that are added to the quantum system. This means that distinguishing typical quantum probability becomes prohibitively expensive as quantum bits (i.e. particles) are added to the system, despite the fact that P_{dist} is of order one.

In short, quantum dynamics tend to rapidly scramble an initial quantum state vector, causing the predictions of different quantum mechanical models to become practically indistinguishable given limited computational resources. We will see below that complexity theory provides strong justification for the impracticality of distinguishing quantum distributions in general. In the following I will assume that the reader is familiar with standard complexity classes P, NP, the polynomial hierarchy, #P and so forth. For brevity, the term efficient will mean "in polynomial time".

3 Boson Sampling

Scott Aaronson and Alex Arkhipov provided the strongest evidence yet that quantum systems exist that are exponentially difficult to simulate using conventional computers in their seminal 2011 paper on "Boson Sampling" [1]. Their work proposes a simple experiment that directly challenges the extended Church–Turing thesis, which conjectures that any physically realistic model of computation can be efficiently simulated using a probabilistic Turing machine. Their Boson sampler device is not universal and is trivial to simulate using a quantum computer. The remarkable feature of this device is that it draws samples from a distribution that a conventional computer cannot efficiently draw samples from under reasonable complexity theoretic assumptions. In particular, an efficient algorithm to sample from an arbitrary Boson sampler's outcome distribution would cause the polynomial hierarchy to collapse to the third level, which is widely conjectured to be impossible.

A Boson sampling experiment involves preparing n photons (which are Bosons) in n different inputs of a network of beam splitters, which are pieces of glass that partially reflect and transmit each incident photon. An illustration of this device is given in Figure 1 (a). The Boson sampler is analogous to the Galton board, which is a device that can be used to approximately sample from the binomial distribution. The main difference between them is that the negativity of the quantum state (manifested as interference between the photons) causes the resultant distribution to be much harder to compute than the binomial distribution, as we will discuss shortly. At the end of the protocol, detectors placed in each of m different possible output "modes" count the number of photons that exit in each mode.

The sampling problem that needs to be solved in order to simulate the action of a Boson sampler is as follows. Let $A \in \mathbb{C}^{m,n}$ be the transition matrix mapping the input state to the output state for the Boson sampling device. Let us define $S = (s_1, \ldots, s_m)$ to be the experimental outcome observed, i.e. a list of the number of photons observed in mode $1, \ldots, m$. Then let $A_S \in \mathbb{C}^{n\times n}$ be the matrix that has s_1 copies of the first row of A, s_2 copies of the second and so forth. Then the probability distribution over S given A is [1]

$$\Pr(S|A) = \frac{|\mathrm{Per(A_S)}|^2}{s_1! \cdots s_m!}, \tag{6}$$

where $\mathrm{Per}(\cdot)$ is the permanent of a matrix, which is like a determinant with the exception that positive cofactors are always used in the expansion by minors step. It is important to note, however, that a quantum computer is not known to be able to efficiently learn these probabilities and in turn the permanent.

Permanent approximation is known to be #P complete [22, 1], where #P is the class of problems associated with counting the number of satisfying assignments to problems whose solutions can be verified efficiently. Such problems can be extremely challenging (much harder than factoring in the worst cases), which strongly suggests that the distribution for these experiments will be hard to find for certain cases if $n \gg 100$. The work of Aaronson and Arkhipov use the hardness of permanent approximation (albeit indirectly) to show that even drawing a sample from this distribution can be computationally hard under reasonable complexity theoretic assumptions. In contrast, drawing a sample from the quantum device

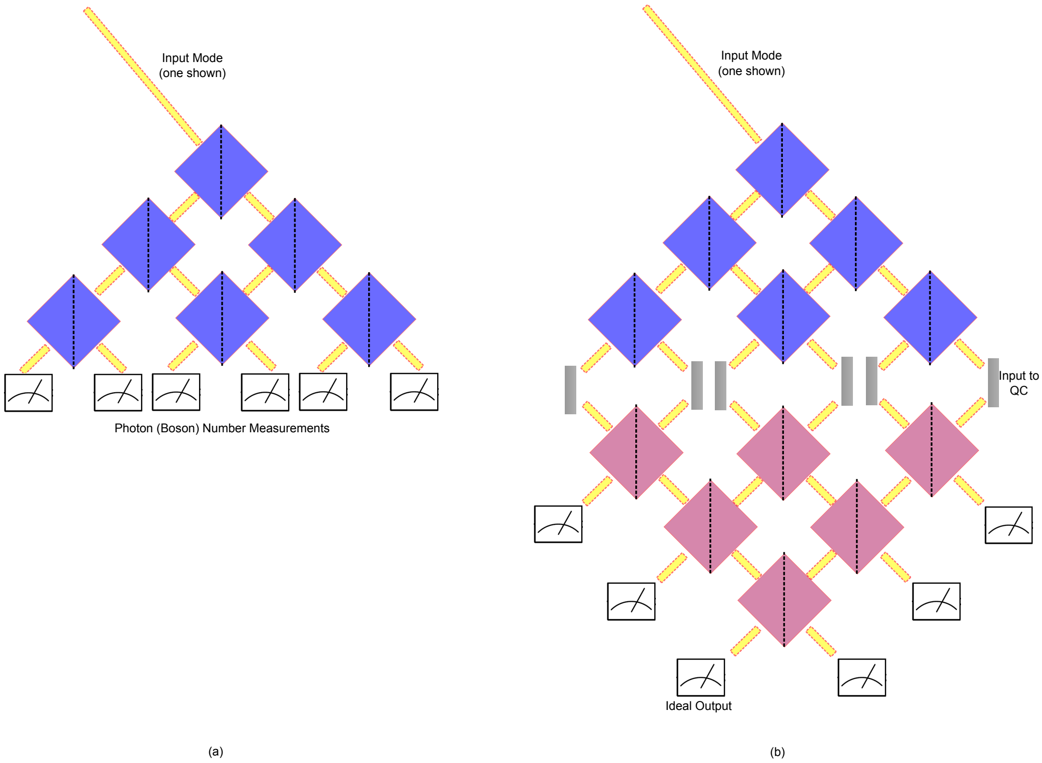

Figure 1: (a) Schematic representation of boson sampling experiment where blue squares are beam splitters. (b) Quantum inference experiment where pink squares represent a quantum computer simulation of the inverse of a hypothetical transformation carried out by (a). If the guess is correct (or nearly so) all the output bosons will be measured in the same mode that they were input with high probability.

is easy (modulo the engineering challenges involved in preparing and detecting single photons [4]) since it involves only measuring the number of photons that leave the Boson sampling device in each of the m possible modes. This provides a compelling reason to believe that there are simple physical processes that are too difficult to simulate using any conceivable conventional computer.

Although the Boson sampling distribution can be hard to compute, it can be easily to distinguished from the uniform distribution because information is known about the asymptotic form of the distribution is known [2]. Despite this, there are other distributions that are easy to sample from but are hard to distinguish from typical Boson sampling distributions [2]. Thus certifying Boson samplers can still be a difficult. A more general paradigm is needed to approach certifying, or more generally learning, an underlying dynamical model. Bayesian inference provides a near ideal framework to solve such problems.

4 Bayesian Inference

Bayes' theorem provides a convenient way to find the probability that a particular model for a system is true given an observed datum D and a set of prior assumptions about the model H. The prior assumptions about the model can be encoded as a probability distribution $\Pr(H)$ that can be interpreted as a probability that you subjectively assign to represent your belief that the true model is H. Constraints on the classes of models allowed can also be naturally included in this framework by changing the support of the prior distribution (i.e. disallowed models are given zero a priori probability). The probability that model H is correct, given the prior distribution and D is

$$\Pr(H|D) = \frac{\Pr(D|H)\Pr(H)}{\Pr(D)}, \tag{7}$$

where $\Pr(D)$ is a normalizing constant. The resultant distribution is known as the posterior distribution, which then becomes the prior distribution for the next experiment. This procedure is called "updating" and $\Pr(D|H)$ is called the likelihood function.

There is a strong connection in Bayesian inference between learning and simulation. This link is made explicit through the likelihood function, $\Pr(D|H)$, whose computation can be thought of as a simulation of H. It further suggests that if the likelihood function cannot be computed efficiently then updating your beliefs about the correct model for the system, given an observation of the system, is also not efficient. This can make Bayesian inference intractable.

Let's return to the problem of testing models for quantum systems. In this context, Bayesian inference can be extremely difficult to perform using conventional computers because the likelihood function is evaluated by simulating a quantum system, which we have argued can be exponentially expensive modulo reasonable complexity theoretic assumptions. This raises an interesting philosophical question: does computational complexity place fundamental limitations on our ability to understand systems in nature? Although this may be difficult to show in general (owing to the plethora of ways that a quantum system can be probed and the data can be analyzed), it is possible to show that this conjecture holds in certain limiting cases:

Claim 1. *There exist finite-dimensional models H_a and H_b for a quantum mechanical system and fixed observables $\mathcal{M}$ such that models H_a and H_b cannot be efficiently distinguished using Bayesian inference based on outcomes of $\mathcal{M}$ without causing the polynomial hierarchy to collapse and yet quantum computing can be used to efficiently distinguish these models using Bayesian inference and the same $\mathcal{M}$.*

The proof of the first half of this claim is straight forward. Let us consider H_a and H_b to be Boson sampling experiments with different matrices A_a and A_b which are taken to be Gaussian random matrices. Let us further take $\mathcal{M}$ to represent the measurement of the number of Bosons in each of the modes. We saw that the likelihood of a particular outcome D being observed involves calculating the permanents of $(A_a)_S$ and $(A_b)_S$ [1]. Efficient approximation of the permanent causes the polynomial hierarchy to collapse [1] and hence, modulo reasonable complexity theoretic assumptions, Bayesian inference cannot be used to distinguish between these models efficiently using $\mathcal{M}$.

A quantum computer cannot be expected to inexpensively compute the likelihood function directly because that entails computing a permanent: a task for which neither a fast quantum or conventional algorithm is known. So how is it that a quantum computer can provide an advantage? The answer is that a quantum computer can be used to change the problem to one that is easily decidable. The key freedom that a quantum computer provides is the ability to perform basis transformations that, in effect, allow $\mathcal{M}$ to be transformed into a different measurement operator whose outcomes provide clear evidence for some models over others. The presence of a quantum computer (that is coupled to the experimental system) therefore gives us the ability to solve the problem of model distinction in a completely different way.

Boson samplers can be efficiently simulated by a quantum computer. Indeed, a Boson sampler is simply a linear optical quantum computer without the ability to adaptively control quantum gates based on prior measurements (which is needed for universality). This means that a universal quantum computer trivially has all of the computational power of a Boson sampler [1]. Furthermore, closed system quantum dynamics are always be invertable (unless the system is measured) as we discussed in the context of quantum computing above. This means that we can scramble the initial state $\boldsymbol{v}$ by running it through the Boson sampler and attempt to approximately unscramble it using a quantum computer that simulates the inverse of the map that the Boson sampler H_a would perform. This inversion experiment is illustrated in Figure 1 (b). The forward evolution under the experimental system and approximate inversion under the quantum computer of the initial quantum state vector is

$$\boldsymbol{v} \mapsto U_a^{-1} U \boldsymbol{v}, \tag{8}$$

where U_a is the quantum transformation generated by H_a, U_b is the quantum transformation generated by H_b and U is an unknown transformation promised to be either U_a or U_b. If, for example, $U = U_a$ then this protocol performs:

$$\boldsymbol{v} \mapsto U_a^{-1} U_a \boldsymbol{v} = \boldsymbol{v}, \tag{9}$$

meaning that we will always observe as output the same state that was used as the input. In contrast, if $U = U_b$ then this process leads to

$$v \mapsto U_a^{-1} U_b v \neq v. \tag{10}$$

In fact, with high probability over the matrices A_a and A_b this transformation will lead to a vector that is nearly orthogonal to v if N is large [2, 11], i.e. $v^* U_a^{-1} U_b v \approx 0$, where $*$ is the conjugate transpose operation. In Boson sampling experiments, v is taken to be a Boson number eigenstate meaning that a measurement of the number of Bosons in each mode will yield the same value every time. This means that we can easily use the same $\mathcal{M}$ to determine whether $U_a^{-1} U$ maps v back to itself, which should happen with probability 1 if $U = U_a$ and with low probability if $U = U_b$. Thus the decision problem can be solved efficiently using Bayesian inference in concert with quantum computing, which justifies the above claim.

Quantum computation can therefore be used to convert seemingly intractable problems in modeling quantum systems into easily decidable ones. Also, by using a quantum computer in place of an experimental device we can assign a computational complexity to performing experiments. This allows us to characterize facts about nature as "easy" or "hard" to learn based on how computationally expensive it would be to do so using an experimental system coupled to a quantum computer; furthermore such insights are valuable *even in absentia of a scalable quantum computer*.

5 A New Way to Approach Physics

A quantum computer is more than just a computational device: it also is a universal toolbox that can emulate any other experimental system permitted by quantum theory. Put simply, if a quantum computer were to be constructed that accepts input from external physical systems then experimental physics would become computer science. When seen in this light, computational complexity becomes vital to the foundations of physics because it gives insights into the limitations of our ability to model, and in turn understand, physical systems that are no less profound than those yielded by the laws of thermodynamics.

It is assumed that the reader is familiar with quantum computing in the following. In particular, Theorem 2 and Algorithm 1 are provided for the benefit of experts in quantum computation. The details of these results can, nonetheless, be safely ignored by the reader.

This approach to learning physics using quantum computers is simply an elaboration on the strategy used above in (8) to (10). As an example of this paradigm (see [26, 25]) consider the following computational problem: assume that you are provided with an experimental quantum system that you can evolve for a specified evolution time t. We denote the transformation that this system enacts by

$\mathcal{E}(t) : \mathbb{C}^N \mapsto \mathbb{C}^N$ that can be applied to an arbitrary input quantum state for any $t \in \mathbb{R}$. The computational problem is to estimate the true model for $\mathcal{E}(t)$, denoted H, in a (potentially continuous) family of models $\{\mathcal{H}\}$ using a minimal number of experiments and only efficient quantum computations.

In order to practically learn H, we need to have a concise representation of the model. We assume that each H is parameterized by a vector $\boldsymbol{x}$ and explicitly denote a particular model H as $H(\boldsymbol{x})$ when necessary. As a clarifying example, let us return to the case of the Schrödinger equation (1) with the case where $V(r) = \frac{1}{2}kr^2$:

$$i\hbar\frac{\partial\psi(r,t)}{\partial t} = \left[\frac{-\hbar^2}{2m}\nabla^2 + \frac{1}{2}kr^2\right]\psi(r,t). \tag{11}$$

This problem, which corresponds to a quantum mechanical mass–spring system (harmonic oscillator), it may be that neither the mass of the particle m nor the spring constant k are known accurately. In this case, the model parameters can be thought of as a vector $\boldsymbol{x} = [m, k]$ and the model itself is specified by $H(\boldsymbol{x}) = \left[\frac{-\hbar^2}{2m}\nabla^2 + \frac{1}{2}kr^2\right]$. Such models are known in quantum mechanics as Hamiltonians and they uniquely specify the quantum dynamical system: $\psi(r,t) = e^{-iH(\boldsymbol{x})t/\hbar}\psi(r,0)$. For finite–dimensional systems the rule is exactly the same: $\boldsymbol{v} \mapsto e^{-iH(\boldsymbol{x})t}\boldsymbol{v}$ (in units where $\hbar = 1$), which means $\mathcal{E}(t) = e^{-iHt}$.

The problem of inferring the model parameters, $\boldsymbol{x}$, can be addressed by using Bayesian inference by following steps identical to those discussed in (8) to (10) in the context of Boson sampling. The following theorem illustrates that the likelihood functions that can be efficiently computed using a quantum computer for many quantum systems.

Theorem 2. *Let $\mathcal{E}(t) = e^{-iHt}$ where $H \in \mathcal{H}$ is the model "Hamiltonian" and we assume that*

1. *For every $X \in \mathcal{H}$, $X \in \mathbb{C}^{N\times N}$ is a d–sparse Hermitian matrix whose non–zero entries in each row can be efficiently located and computed.*
2. *There exists $\Lambda \in \mathbb{R}$ such that every $X \in \mathcal{H}$, $|X_{i,j}| \leq \Lambda$ for all i, j.*
3. *t is an arbitrary real valued evolution time for the quantum system that is specified by the user.*

then for any such hypothetical model, the likelihood of obtaining a measurement outcome D, $\Pr(D|H)$, under the action of $\left(e^{-iH_-t}\right)^{-1}\mathcal{E}(t) = e^{iH_-t}\mathcal{E}(t)$, where $H_- \in \mathcal{H}$, can be computed with high probability within error ϵ using a number of accesses to the entries of the matrices H amd H_- that, at most, scales as

$$\frac{(\log^*(N))^{1+o(1)}d^{3+o(1)}(\Lambda t)^{1+o(1)}}{\epsilon^{1+o(1)}},$$

and a number of elementary operations that scales polynomially in $\log(N)$ *given that the initial state can be efficiently prepared using a quantum computer.*

Proof. The proof follows by combining two well known quantum algorithms. First the Childs and Kothari simulation algorithm [8] shows that the action of e^{-iHt} can be simulated within error at most $\epsilon/2$ using

$$\frac{(\log^*(N))^{1+o(1)}d^{3+o(1)}(\Lambda t)^{1+o(1)}}{\epsilon^{o(1)}},$$

accesses to the matrix elements of H. The likelihood $\Pr(D|H)$ can then be estimated to within precision $\epsilon/2$ by repeating this algorithm $O(1/\epsilon^2)$ times and measuring the fraction of times that outcome D is observed. The overall error is at most ϵ since these errors are at worst additive [18].

A faster quantum algorithm called amplitude estimation exists for such sampling problems [5]. It can be used to estimate the likelihood using a number of queries that scales quadratically better with ϵ than statistical sampling and is successful at least 81% of the time. The caveat is that we need to be able to reflect quantum state vectors about the space orthogonal to the marked state and also the initial state, which is efficient under the assumptions made above. Both algorithms require a number of auxiliary operations that scale polynomially in $\log(N)$ which justifies the above claims of efficiency. □

This shows that the likelihood function can be efficiently approximated within constant error for a broad class of quantum systems using a quantum computer: quantum chemistry, condensed matter systems and many other systems fall into this class. Now let us assume that M possible hypothetical models are posited to describe $\mathcal{E}(t)$ that each satisfy the properties of Theorem 2. Using (7) requires M simulations resulting in an overall cost of

$$\frac{M(\log^*(N))^{1+o(1)}d^{3+o(1)}(\Lambda t)^{1+o(1)}}{\epsilon^{1+o(1)}}.$$

Here N can be as large as 2^{200} for reasonably large quantum systems and d is often on the order of a few hundred. The cost will therefore likely be modest if M is small, the Hamiltonian matrix is sparse and (Λt) is not unreasonably long. These requirements can be met in most physically realistic cases.

These results show efficiency for fixed ϵ, but does ϵ need to be to prohibitively small to guarantee stability? If, for example, D is an outcome such that $\Pr(D|H) \in O(\mathrm{poly}(N^{-1}))$ for hypothetical model H then ϵ will have to be extremely small in order to compute the likelihoods to even a single digit of accuracy. Cases where every model predicts $P(D|H(\boldsymbol{x})) \in \Theta(1/N)$ are therefore anathema to Bayesian inference.

Fortunately, inversion removes this possibility because it reduces each experiment to two effective outcomes. If $H_- = H$ then $e^{iH_-t}\mathcal{E}(t)\boldsymbol{v} = \boldsymbol{v}$ for all $\boldsymbol{v}$, which precludes the possibility of exponentially small likelihoods if $\boldsymbol{v}$ is a computational basis vector (or more generally if $\boldsymbol{v}$ can be efficiently transformed to a computational basis vector using the quantum computer). Conversely, it is well known that with high probability over models, $e^{iH_-t}\mathcal{E}(t)\boldsymbol{v} \neq \boldsymbol{v}$ in the limit as $N \to \infty$ if $H \neq H_-$ and $t > 0$ is taken to be a constant [12]. In contrast if t is small then it is trivial to see that almost all models and choices of H_- will result in $e^{iH_-t}\mathcal{E}(t)\boldsymbol{v} = \boldsymbol{v}$. Thus:

1. Each experiment has two outcomes: either $e^{iH_-t}\mathcal{E}(t) : \boldsymbol{v} \mapsto \boldsymbol{v}$ or $e^{iH_-t}\mathcal{E}(t) : \boldsymbol{v} \mapsto \boldsymbol{v}^\perp$ where $\boldsymbol{v}^\perp$ is in the orthogonal complement of $\boldsymbol{v}$.
2. The variable t can be chosen by the user and hence can be chosen to ensure that roughly half of the most likely models for the quantum system will approximately yield $\boldsymbol{v}$ with high probability over models [26].

These properties will typically allow Bayesian inference to identify the correct model using a logarithmic number of experiments, similar to binary search. Algorithm 1 provides a concrete method that uses these principles to learn an approximate model for a quantum system.

The question is, how well does Algorithm 1 work in practice? Since we lack quantum hardware that can implement it directly, we cannot assess Algorithm 1's performance directly; however, we can estimate how it ought to scale in practice using small numerical experiments. We will see that it works extremely well for the examples considered.

A good benchmark of the performance of the algorithm is given in [26], wherein it is shown that this approach can be used to learn inter–atom couplings of frustrated Ising models (which are used condensed matter physics to model magnetic systems). The Hamiltonian matrix that describes these systems is

$$H(\boldsymbol{x}) = \sum_{i=0}^{n-1} \sum_{j=i+1}^{n} \boldsymbol{x}_{i,j} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}^{(i)} \otimes \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}^{(j)},$$

where this notation means that the state is assumed to be a tensor product of n two–dimensional subsystems and each term in $H(\boldsymbol{x})$ acts non–trivially only on the i^{th} and j^{th} subsystems. This application is slightly more complex than the simple one discussed in Algorithm 1 because the model is continuous rather than discrete, but by discretising the problem and using a technique called resampling, this process essentially reduces to that in Algorithm 1.

Figure 2 shows that the number of updates needed to learn the couplings in an Ising model that has $n = 3, 5, 7, 9$ qubits with pairwise interactions between every qubit in the system (i.e. the interaction graph is complete) where

Algorithm 1: Quantum Hamiltonian learning algorithm.

Input: Prior probabilities w_i, $i \in \{1,\dots,M\}$, hypothetical model specifications $\boldsymbol{x}_i$, $i \in \{1,\dots,M\}$, number of samples used to estimate probabilities $N_{\rm samp}$, total number of updates used $N_{\rm exp}$, state preparation protocol Q for the initial states $\boldsymbol{v}_\mathbf{0}$, protocol for implementing measurement operator P such that $\boldsymbol{v}_\mathbf{0}\boldsymbol{v}_\mathbf{0}^*$ is a POVM element, an error tolerance ϵ and k the number of votes used to boost the success probability of amplitude estimation.

Output: Hamiltonian parameters $\boldsymbol{x}$ such that $H(\boldsymbol{x}) \approx H(\boldsymbol{x}_{\rm true})$.

```
function QHL($\{w_i\}$, $\{\boldsymbol{x}_i\}$, $N_{\rm samp}$, $N_{\rm exp}$, $P$, $Q$,$\epsilon$, $k$)
    for $i \in 1 \to N_{\rm exp}$ do
        $\boldsymbol{v}_\mathbf{0} \leftarrow Q(i)$.                                   ▷ Prepare initial state.
        Draw $\boldsymbol{x}_-$ and $\boldsymbol{x}'$ from $\Pr(\boldsymbol{x}) := w_i/\sum_i w_i$.
        $t \leftarrow 1/\|H(x) - H(x')\|$.              ▷ Choose $t$ according to guess heuristic
        $H_- \leftarrow H(\boldsymbol{x}_-)$.
        $D \leftarrow$ measurement of $e^{iH(\boldsymbol{x}_-)t}\mathcal{E}(t)\boldsymbol{v}_\mathbf{0}$ using $P$.        ▷ Perform experiment on untrusted system.
        for $j \in 1 \to M$ do                  ▷ Compute likelihoods using quantum computer
            for $\kappa \in 1 \to 2k-1$ do
                $Y_\kappa \leftarrow$ result of amplitude estimation         ▷ Estimate $\Pr(D|H(\boldsymbol{x}_j))$
                        on $e^{iH_-t}e^{-iH(x_j)t}\boldsymbol{v}_0$, to within error $\epsilon$.
            end for
            $p_m \leftarrow$ median($Y$).                        ▷ Vote on correct likelihood
        end for
        $Z \leftarrow \sum_{m=1,M} w_m p_m$.
        $w_i \leftarrow w_i p_i/Z$.                                    ▷ Perform update.
    end for
    return $\sum_m w_m \boldsymbol{x}_m$                       ▷ Return Bayes estimate of $\boldsymbol{x}_{\rm true}$.
end function
```

$\boldsymbol{v}_\mathbf{0} = [1/\sqrt{2^n},\dots,1/\sqrt{2^n}]^t$. The error decreases exponentially with the number of updates for any fixed n. In other words, there exist $\gamma(n) > 0$ such that the error scales as $e^{-\gamma(n)N_{\rm samp}}$. Figure 3 shows that the decay constants, $\gamma(n)$, for these exponential curves scale as $\Theta(1/\dim(\boldsymbol{x}))$, where $\dim(\boldsymbol{x})$ is the number of parameters in the model. In general $\dim(\boldsymbol{x})$ need not be a function of n but for the data above, each qubit interacts with all other qubits in the system and hence $\dim(\boldsymbol{x}) = \binom{n}{2} = O(n^2)$. We find that $\boldsymbol{x}$ can be learned to within error Δ using a number of experiments that empirically scales as $\Theta(n^2\log(1/\Delta))$.

The prior results allow us to estimate the complexity of inferring an Ising

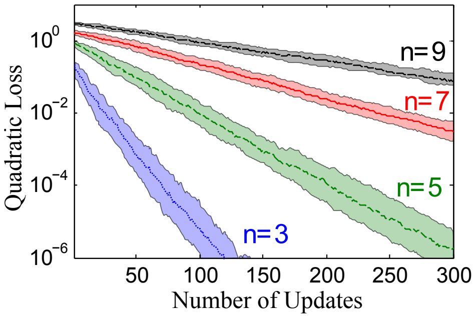

Figure 2: Quadratic loss vs the number of experiments needed to learn Ising models. The shaded regions are a 50% confidence interval and the solid line gives the median. Figure from [26].

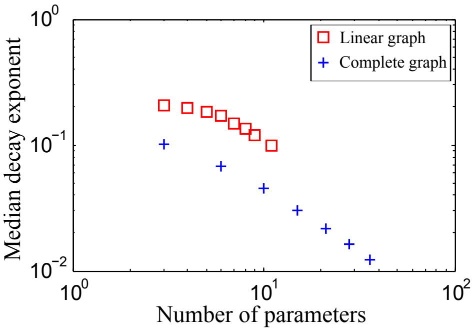

Figure 3: The median decay exponent (γ) for the quadratic loss as a function of the number of parameters in the Ising model on the complete graph as well as the line. Figure originally from [26].

model $H \in \mathbb{C}^{2^n \times 2^n}$ using a quantum computer as an experimental device. The algorithm requires a number of queries to $\mathcal{E}(t)$ that scales as $O(n^2 \log(1/\Delta))$ and a number of elementary operations that scale as

$$\frac{Mn^{2+o(1)}}{(\Delta\epsilon)^{1+o(1)}},$$

where M is the number of models used in the discretization (M is known to scale sub–exponentially with n but experimentally $M \in O(\text{polylog}(n))$ and a constant value of $\epsilon = 0.05$ seems to suffice) and $d = 1$.

These scalings are not only show that a formal complexity can be assigned to the experimental problem of learning a physical model for a large quantum system, but also suggest that quantum computing could make the learning problem tractable even for cases where $n > 100$. The combination of quantum computation and statistical inference may therefore provide our best hope of deeply understanding the physics of massive quantum systems that seem to be too complex to understand directly.

6 Conclusion

Returning now to our original question, we see that there is good reason (based on complexity theoretic conjectures) to suspect that solving the equations of quantum dynamics, such as the Schrödinger equation, may be intractable even for certain modestly large quantum systems. This would seem to suggest that some quantum systems exhibit dynamics that, for all practical purposes, cannot be directly compared to the predictions of quantum theory. Quantum computation offers the possibility that such problems can be circumvented by indirectly comparing the dynamics of the system with simulations performed on the quantum computer. This also provides a surprising insight: experimental quantum physics can be recast in the language of quantum computer science, allowing formal time complexities to be assigned to learning facts about nature and reinforcing the importance of computer science to the foundations of science.

A quantum computer does not, however, allow us to address all of the deep problems facing us in physics: they are not known to be capable of simulating the standard model in quantum field theory [14], which is arguably the most successful physical theory ever tested. Simulation algorithms are also unknown for string theory or quantum loop gravity, meaning that many of the most important physics questions of this generation remain outside of reach of quantum computing. Much more work may be needed before physicists and computer scientists can truly claim that quantum computers (or generalizations thereof) can be used to rapidly infer models for all physical systems.

I would like to thank Christopher Granade and Yuri Gurevitch for valuable feedback and Robert Browne for the inspiration behind this work.

References

[1] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, pages 333–342. ACM, 2011.

[2] Scott Aaronson and Alex Arkhipov. Bosonsampling is far from uniform. *arXiv preprint arXiv:1309.7460*, 2013.

[3] Dorit Aharonov. A simple proof that toffoli and hadamard are quantum universal. *arXiv preprint quant-ph/0301040*, 2003.

[4] Matthew A Broome, Alessandro Fedrizzi, Saleh Rahimi-Keshari, Justin Dove, Scott Aaronson, Timothy C Ralph, and Andrew G White. Photonic boson sampling in a tunable circuit. *Science*, 339(6121):794–798, 2013.

[5] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *arXiv preprint quant-ph/0005055*, 2000.

[6] Joseph W Britton, Brian C Sawyer, Adam C Keith, C-C Joseph Wang, James K Freericks, Hermann Uys, Michael J Biercuk, and John J Bollinger. Engineered two-dimensional ising interactions in a trapped-ion quantum simulator with hundreds of spins. *Nature*, 484(7395):489–492, 2012.

[7] Andrew M Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 59–68. ACM, 2003.

[8] Andrew M Childs and Robin Kothari. Simulating sparse hamiltonians with star decompositions. In *Theory of Quantum Computation, Communication, and Cryptography*, pages 94–103. Springer, 2011.

[9] Richard P Feynman. Simulating physics with computers. *International journal of theoretical physics*, 21(6):467–488, 1982.

[10] Austin G Fowler, Ashley M Stephens, and Peter Groszkowski. High-threshold universal quantum computation on the surface code. *Physical Review A*, 80(5):052312, 2009.

[11] C Gogolin, M Kliesch, L Aolita, and J Eisert. Boson-sampling in the light of sample complexity. *arXiv preprint arXiv:1306.3995*, 2013.

[12] Fritz Haake. *Quantum signatures of chaos*, volume 54. Springer, 2010.

[13] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.

[14] Stephen P Jordan, Keith SM Lee, and John Preskill. Quantum algorithms for quantum field theories. *Science*, 336(6085):1130–1133, 2012.

[15] Ivan Kassal, James D Whitfield, Alejandro Perdomo-Ortiz, Man-Hong Yung, and Alán Aspuru-Guzik. Simulating chemistry using quantum computers. *arXiv preprint arXiv:1007.2648*, 2010.

[16] Seth Lloyd. Universal quantum simulators. *SCIENCE*, pages 1073–1077, 1996.

[17] Stephan Mertens. Computational complexity for physicists. *Computing in Science & Engineering*, 4(3):31–47, 2002.

[18] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

[19] A. J. Short and Farrelly T. C. Quantum equilibration in finite time. *New Journal of Physics*, 14:013063, 2012. arXiv:1110.5759.

[20] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 124–134. IEEE, 1994.

[21] Cozmin Ududec, Nathan Wiebe, and Joseph Emerson. Information-theoretic equilibration: The appearance of irreversibility under complex quantum dynamics. *Physical review letters*, 111(8):080403, 2013.

[22] Leslie G Valiant. The complexity of computing the permanent. *Theoretical computer science*, 8(2):189–201, 1979.

[23] Victor Veitch, Christopher Ferrie, David Gross, and Joseph Emerson. Negative quasi-probability as a resource for quantum computation. *New Journal of Physics*, 14(11):113011, 2012.

[24] Nathan Wiebe, Daniel Braun, and Seth Lloyd. Quantum algorithm for data fitting. *Physical Review Letters*, 109(5):050505, 2012.

[25] Nathan Wiebe, Christopher Granade, Christopher Ferrie, and David G Cory. Quantum hamiltonian learning using imperfect quantum resources. *arXiv preprint arXiv:1311.5269*, 2013.

[26] Nathan Wiebe, Christopher Granade, Christopher Ferrie, and DG Cory. Hamiltonian learning and certification using quantum resources. *arXiv preprint arXiv:1309.0876*, 2013.