The Computational Complexity of Linear Optics*

Scott Aaronson†
MIT

Alex Arkhipov‡
MIT

ABSTRACT

We give new evidence that quantum computers—moreover, rudimentary quantum computers built entirely out of linear-optical elements—cannot be efficiently simulated by classical computers. In particular, we define a model of computation in which identical photons are generated, sent through a linear-optical network, then nonadaptively measured to count the number of photons in each mode. This model is not known or believed to be universal for quantum computation, and indeed, we discuss the prospects for realizing the model using current technology. On the other hand, we prove that the model is able to solve sampling problems and search problems that are classically intractable under plausible assumptions.

Our first result says that, if there exists a polynomial-time classical algorithm that samples from the same probability distribution as a linear-optical network, then $\mathsf{P}^{\#\mathsf{P}} = \mathsf{BPP}^{\mathsf{NP}}$, and hence the polynomial hierarchy collapses to the third level. Unfortunately, this result assumes an extremely accurate simulation.

Our main result suggests that even an approximate or noisy classical simulation would already imply a collapse of the polynomial hierarchy. For this, we need two unproven conjectures: the *Permanent-of-Gaussians Conjecture*, which says that it is #P-hard to approximate the permanent of a matrix A of independent $\mathcal{N}(0,1)$ Gaussian entries, with high probability over A; and the *Permanent Anti-Concentration Conjecture*, which says that $|\operatorname{Per}(A)| \geq \sqrt{n!}/\operatorname{poly}(n)$ with high probability over A. We present evidence for these conjectures, both of which seem interesting even apart from our application.

This paper does not assume knowledge of quantum optics. Indeed, part of its goal is to develop the beautiful theory of noninteracting bosons underlying our model, and its connection to the permanent function, in a self-contained way accessible to theoretical computer scientists.

Categories and Subject Descriptors

F.1.3 [**Theory of Computation**]: Computation by Abstract Devices—*Complexity Measures and Classes*

General Terms

Theory

1. INTRODUCTION

The Extended Church-Turing Thesis says that all computational problems that are efficiently solvable by realistic physical devices, are efficiently solvable by a probabilistic Turing machine. Ever since Shor's algorithm [29], we have known that this thesis is in severe tension with the currently-accepted laws of physics. One way to state Shor's discovery is this:

> *Predicting the (probabilistic) results of a given quantum-mechanical experiment, to finite accuracy, cannot be done by a classical computer in probabilistic polynomial time, unless factoring integers can as well.*

As the above formulation makes clear, Shor's result is not merely about some hypothetical future in which large-scale quantum computers are built. It is also a hardness result for a practical problem. For *simulating quantum systems* is one of the central computational problems of modern science, with applications from drug design to nanofabrication to nuclear physics. It has long been a major application of high-performance computing, and Nobel Prizes have been awarded for methods (such as the Density Functional Theory) to handle special cases. What Shor's result shows is that, if we had an efficient, *general-purpose* solution to the quantum simulation problem, then we could also break widely-used cryptosystems such as RSA.

However, as evidence against the Extended Church-Turing Thesis, Shor's algorithm has two significant drawbacks. The first is that, even by the conjecture-happy standards of complexity theory, it is no means settled that factoring is classically hard. Yes, we believe this enough to base modern

*For the 96-page full version, see www.scottaaronson.com/papers/optics.pdf

†Email: aaronson@csail.mit.edu. This material is based upon work supported by the National Science Foundation under Grant No. 0844626. Also supported by a DARPA YFA grant and a Sloan Fellowship.

‡Email: arkhipov@mit.edu. Supported by an Akamai Foundation Fellowship.

cryptography on it—but as far as anyone knows, factoring could be in BPP without causing any collapse of complexity classes or other disastrous theoretical consequences. Also, of course, there *are* subexponential-time factoring algorithms (such as the number field sieve), and few would express confidence that they cannot be further improved. And thus, ever since Bernstein and Vazirani [7] defined the class BQP of quantumly feasible problems, it has been a dream of quantum computing theory to show (for example) that, if BPP = BQP, then the polynomial hierarchy would collapse, or some other "generic, foundational" assumption of theoretical computer science would fail. In this paper, we do not *quite* achieve that dream, but we come closer than one might have thought possible.

The second, even more obvious drawback of Shor's algorithm is that implementing it scalably is well beyond current technology. To run Shor's algorithm, one needs to be able to perform arithmetic (including modular exponentiation) on a coherent superposition of integers encoded in binary. This does not seem much easier than building a *universal* quantum computer.[1] In particular, it appears one first needs to solve the problem of *fault-tolerant quantum computation*, which is known to be possible in principle if quantum mechanics is valid [5, 23], but might require decoherence rates that are several orders of magnitude below what is achievable today.

Thus, one might suspect that proving a quantum system's computational power by having it factor integers encoded in binary is a bit like proving a dolphin's intelligence by teaching it to solve arithmetic problems. Yes, with heroic effort, we can probably do this, and perhaps we have good reasons to. However, if we just watched the dolphin in its natural habitat, then we might see it display equal intelligence with no special training at all.

Following this analogy, we can ask: are there more "natural" quantum systems that *already* provide evidence against the Extended Church-Turing Thesis? Indeed, there are countless quantum systems accessible to current experiments—including high-temperature superconductors, Bose-Einstein condensates, and even just large nuclei and molecules—that seem intractable to simulate on a classical computer, and largely for the reason a theoretical computer scientist would expect: namely, that the dimension of a quantum state increases exponentially with the number of particles. The difficulty is that it is not clear how to interpret these systems as *solving computational problems*. For example, what is the "input" to a Bose-Einstein condensate? In other words, while these systems might be hard to simulate, we would not know how to justify that conclusion using the one formal tool (reductions) that is currently available to us.

So perhaps the real question is this: do there exist quantum systems that are "intermediate" between Shor's algorithm and a Bose-Einstein condensate—in the sense that

(1) they are significantly closer to experimental reality than universal quantum computers, but

(2) they can be proved, under plausible complexity assumptions (the more "generic" the better), to be intractable to simulate classically?

[1] One caveat is a result of Cleve and Watrous [10], that Shor's algorithm can be implemented using *log-depth* quantum circuits (that is, in $\mathsf{BPP}^{\mathsf{BQNC}}$). But even here, fault-tolerance will presumably be needed, among other reasons because one still has polynomial *latency* (the log-depth circuit does not obey spatial locality constraints).

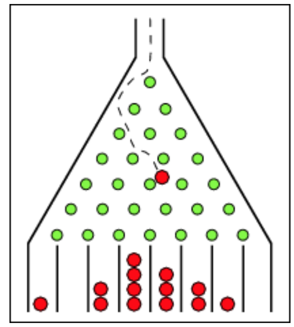

Figure 1: Galton's board, a simple "computer" to output samples from the binomial distribution. From MathWorld.

In this paper, we will argue that the answer is yes.

1.1 Our Model

Building on earlier work [14, 15, 22, 27, 36], we define and study a formal model of *quantum computation with noninteracting bosons*. Physically, our model could be implemented using a *linear-optical network*, in which n identical photons pass through a collection of simple optical elements (beamsplitters and phaseshifters), and are then measured to determine the number of photons in each location. In the full version, we give a detailed exposition of the model that does not presuppose any physics knowledge. For now, though, it is helpful to imagine a rudimentary "computer" consisting of n identical balls, which are dropped one by one into a vertical lattice of pegs, each of which randomly scatters each incoming ball onto one of two other pegs. Such an arrangement—called *Galton's board*—is sometimes used in science museums to illustrate the binomial distribution (see Figure 1). The "input" to the computer is the exact arrangement A of the pegs, while the "output" is the number of balls that have landed at each location on the bottom (or rather, a sample from the joint distribution $\mathcal{D}_A$ over these numbers). There is no interaction between pairs of balls.

Our model is essentially the same as that shown in Figure 1, except that instead of identical balls, we use *identical bosons* governed by quantum statistics. Other differences are that, in our model, the "balls" are each dropped from different starting locations, rather than a single location; and the "pegs," rather than being arranged in a regular lattice, can be arranged arbitrarily to encode a problem of interest.

Mathematically, the key point about our model is that, to find the probability of any particular output of the computer, one needs to calculate the *permanent* of an $n \times n$ matrix. This can be seen even in the classical case: suppose there are n balls and n final locations, and ball i has probability a_{ij} of landing at location j. Then the probability of one ball landing in each of the n locations is

$$\operatorname{Per}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i\sigma(i)},$$

where $A = (a_{ij})_{i,j\in[n]}$. Of course, in the classical case, the a_{ij}'s are nonnegative real numbers—which means that we can approximate $\operatorname{Per}(A)$ in probabilistic polynomial time, by using the celebrated algorithm of Jerrum, Sinclair, and Vigoda [18]. In the quantum case, by contrast, the a_{ij}'s are complex numbers. And it is not hard to show that, given a general matrix $A \in \mathbb{C}^{n\times n}$, even *approximating* $\operatorname{Per}(A)$ to within a constant factor is $\#\mathsf{P}$-complete. This fundamental difference between nonnegative and complex matrices is the starting point for everything we do in this paper.

It is not hard to show that a boson computer can be simulated by a "standard" quantum computer (that is, in BQP). But the other direction seems extremely unlikely—indeed, it even seems unlikely that a boson computer can do universal *classical* computation! Nor do we have any evidence that a boson computer could factor integers, or solve any other decision or promise problem not in BPP. However, if we broaden the notion of a computational problem to encompass *sampling* and *search* problems, then the situation is quite different.

1.2 Our Results

In this paper we study BosonSampling: the problem of sampling, either exactly or approximately, from the output distribution of a boson computer. Our goal is to give evidence that this problem is hard for a classical computer. Crucially, we cannot simply appeal to the fact that the permanent is $\#\mathsf{P}$-complete, since estimating the permanent of a *given* matrix via a linear-optical experiment would generally involve estimating an exponentially-small amplitude, which would require repeating the experiment an exponential number of times. Instead, we need to study BosonSampling as a new problem requiring new hardness arguments.

Our main results fall into three categories:

(1) Hardness results for exact BosonSampling, which give an essentially complete picture of that case.

(2) Hardness results for *approximate* BosonSampling, which depend on plausible conjectures about the permanents of i.i.d. Gaussian matrices.

(3) A program aimed at understanding and proving the conjectures.

We now discuss these in turn.

1.2.1 The Exact Case

Our first (easy) result says the following.

Theorem 1. *The exact* BosonSampling *problem is not efficiently solvable by a classical computer, unless* $\mathsf{P}^{\#\mathsf{P}} = \mathsf{BPP}^{\mathsf{NP}}$ *and the polynomial hierarchy collapses to the third level.*

More generally, let $\mathcal{O}$ be any oracle that "simulates boson computers," in the sense that $\mathcal{O}$ takes as input a random string r (which $\mathcal{O}$ uses as its only source of randomness) and a description of a boson computer A, and returns a sample $\mathcal{O}_A(r)$ from the probability distribution $\mathcal{D}_A$ over possible outputs of A. Then $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{BPP}^{\mathsf{NP}^{\mathcal{O}}}$.

In particular, even if the exact BosonSampling problem were solvable by a classical computer *with an oracle for a* PH *problem*, Theorem 1 would still imply that $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{BPP}^{\mathsf{PH}}$—and therefore that the polynomial hierarchy would collapse, by Toda's Theorem [35]. This provides evidence that quantum computers have capabilities outside the entire polynomial hierarchy, complementing the recent evidence of Aaronson [2] and Fefferman and Umans [11].

At least for a computer scientist, it is tempting to interpret Theorem 1 as saying that "the exact BosonSampling problem is $\#\mathsf{P}$-hard under $\mathsf{BPP}^{\mathsf{NP}}$-reductions." Notice that this would have a shocking implication: that quantum computers (indeed, quantum computers of a particularly simple kind) could efficiently solve a $\#\mathsf{P}$-hard problem!

There is a catch, though, arising from the fact that BosonSampling is a sampling problem rather than a decision problem. Namely, if $\mathcal{O}$ is an oracle for sampling from the boson distribution $\mathcal{D}_A$, then Theorem 1 shows that $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{BPP}^{\mathsf{NP}^{\mathcal{O}}}$—*but only if the* $\mathsf{BPP}^{\mathsf{NP}}$ *machine gets to fix the random bits used by* $\mathcal{O}$. This condition is clearly met if $\mathcal{O}$ is a classical randomized algorithm, since we can always interpret a randomized algorithm as just a deterministic algorithm that takes a random string r as part of its input. On the other hand, the condition would *not* be met if we implemented $\mathcal{O}$ (for example) using the boson computer itself. In other words, our "reduction" from $\#\mathsf{P}$-complete problems to BosonSampling makes essential use of the hypothesis that we have a *classical* BosonSampling algorithm.

In the full version, we give two proofs of Theorem 1. In the first proof, we consider the probability p of some particular basis state when a boson computer is measured. We then prove two facts:

(1) Even *approximating* p to within a multiplicative constant is a $\#\mathsf{P}$-hard problem.

(2) *If* we had a polynomial-time classical algorithm for exact BosonSampling, *then* we could approximate p to within a multiplicative constant in the class $\mathsf{BPP}^{\mathsf{NP}}$, by using a standard technique called *universal hashing*.

Combining facts (1) and (2), we find that, if the classical BosonSampling algorithm exists, then $\mathsf{P}^{\#\mathsf{P}} = \mathsf{BPP}^{\mathsf{NP}}$, and therefore the polynomial hierarchy collapses.

Our second proof was inspired by independent work of Bremner, Jozsa, and Shepherd [8]. In this proof, we start with a result of Knill, Laflamme, and Milburn [22], which says that linear optics with *adaptive measurements* is universal for BQP. A straightforward modification of their construction shows that linear optics with *postselected* measurements is universal for $\mathsf{PostBQP}$ (that is, quantum polynomial-time with postselection on possibly exponentially-unlikely measurement outcomes). Furthermore, Aaronson [1] showed that $\mathsf{PostBQP} = \mathsf{PP}$. On the other hand, if a classical BosonSampling algorithm existed, then we will show that we could simulate postselected linear optics in $\mathsf{PostBPP}$ (that is, *classical* polynomial-time with postselection, also called $\mathsf{BPP}_{\mathsf{path}}$). We would therefore get

$$\mathsf{BPP}_{\mathsf{path}} = \mathsf{PostBPP} = \mathsf{PostBQP} = \mathsf{PP},$$

which is known to imply a collapse of the polynomial hierarchy.

Despite the simplicity of the above arguments, there is something conceptually striking about them. Namely, starting from an algorithm to *simulate quantum mechanics*, we get an algorithm[2] to *solve* $\#\mathsf{P}$*-complete problems*—even though solving $\#\mathsf{P}$-complete problems is believed to be well beyond what a quantum computer itself can do! Of course, one price we pay is that we need to talk about sampling problems rather than decision problems. If we do so, though,

[2]Admittedly, a $\mathsf{BPP}^{\mathsf{NP}}$ algorithm.

then we get to base our belief in the power of quantum computers on $\mathsf{P}^{\#\mathsf{P}} \neq \mathsf{BPP}^{\mathsf{NP}}$, which is a much more "generic" (many would say safer) assumption than FACTORING$\notin$ BPP.

As we see it, the central drawback of Theorem 1 is that it only addresses the consequences of a fast classical algorithm that *exactly* samples the boson distribution $\mathcal{D}_A$. One can relax this condition slightly: if the oracle $\mathcal{O}$ samples from some distribution $\mathcal{D}'_A$ whose probabilities are all *multiplicatively close* to those in $\mathcal{D}_A$, then we still get the conclusion that $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{BPP}^{\mathsf{NP}^{\mathcal{O}}}$. In our view, though, multiplicative closeness is already too strong an assumption. At a minimum, given as input an error parameter $\varepsilon > 0$, we ought to let our simulation algorithm sample from some distribution $\mathcal{D}'_A$ such that $\|\mathcal{D}'_A - \mathcal{D}_A\| \leq \varepsilon$ (where $\|\cdot\|$ represents total variation distance), using $\operatorname{poly}(n, 1/\varepsilon)$ time.

Why are we so worried about this issue? One obvious reason is that noise, decoherence, photon losses, etc. will be unavoidable features in any real implementation of a boson computer. As a result, not even the boson computer *itself* can sample exactly from the distribution $\mathcal{D}_A$! So it seems arbitrary and unfair to require this of a classical simulation algorithm.

A second, more technical reason to allow error is that later, we would like to show that a boson computer can solve classically-intractable *search* problems, in addition to sampling problems. However, while Aaronson [3] proved an extremely general connection between search problems and sampling problems, that connection only works for *approximate* sampling, not exact sampling.

The third, most fundamental reason to allow error is that the connection we are claiming, between quantum computing and #P-complete problems, is so counterintuitive. One's first urge is to dismiss this connection as an artifact of poor modeling choices. So the burden is on us to demonstrate the connection's robustness.

Unfortunately, the proof of Theorem 1 fails completely when we consider approximate sampling algorithms. The reason is that the proof hinges on the #P-completeness of estimating a single, exponentially-small probability p. Thus, if a sampler "knew" which p we wanted to estimate, then it could adversarially choose to corrupt that p. It would still be a perfectly good approximate sampler, but would no longer reveal the solution to the #P-complete instance that we were trying to solve.

1.2.2 The Approximate Case

To get around the above problem, we need to argue that a boson computer can sample from a distribution $\mathcal{D}$ that "robustly" encodes the solution to a #P-complete problem. This means intuitively that, even if a sampler was badly wrong about any ε fraction of the probabilities in $\mathcal{D}$, the remaining $1-\varepsilon$ fraction would still allow the #P-complete problem to be solved.

It is well-known that there exist #P-complete problems with *worst-case/average-case equivalence*, and that one example of such a problem is the permanent, at least over finite fields. This is a reason for optimism that the sort of robust encoding we need might be possible. Indeed, it was precisely our desire to encode the "robustly #P-complete" permanent function into a quantum computer's amplitudes that led us to study the noninteracting-boson model in the first place. That this model also has great experimental interest simply came as a bonus.

In this paper, *our main technical contribution is to prove a connection between the ability of classical computers to solve the approximate* BOSONSAMPLING *problem and their ability to approximate the permanent.* This connection "almost" shows that even approximate classical simulation of boson computers would imply a collapse of the polynomial hierarchy. There is still a gap in the argument, but it has nothing to do with quantum computing. The gap is simply that it is not known, at present, how to extend the worst-case/average-case equivalence of the permanent from finite fields to suitably analogous statements over the reals or complex numbers. We show that, *if* this gap can be bridged, then there exist search problems and approximate sampling problems that are solvable in polynomial time by a boson computer, but not by a BPP machine unless $\mathsf{P}^{\#\mathsf{P}} = \mathsf{BPP}^{\mathsf{NP}}$.

More concretely, consider the following problem, where the GPE stands for GAUSSIAN PERMANENT ESTIMATION:

PROBLEM 2 ($|\mathrm{GPE}|^2_{\pm}$). *Given as input a matrix $X \sim \mathcal{N}(0,1)^{n\times n}_{\mathbb{C}}$ of i.i.d. Gaussians, together with error bounds $\varepsilon, \delta > 0$, estimate $|\operatorname{Per}(X)|^2$ to within additive error $\pm\varepsilon \cdot n!$, with probability at least $1-\delta$ over X, in $\operatorname{poly}(n, 1/\varepsilon, 1/\delta)$ time.*

Then our main result is the following.

THEOREM 3 (MAIN RESULT). *Let $\mathcal{D}_A$ be the probability distribution sampled by a boson computer A. Suppose there exists a classical algorithm C that takes as input a description of A as well as an error bound ε, and that samples from a probability distribution $\mathcal{D}'_A$ such that $\|\mathcal{D}'_A - \mathcal{D}_A\| \leq \varepsilon$ in $\operatorname{poly}(|A|, 1/\varepsilon)$ time. Then the $|\mathrm{GPE}|^2_{\pm}$ problem is solvable in $\mathsf{BPP}^{\mathsf{NP}}$. Indeed, if we treat C as a black box, then $|\mathrm{GPE}|^2_{\pm} \in \mathsf{BPP}^{\mathsf{NP}^C}$.*

In proving Theorem 3, the key idea is to "smuggle" the $|\mathrm{GPE}|^2_{\pm}$ instance X that we want to solve into the probability of a *random* output of a boson computer A. That way, even if the classical sampling algorithm C is adversarial, it will not know which of the exponentially many probabilities in $\mathcal{D}_A$ is the one we care about. And therefore, provided C correctly approximates *most* probabilities in $\mathcal{D}_A$, with high probability it will correctly approximate "our" probability, and will therefore allow $|\operatorname{Per}(X)|^2$ to be estimated in $\mathsf{BPP}^{\mathsf{NP}}$.

Besides this conceptual step, the proof of Theorem 3 also contains a technical component that might find other applications in quantum information. This is that, if we choose an $m \times m$ unitary matrix U randomly according to the Haar measure, then *any $n \times n$ submatrix of U will be close in variation distance to a matrix of i.i.d. Gaussians, provided that $n \leq m^{1/6}$.* Indeed, the fact that i.i.d. Gaussian matrices naturally arise as submatrices of Haar unitaries is the reason why we are so interested in Gaussian matrices in this paper, rather than Bernoulli matrices or other well-studied ensembles.

In our view, Theorem 3 already shows that fast, approximate classical simulation of boson computers would have a surprising complexity consequence. For notice that, if $X \sim \mathcal{N}(0,1)^{n\times n}_{\mathbb{C}}$ is a complex Gaussian matrix, then $\operatorname{Per}(X)$ is a sum of $n!$ complex terms, almost all of which usually cancel each other out, leaving only a tiny residue exponentially smaller than $n!$. *A priori*, there seems to be little reason to expect that residue to be approximable in the polynomial hierarchy, let alone in $\mathsf{BPP}^{\mathsf{NP}}$.

1.2.3 The Permanents of Gaussian Matrices

One could go further, though, and speculate that estimating $\text{Per}(X)$ for Gaussian X is actually $\#\mathsf{P}$-hard. We call this the *Permanent-of-Gaussians Conjecture*, or PGC.[3] We prefer to state the PGC using a more "natural" variant of the Gaussian Permanent Estimation problem than $|\text{GPE}|_{\pm}^2$. The more natural variant talks about estimating $\text{Per}(X)$ itself, rather than $|\text{Per}(X)|^2$, and also asks for a *multiplicative* rather than additive approximation.

Problem 4 $(\text{GPE}_\times)$. *Given as input a matrix $X \sim \mathcal{N}(0,1)_{\mathbb{C}}^{n\times n}$ of i.i.d. Gaussians, together with error bounds $\varepsilon, \delta > 0$, estimate $\text{Per}(X)$ to within error $\pm\varepsilon \cdot |\text{Per}(X)|$, with probability at least $1-\delta$ over X, in $\text{poly}(n, 1/\varepsilon, 1/\delta)$ time.*

Then the main complexity-theoretic challenge we offer is to prove or disprove the following:

Conjecture 5. (Permanent-of-Gaussians Conjecture or PGC) $\text{GPE}_\times$ *is $\#\mathsf{P}$-hard. In other words, if $\mathcal{O}$ is any oracle that solves $\text{GPE}_\times$, then $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{BPP}^{\mathcal{O}}$.*

Of course, a question arises as to whether one can bridge the gap between the $|\text{GPE}|_{\pm}^2$ problem that appears in Theorem 3, and the more "natural" $\text{GPE}_\times$ problem used in Conjecture 5. We are able to do so assuming *another* conjecture, this one an extremely plausible anti-concentration bound for the permanents of Gaussian random matrices.

Conjecture 6. (Permanent Anti-Concentration Conjecture or PACC) *There exists a polynomial p such that for all n and $\delta > 0$,*

$$\Pr_{X\sim\mathcal{N}(0,1)_{\mathbb{C}}^{n\times n}}\left[|\text{Per}(X)| < \frac{\sqrt{n!}}{p(n,1/\delta)}\right] < \delta.$$

In the full version, we give a complicated reduction that proves the following:

Theorem 7. *Suppose the PACC holds. Then $|\text{GPE}|_{\pm}^2$ and $\text{GPE}_\times$ are polynomial-time equivalent.*

Figure 2 summarizes the overall structure of our hardness argument for approximate BosonSampling.

The rest of our main results aim at a better understanding of Conjectures 5 and 6.

First, we give considerable evidence for the Permanent Anti-Concentration Conjecture. This includes numerical results (see Figure 3); a weaker anti-concentration bound for the permanent recently proved by Tao and Vu [32]; another weaker bound that we prove; and the analogue of Conjecture 6 for the determinant.

Next, we examine the less certain state of affairs regarding the Permanent-of-Gaussians Conjecture. On the one hand, we extend the random self-reducibility of permanents over finite fields proved by Lipton [25], to show that *exactly* computing the permanent of *most* Gaussian matrices $X \sim \mathcal{N}(0,1)_{\mathbb{C}}^{n\times n}$ is $\#\mathsf{P}$-hard. On the other hand, we also show that extending this result further, to show that *approximating* $\text{Per}(X)$ for Gaussian X is $\#\mathsf{P}$-hard, will require going beyond Lipton's polynomial interpolation technique in a fundamental way.

[3]The name is a pun on the well-known Unique Games Conjecture (UGC), which says that a certain approximation problem that "ought" to be NP-hard really *is* NP-hard.

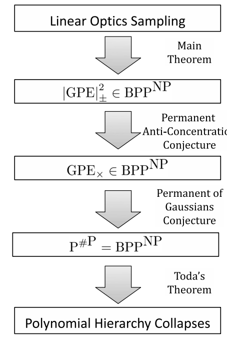

Figure 2: Summary of our hardness argument (modulo conjectures).

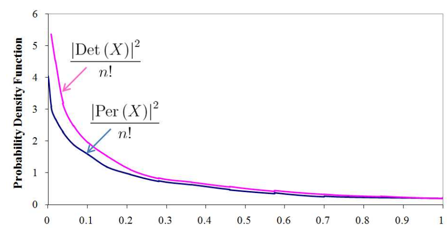

Figure 3: Probability density functions of $D_n := |\text{Det}(X)|^2/n!$ and $P_n := |\text{Per}(X)|^2/n!$, where $X \sim \mathcal{G}^{n\times n}$ is a Gaussian random matrix and $n = 6$. Note that $\text{E}[D_n] = \text{E}[P_n] = 1$ and that the tails continue infinitely to the right.

Let us mention a few additional results that are in the full version. First, we present two remarkable algorithms due to Gurvits [16] (with Gurvits's kind permission) for solving certain problems related to linear-optical networks in classical polynomial time. We also explain why these algorithms do not conflict with our hardness conjecture. Second, we prove a useful fact that is implicit in our proof of Theorem 3, but seems to deserve its own treatment. This is that, if we have n identical bosons scattered among $m \gg n^2$ locations, with no two bosons in the same location, and if we apply a Haar-random $m \times m$ unitary transformation U and then measure the number of bosons in each location, with high probability we will *still* not find two bosons in the same location. In other words, at least asymptotically, the birthday paradox works the same way for identical bosons as for classical particles, in spite of bosons' well-known tendency to cluster in the same state.

1.3 Experimental Implications

An important motivation for our results is that they immediately suggest a linear-optics experiment, which would use simple optical elements (beamsplitters and phaseshifters) to induce a Haar-random $m \times m$ unitary transformation U on an input state of n photons, and would then check that the probabilities of various final states of the photons correspond to the permanents of $n \times n$ submatrices of U, as predicted by quantum mechanics. Were such an experiment successfully scaled to large numbers of photons n, Theorem 3 asserts that no polynomial-time classical algorithm could simulate the experiment even approximately, unless $|\text{GPE}|_{\pm}^2 \in \mathsf{BPP}^{\mathsf{NP}}$.

Of course, the question arises of how large n has to be before one can draw interesting conclusions. An obvious difficulty is that *no* finite experiment can hope to render a decisive verdict on the Extended Church-Turing Thesis, since the ECT is a statement about the asymptotic limit as $n \to \infty$. Indeed, this problem is actually *worse* for us than for (say) Shor's algorithm, since unlike with FACTORING, we do not believe there is any NP witness for BOSONSAMPLING. In other words, if n is large enough that a classical computer cannot solve BOSONSAMPLING, then n is probably *also* large enough that a classical computer cannot even verify that a quantum computer is solving BOSONSAMPLING correctly.

Yet while this sounds discouraging, it is not really an issue from the perspective of near-term experiments. For the foreseeable future, n being *too large* is likely to be the least of one's problems! If one could implement our experiment with (say) $20 \leq n \leq 30$, then certainly a classical computer could verify the answers—but at the same time, one would be getting direct evidence that a quantum computer could efficiently solve an "interestingly difficult" problem, one for which the best-known classical algorithms require many millions of operations. While *disproving* the Extended Church-Turing Thesis is formally impossible, such an experiment would arguably constitute the strongest evidence against the ECT to date.

But the reader might be wondering: *what, if any, are the advantages of doing our experiment, as opposed simply to building a somewhat larger "conventional" quantum computer, able (for example) to factor* 10*-digit numbers using Shor's algorithm?* While a full answer to this question will need to await detailed analysis by experimentalists, let us mention four aspects of BOSONSAMPLING that might make it attractive for quantum computing experiments.

(1) Our proposal does not require any explicit *coupling* between pairs of photons. It therefore bypasses what has long been seen as one of the central technological obstacles to building a scalable quantum computer: namely, how to make arbitrary pairs of particles "talk to each other" (e.g., via two-qubit gates), in a manner that still preserves the particles' coherence. One might ask how there is any possibility of a quantum speedup, if the particles are never entangled. The answer is that, because of the way boson statistics work, every two identical photons are *somewhat* entangled "for free," in the sense that the amplitude for any process involving both photons includes contributions in which the photons swap their states. This "free" entanglement is the only kind that our model ever uses.

(2) Photons traveling through linear-optical networks are known to have some of the best coherence properties of any quantum system accessible to current experiments. From a "traditional" quantum computing standpoint, the disadvantages of photons are that they have no direct coupling to one another, and also that they are extremely difficult to store (they are, after all, traveling at the speed of light). There have been ingenious proposals for working around these problems, including the schemes of Knill, Laflamme, and Milburn [22] and Gottesman, Kitaev, and Preskill [14], both of which require the additional resource of *adaptive measurements*. By contrast, rather than trying to remedy photons' disadvantages as qubits, our proposal simply never uses photons as qubits at all, and thereby gets the coherence advantages of linear optics without having to address the disadvantages.

(3) To implement Shor's algorithm, one needs to perform modular arithmetic on a coherent superposition of integers encoded in binary. Unfortunately, this requirement causes significant constant blowups, and helps to explain why the "world record" for implementations of Shor's algorithm is still the factoring of 15 into 3×5, first demonstrated in 2001 [39]. By contrast, because the BOSONSAMPLING problem is so close to the "native physics" of linear-optical networks, an n-photon experiment corresponds directly to a problem instance of size n, which involves the permanents of $n \times n$ matrices. This raises the hope that, using current technology, one could sample quantum-mechanically from a distribution in which the probabilities depended (for example) on the permanents of 10×10 matrices of complex numbers.

(4) The resources that our experiment *does* demand—including reliable single-photon sources and photodetector arrays—are ones that experimentalists, for their own reasons, have devoted large and successful efforts to improving within the past decade. We see every reason to expect further improvements.

In implementing our experiment, the central difficulty is likely to be getting a reasonably-large probability of an n-*photon coincidence*: that is, of all n photons arriving at the photodetectors at the same time (or rather, within a short enough time interval that interference is seen). If the photons arrive at different times, then they effectively become *distinguishable* particles, and the experiment no longer solves the BOSONSAMPLING problem. Of course, one solution is simply to repeat the experiment many times, then *postselect* on the n-photon coincidences. However, if the probability of an n-photon coincidence decreases exponentially with n, then this "solution" has obvious scalability problems.

If one could scale our experiment to moderately large values of n (say, 10 or 20), without the probability of an n-photon coincidence falling off dramatically, then our experiment would raise the exciting possibility of doing an interestingly-large quantum computation without any need for explicit quantum error-correction. Whether or not this is feasible is the main open problem we leave for experimentalists.

The full version goes into more detail about the physical resource requirements for our proposed experiment, as well as how one would interpret the results. We also show there that the size and depth of the linear-optical network needed for our experiment can both be improved by polynomial factors over the naïve bounds.

1.4 Related Work

By necessity, this paper brings together many ideas from quantum computing, optical physics, and computational complexity. In this section, we try to survey the large relevant literature, organizing it into eight categories.

Quantum computing with linear optics. There is a huge body of work, both experimental and theoretical, on quantum computing with linear optics. Much of that work builds on a seminal 2001 result of Knill, Laflamme, and Milburn [22], showing that linear optics combined with *adaptive measurements* is universal for quantum computation. It is largely because of that result—as well as an alternative scheme due to Gottesman, Kitaev, and Preskill

[14]—that linear optics is considered a viable proposal for building a universal quantum computer.[4]

In the opposite direction, several interesting classes of linear-optics experiments are known to be efficiently simulable on a classical computer. First, it is easy to show that a linear-optical network with *coherent-state inputs*, and possibly-adaptive *demolition measurements* in the photon-number basis, can be simulated in classical polynomial time. Intuitively, a coherent state—the output of a standard laser—is a superposition over different numbers of photons that behaves essentially like a classical wave. Also, a demolition measurement is one that only returns the classical measurement outcome, and not the post-measurement quantum state.

Second, Bartlett and Sanders [6] showed that a linear-optical network with *Gaussian-state inputs* and possibly-adaptive *Gaussian nondemolition measurements* can be simulated in classical polynomial time. Here a Gaussian state is an entangled generalization of a coherent state, and is also relatively easy to produce experimentally. A Gaussian nondemolition measurement is a measurement of a Gaussian state whose outcome is also Gaussian. This result of Bartlett and Sanders can be seen as the linear-optical analogue of the *Gottesman-Knill Theorem* (see [4]).

Third, Gurvits [16] showed that, in any n-photon linear-optical experiment, the probability of measuring a particular basis state can be estimated to within $\pm\varepsilon$ additive error in $\operatorname{poly}(n, 1/\varepsilon)$ time.[5] He also showed that the marginal distribution over any k photon modes can be computed deterministically in $n^{O(k)}$ time. We discuss Gurvits's results in detail in the full version.

Our model seems to be intermediate between the extremes of quantum universality and classical simulability. Unlike Knill et al. [22], we do not allow adaptive measurements, and as a result, our model is probably not BQP-complete. On the other hand, unlike Bartlett and Sanders, we *do* allow single-photon inputs and (nonadaptive) photon-number measurements; and unlike Gurvits [16], we consider sampling from the joint distribution over all $\operatorname{poly}(n)$ photon modes. Our main result gives evidence that the resulting model, while possibly easier to implement than a universal quantum computer, is still intractable to simulate classically.

Intermediate models of quantum computation. By now, several interesting models of quantum computation have been proposed that are neither known to be universal for BQP, nor simulable in classical polynomial time. A few examples, besides the ones mentioned elsewhere in the paper, are the "one-clean-qubit" model of Knill and Laflamme [21]; the permutational quantum computing model of Jordan [19]; and stabilizer circuits with non-stabilizer initial states (such as $\cos\frac{\pi}{8}|0\rangle + \sin\frac{\pi}{8}|0\rangle$) and nonadaptive measurements [4]. The noninteracting-boson model is another addition to this list.

[4] An earlier proposal for building a universal optical quantum computer was to use *nonlinear optics*: in other words, explicit entangling interactions between pairs of photons. (See Nielsen and Chuang [26] for discussion.) The problem is that, at least at low energies, photons *have* no direct coupling to one another. It is therefore necessary to use other particles as intermediaries, which greatly increases decoherence, and negates many of the advantages of using photons in the first place.

[5] While beautiful, this result is of limited use in practice—since in a typical linear-optics experiment, the probability p of measuring any *specific* basis state is so small that 0 is a good additive estimate to p.

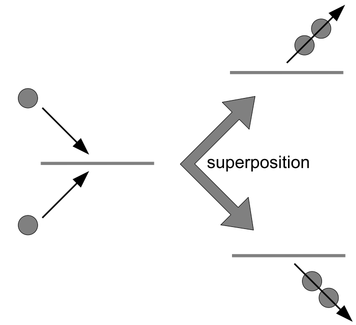

Figure 4: The Hong-Ou-Mandel dip.

The Hong-Ou-Mandel dip. In 1987, Hong, Ou, and Mandel [17] performed a now-standard experiment that, in essence, directly confirms that *two*-photon amplitudes correspond to 2×2 permanents in the way predicted by quantum mechanics. In more detail, two identical photons, which were initially in different locations, become *correlated* after passing through a beamsplitter that applies the Hadamard transformation (see Figure 4). Formally, the basis state $|1,1\rangle$ evolves to

$$\frac{|2,0\rangle - |0,2\rangle}{\sqrt{2}},$$

so that a subsequent measurement reveals either both photons in the first location or else both photons in the second location. The reason the amplitude of the basis state $|1,1\rangle$ "dips" to 0 is that

$$\operatorname{Per}\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} = 0,$$

and hence there is destructive interference between the two paths mapping $|1,1\rangle$ to itself. From an experimental perspective, what we are asking for could be seen as a generalization of this "Hong-Ou-Mandel dip" to the n-photon case, where n is as large as possible. Lim and Beige [24] previously proposed an n-photon generalization of the Hong-Ou-Mandel dip, but without the computational complexity motivation.

Bosons and the permanent. *Bosons* are one of the two basic types of particle in the universe; they include photons and the carriers of nuclear forces. It has been known since work by Caianiello [9] in 1953 (if not earlier) that the amplitudes for n-boson processes can be written as the permanents of $n\times n$ matrices. Meanwhile, Valiant [37] proved in 1979 that the permanent is $\#\mathsf{P}$-complete. Interestingly, according to Valiant (personal communication), he and others put these two facts together immediately, and wondered what they might mean for the computational complexity of simulating bosonic systems. To our knowledge, however, the first authors to discuss this question in print were Troyansky and Tishby [36] in 1996. Given an arbitrary matrix $A \in \mathbb{C}^{n\times n}$, these authors showed how to construct a quantum observable with expectation value equal to

Per(A). However, they correctly pointed out that this did not imply a polynomial-time quantum algorithm to *calculate* Per(A), since the variance of their observable was large enough that exponentially many samples would be needed. Later, Scheel [27] explained how permanents arise as amplitudes in linear-optical networks, and noted that calculations involving linear-optical networks might be intractable because the permanent is #P-complete.

Fermions and the determinant. Besides bosons, the other basic particles in the universe are *fermions*; these include matter particles such as quarks and electrons. Remarkably, the amplitudes for n-fermion processes are given not by permanents but by *determinants* of $n \times n$ matrices. Despite the similarity of their definitions, it is well-known that the permanent and determinant differ dramatically in their computational properties; the former is #P-complete while the latter is in P. In a lecture in 2000, Wigderson called attention to this striking connection between the boson/fermion dichotomy of physics and the permanent/determinant dichotomy of computer science. He joked that, between bosons and fermions, "the bosons got the harder job." One could view this paper as a formalization of Wigderson's joke.

To be fair, *half* the work of formalizing Wigderson's joke has already been carried out. In 2002, Valiant [38] defined a beautiful subclass of quantum circuits called *matchgate circuits*, and showed that these circuits could be efficiently simulated classically, via a nontrivial algorithm that ultimately relied on computing determinants.[6] Shortly afterward, Terhal and DiVincenzo [33] (see also Knill [20]) pointed out that matchgate circuits were equivalent to systems of noninteracting fermions[7]: in that sense, one could say Valiant had "rediscovered fermions"! Indeed, Valiant's matchgate model can be seen as the direct counterpart of the model studied in this paper, but with noninteracting fermions in place of noninteracting bosons.[8] At a very high level, Valiant's model is easy to simulate classically because the determinant is in P, whereas our model is hard to simulate because the permanent is #P-complete.

Ironically, when the *quantum Monte Carlo method* is used to approximate the ground states of many-body systems, the computational situation regarding bosons and fermions is reversed. Bosonic ground states tend to be *easy* to approximate because one can exploit non-negativity, while fermionic ground states tend to be *hard* to approximate because of cancellations between positive and negative terms, what physicists call "the sign problem."

Quantum computing and #P-complete problems. Since amplitudes in quantum mechanics are the sums of exponentially many complex numbers, it is natural to look for some formal connection between quantum computing and the class #P of counting problems. In 1993, Bernstein and Vazirani [7] proved that $\mathsf{BQP} \subseteq \mathsf{P}^{\#\mathsf{P}}$. However, this result says only that #P is an *upper* bound on the power of quantum computation, so the question arises of whether solving #P-complete problems is in any sense *necessary* for simulating quantum mechanics.

To be clear, we do not expect that $\mathsf{BQP} = \mathsf{P}^{\#\mathsf{P}}$; indeed, it would be a scientific revolution even if BQP were found to contain NP. However, already in 1999, Fenner, Green, Homer, and Pruim [12] noticed that, if we ask more refined questions about a quantum circuit than whether it accepts with probability greater than 2/3 or less than 1/3, then we can quickly encounter #P-completeness. In particular, Fenner et al. showed that deciding whether a quantum circuit accepts with *nonzero or zero* probability is complete for the complexity class $\mathsf{coC_{=}P}$. Since $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{NP}^{\mathsf{coC_{=}P}}$, this means that the problem is #P-hard under nondeterministic reductions.

Later, Aaronson [1] defined the class PostBQP, or quantum polynomial-time with *postselection* on possibly exponentially-unlikely measurement outcomes. He showed that PostBQP is equal to the classical class PP. Since $\mathsf{P}^{\mathsf{PP}} = \mathsf{P}^{\#\mathsf{P}}$, this says that quantum computers with postselection can already solve #P-complete problems. Following [8], in the full version we use the PostBQP = PP theorem to give an alternative proof of Theorem 1, which does not require using the #P-completeness of the permanent.

Quantum speedups for sampling and search problems. Ultimately, we want a hardness result for simulating *real* quantum experiments, rather than postselected ones. To achieve that, a crucial step in this paper is to switch attention from *decision* problems to *sampling* and *search* problems. The value of that step in a quantum computing context was recognized in several previous works.

In 2008, Shepherd and Bremner [28] defined and studied a fascinating subclass of quantum computations, which they called "commuting" or "temporally-unstructured." Their model is probably not universal for BQP, and there is no known example of a decision problem solvable by their model that is not also in BPP. However, if we consider *sampling* problems or interactive protocols, then Shepherd and Bremner plausibly argued (without formal evidence) that their model might be hard to simulate classically.

Recently, and independently of us, Bremner, Jozsa, and Shepherd [8] showed that commuting quantum computers can sample from probability distributions that cannot be efficiently sampled classically, unless $\mathsf{PP} = \mathsf{BPP_{path}}$ and hence the polynomial hierarchy collapses to the third level. This is analogous to our Theorem 1, except with commuting quantum computations instead of noninteracting-boson ones.

Previously, in 2002, Terhal and DiVincenzo [34] showed that constant-depth quantum circuits can sample from probability distributions that cannot be efficiently sampled by a classical computer, unless $\mathsf{BQP} \subseteq \mathsf{AM}$. By using our arguments and Bremner et al.'s [8], it is not hard to strengthen Terhal and DiVincenzo's conclusion, to show that exact classical simulation of their model would also imply $\mathsf{PP} = \mathsf{PostBQP} = \mathsf{BPP_{path}}$, and hence that the polynomial hierarchy collapses.

However, all of these results (including our Theorem 1) have the drawback that they only address sampling from *exactly* the same distribution $\mathcal{D}$ as the quantum algorithm—or

[6] Or rather, a closely-related matrix function called the Pfaffian.

[7] Strictly speaking, *unitary* matchgate circuits are equivalent to noninteracting fermions (Valiant also studied matchgates that violated unitarity).

[8] However, the noninteracting-boson model is somewhat more complicated to define, since one can have multiple bosons occupying the same mode, whereas fermions are prohibited from this by the Pauli exclusion principle. This is why the basis states in our model are lists of nonnegative integers, whereas the basis states in Valiant's model are binary strings.

at least, from some distribution in which all the probabilities are multiplicatively close to the ideal ones. Indeed, in these results, everything hinges on the #P-completeness of estimating a single, exponentially-small probability p. For this reason, such results might be considered "cheats": presumably not even the quantum device *itself* can sample perfectly from the ideal distribution $\mathcal{D}$! What if we allow "realistic noise," so that one only needs to sample from some probability distribution $\mathcal{D}'$ that is $1/\operatorname{poly}(n)$-*close* to $\mathcal{D}$ in total variation distance? Is that *still* a classically-intractable problem? This is the question we took as our starting point.

Oracle results. We know of one previous work that addressed the hardness of sampling *approximately* from a quantum computer's output distribution. In 2010, Aaronson [2] showed that, relative to a random oracle A, quantum computers can sample from probability distributions $\mathcal{D}$ that are not even *approximately* samplable in $\mathsf{BPP}^{\mathsf{PH}^A}$ (that is, by classical computers with oracles for the polynomial hierarchy). Relative to a random oracle A, quantum computers can also solve *search* problems not in $\mathsf{BPP}^{\mathsf{PH}^A}$. The point of these results was to give the first formal evidence that quantum computers have "capabilities outside PH."

For us, though, what is more relevant is a striking feature of the *proofs* of these results. Namely, they showed that, if the sampling and search problems in question were in $\mathsf{BPP}^{\mathsf{PH}^A}$, then (via a nonuniform, nondeterministic reduction) one could extract small constant-depth circuits for the 2^n-bit MAJORITY function, thereby violating the celebrated circuit lower bounds of Håstad [31] and others. What made this surprising was that the 2^n-bit MAJORITY function is #P-complete.[9] In other words, even though there is no evidence that quantum computers can solve #P-complete problems, somehow we managed to *prove the hardness of simulating a* BQP *machine by using the hardness of* #P.

Of course, a drawback of Aaronson's results [2] is that they were relative to an oracle. However, just like Simon's oracle algorithm [30] led shortly afterward to Shor's algorithm [29], so too in this case one could hope to "reify the oracle": that is, find a real, unrelativized problem with the same behavior that the oracle problem illustrated more abstractly. That is what we do here.

2. OPEN PROBLEMS

The most exciting challenge we leave is to *do* the experiment we propose, whether in linear optics or in other physical systems that contain excitations that behave as identical bosons. If successful, such an experiment has the potential to provide the strongest evidence to date for violation of the Extended Church-Turing Thesis in nature.

We now list a few theoretical open problems.

(1) The most obvious problem is to prove Conjecture 5: that approximating the permanent of a matrix of i.i.d. Gaussian entries is #P-hard. Failing that, can we prove #P-hardness for *any* problem with a similar "flavor" (roughly speaking, an average-case approximate counting problem over $\mathbb{R}$ or $\mathbb{C}$)? Can we at least find evidence that such a problem is not in $\mathsf{BPP}^{\mathsf{NP}}$?

[9] Here we are abusing terminology (but only slightly) by speaking about the #P-completeness of an oracle problem. Also, strictly speaking we mean PP-complete—but since $\mathsf{P}^{\mathsf{PP}} = \mathsf{P}^{\#\mathsf{P}}$, the distinction is unimportant here.

(2) Another obvious problem is to prove Conjecture 6, that $|\operatorname{Per}(X)|$ almost always exceeds $\sqrt{n!}/\operatorname{poly}(n)$ for Gaussian random matrices $X \sim \mathcal{N}(0,1)_{\mathbb{C}}^{n\times n}$. Failing that, *any* progress on understanding the distribution of $\operatorname{Per}(X)$ for Gaussian X would be interesting.

(3) How does the noninteracting-boson model relate to other models of computation that are believed to be intermediate between BPP and BQP? To give one concrete question, can every boson computation be simulated by a qubit-based quantum circuit of logarithmic depth?

(4) Using quantum fault-tolerance techniques, can one decrease the effective error in our experiment to $1/\exp(n)$—thereby obviating the need for the mathematical work we do in this paper to handle $1/\operatorname{poly}(n)$ error in variation distance? Note that, *if* one had the resources for universal quantum computation, then one could easily combine our experiment with standard fault-tolerance schemes, which are known to push the effective error down to $1/\exp(n)$ using $\operatorname{poly}(n)$ computational overhead. So the interesting question is whether one can make our experiment fault-tolerant using *fewer* resources than are needed for universal quantum computing—and in particular, whether one can do so using linear optics alone.

(5) Can we give evidence against not merely an FPTAS (Fully Polynomial Time Approximation Scheme) for the BOSONSAMPLING problem, but an approximate sampling algorithm that works for some *fixed* error $\varepsilon > 1/\operatorname{poly}(n)$?

(6) For what other interesting quantum systems, besides linear optics, do analogues of our hardness results hold? As mentioned in Section 1.4, the beautiful work of Bremner, Jozsa, and Shepherd [8] shows that exact simulation of "commuting quantum computations" in classical polynomial time would collapse the polynomial hierarchy. What can we say about *approximate* classical simulation of their model?

(7) In this work, we showed that unlikely complexity consequences would follow if classical computers could simulate quantum computers on all *sampling* or *search* problems: that is, that SampP = SampBQP or FBPP = FBQP. An obvious question that remains is, what about *decision* problems? Can we derive some unlikely collapse of classical complexity classes from the assumption that P = BQP or PromiseP = PromiseBQP?

(8) To what extent do our results relativize? One immediate problem is that we do not even know what it *means* to relativize a boson computer! Thus, let us state our results in terms of universal quantum computers instead. In that case, our exact result, Theorem 1, says that $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{BPP}^{\mathsf{NP}^{\mathcal{O}}}$ for every oracle $\mathcal{O}$ that samples exactly from the output distribution of a given quantum circuit. The proof of Theorem 1 is easily seen to relativize. However, we do not know the situation with our *approximate* result, Theorem 3. More concretely, does there exist an oracle A relative to which FBPP = FBQP but PH is infinite? Currently, the closest we have to this is a powerful result of Fortnow and Rogers [13], which gives an oracle A relative to which P = BQP but PH is infinite. However, it is not even known how to extend their construction to get an oracle A relative to which PromiseP = PromiseBQP but PH is infinite.

(9) Is there any plausible candidate for a *decision* problem that is efficiently solvable by a boson computer, but not by a classical computer?

(10) It is not obvious how to convince a skeptic that a quantum computer is really solving the BOSONSAMPLING

problem in a scalable way. This is because, unlike with (say) FACTORING, neither BOSONSAMPLING nor any related problem seems to be in NP. How far can we remedy this? For example, can a prover with a BOSONSAMPLING oracle prove any nontrivial statements to a BPP verifier via an interactive protocol?

(11) Is there a polynomial-time classical algorithm to sample from a probability distribution $\mathcal{D}'$ that *cannot be efficiently distinguished* from the distribution $\mathcal{D}$ sampled by a boson computer?

3. ACKNOWLEDGMENTS

We thank Boris Alexeev, Carlo Beenakker, Andy Drucker, Oded Goldreich, Aram Harrow, Matt Hastings, Gil Kalai, Greg Kuperberg, Anthony Leverrier, Masoud Mohseni, Terry Rudolph, Raul Garcia-Patron Sanchez, Barry Sanders, Madhu Sudan, Terry Tao, Barbara Terhal, Lev Vaidman, Leslie Valiant, and Avi Wigderson for helpful discussions. We especially thank Leonid Gurvits for explaining his polynomial formalism and algorithmic results, and Mick Bremner and Richard Jozsa for discussions of their work [8].

4. REFERENCES

[1] S. Aaronson. Quantum computing, postselection, and probabilistic polynomial-time. *Proc. Roy. Soc. London*, A461(2063):3473–3482, 2005. quant-ph/0412187.

[2] S. Aaronson. BQP and the polynomial hierarchy. In *Proc. ACM STOC*, 2010. arXiv:0910.4698.

[3] S. Aaronson. The equivalence of sampling and searching. In *Proc. Computer Science Symposium in Russia (CSR)*, 2011. arXiv:1009.5104, ECCC TR10-128.

[4] S. Aaronson and D. Gottesman. Improved simulation of stabilizer circuits. *Phys. Rev. A*, 70(052328), 2004. quant-ph/0406196.

[5] D. Aharonov and M. Ben-Or. Fault-tolerant quantum computation with constant error. In *Proc. ACM STOC*, pages 176–188, 1997. quant-ph/9906129.

[6] S. D. Bartlett and B. C. Sanders. Requirement for quantum computation. *Journal of Modern Optics*, 50:2331–2340, 2003. quant-ph/0302125.

[7] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM J. Comput.*, 26(5):1411–1473, 1997. First appeared in ACM STOC 1993.

[8] M. Bremner, R. Jozsa, and D. Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proc. Roy. Soc. London*, A467(2126):459–472, 2010. arXiv:1005.1407.

[9] E. R. Caianiello. On quantum field theory, 1: explicit solution of Dyson's equation in electrodynamics without use of Feynman graphs. *Nuovo Cimento*, 10:1634–1652, 1953.

[10] R. Cleve and J. Watrous. Fast parallel circuits for the quantum Fourier transform. In *Proc. IEEE FOCS*, pages 526–536, 2000. quant-ph/0006004.

[11] B. Fefferman and C. Umans. Pseudorandom generators and the BQP vs. PH problem. www.cs.caltech.edu/˜umans/papers/FU10.pdf, 2010.

[12] S. Fenner, F. Green, S. Homer, and R. Pruim. Determining acceptance possibility for a quantum computation is hard for the polynomial hierarchy. *Proc. Roy. Soc. London*, A455:3953–3966, 1999. quant-ph/9812056.

[13] L. Fortnow and J. Rogers. Complexity limitations on quantum computation. *J. Comput. Sys. Sci.*, 59(2):240–252, 1999. cs.CC/9811023.

[14] D. Gottesman, A. Kitaev, and J. Preskill. Encoding a qubit in an oscillator. *Phys. Rev. A*, (64:012310), 2001. quant-ph/0008040.

[15] L. Gurvits. Classical complexity and quantum entanglement. *J. Comput. Sys. Sci.*, 69(3):448–484, 2004. quant-ph/0201022.

[16] L. Gurvits. On the complexity of mixed discriminants and related problems. In *Mathematical Foundations of Computer Science*, pages 447–458, 2005.

[17] C. K. Hong, Z. Y. Ou, and L. Mandel. Measurement of subpicosecond time intervals between two photons by interference. *Phys. Rev. Lett.*, 59(18):2044–2046, 1987.

[18] M. Jerrum, A. Sinclair, and E. Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries. *J. ACM*, 51(4):671–697, 2004. Earlier version in STOC'2001.

[19] S. P. Jordan. Permutational quantum computing. *Quantum Information and Computation*, 10(5/6):470–497, 2010. arXiv:0906.2508.

[20] E. Knill. Fermionic linear optics and matchgates. quant-ph/0108033, 2001.

[21] E. Knill and R. Laflamme. Power of one bit of quantum information. *Phys. Rev. Lett.*, 81(25):5672–5675, 1998. quant-ph/9802037.

[22] E. Knill, R. Laflamme, and G. J. Milburn. A scheme for efficient quantum computation with linear optics. *Nature*, 409:46–52, 2001. See also quant-ph/0006088.

[23] E. Knill, R. Laflamme, and W. Zurek. Resilient quantum computation. *Science*, 279:342–345, 1998. quant-ph/9702058.

[24] Y. L. Lim and A. Beige. Generalized Hong-Ou-Mandel experiments with bosons and fermions. *New J. Phys.*, 7(155), 2005. quant-ph/0505034.

[25] R. J. Lipton. New directions in testing. In *Distributed Computing and Cryptography*, pages 191–202. AMS, 1991.

[26] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[27] S. Scheel. Permanents in linear optical networks. quant-ph/0406127, 2004.

[28] D. Shepherd and M. J. Bremner. Temporally unstructured quantum computation. *Proc. Roy. Soc. London*, A465(2105):1413–1439, 2009. arXiv:0809.0847.

[29] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997. Earlier version in IEEE FOCS 1994. quant-ph/9508027.

[30] D. Simon. On the power of quantum computation. In *Proc. IEEE FOCS*, pages 116–123, 1994.

[31] J. Håstad. *Computational Limitations for Small Depth Circuits*. MIT Press, 1987.

[32] T. Tao and V. Vu. On the permanent of random Bernoulli matrices. *Advances in Mathematics*, 220(3):657–669, 2009. arXiv:0804.2362.

[33] B. M. Terhal and D. P. DiVincenzo. Classical simulation of noninteracting-fermion quantum circuits. *Phys. Rev. A*, 65(032325), 2002. quant-ph/0108010.

[34] B. M. Terhal and D. P. DiVincenzo. Adaptive quantum computation, constant-depth circuits and Arthur-Merlin games. *Quantum Information and Computation*, 4(2):134–145, 2004. quant-ph/0205133.

[35] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991.

[36] L. Troyansky and N. Tishby. Permanent uncertainty: On the quantum evaluation of the determinant and the permanent of a matrix. In *Proceedings of PhysComp*, 1996.

[37] L. G. Valiant. The complexity of computing the permanent. *Theoretical Comput. Sci.*, 8(2):189–201, 1979.

[38] L. G. Valiant. Quantum circuits that can be simulated classically in polynomial time. *SIAM J. Comput.*, 31(4):1229–1254, 2002. Earlier version in STOC'2001.

[39] L. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414:883–887, 2001. quant-ph/0112176.