THEORY OF COMPUTING, Volume 9 (4), 2013, pp. 143–252
www.theoryofcomputing.org

The Computational Complexity of Linear Optics*

Scott Aaronson† Alex Arkhipov‡

Received December 2, 2011; Revised December 30, 2012; Published February 9, 2013

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*

*An extended abstract of this article appeared in the Proceedings of STOC 2011.

†MIT. 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, a TIBCO chair, and a Sloan Fellowship.

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

ACM Classification: F.1.2, F.1.3

AMS Classification: 81P68, 68Q17, 68Q15

Key words and phrases: #P, bosons, BQP, linear optics, permanent, polynomial hierarchy, random self-reducibility, sampling

 DOI: 10.4086/toc.2013.v009a004

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 $|\mathrm{Per}(A)| \geq \sqrt{n!}/\mathrm{poly}(n)$ 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.

Contents

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 [60], 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 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 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 [11] defined the class BQP of quantumly feasible problems, it has been a dream of quantum computing theory to show (for example) that, if $\mathsf{BPP} = \mathsf{BQP}$ 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 [7, 44], 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 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

[1] One caveat is a result of Cleve and Watrous [17], 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).

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?

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

1.1 Our model

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 (or "mode"). In Section 3, 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 minor 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.

Our model does not require any explicit *interaction* between pairs of bosons. 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). At first, this aspect of our model might seem paradoxical: if there are no interactions, how can we ever generate *entanglement* between pairs of bosons? And if there is no entanglement, how can there

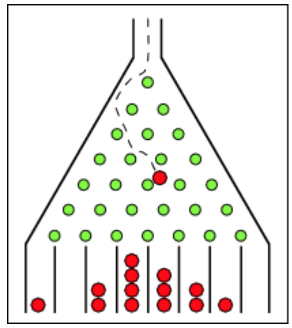

Figure 1: Galton's board, a simple "computer" to output samples from the binomial distribution. From MathWorld, http://mathworld.wolfram.com/GaltonBoard.html

be any possibility of a quantum speedup? The resolution of this puzzle lies in the way boson statistics work, which we explain in Section 3. Briefly, the Hilbert space for n identical bosons is *not* the tensor product of n single-boson Hilbert spaces, but a slightly less-familiar object known to mathematicians as the *symmetric product*. And because of the peculiarities of the symmetric product, an n-boson state $|\psi\rangle$ can *look* entangled when expanded out in various natural ways, even if the bosons in $|\psi\rangle$ were never actually subject to entangling interactions. We discuss this subtlety in more detail in Appendix A, for readers who are interested. For now, though, it suffices to say that the "apparent," "effective," or "illusory" entanglement produced by noninteracting bosons—whatever one wants to call it!—is the only kind of "entanglement" that our computational model ever assumes or needs.

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

$$\operatorname{Per}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i\sigma(i)}\,, \tag{1.1}$$

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 [37]. 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 #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. 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 result, proved in Section 4, says the following.

Theorem 1.1. *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* $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{BPP}^{\mathsf{NP}^{\mathcal{O}}}$.

Recently, and independently of us, Bremner, Jozsa, and Shepherd [12] proved a lovely result directly analogous to Theorem 1.1, but for a *different* weak quantum computing model (based on commuting Hamiltonians rather than bosons). As we discuss later, our original proof of Theorem 1.1 is quite different from Bremner et al.'s, but for completeness, we will also give a proof of Theorem 1.1 along Bremner et al.'s lines. The main respect in which this work goes further than Bremner et al.'s is not in Theorem 1.1, but rather in our treatment of the approximate case (to be discussed in Section 1.2.2).

For now, let us focus on Theorem 1.1 and try to understand what it means. At least for a computer scientist, it is tempting to interpret Theorem 1.1 as saying that "the exact BOSONSAMPLING problem is #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 #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 1.1 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 #P-complete problems to BOSONSAMPLING makes essential use of the hypothesis that we have a *classical* BOSONSAMPLING algorithm.

Note that, even if the exact BOSONSAMPLING problem were solvable by a polynomial-time classical algorithm *with an oracle for a* PH *problem*, Theorem 1.1 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 [69]. This provides evidence that quantum computers have capabilities outside the entire polynomial hierarchy, complementing the recent evidence of Aaronson [3] and Fefferman and Umans [24].

Another point worth mentioning is that, even if the exact BOSONSAMPLING problem were solvable by a polynomial-time *nonuniform* sampling algorithm—that is, by an algorithm that could be different for each boson computer A—we would still get the conclusion $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{BPP}^{\mathsf{NP}}/\mathsf{poly}$, whence the polynomial hierarchy would collapse. This is a consequence of the existence of a "universal BOSONSAMPLING instance," which we point out in Section 4.3.

We will give two proofs of Theorem 1.1. In the first, 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 #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 the independent work of Bremner et al. [12]. Here we start with a result of Knill, Laflamme, and Milburn [43], 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 [2] 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

$$\mathsf{BPP}_{\mathsf{path}} = \mathsf{PostBPP} = \mathsf{PostBQP} = \mathsf{PP}, \tag{1.2}$$

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* #P-*complete problems*—even though solving #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, 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$\notin \mathsf{BPP}$.

As we see it, the central drawback of Theorem 1.1 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:

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

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 [4] 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 1.1 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 will 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 1.2 ($|\mathrm{GPE}|_{\pm}^{2}$)**.** 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 $|\mathrm{Per}(X)|^2$ to within additive error $\pm\varepsilon \cdot n!$ with probability at least $1-\delta$ over X in $\mathrm{poly}(n, 1/\varepsilon, 1/\delta)$ time.

Then our main result is the following.

Theorem 1.3 (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* $\mathrm{poly}(|A|, 1/\varepsilon)$ *time. Then the $|\mathrm{GPE}|_{\pm}^{2}$ problem is solvable in* $\mathsf{BPP}^{\mathsf{NP}}$*. Indeed, if we treat C as a black box, then* $|\mathrm{GPE}|_{\pm}^{2} \in \mathsf{BPP}^{\mathsf{NP}^{C}}$.

Theorem 1.3 is proved in Section 5. The key idea of the proof is to "smuggle" the $|\mathrm{GPE}|_{\pm}^{2}$ 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 $|\mathrm{Per}(X)|^2$ to be estimated in $\mathsf{BPP}^{\mathsf{NP}}$.

Incidentally, the reason why we are so interested here in the permanents of *Gaussian* matrices X rather than Bernoulli matrices or other well-studied random matrix ensembles, is that taking small submatrices of *Haar-random unitary matrices* leads in an extremely natural way to matrices of (approximately) i. i. d. Gaussian entries. This fact was already known in random matrix theory, but we provide a self-contained proof for completeness.

In our view, Theorem 1.3 already shows that fast, approximate classical simulation of boson computers would have a surprising complexity consequence. For notice that, if $X \sim \mathcal{N}(0,1)_{\mathbb{C}}^{n\times n}$ is a complex Gaussian matrix, then $\mathrm{Per}(X)$ 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 $\mathrm{Per}(X)$ for Gaussian X is actually #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 $|\mathrm{GPE}|_{\pm}^{2}$. The more natural variant talks about estimating $\mathrm{Per}(X)$ itself, rather than $|\mathrm{Per}(X)|^2$ and also asks for a *multiplicative* rather than additive approximation.

[3]The name is a pun on the well-known Unique Games Conjecture (UGC) [40], which says that a certain approximation problem that "ought" to be NP-hard really *is* NP-hard.

Problem 1.4 ($\mathrm{GPE}_\times$)**.** 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 $\mathrm{Per}(X)$ to within error $\pm\varepsilon \cdot |\mathrm{Per}(X)|$ with probability at least $1-\delta$ over X in $\mathrm{poly}(n, 1/\varepsilon, 1/\delta)$ time.

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

Conjecture 1.5 (Permanent-of-Gaussians Conjecture or PGC)**.** $\mathrm{GPE}_\times$ *is* #P*-hard. In other words, if* $\mathcal{O}$ *is any oracle that solves* $\mathrm{GPE}_\times$*, 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 $|\mathrm{GPE}|_\pm^2$ problem that appears in Theorem 1.3, and the more "natural" $\mathrm{GPE}_\times$ problem used in Conjecture 1.5. 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 1.6 (Permanent Anti-Concentration Conjecture)**.** *There exists a polynomial p such that for all n and* $\delta > 0$*,*

$$\Pr_{X\sim\mathcal{N}(0,1)_{\mathbb{C}}^{n\times n}}\left[|\mathrm{Per}(X)| < \frac{\sqrt{n!}}{p(n,1/\delta)}\right] < \delta\,. \tag{1.3}$$

In Section 7, we give a complicated reduction that proves the following:

Theorem 1.7. *Suppose the Permanent Anti-Concentration Conjecture holds. Then* $|\mathrm{GPE}|_\pm^2$ *and* $\mathrm{GPE}_\times$ *are polynomial-time equivalent (under nonadaptive reductions).*

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

The rest of the body of the paper aims at a better understanding of Conjectures 1.5 and 1.6.

First, in Section 8, we summarize the considerable evidence for the Permanent Anti-Concentration Conjecture. This includes numerical results; a weaker anti-concentration bound for the permanent recently proved by Tao and Vu [65]; another weaker bound that we prove; and the analogue of Conjecture 1.6 for the determinant.

Next, in Section 9, we discuss 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 [47], to show that *exactly* computing the permanent of *most* Gaussian matrices $X \sim \mathcal{N}(0,1)_{\mathbb{C}}^{n\times n}$ is #P-hard. On the other hand, we also show that extending this result further, to show that *approximating* $\mathrm{Per}(X)$ for Gaussian X is #P-hard, will require going beyond Lipton's polynomial interpolation technique in a fundamental way.

Two appendices give some additional results. First, in Appendix B, we present two remarkable algorithms due to Gurvits [33] (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, in Appendix C, we bring out a useful fact that was implicit in our proof of Theorem 1.3, 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.

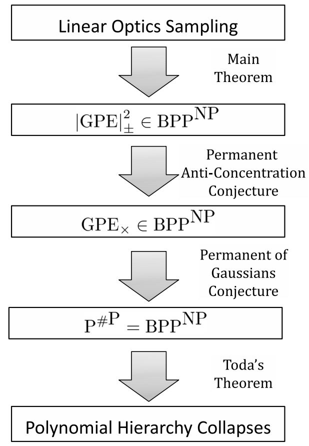

Figure 2: Summary of our hardness argument (modulo conjectures). If there exists a polynomial-time classical algorithm for approximate BOSONSAMPLING, then Theorem 1.3 says that $|\mathsf{GPE}|^2_{\pm} \in \mathsf{BPP}^{\mathsf{NP}}$. Assuming Conjecture 1.6 (the PACC), Theorem 1.7 says that this is equivalent to $\mathsf{GPE}_{\times} \in \mathsf{BPP}^{\mathsf{NP}}$. Assuming Conjecture 1.5 (the PGC), this is in turn equivalent to $\mathsf{P}^{\#\mathsf{P}} = \mathsf{BPP}^{\mathsf{NP}}$ which collapses the polynomial hierarchy by Toda's Theorem [69].

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.[4] Were such an experiment successfully scaled to large values of n Theorem 1.3 asserts that no polynomial-time classical algorithm could simulate the experiment even approximately, unless $|\mathsf{GPE}|^2_{\pm} \in \mathsf{BPP}^{\mathsf{NP}}$.

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

[4] Right before this paper was published, four groups announced the first experimental demonstrations of BosonSampling, each with 3 photons: see Broome et al. [13], Spring et al. [62], Crespi et al. [19], and Tillmann et al. [68]. Unfortunately, a full discussion of these exciting developments is outside the scope of this paper.

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.

Section 6 goes into more detail about the physical resource requirements for our proposed experiment, as well as how one would interpret the results. In Section 6, we also show 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. Complexity theorists who are not interested in the "practical side" of boson computation can safely skip Section 6, while experimentalists who are *only* interested the practical side can skip everything else.

While most further discussion of experimental issues is deferred to Section 6, there is one question we need to address now. Namely: *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, perhaps the most important advantage was already discussed in Section 1.1: our model does not require any explicit *coupling* between pairs of photons. Let us mention three other aspects of BOSONSAMPLING that might make it attractive for quantum computing experiments.

(1) 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 [43] and Gottesman, Kitaev, and Preskill [32], 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.

(2) 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 [73]. 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.

(3) 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.

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 [43], 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 [32]—that linear optics is considered a viable proposal for building a universal quantum computer.[5]

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 [9] 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 [5]).

[5] 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 [50] 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.

Third, Gurvits [33] 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 poly $(n, 1/\varepsilon)$ time.[6] 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 Appendix B.

Our model seems to be intermediate between the extremes of quantum universality and classical simulability. Unlike Knill et al. [43], 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 [33], we consider sampling from the joint distribution over all 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.

The table below summarizes what is known about the power of linear-optical quantum computers, with various combinations of physical resources, in light of this paper's results. The columns show what is achievable if the inputs are (respectively) coherent states, Gaussian states, or single-photon Fock states. The first four rows show what is achievable using measurements in the photon-number basis; such measurements might be either *adaptive* or *nonadaptive* (that is, one might or might not be able to condition future operations on the classical measurement outcomes), and also either *nondemolition* or *demolition* (that is, the post-measurement quantum state might or might not be available after the measurement). The fifth row shows what is achievable using measurements in the Gaussian basis, for any combination of adaptive/nonadaptive and demolition/nondemolition (we do not know of results that work for some combinations but not others). A "P" entry means that a given combination of resources is known to be simulable in classical polynomial time, while a "BQP" entry means it is known to suffice for universal quantum computation. "Exact sampling hard" means that our hardness results for the exact case go through: using these resources, one can sample from a probability distribution that is not samplable in classical polynomial time unless $\mathsf{P}^{\#\mathsf{P}} = \mathsf{BPP}^{\mathsf{NP}}$. "Apx. sampling hard?" means that our hardness results for the *approximate* case go through as well: using these resources, one can sample from a probability distribution that is not even approximately samplable in classical polynomial time unless $|\mathrm{GPE}|_{\pm}^{2} \in \mathsf{BPP}^{\mathsf{NP}}$.

	Available input states		
Available measurements	*Coherent states*	*Gaussian states*	*Single photons*
Adaptive, nondemolition	BQP [43]	BQP [43]	BQP [43]
Adaptive, demolition	P (trivial)	BQP [43]	BQP [43]
Nonadaptive, nondemolition	Exact sampling hard	Exact sampling hard	Apx. sampling hard?
Nonadaptive, demolition	P (trivial)	Exact sampling hard	Apx. sampling hard?
Gaussian basis only	P [9]	P [9]	?

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 [42]; the permutational quantum computing model of Jordan [39];

[6] 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.

and stabilizer circuits with non-stabilizer initial states (such as $\cos(\pi/8)|0\rangle + \sin(\pi/8)|1\rangle$) and nonadaptive measurements [5]. The noninteracting-boson model is another addition to this list.

The Hong-Ou-Mandel dip In 1987, Hong, Ou, and Mandel [36] 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. From an experimental perspective, what we are asking for could be seen as a generalization of the so-called "Hong-Ou-Mandel dip" to the n-photon case, where n is as large as possible. Lim and Beige [46] 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 [15] 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 [71] proved in 1979 that the permanent is #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 [70] 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 $\operatorname{Per}(A)$. However, they correctly pointed out that this did not imply a polynomial-time quantum algorithm to *calculate* $\operatorname{Per}(A)$ since the variance of their observable was large enough that exponentially many samples would be needed. (In this paper, we sidestep the issue raised by Troyansky and Tishby by not even *trying* to calculate $\operatorname{Per}(A)$ for a given A settling instead for *sampling from a probability distribution* in which the probabilities depend on permanents of various $n \times n$ matrices. Our main result gives evidence that this sampling task is already classically intractable.)

Later, Scheel [57] 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 [72] 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.[7] Shortly afterward, Terhal and DiVincenzo [66] (see also Knill [41]) pointed out

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

that matchgate circuits were equivalent to systems of noninteracting fermions[8]: 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.[9,10] 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 $\#\mathsf{P}$-complete.

Ironically, when the *quantum Monte Carlo method* [16] 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 $\#\mathsf{P}$-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 $\#\mathsf{P}$ of counting problems. In 1993, Bernstein and Vazirani [11] proved that $\mathsf{BQP} \subseteq \mathsf{P}^{\#\mathsf{P}}$.[11] However, this result says only that $\#\mathsf{P}$ is an *upper* bound on the power of quantum computation, so the question arises of whether solving $\#\mathsf{P}$-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 [25] noticed that, if we ask more refined questions about a quantum circuit than

> *"does this circuit accept with probability greater than $1-\varepsilon$ or less than ε, promised that one of those is true?,"*

then we can quickly encounter $\#\mathsf{P}$-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 $\#\mathsf{P}$-hard under nondeterministic reductions.

Later, Aaronson [2] defined the class $\mathsf{PostBQP}$ or quantum polynomial-time with *postselection* on possibly exponentially-unlikely measurement outcomes. He showed that $\mathsf{PostBQP}$ 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 $\#\mathsf{P}$-complete problems. Following [12], in Section 4.2 we will use the $\mathsf{PostBQP} = \mathsf{PP}$ theorem to give an alternative proof of Theorem 1.1, which does not require using the $\#\mathsf{P}$-completeness of the permanent.

---

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

[9] 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.

[10] Interestingly, Beenakker et al. [10] have shown that, if we augment the noninteracting-fermion model by adaptive *charge* measurements (which reveal whether 0 1 or 2 of the two spin states at a given spatial location are occupied by an electron), then the model becomes universal for quantum computation.

[11] See also Rudolph [56] for a direct encoding of quantum computations by matrix permanents.

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 will be 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 [58] 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 [12] 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 1.1, except with commuting quantum computations instead of noninteracting-boson ones.

Previously, in 2002, Terhal and DiVincenzo [67] 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 [12], 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 1.1) have the drawback that they only address sampling from *exactly* the same distribution $\mathcal{D}$ as the quantum algorithm—or 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 [3] 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 [35] and others. What made this surprising was that the 2^n-bit MAJORITY function is #P-complete.[12] In other words, even though there is

[12]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.

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 [3] is that they were relative to an oracle. However, just like Simon's oracle algorithm [61] led shortly afterward to Shor's algorithm [60], 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 Preliminaries

Throughout this paper, we use $\mathcal{G}$ to denote $\mathcal{N}(0,1)_{\mathbb{C}}$ the complex Gaussian distribution with mean 0 and variance $\mathrm{E}_{z\sim\mathcal{G}}\left[|z|^2\right]=1$. (We often use the word "distribution" for continuous probability measures, as well as for discrete distributions.) We will be especially interested in $\mathcal{G}^{n\times n}$ the distribution over $n\times n$ matrices with i. i. d. Gaussian entries.

For $m\geq n$ we use $\mathcal{U}_{m,n}$ to denote the set of matrices $A\in\mathbb{C}^{m\times n}$ whose columns are orthonormal vectors—so in particular, $\mathcal{U}_{m,m}$ is the set of $m\times m$ unitary matrices. We also use $\mathcal{H}_{m,n}$ to denote the *Haar measure* over $\mathcal{U}_{m,n}$. Informally, Haar measure just means the "continuous analogue of the uniform distribution": for example, to draw a matrix A from $\mathcal{H}_{m,n}$ we set the first column equal to a random unit vector in $\mathbb{C}^m$ the second column equal to a random unit vector orthogonal to the first column, and so on. Formally, one can define $\mathcal{H}_{m,n}$ by starting from the Haar measure over $\mathcal{U}_{m,m}$ (defined as the unique measure invariant under the action of the $m\times m$ unitary group), and then restricting to the first n columns.

We use $\overline{\alpha}$ to denote the complex conjugate of α. We denote the set $\{1,\ldots,n\}$ by $[n]$. Let $v\in\mathbb{C}^n$ and $A\in\mathbb{C}^{n\times n}$. Then

$$\|v\|:=\sqrt{|v_1|^2+\cdots+|v_n|^2}\quad\text{and}\quad\|A\|:=\max_{\|v\|=1}\|Av\|\,.$$

Equivalently, $\|A\|=\sigma_{\max}(A)$ is the largest singular value of A.

We generally omit floor and ceiling signs, when it is clear that the relevant quantities can be rounded to integers without changing the asymptotic complexity. Likewise, we will talk about a polynomial-time algorithm receiving as input a matrix $A\in\mathbb{C}^{n\times n}$ often drawn from the Gaussian distribution $\mathcal{G}^{n\times n}$. Here it is understood that the entries of A are rounded to $p(n)$ bits of precision, for some polynomial p. In all such cases, it will be straightforward to verify that there exists a fixed polynomial p such that none of the relevant calculations are affected by precision issues.

2.1 Complexity classes

We assume familiarity with standard computational complexity classes such as BQP (Bounded-Error Quantum Polynomial-Time) and PH (the Polynomial Hierarchy).[13] We now define some other complexity classes that will be important in this work.

Definition 2.1 (PostBPP and PostBQP)**.** Say the algorithm $\mathcal{A}$ "succeeds" if its first output bit is measured to be 1 and "fails" otherwise; conditioned on succeeding, say $\mathcal{A}$ "accepts" if its second output bit is

[13] See the Complexity Zoo, www.complexityzoo.com, for definitions of these and other classes.

measured to be 1 and "rejects" otherwise. Then $\mathsf{PostBPP}$ is the class of languages $L \subseteq \{0,1\}^*$ for which there exists a probabilistic polynomial-time algorithm $\mathcal{A}$ such that, for all inputs x:

(i) $\Pr[\mathcal{A}(x) \text{ succeeds}] > 0$.

(ii) If $x \in L$ then $\Pr[\mathcal{A}(x) \text{ accepts} \mid \mathcal{A}(x) \text{ succeeds}] \geq \frac{2}{3}$.

(iii) If $x \notin L$ then $\Pr[\mathcal{A}(x) \text{ accepts} \mid \mathcal{A}(x) \text{ succeeds}] \leq \frac{1}{3}$.

$\mathsf{PostBQP}$ is defined the same way, except that $\mathcal{A}$ is a quantum algorithm rather than a classical one.

$\mathsf{PostBPP}$ is easily seen to equal a complexity class called $\mathsf{BPP_{path}}$ which was defined by Han, Hemaspaandra, and Thierauf [34]. In particular, it follows from Han et al.'s results that

$$\mathsf{MA} \subseteq \mathsf{PostBPP} \subseteq \mathsf{BPP}^{\mathsf{NP}}. \tag{2.1}$$

As for $\mathsf{PostBQP}$ we have the following result of Aaronson [2], which characterizes $\mathsf{PostBQP}$ in terms of the classical complexity class PP (Probabilistic Polynomial-Time).

Theorem 2.2 (Aaronson [2])**.** $\mathsf{PostBQP} = \mathsf{PP}$.

It is well-known that $\mathsf{P}^{\mathsf{PP}} = \mathsf{P}^{\#\mathsf{P}}$—and thus, Theorem 2.2 has the surprising implication that BQP with postselection is as powerful as an oracle for *counting* problems. Aaronson [2] also observed that, just as intermediate measurements do not affect the power of BQP so intermediate postselected measurements do not affect the power of $\mathsf{PostBQP}$.

All the results mentioned above are easily seen to hold relative to any oracle.

2.2 Sampling and search problems

In this work, a central role is played not only by decision problems, but also by *sampling* and *search* problems. By a *sampling problem* S we mean a collection of probability distributions $(\mathcal{D}_x)_{x\in\{0,1\}^*}$ one for each input string $x \in \{0,1\}^n$. Here $\mathcal{D}_x$ is a distribution over $\{0,1\}^{p(n)}$ for some fixed polynomial p. To "solve" S means to sample from $\mathcal{D}_x$ given x as input, while to solve S approximately means (informally) to sample from some distribution that is $1/\operatorname{poly}(n)$-close to $\mathcal{D}_x$ in variation distance. In this paper, we will be interested in both notions, but especially approximate sampling.

We now define the classes SampP and $\mathsf{SampBQP}$ consisting of those sampling problems that are approximately solvable by polynomial-time classical and quantum algorithms respectively.

Definition 2.3 (SampP and $\mathsf{SampBQP}$)**.** SampP is the class of sampling problems $S = (\mathcal{D}_x)_{x\in\{0,1\}^*}$ for which there exists a probabilistic polynomial-time algorithm A that, given $\left\langle x, 0^{1/\varepsilon}\right\rangle$ as input,[14] samples from a probability distribution $\mathcal{D}'_x$ such that $\|\mathcal{D}'_x - \mathcal{D}_x\| \leq \varepsilon$. $\mathsf{SampBQP}$ is defined the same way, except that A is a quantum algorithm rather than a classical one.

[14]Giving $\left\langle x, 0^{1/\varepsilon}\right\rangle$ as input (where $0^{1/\varepsilon}$ represents $1/\varepsilon$ encoded in unary) is a standard trick for forcing an algorithm's running time to be polynomial in n as well as $1/\varepsilon$.

Another class of problems that will interest us are *search problems* (also confusingly called "relation problems" or "function problems"). In a search problem, there is always at least one valid solution, and the problem is to *find* a solution: a famous example is finding a Nash equilibrium of a game, the problem shown to be PPAD-complete by Daskalakis et al. [20, 21]. More formally, a search problem R is a collection of nonempty sets $(B_x)_{x\in\{0,1\}^*}$, one for each input $x \in \{0,1\}^n$. Here $B_x \subseteq \{0,1\}^{p(n)}$ for some fixed polynomial p. To solve R means to output an element of B_x given x as input.

We now define the complexity classes FBPP and FBQP, consisting of those search problems that are solvable by BPP and BQP machines respectively.

Definition 2.4 (FBPP and FBQP)**.** FBPP is the class of search problems $R = (B_x)_{x\in\{0,1\}^*}$ for which there exists a probabilistic polynomial-time algorithm A that, given $\langle x, 0^{1/\varepsilon}\rangle$ as input, produces an output y such that $\Pr[y \in B_x] \geq 1-\varepsilon$, where the probability is over A's internal randomness. FBQP is defined the same way, except that A is a quantum algorithm rather than a classical one.

Recently, and directly motivated by the present work, Aaronson [4] proved a general connection between sampling problems and search problems.

Theorem 2.5 (Sampling/Searching Equivalence Theorem [4])**.** *Let $S = (\mathcal{D}_x)_{x\in\{0,1\}^*}$ be any approximate sampling problem. Then there exists a search problem $R_S = (B_x)_{x\in\{0,1\}^*}$ that is "equivalent" to S in the following two senses.*

(i) Let $\mathcal{O}$ be any oracle that, given $\langle x, 0^{1/\varepsilon}, r\rangle$ as input, outputs a sample from a distribution $\mathcal{C}_x$ such that $\|\mathcal{C}_x - \mathcal{D}_x\| \leq \varepsilon$ as we vary the random string r. Then $R_S \in \mathsf{FBPP}^{\mathcal{O}}$.

(ii) Let M be any probabilistic Turing machine that, given $\langle x, 0^{1/\delta}\rangle$ as input, outputs an element $Y \in B_x$ with probability at least $1-\delta$. Then $S \in \mathsf{SampP}^M$.

Briefly, Theorem 2.5 is proved by using the notion of a "universal randomness test" from algorithmic information theory. Intuitively, given a sampling problem S, we define an "equivalent" search problem R_S as follows: "output a collection of strings $Y = (y_1,\ldots,y_T)$ in the support of $\mathcal{D}_x$, most of which have large probability in $\mathcal{D}_x$ and which *also*, conditioned on that, have close-to-maximal Kolmogorov complexity." Certainly, if we can sample from $\mathcal{D}_x$ then we can solve this search problem as well. But the converse also holds: if a probabilistic Turing machine is solving the search problem R_S it can *only* be doing so by sampling approximately from $\mathcal{D}_x$. For otherwise, the strings $y_1,\ldots,y_T$ would have short Turing machine descriptions, contrary to assumption.

In particular, Theorem 2.5 implies that $S \in \mathsf{SampP}$ if and only if $R_S \in \mathsf{FBPP}$ $S \in \mathsf{SampBQP}$ if and only if $R_S \in \mathsf{FBQP}$ and so on. We therefore obtain the following consequence:

Theorem 2.6 ([4])**.** SampP = SampBQP *if and only if* FBPP = FBQP.

3 The noninteracting-boson model of computation

In this section, we develop a formal model of computation based on *identical, noninteracting bosons*: as a concrete example, a linear-optical network with single-photon inputs and nonadaptive photon-number measurements. As far as we know, this model is incapable of universal quantum computing (or even universal *classical* computing, for that matter!), although a universal quantum computer can certainly simulate *it*. The surprise is that this rudimentary model can already solve certain sampling and search problems that, under plausible assumptions, *cannot* be solved efficiently by a classical computer. The ideas behind the model have been the basis for optical physics for almost a century. To our knowledge, however, this is the first time the model has been presented from a theoretical computer science perspective.

Like quantum mechanics itself, the noninteracting-boson model possesses a mathematical beauty that can be appreciated even independently of its physical origins. In an attempt to convey that beauty, we will define the model in *three* ways, and also prove those ways to be equivalent. The first definition, in Section 3.1, is directly in terms of physical devices (beamsplitters and phaseshifters) and the unitary transformations that they induce. This definition should be easy to understand for those already comfortable with quantum computing, and makes it apparent why our model can be simulated on a standard quantum computer. The second definition, in Section 3.2, is in terms of multivariate polynomials with an unusual inner product. This definition, which we learned from Gurvits [33], is the nicest one mathematically, and makes it easy to prove many statements (for example, that the probabilities sum to 1) that would otherwise require tedious calculation. The third definition is in terms of permanents of $n \times n$ matrices, and is what lets us connect our model to the hardness of the permanent. The second and third definitions do not use any quantum formalism.

Finally, Section 3.4 defines BOSONSAMPLING, the basic computational problem considered in this paper, as well as the complexity class $\mathsf{BosonFP}$ of search problems solvable using a BOSONSAMPLING oracle. It also proves the simple but important fact that $\mathsf{BosonFP} \subseteq \mathsf{FBQP}$: in other words, boson computers can be simulated efficiently by standard quantum computers.

3.1 Physical definition

The model that we are going to define involves a quantum system of n identical photons[15] and m *modes* (intuitively, places that a photon can be in). We will usually be interested in the case where $n \le m \le \mathrm{poly}(n)$ though the model makes sense for arbitrary n and m.[16] Each computational basis state of this system has the form $|S\rangle = |s_1, \dots, s_m\rangle$ where s_i represents the number of photons in the i^{th} mode (s_i is also called the i^{th} *occupation number*). Here the s_i's can be any nonnegative integers summing to n; in particular, the s_i's can be greater than 1. This corresponds to the fact that photons are bosons, and (unlike with fermions) an unlimited number of bosons can be in the same mode at the same time.

During a computation, photons are never created or destroyed, but are only moved from one mode to another. Mathematically, this means that the basis states $|S\rangle$ of our computer will always satisfy $S \in \Phi_{m,n}$ where $\Phi_{m,n}$ is the set of tuples $S = (s_1, \dots, s_m)$ satisfying $s_1, \dots, s_m \ge 0$ and $s_1 + \dots + s_m = n$. Let $M = |\Phi_{m,n}|$ be the total number of basis states; then one can easily check that $M = \binom{m+n-1}{n}$.

Since this is quantum mechanics, a general state of the computer has the form

$$|\psi\rangle = \sum_{S \in \Phi_{m,n}} \alpha_S |S\rangle \ , \tag{3.1}$$

[15] For concreteness, we will often talk about photons in a linear-optical network, but the mathematics would be the same with any other system of identical, noninteracting bosons (for example, bosonic excitations in solid-state).

[16] The one caveat is that our "standard initial state," which consists of one photon in each of the first n modes, is only defined if $n \le m$.

where the α_S's are complex numbers satisfying $\sum_{S\in\Phi_{m,n}} |\alpha_S|^2 = 1$. In other words, $|\psi\rangle$ is a unit vector in the M-dimensional complex Hilbert space spanned by elements of $\Phi_{m,n}$. Call this Hilbert space $H_{m,n}$.

Just like in standard quantum computing, the Hilbert space $H_{m,n}$ is exponentially large (as a function of $m+n$), which means that we can only hope to explore a tiny fraction of it using polynomial-size circuits. On the other hand, one difference from standard quantum computing is that $H_{m,n}$ is *not* built up as the tensor product of smaller Hilbert spaces.

Throughout this paper, we will assume that our computer starts in the state

$$|1_n\rangle := |1,\dots,1,0,\dots,0\rangle\,, \tag{3.2}$$

where the first n modes contain one photon each, and the remaining $m-n$ modes are unoccupied. We call $|1_n\rangle$ the *standard initial state*.

We will also assume that measurement only occurs at the end of the computation, and that what is measured is the number of photons in each mode. In other words, a measurement of the state $|\psi\rangle = \sum_{S\in\Phi_{m,n}} \alpha_S |S\rangle$ returns an element S of $\Phi_{m,n}$, with probability equal to

$$\Pr[S] = |\alpha_S|^2 = |\langle\psi|S\rangle|^2\,. \tag{3.3}$$

But which unitary transformations can we perform on the state $|\psi\rangle$ after the initialization and before the final measurement? For simplicity, let us consider the special case where there is only one photon; later we will generalize to n photons. In the one-photon case, the Hilbert space $H_{m,1}$ has dimension $M=m$ and the computational basis states ($|1,0,\dots,0\rangle$ $|0,1,0,\dots,0\rangle$ etc.) simply record which mode the photon is in. Thus, a general state $|\psi\rangle$ is just a unit vector in $\mathbb{C}^m$: that is, a superposition over the modes. An $m\times m$ unitary transformation U acts on this unit vector in exactly the way one would expect: namely, the vector is left-multiplied by U.

However, this still leaves the question of how an arbitrary $m\times m$ unitary transformation U is *implemented* within this model. In standard quantum computing, we know that any unitary transformation on n qubits can be decomposed as a product of *gates*, each of which acts nontrivially on at most two qubits, and is the identity on the other qubits. Likewise, in the linear-optics model, any unitary transformation on m modes can be decomposed into a product of *optical elements*, each of which acts nontrivially on at most two modes, and is the identity on the other $m-2$ modes. The two best-known optical elements are called *phaseshifters* and *beamsplitters*. A phaseshifter multiplies a single amplitude α_S by $e^{i\theta}$ for some specified angle θ and acts as the identity on the other $m-1$ amplitudes. A beamsplitter modifies two amplitudes α_S and α_T as follows, for some specified angle θ:

$$\begin{pmatrix} \alpha'_S \\ \alpha'_T \end{pmatrix} := \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \alpha_S \\ \alpha_T \end{pmatrix}. \tag{3.4}$$

It acts as the identity on the other $m-2$ amplitudes. It is easy to see that beamsplitters and phaseshifters generate all optical elements (that is, all 2×2 unitaries). Moreover, the optical elements generate all $m\times m$ unitaries, as shown by the following lemma of Reck et al. [54]:

Lemma 3.1 (Reck et al. [54])**.** *Let U be any $m\times m$ unitary matrix. Then one can decompose U as a product $U = U_T\cdots U_1$ where each U_t is an optical element (that is, a unitary matrix that acts nontrivially on at most 2 modes and as the identity on the remaining $m-2$ modes). Furthermore, this decomposition has size $T = O\left(m^2\right)$ and can be found in time polynomial in m.*

Proof Sketch. The task is to produce U starting from the identity matrix—or equivalently, to produce I starting from U—by successively multiplying by block-diagonal unitary matrices, each of which contains a single 2×2 block and $m-2$ blocks consisting of 1.[17] To do so, we use a procedure similar to Gaussian elimination, which zeroes out the m^2-m off-diagonal entries of U one by one. Then, once U has been reduced to a diagonal matrix, we use m phaseshifters to produce the identity matrix. □

We now come to the more interesting part: how do we describe the action of the unitary transformation U on a state with *multiple* photons? As it turns out, there is a natural homomorphism φ which maps an $m\times m$ unitary transformation U acting on a single photon to the corresponding $M\times M$ unitary transformation $\varphi(U)$ acting on n photons. Since φ is a homomorphism, Lemma 3.1 implies that we can specify φ merely by describing its behavior on 2×2 unitaries. For given an arbitrary $m\times m$ unitary matrix U, we can write $\varphi(U)$ as

$$\varphi(U_T\cdots U_1)=\varphi(U_T)\cdots\varphi(U_1)\,, \tag{3.5}$$

where each U_t is an optical element (that is, a block-diagonal unitary that acts nontrivially on at most 2 modes).

In the case of a phaseshifter (that is, a 1×1 unitary), it is relatively obvious what should happen. Namely, the phaseshifter should be applied once for each photon in the mode to which it is applied. In other words, suppose U is an $m\times m$ diagonal matrix such that $u_{ii}=e^{i\theta}$ and $u_{jj}=1$ for all $j\neq i$. Then we ought to have

$$\varphi(U)\left|s_1,\ldots,s_m\right\rangle=e^{i\theta s_i}\left|s_1,\ldots,s_m\right\rangle\,. \tag{3.6}$$

However, it is less obvious how to describe the action of a beamsplitter on multiple photons. Let

$$U=\begin{pmatrix} a & b\\ c & d\end{pmatrix} \tag{3.7}$$

be any 2×2 unitary matrix, which acts on the Hilbert space $H_{2,1}$ spanned by $|1,0\rangle$ and $|0,1\rangle$. Then since $\varphi(U)$ preserves photon number, we know it must be a block-diagonal matrix that satisfies

$$\langle s,t|\,\varphi(U)\,|u,v\rangle=0 \tag{3.8}$$

whenever $s+t\neq u+v$. But what about when $s+t=u+v$? Here the formula for the appropriate entry of $\varphi(U)$ is

$$\langle s,t|\,\varphi(U)\,|u,v\rangle=\sqrt{\frac{u!v!}{s!t!}}\sum_{k+\ell=u,\;k\leq s,\;\ell\leq t}\binom{s}{k}\binom{t}{\ell}a^k b^{s-k}c^{\ell}d^{t-\ell}\,. \tag{3.9}$$

One can verify by calculation that $\varphi(U)$ is unitary; however, a much more elegant proof of unitarity will follow from the results in Section 3.2.

One more piece of notation: let $\mathcal{D}_U$ be the probability distribution over $S\in\Phi_{m,n}$ obtained by measuring the state $\varphi(U)\,|1_n\rangle$ in the computational basis. That is,

$$\Pr_{\mathcal{D}_U}[S]=\left|\langle 1_n|\varphi(U)\,|S\rangle\right|^2\,. \tag{3.10}$$

Notice that $\mathcal{D}_U$ depends only on the first n columns of U. Therefore, instead of writing $\mathcal{D}_U$ it will be better to write $\mathcal{D}_A$ where $A\in\mathcal{U}_{m,n}$ is the $m\times n$ matrix corresponding to the first n columns of U.

[17]Such matrices are the generalizations of the so-called *Givens rotations* to the complex numbers.

3.2 Polynomial definition

In this section, we present a beautiful alternative interpretation of the noninteracting-boson model, in which the "states" are multivariate polynomials, the "operations" are unitary changes of variable, and a "measurement" samples from a probability distribution over monomials weighted by their coefficients. We also prove that this model is well-defined (i. e., that in any measurement, the probabilities of the various outcomes sum to 1), and that it is indeed equivalent to the model from Section 3.1. Combining these facts yields the simplest proof we know that the model from Section 3.1 is well-defined.

Let $m \geq n$. Then the "state" of our computer, at any time, will be represented by a multivariate complex-valued polynomial $p(x_1, \ldots, x_m)$ of degree n. Here the x_i's can be thought of as just formal variables.[18] The standard initial state $|1_n\rangle$ corresponds to the degree-n polynomial $J_{m,n}(x_1, \ldots, x_m) := x_1 \cdots x_n$ where $x_1, \ldots, x_n$ are the first n variables. To transform the state, we can apply any $m \times m$ unitary transformation U we like to the vector of x_i's:

$$\begin{pmatrix} x_1' \\ \vdots \\ x_m' \end{pmatrix} = \begin{pmatrix} u_{11} & \cdots & u_{1m} \\ \vdots & \ddots & \vdots \\ u_{m1} & \cdots & u_{mm} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}. \tag{3.11}$$

The new state of our computer is then equal to

$$U[J_{m,n}](x_1, \ldots, x_m) = J_{m,n}(x_1', \ldots, x_m') = \prod_{i=1}^{n} (u_{i1}x_1 + \cdots + u_{im}x_m). \tag{3.12}$$

Here and throughout, we let $L[p]$ be the polynomial obtained by starting with p and then applying the $m \times m$ linear transformation L to the variables.

After applying one or more unitary transformations to the x_i's, we then get a single opportunity to measure the computer's state. Let the polynomial p at the time of measurement be

$$p(x_1, \ldots, x_m) = \sum_{S=(s_1,\ldots,s_m)} a_S x_1^{s_1} \cdots x_m^{s_m}, \tag{3.13}$$

where S ranges over $\Phi_{m,n}$ (i. e., lists of nonnegative integers such that $s_1 + \cdots + s_m = n$). Then the measurement returns the monomial $x_1^{s_1} \cdots x_m^{s_m}$ (or equivalently, the list of integers $S = (s_1, \ldots, s_m)$) with probability equal to

$$\Pr[S] := |a_S|^2 s_1! \cdots s_m!. \tag{3.14}$$

From now on, we will use x as shorthand for $x_1, \ldots, x_m$ and x^S as shorthand for the monomial $x_1^{s_1} \cdots x_m^{s_m}$. Given two polynomials

$$p(x) = \sum_{S \in \Phi_{m,n}} a_S x^S, \tag{3.15}$$

$$q(x) = \sum_{S \in \Phi_{m,n}} b_S x^S, \tag{3.16}$$

[18] For physicists, they are "creation operators."

we can define an inner product between them—called the *Fock-space inner product*—as follows:

$$\langle p,q\rangle := \sum_{S=(s_1,\dots,s_m)\in\Phi_{m,n}} \overline{a}_S b_S s_1!\cdots s_m!\,. \tag{3.17}$$

The following key result gives a more intuitive interpretation of the Fock-space inner product.

Lemma 3.2 (Interpretation of Fock Inner Product)**.** $\langle p,q\rangle = \mathrm{E}_{x\sim\mathcal{G}^m}\left[\overline{p}(x)\,q(x)\right]$ *where* $\mathcal{G}$ *is the Gaussian distribution* $\mathcal{N}(0,1)_{\mathbb{C}}$.

Proof. Since inner product and expectation are linear, it suffices to consider the case where p and q are monomials. Suppose $p(x) = x^R$ and $q(x) = x^S$ for some $R = (r_1,\dots,r_m)$ and $S = (s_1,\dots,s_m)$ in $\Phi_{m,n}$. Then

$$\mathop{\mathrm{E}}_{x\sim\mathcal{G}^m}\left[\overline{p}(x)\,q(x)\right] = \mathop{\mathrm{E}}_{x\sim\mathcal{G}^m}\left[\overline{x}^R x^S\right]. \tag{3.18}$$

If $p \neq q$—that is, if there exists an i such that $r_i \neq s_i$—then the above expectation is clearly 0 since the Gaussian distribution is uniform over phases. If $p = q$ on the other hand, then the expectation equals

$$\mathop{\mathrm{E}}_{x\sim\mathcal{G}^m}\left[|x_1|^{2s_1}\cdots|x_m|^{2s_m}\right] = \mathop{\mathrm{E}}_{x_1\sim\mathcal{G}}\left[|x_1|^{2s_1}\right]\cdots\mathop{\mathrm{E}}_{x_m\sim\mathcal{G}}\left[|x_m|^{2s_m}\right] \tag{3.19}$$

$$= s_1!\cdots s_m!\,. \tag{3.20}$$

We conclude that

$$\mathop{\mathrm{E}}_{x\sim\mathcal{G}^m}\left[\overline{p}(x)\,q(x)\right] = \sum_{S=(s_1,\dots,s_m)\in\Phi_{m,n}} \overline{a}_S b_S s_1!\cdots s_m! \tag{3.21}$$

as desired. □

Recall that $U[p]$ denotes the polynomial $p(Ux)$ obtained by applying the $m\times m$ linear transformation U to the variables $x = (x_1,\dots,x_m)$ of p. Then Lemma 3.2 has the following important consequence.

Theorem 3.3 (Unitary Invariance of Fock Inner Product)**.** $\langle p,q\rangle = \langle U[p],U[q]\rangle$ *for all polynomials* p,q *and all unitary transformations* U.

Proof. We have

$$\langle U[p],U[q]\rangle = \mathop{\mathrm{E}}_{x\sim\mathcal{G}^m}\left[\overline{U[p]}(x)\,U[q](x)\right] \tag{3.22}$$

$$= \mathop{\mathrm{E}}_{x\sim\mathcal{G}^m}\left[\overline{p}(Ux)\,q(Ux)\right] \tag{3.23}$$

$$= \mathop{\mathrm{E}}_{x\sim\mathcal{G}^m}\left[\overline{p}(x)\,q(x)\right] \tag{3.24}$$

$$= \langle p,q\rangle\,, \tag{3.25}$$

where line (3.24) follows from the rotational invariance of the Gaussian distribution. □

Indeed, we have a more general result:

Theorem 3.4. $\langle p, L[q]\rangle = \left\langle L^{\dagger}[p], q\right\rangle$ *for all polynomials* p,q *and all linear transformations* L. *In particular, if* L *is invertible, then*

$$\langle p,q\rangle = \left\langle \left(L^{-1}\right)^{\dagger}[p], L[q]\right\rangle .$$

Proof. Let $p(x) = \sum_{S\in\Phi_{m,n}} a_S x^S$ and $q(x) = \sum_{S\in\Phi_{m,n}} b_S x^S$. First suppose L is a diagonal matrix, i. e., $L = \operatorname{diag}(\lambda)$ for some $\lambda = (\lambda_1,\ldots,\lambda_m)$. Then

$$\langle p, L[q]\rangle = \sum_{S=(s_1,\ldots,s_m)\in\Phi_{m,n}} \overline{a}_S \left(b_S \lambda^S\right) s_1!\cdots s_m! \tag{3.26}$$

$$= \sum_{S=(s_1,\ldots,s_m)\in\Phi_{m,n}} \left(\overline{a_S \overline{\lambda}^S}\right) b_S s_1!\cdots s_m! \tag{3.27}$$

$$= \left\langle L^{\dagger}[p], q\right\rangle . \tag{3.28}$$

Now note that we can decompose an arbitrary L as $U\Lambda V$ where Λ is diagonal and U,V are unitary. So

$$\langle p, L[q]\rangle = \langle p, U\Lambda V[q]\rangle \tag{3.29}$$

$$= \left\langle U^{\dagger}[p], \Lambda V[q]\right\rangle \tag{3.30}$$

$$= \left\langle \Lambda^{\dagger} U^{\dagger}[p], V[q]\right\rangle \tag{3.31}$$

$$= \left\langle V^{\dagger}\Lambda^{\dagger} U^{\dagger}[p], q\right\rangle \tag{3.32}$$

$$= \left\langle L^{\dagger}[p], q\right\rangle \tag{3.33}$$

where lines (3.30) and (3.32) follow from Theorem 3.3. □

We can also define a *Fock-space norm* as follows:

$$\|p\|^2_{\text{Fock}} = \langle p,p\rangle = \sum_{S=(s_1,\ldots,s_m)} |a_S|^2 s_1!\cdots s_m! . \tag{3.34}$$

Clearly $\|p\|^2_{\text{Fock}} \geq 0$ for all p. We also have the following:

Corollary 3.5. $\|U[J_{m,n}]\|^2_{\text{Fock}} = 1$ *for all unitary matrices* U.

Proof. By Theorem 3.3,

$$\|U[J_{m,n}]\|^2_{\text{Fock}} = \langle U[J_{m,n}], U[J_{m,n}]\rangle = \left\langle UU^{\dagger}[J_{m,n}], J_{m,n}\right\rangle = \langle J_{m,n}, J_{m,n}\rangle = 1 . \tag{3.35}$$

□

Corollary 3.5 implies, in particular, that our model of computation based on multivariate polynomials is well-defined: that is, the probabilities of the various measurement outcomes always sum to $\|U[J_{m,n}]\|^2_{\text{Fock}} = 1$. We now show that the polynomial-based model of this section is *equivalent* to the linear-optics model of Section 3.1. As an immediate consequence, this implies that probabilities sum to 1 in the linear-optics model as well.

Given any pure state

$$|\psi\rangle = \sum_{S\in\Phi_{m,n}} \alpha_S |S\rangle \tag{3.36}$$

in $H_{m,n}$ let $P_{|\psi\rangle}$ be the multivariate polynomial defined by

$$P_{|\psi\rangle}(x) := \sum_{S=(s_1,\ldots,s_m)\in\Phi_{m,n}} \frac{\alpha_S x^S}{\sqrt{s_1!\cdots s_m!}}. \tag{3.37}$$

In particular, for any computational basis state $|S\rangle$, we have

$$P_{|S\rangle}(x) = \frac{x^S}{\sqrt{s_1!\cdots s_m!}}. \tag{3.38}$$

Theorem 3.6 (Equivalence of Physical and Polynomial Definitions)**.** *$|\psi\rangle \longleftrightarrow P_{|\psi\rangle}$ defines an isomorphism between quantum states and polynomials, which commutes with inner products and unitary transformations in the following senses:*

$$\langle\psi|\phi\rangle = \left\langle P_{|\psi\rangle}, P_{|\phi\rangle}\right\rangle, \tag{3.39}$$

$$P_{\varphi(U)|\psi\rangle} = U\left[P_{|\psi\rangle}\right]. \tag{3.40}$$

Proof. That $\langle\psi|\phi\rangle = \left\langle P_{|\psi\rangle}, P_{|\phi\rangle}\right\rangle$ follows immediately from the definitions of $P_{|\psi\rangle}$ and the Fock-space inner product. For $P_{\varphi(U)|\psi\rangle} = U\left[P_\psi\right]$, notice that

$$U\left[P_{|\psi\rangle}\right] = U\left[\sum_{S=(s_1,\ldots,s_m)\in\Phi_{m,n}} \frac{\alpha_S x^S}{\sqrt{s_1!\cdots s_m!}}\right] \tag{3.41}$$

$$= \sum_{S=(s_1,\ldots,s_m)\in\Phi_{m,n}} \frac{\alpha_S}{\sqrt{s_1!\cdots s_m!}} \prod_{i=1}^{m} \left(u_{i1}x_1 + \cdots + u_{im}x_m\right)^{s_i}. \tag{3.42}$$

So in particular, transforming $P_{|\psi\rangle}$ to $U\left[P_{|\psi\rangle}\right]$ simply effects a linear transformation on the coefficients on $P_{|\psi\rangle}$. This means that there must be *some* $M\times M$ linear transformation $\varphi(U)$ depending on U such that $U\left[P_{|\psi\rangle}\right] = P_{\varphi(U)|\psi\rangle}$. Thus, in defining the homomorphism $U\to\varphi(U)$ in equation (3.9), we simply chose it to yield that linear transformation. This can be checked by explicit computation. By Lemma 3.1, we can restrict attention to a 2×2 unitary matrix

$$U = \begin{pmatrix} a & b \\ c & d \end{pmatrix}. \tag{3.43}$$

By linearity, we can also restrict attention to the action of $\varphi(U)$ on a computational basis state $|s,t\rangle$ (or in the polynomial formalism, the action of U on a monomial $x^s y^t$). Then

$$U\left[x^s y^t\right] = (ax+by)^s (cx+dy)^t \tag{3.44}$$

$$= \sum_{k=0}^{s}\sum_{\ell=0}^{t} \binom{s}{k}\binom{t}{\ell} a^k b^{s-k} c^\ell d^{t-\ell} x^{k+\ell} y^{s+t-k-\ell} \tag{3.45}$$

$$= \sum_{u+v=s+t}\ \sum_{k+\ell=u,\ k\le s,\ \ell\le t} \binom{s}{k}\binom{t}{\ell} a^k b^{s-k} c^\ell d^{t-\ell} x^u y^v. \tag{3.46}$$

Thus, inserting normalization,

$$U\left[\frac{x^s y^t}{\sqrt{s!t!}}\right]=\sum_{u+v=s+t}\left(\sqrt{\frac{u!v!}{s!t!}}\sum_{k+\ell=u,\ k\leq s,\ \ell\leq t}\binom{s}{k}\binom{t}{\ell}a^k b^{s-k}c^{\ell}d^{t-\ell}\right)\frac{x^u y^v}{\sqrt{u!v!}}, \tag{3.47}$$

which yields precisely the definition of $\varphi(U)$ from equation (3.9). □

As promised in Section 3.1, we can also show that $\varphi(U)$ is unitary.

Corollary 3.7. *$\varphi(U)$ is unitary.*

Proof. One definition of a unitary matrix is that it preserves inner products. Let us check that this is the case for $\varphi(U)$. For all U, we have

$$\langle\psi|\phi\rangle=\left\langle P_{|\psi\rangle},P_{|\phi\rangle}\right\rangle \tag{3.48}$$

$$=\left\langle U\left[P_{|\psi\rangle}\right],U\left[P_{|\phi\rangle}\right]\right\rangle \tag{3.49}$$

$$=\left\langle P_{\varphi(U)|\psi\rangle},P_{\varphi(U)|\phi\rangle}\right\rangle \tag{3.50}$$

$$=\langle\psi|\,\varphi(U)^{\dagger}\,\varphi(U)\,|\phi\rangle \tag{3.51}$$

where line (3.49) follows from Theorem 3.3, and all other lines from Theorem 3.6. □

3.3 Permanent definition

This section gives a *third* interpretation of the noninteracting-boson model, which makes clear its connection to the permanent. Given an $n\times n$ matrix $A=(a_{ij})\in\mathbb{C}^{n\times n}$ recall that the permanent is

$$\operatorname{Per}(A)=\sum_{\sigma\in S_n}\prod_{i=1}^{n}a_{i,\sigma(i)}\,. \tag{3.52}$$

Also, given an $m\times m$ matrix V let $V_{n,n}$ be the top-left $n\times n$ submatrix of V. Then the following lemma establishes a direct connection between $\operatorname{Per}(V_{n,n})$ and the Fock-space inner product defined in Section 3.2.

Lemma 3.8. $\operatorname{Per}(V_{n,n})=\langle J_{m,n},V[J_{m,n}]\rangle$ *for any $m\times m$ matrix V.*

Proof. By definition,

$$V[J_{m,n}]=\prod_{i=1}^{n}(v_{i1}x_1+\cdots+v_{im}x_m)\,. \tag{3.53}$$

Then $\langle J_{m,n},V[J_{m,n}]\rangle$ is just the coefficient of $J_{m,n}=x_1\cdots x_n$ in the above polynomial. This coefficient can be calculated as

$$\sum_{\sigma\in S_n}\prod_{i=1}^{n}v_{i,\sigma(i)}=\operatorname{Per}(V_{n,n})\,. \tag{3.54}$$

□

Combining Lemma 3.8 with Theorem 3.4, we immediately obtain the following:

Corollary 3.9. Per$\left(\left(V^{\dagger}W\right)_{n,n}\right) = \langle V[J_{m,n}], W[J_{m,n}]\rangle$ *for any two matrices* $V, W \in \mathbb{C}^{m\times m}$.

Proof.

$$\operatorname{Per}\left(\left(V^{\dagger}W\right)_{n,n}\right) = \left\langle J_{m,n}, V^{\dagger}W[J_{m,n}]\right\rangle = \langle V[J_{m,n}], W[J_{m,n}]\rangle . \tag{3.55}$$

□

Now let U be any $m\times m$ unitary matrix, and let $S=(s_1,\ldots,s_m)$ and $T=(t_1,\ldots,t_m)$ be any two computational basis states (that is, elements of $\Phi_{m,n}$). Then we define an $n\times n$ matrix $U_{S,T}$ in the following manner. First form an $m\times n$ matrix U_T by taking t_j copies of the j^{th} column of U, for each $j\in[m]$. Then form the $n\times n$ matrix $U_{S,T}$ by taking s_i copies of the i^{th} row of U_T for each $i\in[m]$. As an example, suppose

$$U = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} \tag{3.56}$$

and $S=T=(0,1,2)$. Then

$$U_{S,T} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & -1 \\ 0 & -1 & -1 \end{pmatrix} . \tag{3.57}$$

Note that if the s_i's and t_j's are all 0 or 1 then $U_{S,T}$ is simply an $n\times n$ submatrix of U. If some s_i's or t_j's are greater than 1 then $U_{S,T}$ is like a submatrix of U but with repeated rows and/or columns.

Here is an alternative way to define $U_{S,T}$. Given any $S\in\Phi_{m,n}$, let I_S be a linear substitution of variables, which maps the variables $x_1,\ldots,x_{s_1}$ to x_1 the variables $x_{s_1+1},\ldots,x_{s_1+s_2}$ to x_2, and so on, so that $I_S[x_1\cdots x_n] = x_1^{s_1}\cdots x_m^{s_m}$. (If $i>n$, then $I_S[x_i]=0$.) Then one can check that

$$U_{S,T} = \left(I_S^{\dagger} U I_T\right)_{n,n} . \tag{3.58}$$

(Note also that $\varphi(I_S)|1_n\rangle = |S\rangle$.)

Theorem 3.10 (Equivalence of All Three Definitions)**.** *For all* $m\times m$ *unitaries* U *and basis states* $S,T\in\Phi_{m,n}$,

$$\operatorname{Per}(U_{S,T}) = \left\langle x^S, U\left[x^T\right]\right\rangle = \langle S|\varphi(U)|T\rangle \sqrt{s_1!\cdots s_m! t_1!\cdots t_m!} . \tag{3.59}$$

Proof. For the first equality, from Corollary 3.9 we have

$$\left\langle x^S, U\left[x^T\right]\right\rangle = \langle I_S[J_{m,n}], UI_T[J_{m,n}]\rangle \tag{3.60}$$
$$= \operatorname{Per}\left(\left(I_S^{\dagger}UI_T\right)_{n,n}\right) \tag{3.61}$$
$$= \operatorname{Per}(U_{S,T}) . \tag{3.62}$$

For the second equality, from Theorem 3.6 we have

$$\langle S|\varphi(U)|T\rangle = \left\langle P_{|S\rangle}, P_{\varphi(U)|T\rangle}\right\rangle \tag{3.63}$$
$$= \left\langle P_{|S\rangle}, U\left[P_{|T\rangle}\right]\right\rangle \tag{3.64}$$
$$= \frac{\left\langle x^S, U\left[x^T\right]\right\rangle}{\sqrt{s_1!\cdots s_m! t_1!\cdots t_m!}} . \tag{3.65}$$

□

3.4 Bosonic complexity theory

Having presented the noninteracting-boson model from three perspectives, we are finally ready to define BOSONSAMPLING, the central computational problem considered in this work. The input to the problem will be an $m \times n$ column-orthonormal matrix $A \in \mathcal{U}_{m,n}$.[19] Given A together with a basis state $S \in \Phi_{m,n}$—that is, a list $S = (s_1, \ldots, s_m)$ of nonnegative integers, satisfying $s_1 + \cdots + s_m = n$—let A_S be the $n \times n$ matrix obtained by taking s_i copies of the i^{th} row of A for all $i \in [m]$. Then let $\mathcal{D}_A$ be the probability distribution over $\Phi_{m,n}$ defined as follows:

$$\Pr_{\mathcal{D}_A}[S] = \frac{|\mathrm{Per}(A_S)|^2}{s_1! \cdots s_m!}. \tag{3.66}$$

(Theorem 3.10 implies that $\mathcal{D}_A$ is indeed a probability distribution, for every $A \in \mathcal{U}_{m,n}$.) The goal of BOSONSAMPLING is to sample either exactly or approximately from $\mathcal{D}_A$ given A as input.

Of course, we *also* could have defined $\mathcal{D}_A$ as the distribution over $\Phi_{m,n}$ obtained by first completing A to any $m \times m$ unitary matrix U then measuring the quantum state $\varphi(U)|1_n\rangle$ in the computational basis. Or we could have defined $\mathcal{D}_A$ as the distribution obtained by first applying the linear change of variables U to the polynomial $x_1 \cdots x_n$ (where again U is any $m \times m$ unitary completion of A), to obtain a new m-variable polynomial

$$U[x_1 \cdots x_n] = \sum_{S \in \Phi_{m,n}} \alpha_S x^S, \tag{3.67}$$

and then letting

$$\Pr_{\mathcal{D}_A}[S] = |\alpha_S|^2 s_1! \cdots s_m! = \frac{\left|\left\langle x^S, U[x_1 \cdots x_n]\right\rangle\right|^2}{s_1! \cdots s_m!}. \tag{3.68}$$

For most of the paper, though, we will find it most convenient to use the definition of $\mathcal{D}_A$ in terms of permanents.

Besides the BOSONSAMPLING problem, we will also need the concept of an exact or approximate BOSONSAMPLING *oracle*. Intuitively, a BOSONSAMPLING oracle is simply an oracle $\mathcal{O}$ that solves the BOSONSAMPLING problem: that is, $\mathcal{O}$ takes as input a matrix $A \in \mathcal{U}_{m,n}$ and outputs a sample from $\mathcal{D}_A$. However, there is a subtlety, arising from the fact that $\mathcal{O}$ is an oracle for a *sampling* problem. Namely, it is essential that $\mathcal{O}$'s *only* source of randomness be a string $r \in \{0,1\}^{\mathrm{poly}(n)}$ that is also given to $\mathcal{O}$ as input. In other words, if we fix r then $\mathcal{O}(A,r)$ must be deterministic, just like a conventional oracle that decides a language. Of course, if $\mathcal{O}$ were implemented by a classical algorithm, this requirement would be trivial to satisfy.

More formally:

[19]Here we assume each entry of A is represented in binary, so that it has the form $(x+yi)/2^{p(n)}$ where x and y are integers and p is some fixed polynomial. As a consequence, A might not be *exactly* column-orthonormal—but as long as $A^\dagger A$ is exponentially close to the identity, A can easily be "corrected" to an element of $\mathcal{U}_{m,n}$ using Gram-Schmidt orthogonalization. Furthermore, it is not hard to show that every element of $\mathcal{U}_{m,n}$ can be approximated in this manner. See for example Aaronson [1] for a detailed error analysis.

Definition 3.11 (BOSONSAMPLING oracle)**.** Let $\mathcal{O}$ be an oracle that takes as input a string $r \in \{0,1\}^{\text{poly}(n)}$ an $m \times n$ matrix $A \in \mathcal{U}_{m,n}$ and an error bound $\varepsilon > 0$ encoded as $0^{1/\varepsilon}$. Also, let $\mathcal{D}_{\mathcal{O}}(A,\varepsilon)$ be the distribution over outputs of $\mathcal{O}$ if A and ε are fixed but r is uniformly random. We call $\mathcal{O}$ an exact BOSONSAMPLING oracle if $\mathcal{D}_{\mathcal{O}}(A,\varepsilon) = \mathcal{D}_A$ for all $A \in \mathcal{U}_{m,n}$. Also, we call $\mathcal{O}$ an approximate BOSONSAMPLING oracle if $\|\mathcal{D}_{\mathcal{O}}(A,\varepsilon) - \mathcal{D}_A\| \leq \varepsilon$ for all $A \in \mathcal{U}_{m,n}$ and $\varepsilon > 0$.

If we like, we can define the complexity class $\mathsf{BosonFP}$ to be the set of search problems $R = (B_x)_{x\in\{0,1\}^*}$ that are in $\mathsf{FBPP}^{\mathcal{O}}$ for every exact BOSONSAMPLING oracle $\mathcal{O}$. We can also define $\mathsf{BosonFP}_\varepsilon$ to be the set of search problems that are in $\mathsf{FBPP}^{\mathcal{O}}$ for every approximate BOSONSAMPLING oracle $\mathcal{O}$. We then have the following basic inclusions:

Theorem 3.12. $\mathsf{FBPP} \subseteq \mathsf{BosonFP}_\varepsilon = \mathsf{BosonFP} \subseteq \mathsf{FBQP}$.

Proof. For $\mathsf{FBPP} \subseteq \mathsf{BosonFP}_\varepsilon$ just ignore the BOSONSAMPLING oracle. For $\mathsf{BosonFP}_\varepsilon \subseteq \mathsf{BosonFP}$ note that any exact BOSONSAMPLING oracle is also an ε-approximate one for every ε. For the other direction, $\mathsf{BosonFP} \subseteq \mathsf{BosonFP}_\varepsilon$, let M be a $\mathsf{BosonFP}$ machine, and let $\mathcal{O}$ be M's exact BOSONSAMPLING oracle. Since M has to work for *every* $\mathcal{O}$ we can assume without loss of generality that $\mathcal{O}$ is chosen uniformly at random, consistent with the requirement that $\mathcal{D}_{\mathcal{O}}(A) = \mathcal{D}_A$ for every A. We claim that we can simulate $\mathcal{O}$ to sufficient accuracy using an *approximate* BOSONSAMPLING oracle. To do so, we simply choose $\varepsilon \ll \delta/p(n)$ where $p(n)$ is an upper bound on the number of queries to $\mathcal{O}$ made by M and δ is the desired failure probability of M. For $\mathsf{BosonFP} \subseteq \mathsf{FBQP}$ we use an old observation of Feynman [26] and Abrams and Lloyd [6]: that fermionic and bosonic systems can be simulated efficiently on a standard quantum computer. In more detail, our quantum computer's state at any time step will have the form

$$|\psi\rangle = \sum_{(s_1,\ldots,s_m)\in\Phi_{m,n}} \alpha_{s_1,\ldots,s_m} |s_1,\ldots,s_m\rangle \,. \tag{3.69}$$

That is, we simply encode each occupation number $0 \leq s_i \leq n$ in binary using $\lceil \log_2 n \rceil$ qubits. (Thus, the total number of qubits in our simulation is $m\lceil \log_2 n \rceil$.) To initialize, we prepare the state $|1_n\rangle = |1,\ldots,1,0,\ldots,0\rangle$; to measure, we measure in the computational basis. As for simulating an optical element: recall that such an element acts nontrivially only on two modes i and j and hence on $2\lceil \log_2 n \rceil$ qubits. So we can describe an optical element by an $O(n^2) \times O(n^2)$ unitary matrix U—and furthermore, we gave an explicit formula (3.9) for the entries of U. It follows immediately, from the Solovay-Kitaev Theorem (see [50]), that we can simulate U with error ε using $\text{poly}(n, \log 1/\varepsilon)$ qubit gates. Therefore an FBQP machine can simulate each call that a $\mathsf{BosonFP}$ machine makes to the BOSONSAMPLING oracle. □

4 Efficient classical simulation of linear optics collapses PH

In this section we prove Theorem 1.1, our hardness result for exact BOSONSAMPLING. First, in Section 4.1, we prove that $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{BPP}^{\mathsf{NP}^{\mathcal{O}}}$, where $\mathcal{O}$ is any exact BOSONSAMPLING oracle. In particular, this implies that, if there exists a polynomial-time classical algorithm for exact BOSONSAMPLING, then $\mathsf{P}^{\#\mathsf{P}} = \mathsf{BPP}^{\mathsf{NP}}$ and hence the polynomial hierarchy collapses to the third level. The proof in Section 4.1 directly exploits the fact that boson amplitudes are given by the permanents of complex matrices $X \in \mathbb{C}^{n\times n}$

and that approximating $\text{Per}(X)$ given such an X is #P-complete. The main lemma we need to prove is simply that approximating $|\text{Per}(X)|^2$ is *also* #P-complete. Next, in Section 4.2, we give a completely different proof of Theorem 1.1. This proof repurposes two existing results in quantum computation: the scheme for universal quantum computing with *adaptive* linear optics due to Knill, Laflamme, and Milburn [43], and the PostBQP = PP theorem of Aaronson [2]. Finally, in Section 4.3, we observe two improvements to the basic result.

4.1 Basic result

First, we will need a classic result of Stockmeyer [63].

Theorem 4.1 (Stockmeyer [63]). *Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$ let*

$$p = \Pr_{x \in \{0,1\}^n} [f(x) = 1] = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x) . \tag{4.1}$$

Then for all $g \geq 1 + 1/\text{poly}(n)$ there exists an $\mathsf{FBPP}^{\mathsf{NP}^f}$ machine that approximates p to within a multiplicative factor of g.

Intuitively, Theorem 4.1 says that a $\mathsf{BPP}^{\mathsf{NP}}$ machine can always estimate the probability p that a polynomial-time randomized algorithm accepts to within a $1/\text{poly}(n)$ multiplicative factor, even if p is exponentially small. Note that Theorem 4.1 does *not* generalize to estimating the probability that a quantum algorithm accepts, since the randomness is "built in" to a quantum algorithm, and the $\mathsf{BPP}^{\mathsf{NP}}$ machine does not get to choose or control it.

Another interpretation of Theorem 4.1 is that any counting problem that involves *estimating the sum of 2^n nonnegative real numbers*[20] can be approximately solved in $\mathsf{BPP}^{\mathsf{NP}}$.

By contrast, if a counting problem involves estimating a sum of *both positive and negative numbers*—for example, if one wanted to approximate $\text{E}_{x \in \{0,1\}^n}[f(x)]$ for some function $f : \{0,1\}^n \to \{-1,1\}$—then the situation is completely different. In that case, it is easy to show that even multiplicative approximation is #P-hard, and hence unlikely to be in $\mathsf{FBPP}^{\mathsf{NP}}$.

We will show this phenomenon in the special case of the permanent. If X is a non-negative matrix, then Jerrum, Sinclair, and Vigoda [37] famously showed that one can approximate $\text{Per}(X)$ to within multiplicative error ε in $\text{poly}(n, 1/\varepsilon)$ time (which improves on Theorem 4.1 by getting rid of the NP oracle). On the other hand, let $X \in \mathbb{R}^{n \times n}$ be an arbitrary real matrix, with both positive and negative entries. Then we will show that multiplicatively approximating $\text{Per}(X)^2 = |\text{Per}(X)|^2$ is #P-hard. The reason why we are interested in $|\text{Per}(X)|^2$ rather than $\text{Per}(X)$ itself, is that measurement probabilities in the noninteracting-boson model are the absolute squares of permanents.

Our starting point is a famous result of Valiant [71]:

Theorem 4.2 (Valiant [71]). *The following problem is* #P*-complete: given a matrix $X \in \{0,1\}^{n \times n}$ compute* $\text{Per}(X)$.

[20] Strictly speaking, Theorem 4.1 talks about estimating the sum of 2^n *binary* ($\{0,1\}$-valued) numbers, but it is easy to generalize to arbitrary nonnegative reals.

We now show that $\mathrm{Per}(X)^2$ is #P-hard to approximate.

Theorem 4.3 (Hardness of Approximating $\mathrm{Per}(X)^2$)**.** *The following problem is #P-hard, for any $g \in [1, \mathrm{poly}(n)]$: given a real matrix $X \in \mathbb{R}^{n\times n}$ approximate $\mathrm{Per}(X)^2$ to within a multiplicative factor of g.*

Proof. Let $\mathcal{O}$ be an oracle that, given a matrix $M \in \mathbb{R}^{n\times n}$ outputs a nonnegative real number $\mathcal{O}(M)$ such that

$$\frac{\mathrm{Per}(M)^2}{g} \leq \mathcal{O}(M) \leq g\,\mathrm{Per}(M)^2 . \tag{4.2}$$

Also, let $X = (x_{ij}) \in \{0,1\}^{n\times n}$ be an input matrix, which we assume for simplicity consists only of 0s and 1s. Then we will show how to compute $\mathrm{Per}(X)$ exactly, in polynomial time and using $O\left(gn^2 \log n\right)$ adaptive queries to $\mathcal{O}$. Since $\mathrm{Per}(X)$ is #P-complete by Theorem 4.2, this will immediately imply the lemma. Since X is non-negative, we can check in polynomial time whether $\mathrm{Per}(X) = 0$. If $\mathrm{Per}(X) = 0$ we are done, so assume $\mathrm{Per}(X) \geq 1$. Then there exists a permutation σ such that $x_{1,\sigma(1)} = \cdots = x_{n,\sigma(n)} = 1$. Moreover, we can find such a σ in polynomial time; indeed, this is equivalent to the standard problem of finding a *perfect matching* in a bipartite graph. By permuting the rows and columns, we can assume without loss of generality that $x_{11} = \cdots = x_{nn} = 1$. Our reduction will use recursion on n. Let $Y = (y_{ij})$ be the bottom-right $(n-1)\times(n-1)$ submatrix of X. Then we will assume inductively that we already know $\mathrm{Per}(Y)$. We will use that knowledge, together with $O(gn\log n)$ queries to $\mathcal{O}$ to find $\mathrm{Per}(X)$. Given a real number r let $X^{[r]} \in \mathbb{R}^{n\times n}$ be a matrix identical to X except that the top-left entry is $x_{11} - r$ instead of x_{11}. Then it is not hard to see that

$$\mathrm{Per}\left(X^{[r]}\right) = \mathrm{Per}(X) - r\,\mathrm{Per}(Y) . \tag{4.3}$$

Note that $y_{11} = \cdots = y_{(n-1),(n-1)} = 1$ so $\mathrm{Per}(Y) \geq 1$. Hence there must be a unique value $r = r^*$ such that $\mathrm{Per}\left(X^{[r^*]}\right) = 0$. Furthermore, if we can find that r^*, then we are done, since $\mathrm{Per}(X) = r^*\,\mathrm{Per}(Y)$. To find

$$r^* = \frac{\mathrm{Per}(X)}{\mathrm{Per}(Y)} , \tag{4.4}$$

we will use a procedure based on binary search. Let $r(0) := 0$ be our "initial guess"; then we will repeatedly improve this guess to $r(1)$ $r(2)$, etc. The invariant we want to maintain is that

$$\mathcal{O}\left(X^{[r(t+1)]}\right) \leq \frac{\mathcal{O}\left(X^{[r(t)]}\right)}{2} \tag{4.5}$$

for all t. To find $r(t+1)$ starting from $r(t)$: first observe that

$$|r(t) - r^*| = \frac{|r(t)\,\mathrm{Per}(Y) - \mathrm{Per}(X)|}{\mathrm{Per}(Y)} \tag{4.6}$$

$$= \frac{\left|\mathrm{Per}\left(X^{[r(t)]}\right)\right|}{\mathrm{Per}(Y)} \tag{4.7}$$

$$\leq \frac{\sqrt{g \cdot \mathcal{O}\left(X^{[r(t)]}\right)}}{\mathrm{Per}(Y)} , \tag{4.8}$$

where line (4.8) follows from $\operatorname{Per}(M)^2/g \leq \mathcal{O}(M)$. So setting

$$\beta := \frac{\sqrt{g \cdot \mathcal{O}\left(X^{[r(t)]}\right)}}{\operatorname{Per}(Y)}, \tag{4.9}$$

we find that r^* is somewhere in the interval $I := [r(t)-\beta, r(t)+\beta]$. Divide I into L equal segments (for some L to be determined later), and let $s(1), \ldots, s(L)$ be their left endpoints. Then the procedure is to evaluate $\mathcal{O}\left(X^{[s(i)]}\right)$ for each $i \in [L]$ and set $r(t+1)$ equal to the $s(i)$ for which $\mathcal{O}\left(X^{[s(i)]}\right)$ is minimized (breaking ties arbitrarily). Clearly there exists an $i \in [L]$ such that $|s(i)-r^*| \leq \beta/L$—and for that particular choice of i we have

$$\begin{aligned}
\mathcal{O}\left(X^{[s(i)]}\right) &\leq g \operatorname{Per}\left(X^{[s(i)]}\right)^2 && (4.10)\\
&= g\left(\operatorname{Per}(X) - s(i)\operatorname{Per}(Y)\right)^2 && (4.11)\\
&= g\left(\operatorname{Per}(X) - (s(i)-r^*)\operatorname{Per}(Y) - r^*\operatorname{Per}(Y)\right)^2 && (4.12)\\
&= g\left(s(i)-r^*\right)^2 \operatorname{Per}(Y)^2 && (4.13)\\
&\leq g\frac{\beta^2}{L^2}\operatorname{Per}(Y)^2 && (4.14)\\
&= \frac{g^2}{L^2}\mathcal{O}\left(X^{[r(t)]}\right). && (4.15)
\end{aligned}$$

Therefore, so long as we choose $L \geq \sqrt{2}g$ we find that

$$\mathcal{O}\left(X^{[r(t+1)]}\right) \leq \mathcal{O}\left(X^{[s(i)]}\right) \leq \frac{\mathcal{O}\left(X^{[r(t)]}\right)}{2}, \tag{4.16}$$

which is what we wanted. Now observe that

$$\mathcal{O}\left(X^{[r(0)]}\right) = \mathcal{O}(X) \leq g\operatorname{Per}(X)^2 \leq g(n!)^2. \tag{4.17}$$

So for some $T = O(n\log n)$,

$$\mathcal{O}\left(X^{[r(T)]}\right) \leq \frac{\mathcal{O}\left(X^{[r(0)]}\right)}{2^T} \leq \frac{g(n!)^2}{2^T} \ll \frac{1}{4g}. \tag{4.18}$$

By lines (4.6)-(4.8), this in turn implies that

$$|r(T)-r^*| \leq \frac{\sqrt{g\cdot\mathcal{O}\left(X^{[r(T)]}\right)}}{\operatorname{Per}(Y)} \ll \frac{1}{2\operatorname{Per}(Y)}. \tag{4.19}$$

But this means that we can find r^* exactly, since r^* equals a rational number $\operatorname{Per}(X)/\operatorname{Per}(Y)$ where $\operatorname{Per}(X)$ and $\operatorname{Per}(Y)$ are both positive integers and $\operatorname{Per}(Y)$ is known. □

Let us remark that one can improve Theorem 4.3, to ensure that the entries of X are all at most $\operatorname{poly}(n)$ in absolute value. We do not pursue that here, since it will not be needed for our application.

Lemma 4.4. *Let $X \in \mathbb{C}^{n\times n}$. Then for all $m \geq 2n$ and $\varepsilon \leq 1/\|X\|$ there exists an $m \times m$ unitary matrix U that contains εX as a submatrix. Furthermore, U can be computed in polynomial time given X.*

Proof. Let $Y = \varepsilon X$. Then it suffices to show how to construct a $2n \times n$ matrix W whose columns are orthonormal vectors, and that contains Y as its top $n \times n$ submatrix. For such a W can easily be completed to an $m \times n$ matrix whose columns are orthonormal (by filling the bottom $m-2n$ rows with zeroes), which can in turn be completed to an $m \times m$ unitary matrix in $O\left(m^3\right)$ time. Since $\|Y\| \leq \varepsilon \|X\| \leq 1$, we have $Y^\dagger Y \preceq I$ in the semidefinite ordering. Hence $I - Y^\dagger Y$ is positive semidefinite. So $I - Y^\dagger Y$ has a Cholesky decomposition $I - Y^\dagger Y = Z^\dagger Z$ for some $Z \in \mathbb{C}^{n\times n}$. Let us set

$$W := \begin{pmatrix} Y \\ \hline Z \end{pmatrix}.$$

Then $W^\dagger W = Y^\dagger Y + Z^\dagger Z = I$ so the columns of W are orthonormal as desired. □

We are now ready to prove Theorem 1.1: that $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{BPP}^{\mathsf{NP}^{\mathcal{O}}}$ for any exact BOSONSAMPLING oracle $\mathcal{O}$.

Proof of Theorem 1.1. Given a matrix $X \in \mathbb{R}^{n\times n}$ and a parameter

$$g \in \left[1 + \frac{1}{\operatorname{poly}(n)}, \operatorname{poly}(n)\right]$$

we know from Theorem 4.3 that it is #P-hard to approximate $\operatorname{Per}(X)^2$ to within a multiplicative factor of g. So to prove the theorem, it suffices to show how to approximate $\operatorname{Per}(X)^2$ in $\mathsf{FBPP}^{\mathsf{NP}^{\mathcal{O}}}$. Set $m := 2n$ and $\varepsilon := 1/\|X\| \geq 2^{-\operatorname{poly}(n)}$. Then by Lemma 4.4, we can efficiently construct an $m \times m$ unitary matrix U with $U_{n,n} = \varepsilon X$ as its top-left $n \times n$ submatrix. Let A be the $m \times n$ column-orthonormal matrix corresponding to the first n columns of U. Let us feed A as input to $\mathcal{O}$ and consider the probability p_A that $\mathcal{O}$ outputs 1_n. We have

$$\begin{aligned} p_A &= \Pr_r\left[\mathcal{O}(A, r) = 1_n\right] && (4.20)\\ &= \left|\langle 1_n|\varphi(U)|1_n\rangle\right|^2 && (4.21)\\ &= \left|\operatorname{Per}(U_{n,n})\right|^2 && (4.22)\\ &= \varepsilon^{2n}\left|\operatorname{Per}(X)\right|^2, && (4.23)\end{aligned}$$

where line (4.22) follows from Theorem 3.10. But by Theorem 4.1, we can approximate p_A to within a multiplicative factor of g in $\mathsf{FBPP}^{\mathsf{NP}^{\mathcal{O}}}$. It follows that we can approximate $\left|\operatorname{Per}(X)\right|^2 = \operatorname{Per}(X)^2$ in $\mathsf{FBPP}^{\mathsf{NP}^{\mathcal{O}}}$ as well. □

The main fact that we wanted to prove is an immediate corollary of Theorem 1.1:

Corollary 4.5. *Suppose exact* BOSONSAMPLING *can be done in classical polynomial time. Then* $\mathsf{P}^{\#\mathsf{P}} = \mathsf{BPP}^{\mathsf{NP}}$ *and hence the polynomial hierarchy collapses to the third level.*

Proof. Combining the assumption with Theorem 1.1, we get that $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{BPP}^{\mathsf{NP}}$ which by Toda's Theorem [69] implies that $\mathsf{P}^{\#\mathsf{P}} = \mathsf{PH} = \Sigma_3^{\mathsf{P}} = \mathsf{BPP}^{\mathsf{NP}}$. □

Likewise, even if exact BOSONSAMPLING can be done in $\mathsf{BPP}^{\mathsf{PH}}$ (that is, using an oracle for some fixed level of the polynomial hierarchy), we still get that

$$\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{BPP}^{\mathsf{NP}^{\mathsf{PH}}} = \mathsf{BPP}^{\mathsf{PH}} = \mathsf{PH}\,, \tag{4.24}$$

and hence PH collapses.

As another application of Theorem 1.1, suppose that exact BOSONSAMPLING can be done in $\mathsf{BPP}^{\mathsf{PromiseBQP}}$: that is, using an oracle for BQP *decision* problems. Then we get the containment

$$\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{BPP}^{\mathsf{NP}^{\mathsf{PromiseBQP}}}. \tag{4.25}$$

Such a containment seems unlikely (though we admit to lacking a strong intuition here), thereby providing possible evidence for a separation between BQP sampling problems and BQP decision problems.

4.2 Alternate proof using KLM

Inspired by recent work of Bremner et al. [12], in this section we give a different proof of Theorem 1.1. This proof makes no use of permanents or approximate counting; instead, it invokes two previous quantum computing results—the KLM Theorem [43] and the $\mathsf{PostBQP} = \mathsf{PP}$ theorem [2]—as black boxes. Compared to the first proof, the second one has the advantage of being shorter and completely free of calculations; also, it easily generalizes to many other quantum computing models, besides noninteracting bosons. The disadvantage is that, to those unfamiliar with [43, 2], the second proof gives less intuition about why Theorem 1.1 is true. Also, we do not know how to generalize the second proof to say anything about the hardness of *approximate* sampling. For that, it seems essential to talk about the PERMANENT or some other concrete $\#\mathsf{P}$-complete problem.

Our starting point is the KLM Theorem, which says informally that linear optics augmented with single-photon inputs, as well as adaptive demolition measurements in the photon-number basis, is universal for quantum computation. A bit more formally, let $\mathsf{BosonP}_{\text{adap}}$ be the class of languages that are decidable in BPP (that is, classical probabilistic polynomial-time), augmented with the ability to:

(1) Prepare single-photon Fock states in any of $m = \text{poly}(n)$ modes.

(2) Apply arbitrary optical elements to pairs of modes.

(3) Measure the photon number of any mode at any time (in a way that destroys the photons in that mode).

(4) Condition future optical elements and classical computations on the outcomes of the measurements.

From Theorem 3.12, it is not hard to see that $\mathsf{BosonP}_{\text{adap}} \subseteq \mathsf{BQP}$. The amazing discovery of Knill et al. [43] was that the other direction holds as well:

Theorem 4.6 (KLM Theorem [43])**.** $\mathsf{BosonP}_{\text{adap}} = \mathsf{BQP}$.

In the proof of Theorem 4.6, a key step is to consider a model of linear optics with *postselected* demolition measurements. This is similar to the model with adaptive measurements described above, except that here we *guess* the outcomes of all the photon-number measurements at the very beginning, and then only proceed with the computation if the guesses turn out to be correct. In general, the resulting computation will only succeed with exponentially-small probability, but we know when it does succeed.

Notice that, in this model, there is never any need to condition later computational steps on the outcomes of measurements—since *if* the computation succeeds, then we know in advance what all the measurement outcomes are anyway! One consequence is that, without loss of generality, we can postpone all measurements until the end of the computation.[21]

Along the way to proving Theorem 4.6, Knill et al. [43] showed how to simulate any *postselected* quantum computation using a *postselected* linear-optics computation.[22] To formalize the "Postselected KLM Theorem," we now define the complexity class $\mathsf{PostBosonP}$ which consists of all problems solvable in polynomial time using linear optics with postselected demolition measurements.

Definition 4.7 ($\mathsf{PostBosonP}$)**.** $\mathsf{PostBosonP}$ is the class of languages $L \subseteq \{0,1\}^*$ for which there exist deterministic polynomial-time algorithms $\mathcal{V}, \mathcal{A}, \mathcal{B}$ such that for all inputs $x \in \{0,1\}^N$:

(i) The output of $\mathcal{V}$ is an $m \times n$ matrix $V(x) \in \mathcal{U}_{m,n}$ (for some $m, n = \mathrm{poly}(N)$), corresponding to a linear-optical network that samples from the probability distribution $\mathcal{D}_{V(x)}$.

(ii) $\Pr_{y \sim \mathcal{D}_{V(x)}}[\mathcal{A}(y) \text{ accepts}] > 0$.

(iii) If $x \in L$ then $\Pr_{y \sim \mathcal{D}_{V(x)}}[\mathcal{B}(y) \text{ accepts} \mid \mathcal{A}(y) \text{ accepts}] \geq \frac{2}{3}$.

(iv) If $x \notin L$ then $\Pr_{y \sim \mathcal{D}_{V(x)}}[\mathcal{B}(y) \text{ accepts} \mid \mathcal{A}(y) \text{ accepts}] \leq \frac{1}{3}$.

In our terminology, Knill et al. [43] showed that $\mathsf{PostBosonP}$ captures the full power of postselected quantum computation—in other words, of the class $\mathsf{PostBQP}$ defined in Section 2. We now sketch a proof for completeness.

Theorem 4.8 (Postselected KLM Theorem [43])**.** $\mathsf{PostBosonP} = \mathsf{PostBQP}$.

Proof Sketch. For $\mathsf{PostBosonP} \subseteq \mathsf{PostBQP}$ use the procedure from Theorem 3.12, to create an ordinary quantum circuit C that simulates a given linear-optical network U. Note that the algorithms $\mathcal{A}$ and $\mathcal{B}$ from Definition 4.7 can simply be "folded" into C so that $\mathcal{A}(y)$ accepting corresponds to the first qubit of C's output being measured to be $|1\rangle$ and $\mathcal{B}(y)$ accepting corresponds to the second qubit of C's output being measured to be $|1\rangle$. The more interesting direction is $\mathsf{PostBQP} \subseteq \mathsf{PostBosonP}$. To simulate BQP in $\mathsf{PostBosonP}$, the basic idea of KLM is to use "nondeterministic gates," which consist of sequences of beamsplitters and phaseshifters followed by postselected demolition measurements in the photon-number basis. *If* the measurements return a particular outcome, *then* the effect of the beamsplitters

[21]For this argument to work, it was essential that the measurements were *demolition* measurements. Nondemolition measurements—even if they are nonadaptive—*cannot* generally be postponed to the end of the computation, since for them the post-measurement quantum state matters as well.

[22]Terhal and DiVincenzo [67] later elaborated on their result, using the term "nonadaptive quantum computation" (or QC_{nad}) for what we call postselection.

and phaseshifters is to implement (perfectly) a 2-qubit gate that is known to be universal for standard quantum computation. We refer the reader to [43] for the details of how such gates are constructed; for now, assume we have them. Then for any BQP machine M it is easy to create a $\mathsf{PostBosonP}$ machine M' that simulates M. But once we have BQP we also get $\mathsf{PostBQP}$ essentially "free of charge." This is because the simulating machine M' can postselect, not only on its nondeterministic gates working correctly, but also (say) on M reaching a final configuration whose first qubit is $|1\rangle$. □

We can now complete our alternative proof of Theorem 1.1, that $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{BPP}^{\mathsf{NP}^{\mathcal{O}}}$ for any exact BOSONSAMPLING oracle $\mathcal{O}$.

Proof of Theorem 1.1. Let $\mathcal{O}$ be an exact BOSONSAMPLING oracle. Then we claim that $\mathsf{PostBosonP} \subseteq \mathsf{PostBPP}^{\mathcal{O}}$. To see this, let $\mathcal{V}, \mathcal{A}, \mathcal{B}$ be the polynomial-time Turing machines from Definition 4.7. Then we can create a $\mathsf{PostBPP}^{\mathcal{O}}$ machine that, given an input x and random string r:

(i) "Succeeds" if $\mathcal{A}(\mathcal{O}(V(x), r))$ accepts, and "fails" otherwise.

(ii) Conditioned on succeeding, accepts if $\mathcal{B}(\mathcal{O}(V(x), r))$ accepts and rejects otherwise.

Hence

$$\begin{aligned} \mathsf{PP} &= \mathsf{PostBQP} && (4.26)\\ &= \mathsf{PostBosonP} && (4.27)\\ &\subseteq \mathsf{PostBPP}^{\mathcal{O}} && (4.28)\\ &\subseteq \mathsf{BPP}^{\mathsf{NP}^{\mathcal{O}}}. && (4.29) \end{aligned}$$

Here line (4.26) comes from Theorem 2.2, line (4.27) from Theorem 4.8, line (4.28) from the claim, and line (4.29) from equation (2.1). Therefore $\mathsf{P}^{\#\mathsf{P}} = \mathsf{P}^{\mathsf{PP}}$ is contained in $\mathsf{BPP}^{\mathsf{NP}^{\mathcal{O}}}$ as well. □

4.3 Strengthening the result

In this section, we make two simple but interesting improvements to Theorem 1.1.

The first improvement is this: instead of considering a whole collection of distributions, we can give a *fixed* distribution $\mathcal{D}_n$ (depending only on the input size n) that can be sampled by a boson computer, but that cannot be efficiently sampled classically unless the polynomial hierarchy collapses. This $\mathcal{D}_n$ will effectively be a "complete distribution" for the noninteracting-boson model under nondeterministic reductions. Let us discuss how to construct such a $\mathcal{D}_n$ using the approach of Section 4.2.

Let $p(n)$ be some fixed polynomial (say n^2), and let $\mathcal{C}$ be the set of all quantum circuits on n qubits with at most $p(n)$ gates (over some finite universal basis, such as $\{$HADAMARD, TOFFOLI$\}$ [59]). Then consider the following $\mathsf{PostBQP}$ algorithm $\mathcal{A}$ which takes as input a description of a circuit $C^* \in \mathcal{C}$. First, generate a uniform superposition

$$|\mathcal{C}\rangle = \frac{1}{\sqrt{|\mathcal{C}|}} \sum_{C \in \mathcal{C}} |C\rangle \qquad (4.30)$$

over descriptions of all circuits $C \in \mathcal{C}$. Then measure $|\mathcal{C}\rangle$ in the standard basis, and postselect on the outcome being $|C^*\rangle$. Finally, assuming $|C^*\rangle$ was obtained, take some fixed universal circuit U with the property that

$$\Pr[U(|C\rangle) \text{ accepts}] \approx \Pr[C(0^n) \text{ accepts}] \tag{4.31}$$

for all $C \in \mathcal{C}$ and run U on input $|C^*\rangle$. Now, since $\mathsf{PostBQP} = \mathsf{PostBosonP}$ by Theorem 4.8, it is clear that $\mathcal{A}$ can be "compiled" into a postselected linear-optical network $\mathcal{A}'$. Let $\mathcal{D}_{\mathcal{A}'}$ be the probability distribution sampled by $\mathcal{A}'$ if we ignore the postselection steps. Then $\mathcal{D}_{\mathcal{A}'}$ is our desired universal distribution $\mathcal{D}_n$.

More concretely, we claim that, if $\mathcal{D}_n$ can be sampled in FBPP then $\mathsf{P}^{\#\mathsf{P}} = \mathsf{PH} = \mathsf{BPP}^{\mathsf{NP}}$. To see this, let $\mathcal{O}(r)$ be a polynomial-time classical algorithm that outputs a sample from $\mathcal{D}_n$ given as input a random string $r \in \{0,1\}^{\operatorname{poly}(n)}$. Then, as in the proof of Theorem 1.1 in Section 4.2, we have $\mathsf{PostBosonP} \subseteq \mathsf{PostBPP}$. For let $\mathcal{V}, \mathcal{A}, \mathcal{B}$ be the polynomial-time algorithms from Definition 4.7. Then we can create a $\mathsf{PostBPP}$ machine that, given an input x and random string r:

(1) Postselects on $\mathcal{O}(r)$ containing an encoding of the linear-optical network $V(x)$.

(2) Assuming $|V(x)\rangle$ is observed, simulates the $\mathsf{PostBosonP}$ algorithm: that is, "succeeds" if $\mathcal{A}(\mathcal{O}(r))$ accepts and fails otherwise, and "accepts" if $\mathcal{B}(\mathcal{O}(r))$ accepts and rejects otherwise.

Our second improvement to Theorem 1.1 weakens the physical resource requirements needed to sample from a hard distribution. Recall that we assumed our boson computer began in the "standard initial state" $|1_n\rangle := |1,\dots,1,0,\dots,0\rangle$ in which the first n modes were occupied by a single boson each. Unfortunately, in the optical setting, it is notoriously difficult to produce a single photon on demand (see Section 6 for more about this). Using a standard laser, it is much easier to produce so-called *coherent states*, which have the form

$$|\alpha\rangle := e^{-|\alpha|^2/2} \sum_{n=0}^{\infty} \frac{\alpha^n}{\sqrt{n!}} |n\rangle \tag{4.32}$$

for some complex number α. (Here $|n\rangle$ represents a state of n photons.) However, we now observe that the KLM-based proof of Theorem 1.1 goes through almost without change, if the inputs are coherent states rather than single-photon Fock states, *and* nondemolition measurements are available. The reason is that, in the $\mathsf{PostBosonP}$ model, we can first prepare a coherent state (say $|\alpha = 1\rangle$), then measure it and *postselect* on getting a single photon. In this way, we can use postselection to generate the standard initial state $|1_n\rangle$ then run the rest of the computation as before.

Summarizing the improvements:

Theorem 4.9. *There exists a family of distributions $\{\mathcal{D}_n\}_{n \ge 1}$ depending only on n such that:*

(i) For all n a boson computer with single-photon inputs and demolition measurements, or coherent-state inputs and nondemolition measurements, can sample from $\mathcal{D}_n$ in $\operatorname{poly}(n)$ time.

(ii) Let $\mathcal{O}$ be any oracle that takes as input a random string r (which $\mathcal{O}$ uses as its only source of randomness) together with n and that outputs a sample $\mathcal{O}_n(r)$ from $\mathcal{D}_n$. Then $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{BPP}^{\mathsf{NP}^{\mathcal{O}}}$.

5 Main result

We now move on to prove our main result: that even *approximate* classical simulation of boson computations would have surprising complexity consequences.

5.1 Truncations of Haar-random unitaries

In this section, for completeness, we prove a statement we will need from random matrix theory. Namely: *any $m^{1/6} \times m^{1/6}$ submatrix of an $m \times m$ Haar-random unitary matrix is close, in variation distance, to a matrix of* i. i. d. *Gaussians.* It is easy to see that any individual *entry* of a Haar unitary matrix is approximately Gaussian. Thus, our result just says that any small enough *set* of entries is approximately independent—and that here, "small enough" can mean not only a constant number of entries, but even $m^{\Omega(1)}$ of them. This is not surprising: it simply means that one needs to examine a significant fraction of the entries before one "notices" the unitarity constraint.

Given $m \geq n$ recall from Section 2 that $\mathcal{U}_{m,n}$ is the set of $m \times n$ complex matrices whose columns are orthonormal vectors, and $\mathcal{H}_{m,n}$ is the Haar measure over $\mathcal{U}_{m,n}$. Define $\mathcal{S}_{m,n}$ to be the distribution over $n \times n$ matrices obtained by first drawing a unitary U from $\mathcal{H}_{m,m}$ and then outputting $\sqrt{m}U_{n,n}$ where $U_{n,n}$ is the top-left $n \times n$ submatrix of U. In other words, $\mathcal{S}_{m,n}$ is the distribution over $n \times n$ truncations of $m \times m$ Haar unitary matrices, where the entries have been scaled up by a factor of $\sqrt{m}$ so that they have mean 0 and variance 1. Also, recall that $\mathcal{G}^{n\times n}$ is the probability distribution over $n \times n$ complex matrices whose entries are independent Gaussians with mean 0 and variance 1. Then our main result states that $\mathcal{S}_{m,n}$ is close in variation distance to $\mathcal{G}^{n\times n}$:

Theorem 5.1. *Let $m \geq \frac{n^5}{\delta}\log^2\frac{n}{\delta}$ for any $\delta > 0$. Then $\|\mathcal{S}_{m,n} - \mathcal{G}^{n\times n}\| = O(\delta)$.*

The bound $m \geq (n^5/\delta)\log^2(n/\delta)$ is almost certainly not tight. For our purposes, however, what is important is simply that m is polynomial in n and $1/\delta$. We strongly conjecture that the bound can be improved to $m \geq n^{2+\varepsilon}/\delta$. Indeed, right before this paper's publication, we learned of a paper by Jiang [38], who essentially shows that $m = \omega\left(n^2\right)$ suffices for the analogous problem with *real orthogonal matrices* instead of unitary matrices. More precisely, Jiang shows that when $m = \omega\left(n^2\right)$ the variation distance between an $n \times n$ real Gaussian matrix and an $n \times n$ truncation of an $m \times m$ scaled Haar-random real orthogonal matrix tends to 0 as $n \to \infty$. (On the other hand, Jiang does not bound the *rate* at which the variation distance goes to 0 which we do and which is needed for our application.) Jiang also proves his bound to be tight, in the sense that $m = O\left(n^2\right)$ no longer suffices. We conjecture both that his results should carry over to the complex case, and that the variation distance should tend to 0 sufficiently rapidly.

Let $p_G, p_S : \mathbb{C}^{n\times n} \to \mathbb{R}^+$ be the probability density functions of $\mathcal{G}^{n\times n}$ and $\mathcal{S}_{m,n}$ respectively (for convenience, we drop the subscripts m and n). Then for our application, we will actually need the following stronger version of Theorem 5.1:

Theorem 5.2 (Haar-Unitary Hiding Theorem)**.** *Let $m \geq \frac{n^5}{\delta}\log^2\frac{n}{\delta}$. Then*

$$p_S(X) \leq (1 + O(\delta))\, p_G(X) \tag{5.1}$$

for all $X \in \mathbb{C}^{n\times n}$.

Fortunately, Theorem 5.2 will follow fairly easily from our proof of Theorem 5.1.

Truncations of Haar unitary matrices have been studied in detail, but our specific results in Theorems 5.1 and 5.2 bound the asymptotics of convergence in a different way that what seems to have appeared in the random matrix theory literature. Petz and Réffy [52] showed that the truncated Haar-unitary distribution $\mathcal{S}_{m,n}$ converges to the Gaussian distribution, when n is fixed and $m \to \infty$. (Mastrodonato and Tumulka [49] later gave an elementary proof of this fact.) In a followup paper, Petz and Réffy [53] proved a large deviation bound for the empirical eigenvalue density of matrices drawn from $\mathcal{S}_{m,n}$ (see also Réffy's Ph. D. thesis [55]). We will use some observations from those papers, especially an explicit formula in [55] for the probability density function of $\mathcal{S}_{m,n}$.

We now give an overview of the proof of Theorem 5.1. Our goal is to prove that

$$\Delta(p_G, p_S) := \int_{X \in \mathbb{C}^{n \times n}} |p_G(X) - p_S(X)|\, dX \tag{5.2}$$

is small, where the integral (like all others in this section) is with respect to the Lebesgue measure over the entries of X.

The first crucial observation is that the probability distributions $\mathcal{G}^{n \times n}$ and $\mathcal{S}_{m,n}$ are both invariant under left-multiplication or right-multiplication by a unitary matrix. It follows that $p_G(X)$ and $p_S(X)$ both depend only on the *list of singular values* of X. For we can always write $X = (x_{ij})$ as UDV where U, V are unitary and $D = (d_{ij})$ is a diagonal matrix of singular values; then $p_G(X) = p_G(D)$ and $p_S(X) = p_S(D)$. Let $\lambda_i := d_{ii}^2$ be the square of the i^{th} singular value of X. Then from the identity

$$\sum_{i,j \in [n]} |x_{ij}|^2 = \sum_{i \in [n]} \lambda_i\,, \tag{5.3}$$

we get the following formula for p_G:

$$p_G(X) = \prod_{i,j \in [n]} \frac{1}{\pi} e^{-|x_{ij}|^2} = \frac{1}{\pi^{n^2}} \prod_{i \in [n]} e^{-\lambda_i}\,. \tag{5.4}$$

Also, Réffy [55, p. 61] has shown that, provided $m \geq 2n$, we have

$$p_S(X) = c_{m,n} \prod_{i \in [n]} \left(1 - \frac{\lambda_i}{m}\right)^{m-2n} I_{\lambda_i \leq m} \tag{5.5}$$

for some constant $c_{m,n}$ where $I_{\lambda_i \leq m}$ equals 1 if $\lambda_i \leq m$ and 0 otherwise. Here and throughout, the λ_i's should be understood as functions $\lambda_i(X)$ of X.

Let $\lambda_{\max} := \max_i \lambda_i$ be the greatest squared spectral value of X. Then we can divide the space $\mathbb{C}^{n \times n}$ of matrices into two parts: the *head* R_{head}, consisting of matrices X such that $\lambda_{\max} \leq k$ and the *tail* R_{tail} consisting of matrices X such that $\lambda_{\max} > k$, for a value $k \leq m/(2n^2)$ that we will set later. At a high level, our strategy for upper-bounding $\Delta(p_G, p_S)$ will be to show that the head distributions are close and the tail distributions are small. More formally, define

$$g_{\text{head}} := \int_{X \in R_{\text{head}}} p_G(X)\, dX, \tag{5.6}$$

$$s_{\text{head}} := \int_{X \in R_{\text{head}}} p_S(X)\, dX, \tag{5.7}$$

$$\Delta_{\text{head}} := \int_{X \in R_{\text{head}}} |p_G(X) - p_S(X)|\, dX, \tag{5.8}$$

and define g_{tail} s_{tail} and Δ_{tail} similarly with integrals over R_{tail}. Note that $g_{\text{head}} + g_{\text{tail}} = s_{\text{head}} + s_{\text{tail}} = 1$ by normalization. Also, by the triangle inequality,

$$\Delta(p_G, p_S) = \Delta_{\text{head}} + \Delta_{\text{tail}} \leq \Delta_{\text{head}} + g_{\text{tail}} + s_{\text{tail}}. \tag{5.9}$$

So to upper-bound $\Delta(p_G, p_S)$ it suffices to upper-bound g_{tail} s_{tail} and Δ_{head} separately, which we now proceed to do in that order.

Lemma 5.3. $g_{\text{tail}} \leq n^2 e^{-k/n^2}$.

Proof. We have

$$\begin{aligned} g_{\text{tail}} &= \Pr_{X \sim \mathcal{G}^{n\times n}} [\lambda_{\max} > k] && (5.10)\\ &\leq \Pr_{X \sim \mathcal{G}^{n\times n}} \left[\textstyle\sum_{i,j\in[n]} \left|x_{ij}\right|^2 > k\right] && (5.11)\\ &\leq \sum_{i,j\in[n]} \Pr_{X \sim \mathcal{G}^{n\times n}} \left[\left|x_{ij}\right|^2 > \frac{k}{n^2}\right] && (5.12)\\ &= n^2 e^{-k/n^2}, && (5.13) \end{aligned}$$

where line (5.11) uses the identity (5.3) and line (5.12) uses the union bound. □

Lemma 5.4. $s_{\text{tail}} \leq n^2 e^{-k/(2n^2)}$.

Proof. Recall that $\mathcal{H}_{m,m}$ is the Haar measure over $m \times m$ unitary matrices. Then for a single entry (say u_{11}) of a matrix $U = (u_{ij})$ drawn from $\mathcal{H}_{m,m}$,

$$\Pr_{U \sim \mathcal{H}_{m.m}} \left[|u_{11}|^2 \geq r\right] = (1-r)^{m-1} \tag{5.14}$$

for all $r \in [0,1]$ which can be calculated from the density function given by Réffy [55] for the case $n = 1$.

So as in Lemma 5.3,

$$
\begin{aligned}
s_{\text{tail}} &= \Pr_{X\sim\mathcal{S}_{m,n}}\left[\lambda_{\max} > k\right] && (5.15)\\
&\le \Pr_{X\sim\mathcal{S}_{m,n}}\left[\textstyle\sum_{i,j\in[n]}\left|x_{ij}\right|^2 > k\right] && (5.16)\\
&\le \sum_{i,j\in[n]}\Pr_{X\sim\mathcal{S}_{m,n}}\left[\left|x_{ij}\right|^2 > \frac{k}{n^2}\right] && (5.17)\\
&= n^2\Pr_{U\sim\mathcal{H}_{m,m}}\left[\left|u_{11}\right|^2 > \frac{k}{mn^2}\right] && (5.18)\\
&= n^2\left(1-\frac{k}{mn^2}\right)^{m-1} && (5.19)\\
&< n^2 e^{-k(1-1/m)/n^2} && (5.20)\\
&< n^2 e^{-k/(2n^2)}\,. && (5.21)
\end{aligned}
$$

□

The rest of the proof is devoted to upper-bounding Δ_{head} the distance between the two head distributions. Recall that Réffy's formula for the density function $p_S(X)$ (equation (5.5)) involved a multiplicative constant $c_{m,n}$. Since it is difficult to compute the value of $c_{m,n}$ explicitly, we will instead define

$$\zeta := \frac{(1/\pi)^{n^2}}{c_{m,n}}, \tag{5.22}$$

and consider the scaled density function

$$\widetilde{p}_S(X) := \zeta\cdot p_S(X) = \frac{1}{\pi^{n^2}}\prod_{i\in[n]}\left(1-\frac{\lambda_i}{m}\right)^{m-2n} I_{\lambda_i\le m}\,. \tag{5.23}$$

We will first show that p_G and $\widetilde{p}_S$ are close on R_{head}. We will then deduce from that result, together with the fact that g_{tail} and s_{tail} are small, that p_G and p_S must be close on R_{head} which is what we wanted to show. Strangely, nowhere in this argument do we ever bound ζ directly. *After* proving Theorem 5.1, however, we will then need to go back and show that ζ is close to 1 on the way to proving Theorem 5.2.

Let

$$\widetilde{\Delta}_{\text{head}} := \int_{X\in R_{\text{head}}}\left|p_G(X)-\widetilde{p}_S(X)\right|dX. \tag{5.24}$$

Then our first claim is the following.

Lemma 5.5. $\widetilde{\Delta}_{\text{head}} \le \frac{4nk(n+k)}{m}$.

Proof. As a first observation, when we restrict to R_{head} we have $\lambda_i \le k \le \frac{m}{2n^2} < m$ for all $i\in[n]$ by assumption. So we can simplify the expression for $\widetilde{p}_S(X)$ by removing the indicator variable $I_{\lambda_i\le m}$:

$$\widetilde{p}_S(X) = \frac{1}{\pi^{n^2}}\prod_{i\in[n]}\left(1-\frac{\lambda_i}{m}\right)^{m-2n}. \tag{5.25}$$

Now let us rewrite equation (5.24) in the form

$$\widetilde{\Delta}_{\text{head}} = \int_{X \in R_{\text{head}}} p_G(X) \left| 1 - \frac{\widetilde{p}_S(X)}{p_G(X)} \right| dX\,. \tag{5.26}$$

Then plugging in the expressions for $\widetilde{p}_S(X)$ and $p_G(X)$ respectively gives the ratio

$$\frac{\widetilde{p}_S(X)}{p_G(X)} = \frac{\pi^{-n^2} \prod_{i \in [n]} (1-\lambda_i/m)^{m-2n}}{\pi^{-n^2} \prod_{i \in [n]} e^{-\lambda_i}} \tag{5.27}$$

$$= \exp\left(\sum_{i \in [n]} f(\lambda_i) \right), \tag{5.28}$$

where

$$f(\lambda_i) = \ln \frac{(1-\lambda_i/m)^{m-2n}}{e^{-\lambda_i}} \tag{5.29}$$

$$= \lambda_i - (m-2n)(-\ln(1-\lambda_i/m))\,. \tag{5.30}$$

Since $0 \le \lambda_i < m$ we may use the Taylor expansion

$$-\ln(1-\lambda_i/m) = \frac{\lambda_i}{m} + \frac{1}{2}\frac{\lambda_i^2}{m^2} + \frac{1}{3}\frac{\lambda_i^3}{m^3} + \cdots \tag{5.31}$$

So we can upper-bound $f(\lambda_i)$ by

$$f(\lambda_i) \le \lambda_i - (m-2n)\frac{\lambda_i}{m} \tag{5.32}$$

$$= \frac{2n\lambda_i}{m} \tag{5.33}$$

$$\le \frac{2nk}{m}, \tag{5.34}$$

and can *lower*-bound $f(\lambda_i)$ by

$$f(\lambda_i) \ge \lambda_i - (m-2n)\left(\frac{\lambda_i}{m} + \frac{1}{2}\frac{\lambda_i^2}{m^2} + \frac{1}{3}\frac{\lambda_i^3}{m^3} + \cdots \right) \tag{5.35}$$

$$> \lambda_i - (m-2n)\left(\frac{\lambda_i}{m} + \frac{\lambda_i^2}{m^2} + \frac{\lambda_i^3}{m^3} + \cdots \right) \tag{5.36}$$

$$= \lambda_i - \frac{(m-2n)\lambda_i}{m(1-\lambda_i/m)} \tag{5.37}$$

$$> \lambda_i - \frac{\lambda_i}{1-\lambda_i/m} \tag{5.38}$$

$$= -\frac{\lambda_i^2}{m-\lambda_i} \tag{5.39}$$

$$\ge -\frac{2k^2}{m}. \tag{5.40}$$

Here line (5.40) used the fact that $\lambda_i \le k \le \frac{m}{2n^2} < \frac{m}{2}$ since $X \in R_{\text{head}}$. It follows that

$$-\frac{2nk^2}{m} \le \sum_{i\in[n]} f(\lambda_i) \le \frac{2n^2k}{m}. \tag{5.41}$$

So

$$\begin{aligned}
\left|1-\frac{\widetilde{p}_S(X)}{p_G(X)}\right| &= \left|1-\exp\left(\sum_{i\in[n]} f(\lambda_i)\right)\right| && (5.42)\\
&\le \max\left\{1-\exp\left(-\frac{2nk^2}{m}\right), \exp\left(\frac{2n^2k}{m}\right)-1\right\} && (5.43)\\
&\le \max\left\{\frac{2nk^2}{m}, \frac{4n^2k}{m}\right\} && (5.44)\\
&\le \frac{4nk(n+k)}{m} && (5.45)
\end{aligned}$$

where line (5.44) used the fact that $e^{\delta}-1 < 2\delta$ for all $\delta \le 1$. To conclude,

$$\begin{aligned}
\widetilde{\Delta}_{\text{head}} &\le \int_{X\in R_{\text{head}}} p_G(X)\left[\frac{4nk(n+k)}{m}\right] dX && (5.46)\\
&\le \frac{4nk(n+k)}{m}. && (5.47)
\end{aligned}$$

□

Combining Lemmas 5.3, 5.4, and 5.5, and making repeated use of the triangle inequality, we find that

$$\begin{aligned}
\Delta_{\text{head}} &= \int_{X\in R_{\text{head}}} |p_G(X)-p_S(X)|\, dX && (5.48)\\
&\le \widetilde{\Delta}_{\text{head}} + \int_{X\in R_{\text{head}}} |\widetilde{p}_S(X)-p_S(X)|\, dX && (5.49)\\
&= \widetilde{\Delta}_{\text{head}} + |\zeta s_{\text{head}} - s_{\text{head}}| && (5.50)\\
&\le \widetilde{\Delta}_{\text{head}} + |\zeta s_{\text{head}} - g_{\text{head}}| + |g_{\text{head}}-1| + |1-s_{\text{head}}| && (5.51)\\
&\le 2\widetilde{\Delta}_{\text{head}} + g_{\text{tail}} + s_{\text{tail}} && (5.52)\\
&\le \frac{8nk(n+k)}{m} + n^2e^{-k/n^2} + n^2e^{-k/(2n^2)}. && (5.53)
\end{aligned}$$

Therefore

$$\begin{aligned}
\Delta(p_G, p_S) &\le \Delta_{\text{head}} + g_{\text{tail}} + s_{\text{tail}} && (5.54)\\
&\le \frac{8nk(n+k)}{m} + 2n^2e^{-k/n^2} + 2n^2e^{-k/(2n^2)}. && (5.55)
\end{aligned}$$

Recalling that $m \geq (n^5/\delta)\log^2(n/\delta)$ let us now make the choice $k := 6n^2 \log(n/\delta)$. Then the constraint $k \leq m/(2n^2)$ is satisfied, and furthermore $\Delta(p_G, p_S) = O(\delta)$. This completes the proof of Theorem 5.1.

The above derivation "implicitly" showed that ζ is close to 1. As a first step toward proving Theorem 5.2, let us now make the bound on ζ explicit.

Lemma 5.6. $|\zeta - 1| = O(\delta)$.

Proof. We have

$$\begin{aligned} |\zeta s_{\text{head}} - s_{\text{head}}| &\leq |\zeta s_{\text{head}} - g_{\text{head}}| + |g_{\text{head}} - 1| + |1 - s_{\text{head}}| && (5.56)\\ &= \widetilde{\Delta}_{\text{head}} + g_{\text{tail}} + s_{\text{tail}} && (5.57)\\ &\leq \frac{4nk(n+k)}{m} + n^2 e^{-k/n^2} + n^2 e^{-k/(2n^2)} && (5.58)\end{aligned}$$

and

$$s_{\text{head}} = 1 - s_{\text{tail}} \geq 1 - n^2 e^{-k/(2n^2)}. \tag{5.59}$$

As before, recall that $m \geq \frac{n^5}{\delta}\log^2\frac{n}{\delta}$ and set $k := 6n^2 \log\frac{n}{\delta}$. Then

$$\begin{aligned} |\zeta - 1| &= \frac{|\zeta s_{\text{head}} - s_{\text{head}}|}{s_{\text{head}}} && (5.60)\\ &\leq \frac{4nk(n+k)/m + n^2 e^{-k/n^2} + n^2 e^{-k/(2n^2)}}{1 - n^2 e^{-k/(2n^2)}} && (5.61)\\ &= O(\delta). && (5.62)\end{aligned}$$

□

We can now prove Theorem 5.2, that $p_S(X) \leq (1 + O(\delta))\, p_G(X)$ for all $X \in \mathbb{C}^{n\times n}$.

Proof of Theorem 5.2. Our goal is to upper-bound

$$C := \max_{X \in \mathbb{C}^{n\times n}} \frac{p_S(X)}{p_G(X)}. \tag{5.63}$$

Using the notation of Lemma 5.5, we can rewrite C as

$$\frac{1}{\zeta} \max_{X \in \mathbb{C}^{n\times n}} \frac{\widetilde{p}_S(X)}{p_G(X)} = \frac{1}{\zeta} \max_{\lambda_1,\ldots,\lambda_n \geq 0} \exp\left(\sum_{i\in[n]} f(\lambda_i)\right), \tag{5.64}$$

where

$$f(\lambda_i) := \lambda_i + (m - 2n)\ln(1 - \lambda_i/m). \tag{5.65}$$

By elementary calculus, the function $f(\lambda)$ achieves its maximum at $\lambda = 2n$; note that this is a valid maximum since $m \geq 2n$. Setting $\lambda_i = 2n$ for all i then yields

$$
\begin{aligned}
C &= \frac{1}{\zeta} \exp\left(2n^2 + n(m-2n)\ln\left(1 - \frac{2n}{m}\right)\right) && (5.66)\\
&= \frac{1}{\zeta} e^{2n^2}\left(1 - \frac{2n}{m}\right)^{n(m-2n)} && (5.67)\\
&< \frac{1}{\zeta} e^{2n^2} e^{-2n^2(m-2n)/m} && (5.68)\\
&= \frac{1}{\zeta} e^{4n^3/m} && (5.69)\\
&\leq \frac{1}{1-O(\delta)}(1+O(\delta)) && (5.70)\\
&= 1 + O(\delta)\,. && (5.71)
\end{aligned}
$$

Here line (5.70) used Lemma 5.6, together with the fact that $m \gg \frac{4n^3}{\delta}$. □

5.2 Hardness of approximate BOSONSAMPLING

Having proved Theorem 5.2, we are finally ready to prove the main result of the paper: that $|\mathrm{GPE}|_{\pm}^{2} \in \mathsf{FBPP}^{\mathsf{NP}^{\mathcal{O}}}$ where $\mathcal{O}$ is any approximate BOSONSAMPLING oracle. In other words, if there is a fast classical algorithm for approximate BOSONSAMPLING, then there is also a $\mathsf{BPP}^{\mathsf{NP}}$ algorithm to estimate $|\mathrm{Per}(X)|^2$ with high probability for a Gaussian random matrix $X \sim \mathcal{G}^{n\times n}$.

We first need a technical lemma, which formalizes the well-known concept of rejection sampling.

Lemma 5.7 (Rejection Sampling)**.** *Let $\mathcal{D} = \{p_x\}$ and $\mathcal{E} = \{q_x\}$ be any two distributions over a finite set S. Suppose that there exists a polynomial-time algorithm to compute $\zeta q_x/p_x$ given $x \in S$ where ζ is some constant independent of x such that $|\zeta - 1| \leq \delta$. Suppose also that $q_x/p_x \leq 1+\delta$ for all $x \in S$. Then there exists a* BPP *algorithm $\mathcal{R}$ that takes a sample $x \sim \mathcal{D}$ as input, and either accepts or rejects. $\mathcal{R}$ has the following properties:*

(i) Conditioned on $\mathcal{R}$ accepting, x is distributed according to $\mathcal{E}$.

(ii) The probability that $\mathcal{R}$ rejects (over both its internal randomness and $x \sim \mathcal{D}$) is $O(\delta)$.

Proof. $\mathcal{R}$ works as follows: first compute $\zeta q_x/p_x$; then accept with probability

$$\frac{\zeta q_x/p_x}{(1+\delta)^2} \leq 1\,.$$

Property (i) is immediate. For property (ii),

$$\Pr[\mathcal{R} \text{ rejects}] = \sum_{x \in S} p_x \left(1 - \frac{\zeta q_x / p_x}{(1+\delta)^2}\right) \tag{5.72}$$

$$= \sum_{x \in S} \left(p_x - \frac{\zeta q_x}{(1+\delta)^2}\right) \tag{5.73}$$

$$= 1 - \frac{\zeta}{(1+\delta)^2} \tag{5.74}$$

$$= O(\delta)\,. \tag{5.75}$$

□

By combining Lemma 5.7 with Theorem 5.2, we now show how it is possible to "hide" a matrix $X \sim \mathcal{G}^{n\times n}$ of i. i. d. Gaussians as a random $n \times n$ submatrix of a Haar-random $m \times n$ column-orthonormal matrix A, provided $m = \Omega\left(n^5 \log^2 n\right)$. Our hiding procedure does not involve any distortion of X. We believe that the hiding procedure could be implemented in BPP; however, we will show only that it can be implemented in $\mathsf{BPP}^{\mathsf{NP}}$ since that is easier and suffices for our application.

Lemma 5.8 (Hiding Lemma)**.** *Let*

$$m \geq \frac{n^5}{\delta} \log^2 \frac{n}{\delta}$$

for some $\delta > 0$. Then there exists a $\mathsf{BPP}^{\mathsf{NP}}$ algorithm $\mathcal{A}$ that takes as input a matrix $X \sim \mathcal{G}^{n\times n}$ that "succeeds" with probability $1 - O(\delta)$ over X and that, conditioned on succeeding, samples a matrix $A \in \mathcal{U}_{m,n}$ from a probability distribution $\mathcal{D}_X$, such that the following properties hold:

(i) $X/\sqrt{m}$ occurs as a uniformly-random $n \times n$ submatrix of $A \sim \mathcal{D}_X$ for every X such that

$$\Pr[\mathcal{A}(X) \text{ succeeds}] > 0\,.$$

(ii) The distribution over $A \in \mathbb{C}^{m\times n}$ induced by drawing $X \sim \mathcal{G}^{n\times n}$ running $\mathcal{A}(X)$ and conditioning on $\mathcal{A}(X)$ succeeding is simply $\mathcal{H}_{m,n}$ (the Haar measure over $m \times n$ column-orthonormal matrices).

Proof. Given a sample $X \sim \mathcal{G}^{n\times n}$ the first step is to "convert" X into a sample from the truncated Haar measure $\mathcal{S}_{m,n}$. To do so, we use the rejection sampling procedure from Lemma 5.7. By Theorem 5.2, we have $p_S(X)/p_G(X) \leq 1 + O(\delta)$ for all $X \in \mathbb{C}^{n\times n}$ where p_S and p_G are the probability density functions of $\mathcal{S}_{m,n}$ and $\mathcal{G}^{n\times n}$ respectively. Also, letting $\zeta := (1/\pi)^{n^2}/c_{m,n}$ be the constant from Section 5.1, we have

$$\frac{\zeta \cdot p_S(X)}{p_G(X)} = \frac{\widetilde{p}_S(X)}{p_G(X)} = \frac{\prod_{i\in[n]} (1 - \lambda_i/m)^{m-2n}}{\prod_{i\in[n]} e^{-\lambda_i}}\,, \tag{5.76}$$

which is clearly computable in polynomial time (to any desired precision) given X. Finally, we saw from Lemma 5.6 that $|\zeta - 1| = O(\delta)$. So by Lemma 5.7, the rejection sampling procedure $\mathcal{R}$ has the following properties:

(1) $\mathcal{R}$ can be implemented in BPP.

(2) $\mathcal{R}$ rejects with probability $O(\delta)$.

(3) Conditioned on $\mathcal{R}$ accepting, we have $X \sim \mathcal{S}_{m,n}$.

Now suppose $\mathcal{R}$ accepts, and let $X' := X/\sqrt{m}$. Then our problem reduces to embedding X' as a random submatrix of a sample A from $\mathcal{H}_{m,n}$. We do this as follows. Given a matrix $A \in \mathcal{U}_{m,n}$ let $E_X(A)$ be the event that X' occurs as an $n \times n$ submatrix of A. Then let $\mathcal{D}_X$ be the distribution over $A \in \mathcal{U}_{m,n}$ obtained by first sampling A from $\mathcal{H}_{m,n}$ and then conditioning on $E_X(A)$ holding. Note that $\mathcal{D}_X$ is well-defined, since for every X in the support of $\mathcal{S}_{m,n}$ there is *some* $A \in \mathcal{U}_{m,n}$ satisfying $E_X(A)$. We now check that $\mathcal{D}_X$ satisfies properties (i) and (ii). For (i), every element in the support of $\mathcal{D}_X$ contains X' as a submatrix by definition, and by symmetry, this X' occurs at a uniformly-random location. For (ii), notice that we could equally well have sampled $A \sim \mathcal{D}_X$ by first sampling $X \sim \mathcal{S}_{m,n}$, then placing X' at a uniformly-random location within A and finally "filling in" the remaining $(m-n) \times n$ block of A by drawing it from $\mathcal{H}_{m,n}$ conditioned on X'. From this perspective, however, it is clear that A is Haar-random, since $\mathcal{S}_{m,n}$ was just a truncation of $\mathcal{H}_{m,n}$ to begin with. What remains to show is that, given X as input, we can sample from $\mathcal{D}_X$ in $\mathsf{BPP}^{\mathsf{NP}}$. As a first step, we can certainly sample from $\mathcal{H}_{m,n}$ in BPP. To do so, for example, we can first generate a matrix $A \sim \mathcal{G}^{m \times n}$ of independent Gaussians, and then apply the Gram-Schmidt orthogonalization procedure to A. Now, given a BPP algorithm that samples $A \sim \mathcal{H}_{m,n}$ the remaining task is to condition on the event $E_X(A)$. Given X and A it is easy to check whether $E_X(A)$ holds. But this means that we can sample from the conditional distribution $\mathcal{D}_X$ in the complexity class PostBPP and hence also in $\mathsf{BPP}^{\mathsf{NP}}$ by equation (2.1). So, combining a BPP algorithm with a $\mathsf{BPP}^{\mathsf{NP}}$ algorithm, we get an overall $\mathsf{BPP}^{\mathsf{NP}}$ algorithm. □

The final step is to prove that, *if* we had an oracle $\mathcal{O}$ for approximate BOSONSAMPLING, then by using $\mathcal{O}$ in conjunction with the hiding procedure from Lemma 5.8, we could estimate $|\mathrm{Per}(X)|^2$ in $\mathsf{BPP}^{\mathsf{NP}}$ where $X \sim \mathcal{G}^{n \times n}$ is a Gaussian input matrix.

To prove this theorem, we need to recall some definitions from previous sections. The set of tuples $S = (s_1, \ldots, s_m)$ satisfying $s_1, \ldots, s_m \geq 0$ and $s_1 + \cdots + s_m = n$ is denoted $\Phi_{m,n}$. Given a matrix $A \in \mathcal{U}_{m,n}$ we denote by $\mathcal{D}_A$ the distribution over $\Phi_{m,n}$ where each S occurs with probability

$$\Pr_{\mathcal{D}_A}[S] = \frac{|\mathrm{Per}(A_S)|^2}{s_1! \cdots s_m!}. \tag{5.77}$$

Also, recall that in the $|\mathrm{GPE}|^2_{\pm}$ problem, we are given an input of the form $\langle X, 0^{1/\varepsilon}, 0^{1/\delta} \rangle$ where X is an $n \times n$ matrix drawn from the Gaussian distribution $\mathcal{G}^{n \times n}$. The goal is to approximate $|\mathrm{Per}(X)|^2$ to within an additive error $\varepsilon \cdot n!$ with probability at least $1 - \delta$ over X.

We now prove Theorem 1.3, our main result. Let us restate the theorem for convenience:

Let $\mathcal{O}$ be any approximate BOSONSAMPLING *oracle. Then* $|\mathrm{GPE}|^2_{\pm} \in \mathsf{FBPP}^{\mathsf{NP}^{\mathcal{O}}}$.

Proof of Theorem 1.3. Let $X \sim \mathcal{G}^{n \times n}$ be an input matrix, and let $\varepsilon, \delta > 0$ be error parameters. Then we need to show how to approximate $|\mathrm{Per}(X)|^2$ to within an additive error $\varepsilon \cdot n!$, with probability at least

$1-\delta$ over X in the complexity class $\mathsf{FBPP}^{\mathsf{NP}^{\mathcal{O}}}$. The running time should be polynomial in n, $1/\varepsilon$ and $1/\delta$. Let

$$m := \frac{K}{\delta} n^5 \log^2 \frac{n}{\delta}$$

where K is a suitably large constant. Also, let $X' := X/\sqrt{m}$ be a scaled version of X. Then we can state our problem equivalently as follows: approximate

$$\left|\operatorname{Per}\left(X'\right)\right|^2 = \frac{\left|\operatorname{Per}(X)\right|^2}{m^n} \tag{5.78}$$

to within an additive error $\varepsilon \cdot n!/m^n$. As a first step, Lemma 5.8 says that in $\mathsf{BPP}^{\mathsf{NP}}$ and with high probability over X' we can generate a matrix $A \in \mathcal{U}^{m\times n}$ that is exactly Haar-random, and that contains X' as a random $n \times n$ submatrix. So certainly we can generate such an A in $\mathsf{FBPP}^{\mathsf{NP}^{\mathcal{O}}}$ (indeed, without using the oracle $\mathcal{O}$). Provided we chose K sufficiently large, this procedure will succeed with probability at least (say) $1-\delta/4$. Set $\beta := \varepsilon\delta/24$. Suppose we feed $\left\langle A, 0^{1/\beta}, r\right\rangle$ to the approximate BOSONSAMPLING oracle $\mathcal{O}$ where $r \in \{0,1\}^{\operatorname{poly}(m)}$ is a random string. Then by definition, as r is varied, $\mathcal{O}$ returns a sample from a probability distribution $\mathcal{D}'_A$ such that $\|\mathcal{D}_A - \mathcal{D}'_A\| \leq \beta$. Let $p_S := \Pr_{\mathcal{D}_A}[S]$ and $q_S := \Pr_{\mathcal{D}'_A}[S]$ for all $S \in \Phi_{m,n}$. Also, let $W \subset [m]$ be the subset of n rows of A in which X' occurs as a submatrix. Then we will be particularly interested in the basis state $S^* = (s_1, \ldots, s_m)$, which is defined by $s_i = 1$ if $i \in W$ and $s_i = 0$ otherwise. Notice that

$$p_{S^*} = \frac{\left|\operatorname{Per}(A_{S^*})\right|^2}{s_1! \cdots s_m!} = \left|\operatorname{Per}\left(X'\right)\right|^2, \tag{5.79}$$

and that

$$q_{S^*} = \Pr_{\mathcal{D}'_A}[S^*] = \Pr_{r \in \{0,1\}^{\operatorname{poly}(m)}}\left[\mathcal{O}\left(A, 0^{1/\beta}, r\right) = S^*\right]. \tag{5.80}$$

In other words: p_{S^*} encodes the squared permanent that we are trying to approximate, while q_{S^*} can be approximated in $\mathsf{FBPP}^{\mathsf{NP}^{\mathcal{O}}}$ using Stockmeyer's approximate counting method (Theorem 4.1). Therefore, to show that with high probability we can approximate p_{S^*} in $\mathsf{FBPP}^{\mathsf{NP}^{\mathcal{O}}}$ it suffices to show that p_{S^*} and q_{S^*} are close with high probability over X and A. Call a basis state $S \in \Phi_{m,n}$ *collision-free* if each s_i is either 0 or 1. Let $G_{m,n}$ be the set of collision-free S's, and notice that $S^* \in G_{m,n}$. From now on, we will find it convenient to restrict attention to $G_{m,n}$. Let $\Delta_S := |p_S - q_S|$ so that

$$\left\|\mathcal{D}_A - \mathcal{D}'_A\right\| = \frac{1}{2}\sum_{S \in \Phi_{m,n}} \Delta_S. \tag{5.81}$$

Then

$$\mathop{\mathrm{E}}_{S\in G_{m,n}}[\Delta_S] \leq \frac{\sum_{S\in\Phi_{m,n}}\Delta_S}{|G_{m,n}|} \tag{5.82}$$

$$= \frac{2\left\|\mathcal{D}_A - \mathcal{D}'_A\right\|}{|G_{m,n}|} \tag{5.83}$$

$$\leq \frac{2\beta}{\binom{m}{n}} \tag{5.84}$$

$$< 3\beta\cdot\frac{n!}{m^n}, \tag{5.85}$$

where line (5.85) used the fact that $m = \omega\left(n^2\right)$. So by Markov's inequality, for all $k > 1$,

$$\Pr_{S\in G_{m,n}}\left[\Delta_S > 3\beta k\cdot\frac{n!}{m^n}\right] < \frac{1}{k}. \tag{5.86}$$

In particular, if we set $k := 4/\delta$ and notice that $3\beta k = 12\beta/\delta = \varepsilon/2$,

$$\Pr_{S\in G_{m,n}}\left[\Delta_S > \frac{\varepsilon}{2}\cdot\frac{n!}{m^n}\right] < \frac{\delta}{4}. \tag{5.87}$$

Of course, our goal is to upper-bound Δ_{S^*} not Δ_S for a randomly-chosen $S \in G_{m,n}$. However, a crucial observation is that, from the perspective of $\mathcal{O}$—which sees only A and not S^* or X'—the distribution over possible values of S^* is simply the uniform one. To see this, notice that instead of sampling X *and then* A (as in Lemma 5.8), we could have equally well generated the pair $\langle X, A\rangle$ by first sampling A from the Haar measure $\mathcal{H}_{m,n}$ and then setting $X := \sqrt{m}A_{S^*}$ for S^* chosen uniformly from $G_{m,n}$. It follows that seeing A gives $\mathcal{O}$ no information whatsoever about the identity of S^*. So even if $\mathcal{O}$ is trying adversarially to maximize Δ_{S^*} we still have

$$\Pr_{X,A}\left[\Delta_{S^*} > \frac{\varepsilon}{2}\cdot\frac{n!}{m^n}\right] < \frac{\delta}{4}. \tag{5.88}$$

Now suppose we use Stockmeyer's algorithm to approximate q_{S^*} in $\mathsf{FBPP}^{\mathsf{NP}^{\mathcal{O}}}$. Then by Theorem 4.1, for all $\alpha > 0$ we can obtain an estimate $\widetilde{q}_{S^*}$ such that

$$\Pr\left[\left|\widetilde{q}_{S^*} - q_{S^*}\right| > \alpha\cdot q_{S^*}\right] < \frac{1}{2^m}, \tag{5.89}$$

in time polynomial in m and $1/\alpha$. Note that

$$\mathop{\mathrm{E}}_{S\in G_{m,n}}[q_S] \leq \frac{1}{|G_{m,n}|} = \frac{1}{\binom{m}{n}} < 2\frac{n!}{m^n}, \tag{5.90}$$

so

$$\Pr_{S\in G_{m,n}}\left[q_S > 2k\cdot\frac{n!}{m^n}\right] < \frac{1}{k} \tag{5.91}$$

for all $k > 1$ by Markov's inequality, so

$$\Pr_{X,A}\left[q_{S^*} > 2k \cdot \frac{n!}{m^n}\right] < \frac{1}{k} \tag{5.92}$$

by the same symmetry principle used previously for Δ_{S^*}. Let us now make the choice $\alpha := \varepsilon\delta/16$ and $k := 4/\delta$. Then putting everything together and applying the union bound,

$$\begin{aligned}\Pr\left[|\widetilde{q}_{S^*} - p_{S^*}| > \varepsilon \cdot \frac{n!}{m^n}\right] &\leq \Pr\left[|\widetilde{q}_{S^*} - q_{S^*}| > \frac{\varepsilon}{2} \cdot \frac{n!}{m^n}\right] + \Pr\left[|q_{S^*} - p_{S^*}| > \frac{\varepsilon}{2} \cdot \frac{n!}{m^n}\right] &(5.93)\\ &\leq \Pr\left[q_{S^*} > 2k \cdot \frac{n!}{m^n}\right] + \Pr\left[|\widetilde{q}_{S^*} - q_{S^*}| > \alpha \cdot q_{S^*}\right] + \Pr\left[\Delta_{S^*} > \frac{\varepsilon}{2} \cdot \frac{n!}{m^n}\right] &(5.94)\\ &< \frac{1}{k} + \frac{1}{2^m} + \frac{\delta}{4} &(5.95)\\ &= \frac{\delta}{2} + \frac{1}{2^m}, &(5.96)\end{aligned}$$

where the probabilities are over X and A as well as the internal randomness used by the approximate counting procedure. So, including the probability that the algorithm $\mathcal{A}$ from Lemma 5.8 fails, the total probability that our $\mathsf{FBPP}^{\mathsf{NP}^{\mathcal{O}}}$ machine fails to output a good enough approximation to $p_{S^*} = |\mathrm{Per}(X')|^2$ is at most

$$\frac{\delta}{4} + \left(\frac{\delta}{2} + \frac{1}{2^m}\right) < \delta, \tag{5.97}$$

as desired. This completes the proof. □

5.3 Implications

In this section, we harvest some implications of Theorem 1.3 for quantum complexity theory. First, if a fast classical algorithm for BOSONSAMPLING exists, then it would have a surprising consequence for the classical complexity of the $|\mathrm{GPE}|_{\pm}^2$ problem.

Corollary 5.9. *Suppose* BOSONSAMPLING $\in \mathsf{SampP}$. *Then* $|\mathrm{GPE}|_{\pm}^2 \in \mathsf{FBPP}^{\mathsf{NP}}$. *Indeed, even under the hypothesis* BOSONSAMPLING $\in \mathsf{SampP}^{\mathsf{PH}}$ *we have* $|\mathrm{GPE}|_{\pm}^2 \in \mathsf{FBPP}^{\mathsf{PH}}$.

However, we would also like evidence that a boson computer can solve *search* problems that are intractable classically. Fortunately, by using Theorem 2.5—the "Sampling/Searching Equivalence Theorem"—we can obtain such evidence in a completely automatic way. In particular, combining Corollary 5.9 with Theorem 2.5 yields the following conclusion.

Corollary 5.10. *There exists a search problem* $R \in \mathsf{BosonFP}$ *such that* $|\mathrm{GPE}|_{\pm}^2 \in \mathsf{FBPP}^{\mathsf{NP}^{\mathcal{O}}}$ *for all computable oracles* $\mathcal{O}$ *that solve R. So in particular, if* $\mathsf{BosonFP} \subseteq \mathsf{FBPP}$ *(that is, all search problems solvable by a boson computer are also solvable classically), then* $|\mathrm{GPE}|_{\pm}^2 \in \mathsf{FBPP}^{\mathsf{NP}}$.

Recall from Theorem 3.12 that $\mathsf{BosonFP} \subseteq \mathsf{FBQP}$: that is, linear-optics computers can be simulated efficiently by "ordinary" quantum computers. Thus, Corollary 5.10 implies in particular that, if

$\mathsf{FBPP} = \mathsf{FBQP}$, then $|\mathrm{GPE}|_{\pm}^{2} \in \mathsf{FBPP}^{\mathsf{NP}}$. Or in other words: if $|\mathrm{GPE}|_{\pm}^{2}$ is #P-hard, then FBPP cannot equal FBQP unless $\mathsf{P}^{\#\mathsf{P}} = \mathsf{BPP}^{\mathsf{NP}}$ and the polynomial hierarchy collapses. This would arguably be our strongest evidence to date against the Extended Church-Turing Thesis.

In Sections 7, 8, and 9, we initiate a program aimed at proving $|\mathrm{GPE}|_{\pm}^{2}$ is #P-hard.

6 Experimental prospects

Our main goal in this paper was to define and study a *theoretical* model of quantum computing with noninteracting bosons. There are several ways to motivate this model other than practical realizability: for example, it abstracts a basic class of physical systems, it leads to interesting new complexity classes between BPP and BQP, and it helped us provide evidence that quantum mechanics *in general* is hard to simulate classically. (In other words, even if we only cared about "standard" quantum computing, we would not know how to prove results like Theorem 1.3 without using linear optics as a proof tool.)

Clearly, though, a major motivation for our results is that they raise the possibility of actually *building* a scalable linear-optics computer, and using it to solve the BOSONSAMPLING problem. By doing this, one could hope to give evidence that nontrivial quantum computation is possible, *without* having to solve all the technological problems of building a universal quantum computer. In other words, one could see our results as suggesting a new path to testing the Extended Church-Turing Thesis, which might be more experimentally accessible than alternative paths.

A full discussion of implementation issues is outside the scope of this paper. Here, though, we offer some preliminary observations that emerged from our discussions with quantum optics experts. These observations concern both the challenges of performing a BOSONSAMPLING experiment, and the implications of such an experiment for complexity theory.

6.1 The generalized Hong-Ou-Mandel dip

From a physics standpoint, the experiment that we are asking for is essentially a generalization of the *Hong-Ou-Mandel dip* [36] to three or more photons. The Hong-Ou-Mandel dip (see Figure 3) is a well-known effect in quantum optics whereby two identical photons, which were initially in different modes, become *correlated* after passing through a beamsplitter that applies the Hadamard transformation. More formally, the basis state $|1,1\rangle$ evolves to

$$\frac{|2,0\rangle - |0,2\rangle}{\sqrt{2}}, \tag{6.1}$$

so that a subsequent measurement reveals either both photons in the first mode or else both photons in the second mode. This behavior is exactly what one would predict from the model in Section 3, in which n-photon transition amplitudes are given by the permanents of $n \times n$ matrices. More concretely, the amplitude of the basis state $|1,1\rangle$ "dips" to 0 because

$$\operatorname{Per}\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} = 0, \tag{6.2}$$

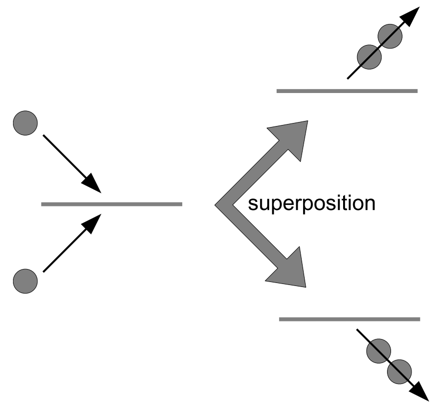

Figure 3: The Hong-Ou-Mandel dip.

and hence there is destructive interference between the two paths mapping $|1,1\rangle$ to itself.

Our challenge to experimentalists is to *confirm directly that the quantum-mechanical formula for n-boson transition amplitudes in terms of $n \times n$ permanents given in Section 3.3, namely*

$$\langle S|\varphi(U)|T\rangle = \frac{\operatorname{Per}(U_{S,T})}{\sqrt{s_1!\cdots s_m!t_1!\cdots t_m!}}, \tag{6.3}$$

continues to hold for large values of n. In other words, demonstrate a Hong-Ou-Mandel interference pattern involving as many identical bosons as possible (though even 3 or 4 bosons would be of interest here).

The point of such an experiment would be to produce evidence that a linear-optical network can indeed solve the BOSONSAMPLING problem in a scalable way—and that therefore, no polynomial-time classical algorithm can sample the observed distribution over photon numbers (modulo our conjectures about the computational complexity of the permanent).

Admittedly, since complexity theory deals only with *asymptotic* statements, no finite experiment can answer the relevant questions definitively. That is, even if formula (6.3) were confirmed for 30 identical bosons, a true-believer in the Extended Church-Turing Thesis could always maintain that the formula would break down for 31 bosons, and so on. Thus, the goal here is simply to collect enough evidence, for large enough n that the ECT becomes less tenable as a scientific hypothesis.

Of course, one should not choose *n so* large that a classical computer cannot even efficiently *verify* that the formula (6.3) holds! It is important to understand this difference between the BOSONSAMPLING problem on the one hand, and NP problems such as FACTORING on the other. Unlike with FACTORING, we do not know of any *witness* for BOSONSAMPLING that a classical computer can efficiently verify, much less a witness that a boson computer can produce.[23] This means that, when n is very large (say, more than 100), even if a linear-optics device is correctly solving BOSONSAMPLING, there might be no

[23]Indeed, given a matrix $X \in \mathbb{C}^{n\times n}$ there *cannot* in general be an NP witness proving the value of $\operatorname{Per}(X)$ unless $\mathsf{P}^{\#\mathsf{P}} = \mathsf{P}^{\mathsf{NP}}$ and the polynomial hierarchy collapses. Nor, under our conjectures, can there even be such a witness for *most* Gaussian matrices X. On the other hand, these arguments do not rule out an *interactive protocol* with a BPP verifier and a BOSONSAMPLING prover. Whether any such protocol exists for verifying statements not in BPP is an extremely interesting open problem.

feasible way to prove this without presupposing the truth of the physical laws being tested! Thus, for experimental purposes, the most useful values of n are presumably those for which a classical computer has some difficulty computing an $n \times n$ permanent, but can nevertheless do so in order to confirm the results. We estimate this range as $10 \leq n \leq 50$.

But how exactly should one verify formula (6.3)? One approach would be to perform full quantum state tomography on the output state of a linear-optical network, or at least to characterize the distribution over photon numbers. However, this approach would require a number of experimental runs that grows exponentially with n and is probably not needed.

Instead, given a system with n identical photons and $m \geq n$ modes, one could do something like the following:

(1) Prepare the "standard initial state" $|1_n\rangle$ in which modes $1, \ldots, n$ are occupied with a single photon each and modes $n+1, \ldots, m$ are unoccupied.

(2) By passing the photons through a suitable network of beamsplitters and phaseshifters, apply an $m \times m$ mode-mixing unitary transformation U. This maps the state $|1_n\rangle$ to $\varphi(U)|1_n\rangle$ where $\varphi(U)$ is the induced action of U on n-photon states.

(3) For each mode $i \in [m]$ measure the number of photons s_i in the i^{th} mode. This collapses the state $\varphi(U)|1_n\rangle$ to some $|S\rangle = |s_1, \ldots, s_m\rangle$ where $s_1, \ldots, s_m$ are nonnegative integers summing to n.

(4) Using a classical computer, calculate $|\mathrm{Per}(U_{1_n,S})|^2 / s_1! \cdots s_m!$ the theoretical probability of observing the basis state $|S\rangle$.

(5) Repeat steps (1) to (4), for a number of repetitions that scales polynomially with n and m.

(6) Plot the empirical frequency of $|\mathrm{Per}(U_{1_n,S})|^2 / s_1! \cdots s_m! > x$ for all $x \in [0,1]$ with particular focus on the range $x \approx 1/\binom{m+n-1}{n}$. Check for agreement with the frequencies predicted by quantum mechanics (which can again be calculated using a classical computer, either deterministically or via Monte Carlo simulation).

The procedure above does not prove that the final state is $\varphi(U)|1_n\rangle$. However, it at least checks that the basis states $|S\rangle$ with large values of $|\mathrm{Per}(U_{1_n,S})|^2$ are more likely to be observed than those with small values of $|\mathrm{Per}(U_{1_n,S})|^2$ in the manner predicted by formula (6.3).

6.2 Physical resource requirements

We now make some miscellaneous remarks about the physical resource requirements for our experiment.

Platform The obvious platform for our proposed experiment is linear optics. However, one could also do the experiment (for example) in a solid-state system, using bosonic excitations. What is essential is just that the excitations behave as *indistinguishable* bosons when they are far apart. In other words, the amplitude for n excitations to transition from one basis state to another must be given by the permanent of an $n \times n$ matrix of transition amplitudes for the individual excitations. On the other hand, the more general formula (6.3) need not hold; that is, it is acceptable for the bosonic approximation to break down for processes that involve multiple excitations in the same mode. (The reason is that the events that most interest us do not involve collisions anyway.)

Initial state In our experiment, the initial state would ideally consist of at most *one photon per mode*: that is, single-photon Fock states. This is already a nontrivial requirement, since a standard laser outputs not Fock states but *coherent states*, which have the form

$$|\alpha\rangle = e^{-|\alpha|^2/2} \sum_{n=0}^{\infty} \frac{\alpha^n}{\sqrt{n!}} |n\rangle \tag{6.4}$$

for some $\alpha \in \mathbb{C}$. (In other words, sometimes there are zero photons, sometimes one, sometimes two, etc., with the number of photons following a Poisson distribution.) Fortunately, the task of building reliable single-photon sources is an extremely well-known one in quantum optics [48], and the technology to generate single-photon Fock states has been steadily improving over the past decade.

Still, one can ask whether an analogue of our computational hardness results goes through, if the inputs are coherent states rather than Fock states. As mentioned in Section 1.4, if the inputs are coherent states and the measurements are demolition, *or* the inputs are Gaussian states (a generalization of coherent states) and the measurements are Gaussian, then the probability distribution over measurement outcomes can be sampled in classical polynomial time. By contrast, if the inputs are coherent states and we have *nondemolition photon-number measurements*, then Theorem 4.9 shows that exact classical simulation of the linear-optics experiment would collapse the polynomial hierarchy. However, we do not know whether *approximate* classical simulation would have surprising complexity consequences in that case.

Measurements For our experiment, it is desirable to have an array of m photodetectors, which reliably measure the number of photons s_i in each mode $i \in [m]$. However, it would also suffice to use detectors that only measure whether each s_i is zero or nonzero. This is because our hardness results talk only about basis states $|S\rangle = |s_1, \ldots, s_m\rangle$ that are *collision-free*, meaning that $s_i \in \{0,1\}$ for all $i \in [m]$. Thus, one could simply postselect on the runs in which exactly n of the m detectors record a photon, in which case one knows that $s_i = 1$ for the corresponding modes i while $s_i = 0$ for the remaining $m - n$ modes. (In Appendix C, we will prove a "Boson Birthday Bound," which shows that as long as m is sufficiently large and the mode-mixing unitary U is Haar-random, this postselection step succeeds with probability close to 1. Intuitively, if m is large enough, then collision-free basis states are the overwhelming majority.)

What might *not* suffice are Gaussian measurements. As mentioned earlier, if both the input states *and* the measurements are Gaussian, then Bartlett and Sanders [9] showed that no superpolynomial quantum speedup is possible. We do not know what the situation is if the measurements are Gaussian and the inputs are single-photon Fock states.

Like single-photon sources, photodetectors have improved dramatically over the past decade, but of course no detector will be 100% efficient.[24] As we discuss later, the higher the photodetector efficiencies, the less need there is for *postselection*, and therefore, the more easily one can scale to larger numbers of photons.

[24] Here the "efficiency" of a photodetector refers to the probability of its detecting a photon that is present.

Number of photons n An obvious question is how many photons are needed for our experiment. The short answer is simply "the more, the better!" The goal of the experiment is to confirm that, for *every* positive integer n the transition amplitudes for n identical bosons are given by $n \times n$ permanents, as quantum mechanics predicts. So the larger the n the stronger the evidence for this claim, and the greater the strain on any competing interpretation.

At present, it seems fair to say that our experiment has already been done for $n = 2$ (this is the Hong-Ou-Mandel dip [36]). However, we are not aware of any experiment directly testing formula (6.3) even for $n = 3$. Experimentalists we consulted expressed the view that this is mostly just a matter of insufficient motivation before now, and that the $n = 3$ and even $n = 4$ cases ought to be feasible with current technology.[25]

Of course, the most interesting regime for computer science is the one where n is large enough that a classical computer would have difficulty computing an $n \times n$ permanent. The best known classical algorithm for the permanent, *Ryser's algorithm*, uses about $2^{n+1}n^2$ floating-point operations. If $n = 10$ then this is about $200,000$ operations; if $n = 20$ it is about 800 million; if $n = 30$ it is about 2 trillion. In any of these cases, it would be exciting to perform a linear-optics experiment that "almost-instantly" sampled from a distribution in which the probabilities were given by $n \times n$ permanents.

Number of modes m Another important question is how many *modes* are needed for our experiment. We showed in our proof of Theorem 1.3—see in particular Theorem 5.1—that it suffices to use

$$m = O\left(\frac{1}{\delta} n^5 \log^2 \frac{n}{\delta}\right)$$

modes. This bound is polynomial in n but clearly impractical. As we mentioned in Section 5.1, results of Jiang [38] (which we learned about only recently) strongly suggest that an improved analysis would yield $m = O\left(n^2\right)$. On the other hand, by the birthday paradox, we cannot have *fewer* than $m = \Omega\left(n^2\right)$ modes, if we want the state $\varphi(U)|1_n\rangle$ to be dominated by *collision-free* photon configurations (meaning those containing at most one photon per mode).

Unfortunately, a quadratic number of modes might still be difficult to arrange in practice. So the question arises: what would happen if we ran our experiment with a *linear* number of modes, $m = O(n)$? In that case, almost every basis state would contain collisions, so our formal argument for the classical hardness of approximate BOSONSAMPLING, based on Conjectures 1.6 and 1.5, would no longer apply. On the other hand, we suspect it would still be *true* that sampling is classically hard! Giving a formal argument for the hardness of approximate BOSONSAMPLING, with n photons and $m = O(n)$ modes, is an interesting challenge that we leave.

In the meantime, if the goal of one's experiment is just to verify that the permanent formula (6.3) remains correct for large values of n, then large numbers of photon collisions are presumably acceptable. In this case, it should suffice to set $m \approx n$ or possibly even $m \ll n$ (though note that it is easy to give a classical simulation algorithm that runs in $n^{O(m)}$ time).

[25]As mentioned earlier, right before this paper was published, Broome et al. [13], Spring et al. [62], Tillmann et al. [68], and Crespi et al. [19] announced the successful demonstration of $n = 3$ case. Spring et al. [62] even report results for the $n = 4$ case, albeit with initial states of the form $|2,2,0,\ldots\rangle$ rather than $|1,1,1,1,0,\ldots\rangle$.

Choice of unitary transformation U One could look for an n-photon Hong-Ou-Mandel dip using *any* unitary transformation U that produces nontrivial interference among n of the m modes. However, some choices of U are more interesting than others. The prescription suggested by our results is to choose U *randomly*, according to the Haar measure over $m \times m$ unitaries. Once U is chosen, one can then "hardwire" a network of beamsplitters and phaseshifters that produces U.

There are at least three reasons why using a Haar-random U seems like a good idea:

(1) Theorem 5.1 showed that any sufficiently small submatrix of a Haar-random unitary matrix U is close to a matrix of i. i. d. Gaussians. This extremely useful fact is what let us prove Theorem 1.3, which relates the hardness of approximate BOSONSAMPLING to the hardness of more "natural" problems that have nothing to do with unitary matrices.

(2) Setting aside our results, the Haar measure is the unique rotationally-invariant measure over unitaries. This makes it an obvious choice, if the goal is to avoid any "special structure" that might make the BOSONSAMPLING problem easy.

(3) In the linear-optics model, one simple way to apply a Haar-random $m \times m$ unitary matrix U is via a network of $\operatorname{poly}(m)$ *randomly-chosen* beamsplitters and phaseshifters.

Optical elements One might worry about the *number* of beamsplitters and phaseshifters needed to implement an arbitrary $m \times m$ unitary transformation U or a Haar-random U in particular. And indeed, the upper bound of Reck et al. [54] (Lemma 3.1) shows only that $O\left(m^2\right)$ beamsplitters and phaseshifters suffice to implement any unitary, and this is easily seen to be tight by a dimension argument. Unfortunately, a network of $\sim m^2$ optical elements might already strain the limits of practicality, especially if m has been chosen to be quadratically larger than n.

Happily, Section 6.3 will show how to reduce the number of optical elements from $O\left(m^2\right)$ to $O(mn)$ by exploiting a simple observation: namely, *we only care about the optical network's behavior on the first n modes, since the standard initial state $|1_n\rangle$ has no photons in the remaining $m-n$ modes anyway.* Section 6.3 will also show how to "parallelize" the resulting optical network, so that the $O(mn)$ beamsplitters and phaseshifters are arranged into only $O(n \log m)$ layers.

Whether one can parallelize linear-optics computations still further, and whether one can sample from hard distributions using even fewer optical elements (say, $O(m \log m)$), are interesting topics for future work.

Error There are many sources of error in our experiment; understanding and controlling the errors is perhaps the central challenge an experimentalist will face. At the most obvious level:

(1) Generation of single-photon Fock states will not be perfectly reliable.

(2) The beamsplitters and phaseshifters will not induce exactly the desired unitary transformations.

(3) Each photon will have some probability of "getting lost along the way."

(4) The photodetectors will not have perfect efficiency.

(5) If the lengths of the optical fibers are not well-calibrated, or the single-photon sources are not synchronized, or there is vibration, etc., then the photons will generally arrive at the photodetectors at different times.

If (5) occurs, then the photons effectively become *distinguishable*, and the amplitudes will no longer correspond to $n \times n$ permanents. So then how well-synchronized do the photons need to be? To answer this question, recall that each photon is actually a Gaussian wavepacket in the position basis, rather than a localized point. For formula (6.3) to hold, what is necessary is that the photons arrive at the photodetectors within a short enough time interval that their wavepackets have large pairwise overlaps.

The fundamental worry is that, as we increase the number of photons n the probability of a successful run of the experiment might decrease like c^{-n}. In practice, experimentalists usually deal with such behavior by *postselecting* on the successful runs. In our context, that could mean (for example) that we only count the runs in which n detectors register a photon simultaneously, even if such runs are exponentially unlikely. We expect that any realistic implementation of our experiment would involve at least some postselection. However, if the eventual goal is to scale to large values of n then any need to postselect on an event with probability c^{-n} presents an obvious barrier. Indeed, from an asymptotic perspective, this sort of postselection defeats the entire purpose of using a quantum computer rather than a classical computer.

For this reason, while even a heavily-postselected Hong-Ou-Mandel dip with (say) $n = 3$ 4 or 5 photons would be interesting, our real hope is that it will ultimately be possible to scale our experiment to interestingly large values of n while maintaining a total error that is closer to 0 than to 1. However, supposing this turns out to be possible, one can still ask: *how close to* 0 *does the error need to be?*

Unfortunately, just like with the question of how many photons are needed, it is difficult to give a direct answer, because of the reliance of our results on asymptotics. What Theorem 1.3 shows is that, *if* one can scale the BOSONSAMPLING experiment to n photons and error δ in total variation distance, using an amount of "experimental effort" that scales polynomially with both n and $1/\delta$ then modulo our complexity conjectures, the Extended Church-Turing Thesis is false. The trouble is that no finite experiment can ever prove (or disprove) the claim that scaling to n photons and error δ takes $\operatorname{poly}(n, 1/\delta)$ experimental effort. One can, however, build a circumstantial case for this claim—by increasing n decreasing δ and making it clear that, with reasonable effort, one could have increased n and decreased δ still further.

One challenge we leave is to prove a computational hardness result that works for a *fixed* (say, constant) error δ rather than treating $1/\delta$ as an input parameter to the sampling algorithm along with n. A second challenge is whether any nontrivial error-correction is possible within the noninteracting-boson model. In standard quantum computing, the famous Threshold Theorem [7, 44] asserts that there exists a constant $\tau > 0$ such that, even if each qubit fails with independent probability τ at each time step, one can still "correct errors faster than they happen," and thereby perform an arbitrarily long quantum computation. In principle, the Threshold Theorem could be applied to our experiment, to deal with all the sources of error listed above. The issue is that, if we have the physical resources available for fault-tolerant quantum computing, then perhaps we ought to forget about BOSONSAMPLING, and simply run a universal quantum computation! What we want, ideally, is a way to reduce the error in our experiment, *without* giving up on the implementation advantages that make the experiment attractive in the first place.

6.3 Reducing the size and depth of optical networks

In this section, we discuss how best to realize an $m \times m$ unitary transformation U acting on the initial state $|1_n\rangle$ as a product of beamsplitters and phaseshifters. If we implement U in the "obvious" way—by appealing to Lemma 3.1—then the number of optical elements and the depth will both be $O(m^2)$. However, we can obtain a significant improvement by noticing that our goal is just to apply *some* unitary transformation $\widetilde{U}$ such that $\varphi(\widetilde{U})|1_n\rangle = \varphi(U)|1_n\rangle$: we do not care about the behavior on $\widetilde{U}$ on inputs other than $|1_n\rangle$. This yields a network in which the number of optical elements and the depth are both $O(mn)$.

The following theorem shows that we can reduce the depth further, to $O(n \log m)$ by exploiting parallelization.

Theorem 6.1 (Parallelization of Linear-Optical Networks)**.** *Given any $m \times m$ unitary operation U one can map the initial state $|1_n\rangle$ to $\varphi(U)|1_n\rangle$ using a linear-optical network of depth $O(n \log m)$ consisting of $O(mn)$ beamsplitters and phaseshifters.*

Proof. We will consider a linear-optics system with $m+n$ modes. Let

$$V = \begin{pmatrix} U & 0 \\ 0 & I \end{pmatrix} \tag{6.5}$$

be a unitary transformation that acts as U on the first m modes, and as the identity on the remaining n modes. Then our goal will be to map $|1_n\rangle$ to $\varphi(V)|1_n\rangle$. Let $|e_i\rangle$ be the basis state that consists of a single photon in mode i and no photons in the remaining $m+n-1$ modes. Also, let $|\psi_i\rangle = V|e_i\rangle$. Then it clearly suffices to implement *some* unitary transformation $\widetilde{V}$ that maps $|e_i\rangle$ to $|\psi_i\rangle$ for all $i \in [n]$—for then $\varphi(\widetilde{V})|1_n\rangle = \varphi(V)|1_n\rangle$ by extension. Our first claim is that, for each $i \in [n]$ *individually*, there exists a unitary transformation V_i that maps $|e_i\rangle$ to $|\psi_i\rangle$, and that can be implemented by a linear-optical network of depth $\log_2 m + O(1)$ with $O(m)$ optical elements. To implement V_i we use a binary doubling strategy: first map $|e_i\rangle$ to a superposition of the first two modes,

$$|z_1\rangle = \alpha_1 |e_1\rangle + \alpha_2 |e_2\rangle\,. \tag{6.6}$$

Then, by using two beamsplitters in parallel, map the above state $|z_1\rangle$ to a superposition of the first four modes,

$$|z_2\rangle = \alpha_1 |e_1\rangle + \alpha_2 |e_2\rangle + \alpha_3 |e_3\rangle + \alpha_4 |e_4\rangle\,. \tag{6.7}$$

Next, by using four beamsplitters in parallel, map $|z_2\rangle$ to a superposition $|z_3\rangle$ of the first eight modes, and so on until $|\psi_i\rangle$ is reached. It is clear that the total depth required is $\log_2 m + O(1)$ while the number of optical elements required is $O(m)$. This proves the claim. Now let S_i be a unitary transformation that swaps modes i and $m+i$, and that acts as the identity on the remaining $m+n-2$ modes. Then we will implement $\widetilde{V}$ as follows:

$$\widetilde{V} = V_n S_n V_n^\dagger \cdot \cdots \cdot V_2 S_2 V_2^\dagger \cdot V_1 S_1 V_1^\dagger \cdot S_n \cdots S_1\,. \tag{6.8}$$

In other words: first swap modes $1,\ldots,n$ with modes $m+1,\ldots,m+n$. Then, for all $i := 1$ to n apply $V_i S_i V_i^\dagger$. Since each S_i involves only one optical element, while each V_i and $V_i^\dagger$ involves $O(m)$ optical elements and $O(\log m)$ depth, it is clear that we can implement $\widetilde{V}$ using a linear-optical network of depth

$O(n \log m)$ with $O(mn)$ optical elements. To prove the theorem, we need to verify that $\widetilde{V}|e_i\rangle = |\psi_i\rangle$ for all $i \in [n]$. We do so in three steps. First, notice that for all $i \in [n]$,

$$V_i S_i V_i^\dagger (S_i |e_i\rangle) = V_i S_i V_i^\dagger |e_{m+i}\rangle \tag{6.9}$$

$$= V_i S_i |e_{m+i}\rangle \tag{6.10}$$

$$= V_i |e_i\rangle \tag{6.11}$$

$$= |\psi_i\rangle \, . \tag{6.12}$$

where line (6.10) follows since $V_i^\dagger$ acts only on the first m modes. Second, for all $i, j \in [n]$ with $i \neq j$,

$$V_j S_j V_j^\dagger |e_{m+i}\rangle = |e_{m+i}\rangle \, , \tag{6.13}$$

since V_j and S_j both act as the identity on $|e_{m+i}\rangle$. Third, notice that $\langle \psi_i | \psi_j \rangle = 0$ for all $i \neq j$ since $|\psi_i\rangle$ and $|\psi_j\rangle$ correspond to two different columns of the unitary matrix U. Since unitaries preserve inner product, this means that $V_j^\dagger |\psi_i\rangle$ is also orthogonal to $V_j^\dagger |\psi_j\rangle = V_j^\dagger V_j |e_j\rangle = |e_j\rangle$: in other words, the state $V_j^\dagger |\psi_i\rangle$ has no support on the j^{th} mode. It follows that S_j acts as the identity on $V_j^\dagger |\psi_i\rangle$—and therefore, for all $i, j \in [n]$ with $i \neq j$ we have

$$V_j S_j V_j^\dagger |\psi_i\rangle = V_j V_j^\dagger |\psi_i\rangle = |\psi_i\rangle \, . \tag{6.14}$$

Summarizing, we find that for all $i \in [n]$:

- $V_i S_i V_i^\dagger$ maps $|e_{m+i}\rangle$ to $|\psi_i\rangle$.
- $V_j S_j V_j^\dagger$ maps $|e_{m+i}\rangle$ to itself for all $j < i$.
- $V_j S_j V_j^\dagger$ maps $|\psi_i\rangle$ to itself for all $j > i$.

We conclude that $\widetilde{V}|e_i\rangle = V_i S_i V_i^\dagger |e_{m+i}\rangle = |\psi_i\rangle$ for all $i \in [n]$. This proves the theorem. ☐

7 Reducing $\mathrm{GPE}_\times$ to $|\mathrm{GPE}|_\pm^2$

The goal of this section is to prove Theorem 1.7: that, assuming Conjecture 1.6 (the Permanent Anti-Concentration Conjecture), the $\mathrm{GPE}_\times$ and $|\mathrm{GPE}|_\pm^2$ problems are polynomial-time equivalent. Or in words: if we can additively estimate $|\mathrm{Per}(X)|^2$ with high probability over a Gaussian matrix $X \sim \mathcal{G}^{n\times n}$ then we can also multiplicatively estimate $\mathrm{Per}(X)$ with high probability over a Gaussian matrix X.

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 $\varepsilon, \delta > 0$ recall that the $\mathrm{GPE}_\times$ problem (Problem 1.4) asks us to estimate $\mathrm{Per}(X)$ to within error $\pm\varepsilon \cdot |\mathrm{Per}(X)|$ with probability at least $1-\delta$ over X in $\mathrm{poly}(n, 1/\varepsilon, 1/\delta)$ time. Meanwhile, the $|\mathrm{GPE}|_\pm^2$ problem (Problem 1.2) asks us to estimate $|\mathrm{Per}(X)|^2$ to within error $\pm\varepsilon \cdot n!$ with probability at least $1-\delta$ over X in $\mathrm{poly}(n, 1/\varepsilon, 1/\delta)$ time. It is easy to give a reduction from $|\mathrm{GPE}|_\pm^2$ to $\mathrm{GPE}_\times$. The hard direction, and the one that requires Conjecture 1.6, is to reduce $\mathrm{GPE}_\times$ to $|\mathrm{GPE}|_\pm^2$.

While technical, this reduction is essential for establishing the connection we want between

(1) Theorem 1.3 (our main result), which relates the classical hardness of BOSONSAMPLING to $|\mathrm{GPE}|_{\pm}^{2}$ and

(2) Conjecture 1.5 (the Permanent-of-Gaussians Conjecture), which asserts that the GAUSSIAN PERMANENT ESTIMATION problem is #P-hard, in the more "natural" setting of multiplicative rather than additive estimation, and $\mathrm{Per}(X)$ rather than $|\mathrm{Per}(X)|^{2}$.

Besides $\mathrm{GPE}_{\times}$ and $|\mathrm{GPE}|_{\pm}^{2}$ one can of course also define two "hybrid" problems:

- $\mathrm{GPE}_{\pm}$ the problem of estimating $\mathrm{Per}(X)$ additively (i. e., to within error $\pm\varepsilon\sqrt{n!}$), with probability at least $1-\delta$ over X in $\mathrm{poly}(n,1/\varepsilon,1/\delta)$ time.
- $|\mathrm{GPE}|_{\times}^{2}$ the problem of estimating $|\mathrm{Per}(X)|^{2}$ multiplicatively (i. e., to within error $\pm\varepsilon\cdot|\mathrm{Per}(X)|^{2}$), with probability at least $1-\delta$ over X in $\mathrm{poly}(n,1/\varepsilon,1/\delta)$ time.

The $\mathrm{GPE}_{\pm}$ problem is not directly used in this paper, but it does play a central role in the recent followup work of Arora et al. [8]. The $|\mathrm{GPE}|_{\times}^{2}$ problem *will* be useful to us, as an "intermediate stepping-stone" in reducing $\mathrm{GPE}_{\times}$ to $|\mathrm{GPE}|_{\pm}^{2}$. Note that, assuming Conjecture 1.6, the $\mathrm{GPE}_{\pm}$ and $|\mathrm{GPE}|_{\times}^{2}$ problems both become equivalent to $\mathrm{GPE}_{\times}$ and $|\mathrm{GPE}|_{\pm}^{2}$ as a byproduct.

Let us start by proving the "easy" reductions. In what follows, $\leq_{\mathsf{P}}$ means "is polynomial-time reducible to" (though in the next two lemmas, the reductions are all extremely simple one-to-one mappings).

Lemma 7.1. *We have the following "square" of reductions, from additive to multiplicative approximation, and from approximation of* $|\mathrm{Per}(X)|^{2}$ *to approximation of* $\mathrm{Per}(X)$*:*

$$\mathrm{GPE}_{\pm} \leq_{\mathsf{P}} \mathrm{GPE}_{\times}, \tag{7.1}$$

$$|\mathrm{GPE}|_{\pm}^{2} \leq_{\mathsf{P}} |\mathrm{GPE}|_{\times}^{2}, \tag{7.2}$$

$$|\mathrm{GPE}|_{\times}^{2} \leq_{\mathsf{P}} \mathrm{GPE}_{\times}, \tag{7.3}$$

$$|\mathrm{GPE}|_{\pm}^{2} \leq_{\mathsf{P}} \mathrm{GPE}_{\pm}. \tag{7.4}$$

As a corollary, of course, $|\mathrm{GPE}|_{\pm}^{2} \leq_{\mathsf{P}} \mathrm{GPE}_{\times}$*. None of these reductions rely on unproved conjectures.*

Proof. We start with $\mathrm{GPE}_{\pm} \leq_{\mathsf{P}} \mathrm{GPE}_{\times}$. Suppose we have a polynomial-time algorithm M that, given $\langle X, 0^{1/\varepsilon}, 0^{1/\delta}\rangle$ outputs a good multiplicative approximation to $\mathrm{Per}(X)$—that is, a z such that

$$|z-\mathrm{Per}(X)| \leq \varepsilon\,|\mathrm{Per}(X)| \tag{7.5}$$

—with probability at least $1-\delta$ over $X \sim \mathcal{G}^{n\times n}$. We claim that z is also a good *additive* approximation to $\mathrm{Per}(X)$ with high probability over X. For by Markov's inequality,

$$\Pr_{X}\left[|\mathrm{Per}(X)| > k\sqrt{n!}\right] < \frac{1}{k^{2}}. \tag{7.6}$$

So by the union bound,

$$\Pr_X\left[|z-\mathrm{Per}(X)|>\varepsilon k\sqrt{n!}\right]\leq\Pr_X\left[|z-\mathrm{Per}(X)|>\varepsilon|\mathrm{Per}(X)|\right]+\Pr_X\left[\varepsilon|\mathrm{Per}(X)|>\varepsilon k\sqrt{n!}\right] \tag{7.7}$$

$$\leq\delta+\frac{1}{k^2}. \tag{7.8}$$

Thus, we can achieve any desired additive error bounds (ε',δ') by setting $\delta:=\delta'/2$, $k:=\sqrt{2/\delta'}$ and $\varepsilon:=\varepsilon'/k$ so that $\varepsilon k=\varepsilon'$ and $\delta+1/k^2=\delta'$. Clearly this increases M's running time by at most a polynomial factor. The reduction $|\mathrm{GPE}|_\pm^2\leq_{\mathsf{P}}|\mathrm{GPE}|_\times^2$ is completely analogous, and is omitted for brevity.

Next we prove $|\mathrm{GPE}|_\times^2\leq_{\mathsf{P}}\mathrm{GPE}_\times$. Suppose z is a good multiplicative approximation to $\mathrm{Per}(X)$:

$$|z-\mathrm{Per}(X)|\leq\varepsilon|\mathrm{Per}(X)|. \tag{7.9}$$

Then certainly $|z|^2$ is a good multiplicative approximation to $|\mathrm{Per}(X)|^2$:

$$\left||z|^2-|\mathrm{Per}(X)|^2\right|=||z|-|\mathrm{Per}(X)||\,(|z|+|\mathrm{Per}(X)|) \tag{7.10}$$

$$\leq\varepsilon|\mathrm{Per}(X)|\cdot[(1+\varepsilon)|\mathrm{Per}(X)|+|\mathrm{Per}(X)|] \tag{7.11}$$

$$=\left(2\varepsilon+\varepsilon^2\right)|\mathrm{Per}(X)|^2 \tag{7.12}$$

$$\leq3\varepsilon|\mathrm{Per}(X)|^2. \tag{7.13}$$

Finally we prove $|\mathrm{GPE}|_\pm^2\leq_{\mathsf{P}}\mathrm{GPE}_\pm$. Suppose z is a good additive approximation to $\mathrm{Per}(X)$:

$$|z-\mathrm{Per}(X)|\leq\varepsilon\sqrt{n!} \tag{7.14}$$

with probability at least $1-\delta$ over X. Then provided that occurs,

$$\left||z|^2-|\mathrm{Per}(X)|^2\right|=||z|-|\mathrm{Per}(X)||\,(|z|+|\mathrm{Per}(X)|) \tag{7.15}$$

$$\leq\varepsilon\sqrt{n!}\cdot\left[|\mathrm{Per}(X)|+\sqrt{n!}+|\mathrm{Per}(X)|\right] \tag{7.16}$$

$$=\varepsilon\left[n!+2\sqrt{n!}\,|\mathrm{Per}(X)|\right]. \tag{7.17}$$

So again using equation (7.6) together with the union bound,

$$\Pr_X\left[\left||z|^2-|\mathrm{Per}(X)|^2\right|>\varepsilon k\cdot n!\right]\leq\Pr_X\left[\left||z|^2-|\mathrm{Per}(X)|^2\right|>\varepsilon\left[n!+2\sqrt{n!}\,|\mathrm{Per}(X)|\right]\right] \tag{7.18}$$

$$+\Pr_X\left[\varepsilon\left[n!+2\sqrt{n!}\,|\mathrm{Per}(X)|\right]>\varepsilon k\cdot n!\right]$$

$$\leq\delta+\Pr_X\left[|\mathrm{Per}(X)|>\frac{k-1}{2}\sqrt{n!}\right] \tag{7.19}$$

$$\leq\delta+\frac{4}{(k-1)^2}. \tag{7.20}$$

Once again, we can achieve any desired error bounds (ε',δ') by appropriate choices of δ k, and ε. □

Next we show that, assuming the Permanent Anti-Concentration Conjecture, the reductions from additive to multiplicative approximation in Lemma 7.1 can be reversed.

Lemma 7.2. *Assuming Conjecture 1.6, we have* $\mathrm{GPE}_{\times} \leq_{\mathsf{P}} \mathrm{GPE}_{\pm}$ *(so that* $\mathrm{GPE}_{\times}$ *and* $\mathrm{GPE}_{\pm}$ *become polynomial-time equivalent), and likewise* $|\mathrm{GPE}|^2_{\times} \leq_{\mathsf{P}} |\mathrm{GPE}|^2_{\pm}$ *(so that* $|\mathrm{GPE}|^2_{\times}$ *and* $|\mathrm{GPE}|^2_{\pm}$ *become polynomial-time equivalent).*

Proof. We show $\mathrm{GPE}_{\times} \leq_{\mathsf{P}} \mathrm{GPE}_{\pm}$; the reduction $|\mathrm{GPE}|^2_{\times} \leq_{\mathsf{P}} |\mathrm{GPE}|^2_{\pm}$ is completely analogous and is omitted for brevity. Suppose z is a good additive approximation to $\mathrm{Per}(X)$:

$$|z - \mathrm{Per}(X)| \leq \varepsilon\sqrt{n!} \tag{7.21}$$

with probability at least $1-\delta$ over $X \sim \mathcal{G}^{n\times n}$. Then assuming Conjecture 1.6, we claim that z is also a good *multiplicative* approximation to $\mathrm{Per}(X)$ with high probability over X. By the conjecture, there exists a polynomial p such that

$$\Pr_X\left[|\mathrm{Per}(X)| < \frac{\sqrt{n!}}{p(n,1/\delta)}\right] < \delta\,. \tag{7.22}$$

So by the union bound,

$$\begin{aligned}\Pr_X\left[|z-\mathrm{Per}(X)| > \varepsilon\cdot p(n,1/\delta)\,|\mathrm{Per}(X)|\right] &\leq \Pr_X\left[|z-\mathrm{Per}(X)| > \varepsilon\sqrt{n!}\right] && (7.23)\\ &\quad + \Pr_X\left[\varepsilon\sqrt{n!} > \varepsilon\cdot p(n,1/\delta)\,|\mathrm{Per}(X)|\right] \\ &\leq 2\delta\,. && (7.24)\end{aligned}$$

Thus, we can achieve any desired multiplicative error bounds (ε',δ') by setting $\delta := \delta'/2$ and $\varepsilon := \varepsilon'/p(n,1/\delta)$ incurring at most a polynomial blowup in running time. □

We now proceed to proving the main result of the section: that assuming the Permanent Anti-Concentration Conjecture, approximating $|\mathrm{Per}(X)|^2$ for a Gaussian random matrix $X \sim \mathcal{G}^{n\times n}$ is as hard as approximating $\mathrm{Per}(X)$ itself. This result can be seen as an average-case analogue of Theorem 4.3. To prove it, we need to give a reduction that estimates the phase $\mathrm{Per}(X)/|\mathrm{Per}(X)|$ of a permanent $\mathrm{Per}(X)$ given only the ability to estimate $|\mathrm{Per}(X)|$ (for most Gaussian matrices X). As in the proof of Theorem 4.3, our reduction proceeds by induction on n: we assume the ability to estimate $\mathrm{Per}(Y)$ for a certain $(n-1)\times(n-1)$ submatrix Y of X and then use that (together with estimates of $|\mathrm{Per}(X')|$ for various $n\times n$ matrices X') to estimate $\mathrm{Per}(X)$. Unfortunately, the reduction and its analysis are more complicated than in Theorem 4.3, since in this case, we can only assume that our oracle estimates $|\mathrm{Per}(X)|^2$ with high probability if X "looks like" a Gaussian matrix. This rules out the adaptive reduction of Theorem 4.3, which even starting with a Gaussian matrix X would vary the top-left entry so as to produce new matrices X' that look nothing like Gaussian matrices. Instead, we will use a *nonadaptive* reduction, which in turn necessitates a more delicate error analysis, as well as an appeal to Conjecture 1.6.

To do the error analysis, we first need a technical lemma about the numerical stability of *triangulation*. Here triangulation means the procedure that determines a point $z \in \mathbb{R}^d$ given the Euclidean distances

$\Delta(z,y_i)$ between z and $d+1$ known points $y_1,\ldots,y_{d+1}\in\mathbb{R}^d$ that are in general position. So for example, the $d=3$ case corresponds to how a GPS receiver calculates its position given its measured distances to four satellites. (Note that the distances to any d of the y_i's are actually enough to narrow z down to two possibilities; the $(d+1)^{st}$ distance is only needed to eliminate one of those possibilities.) Here we are interested in the case $d=2$ which corresponds to calculating an unknown complex number $z=\operatorname{Per}(X)\in\mathbb{C}$ given its squared Euclidean distances $|z-y_1|^2$, $|z-y_2|^2$, $|z-y_3|^2$ to some "fixed" complex numbers $y_1,y_2,y_3\in\mathbb{C}$. The question that interests us is this:

> *Suppose our estimates of the squared distances $|z-y_1|^2$, $|z-y_2|^2$, $|z-y_3|^2$ are noisy, and our estimates of the points y_1,y_2,y_3 are also noisy. Can we upper-bound the error that noise induces in our triangulated value of z?*

The following lemma answers that question, in the special case where $y_1=0$, $y_2=w$, $y_3=iw$ for some complex number w.

Lemma 7.3 (Stability of Triangulation)**.** *Let $z=re^{i\theta}\in\mathbb{C}$ be a hidden complex number that we are trying to estimate, and let $w=ce^{i\tau}\in\mathbb{C}$ be a second "reference" number ($r,c>0$ and $\theta,\tau\in(-\pi,\pi]$). For some known constant $\lambda>0$ let*

$$R:=|z|^2=r^2\,, \tag{7.25}$$
$$S:=|z-\lambda w|^2=r^2+\lambda^2c^2-2\lambda rc\cos(\theta-\tau)\,, \tag{7.26}$$
$$T:=|z-i\lambda w|^2=r^2+\lambda^2c^2-2\lambda rc\sin(\theta-\tau)\,, \tag{7.27}$$
$$C:=|w|^2=c^2\,. \tag{7.28}$$

Suppose we are given approximations $\widetilde{R},\widetilde{S},\widetilde{T},\widetilde{C},\widetilde{\tau}$ to R,S,T,C,τ respectively, such that

$$\left|\widetilde{R}-R\right|,\left|\widetilde{S}-S\right|,\left|\widetilde{T}-T\right|<\varepsilon\lambda^2C\,, \tag{7.29}$$
$$\left|\widetilde{C}-C\right|<\varepsilon C\,. \tag{7.30}$$

Suppose also that

$$\varepsilon\leq\frac{1}{10}\min\left\{1,\frac{R}{\lambda^2C}\right\}. \tag{7.31}$$

Then the approximation

$$\widetilde{\theta}:=\widetilde{\tau}+\operatorname{sgn}\left(\widetilde{R}+\widetilde{C}-\widetilde{T}\right)\arccos\left(\frac{\widetilde{R}+\widetilde{C}-\widetilde{S}}{2\sqrt{\widetilde{R}\widetilde{C}}}\right) \tag{7.32}$$

satisfies

$$\left|\widetilde{\theta}-\theta\right|\bmod 2\pi\leq|\widetilde{\tau}-\tau|+3\sqrt{\varepsilon}\left(\lambda\sqrt{\frac{C}{R}}+1\right). \tag{7.33}$$

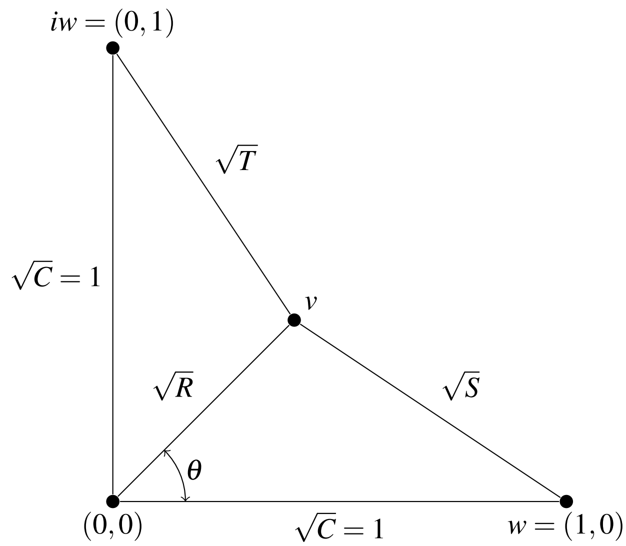

Figure 4: Triangulating the point $v = \sqrt{R}e^{i\theta}$ given its squared distances to the origin, $w = (1,0)$ and $iw = (0,1)$.

Proof. Notice that without loss of generality, we can fix $\lambda := 1$. To obtain the result for general λ we then apply the $\lambda = 1$ case of the lemma, except that we set $w := \lambda w$ $C := \lambda^2 C$ and $\widetilde{C} := \lambda^2 \widetilde{C}$. A second simplification that we can make without loss of generality is to fix $C := 1$ so that

$$\left|\widetilde{R} - R\right|, \left|\widetilde{S} - S\right|, \left|\widetilde{T} - T\right|, \left|\widetilde{C} - 1\right| < \varepsilon \leq \frac{1}{10} \min\{1, R\}. \tag{7.34}$$

A third simplification is to fix $\tau := 0$ so that $w = ce^{i\tau} = 1$. To obtain the result for general C and τ we simply rescale and rotate. After these simplifications, we have situation depicted in Figure 4, which satisfies the geometric identities

$$\cos\theta = \frac{R + 1 - S}{2\sqrt{R}}, \tag{7.35}$$

$$\sin\theta = \frac{R + 1 - T}{2\sqrt{R}}. \tag{7.36}$$

So we can write

$$\theta = b \arccos\left(\frac{R + 1 - S}{2\sqrt{R}}\right) \tag{7.37}$$

where $b \in \{-1, 1\}$ is a sign term given by

$$b := \operatorname{sgn}\theta = \operatorname{sgn}(\sin\theta) = \operatorname{sgn}(R + 1 - T). \tag{7.38}$$

Let

$$\widetilde{b} := \operatorname{sgn}\left(\widetilde{R}+\widetilde{C}-\widetilde{T}\right), \tag{7.39}$$

$$\widetilde{\theta} := \widetilde{\tau}+\widetilde{b}\arccos\left(\frac{\widetilde{R}+\widetilde{C}-\widetilde{S}}{2\sqrt{\widetilde{R}\widetilde{C}}}\right). \tag{7.40}$$

We now consider two cases. First suppose $|R+1-T| \le 3\varepsilon$. Then we have the following sequence of inequalities:

$$\left|2\sqrt{R}\sin\theta\right| \le 3\varepsilon, \tag{7.41}$$

$$\sin^2\theta \le \frac{9\varepsilon^2}{4R}, \tag{7.42}$$

$$\cos^2\theta \ge 1-\frac{9\varepsilon^2}{4R}, \tag{7.43}$$

$$\frac{(R+1-S)^2}{4R} \ge 1-\frac{9\varepsilon^2}{4R}, \tag{7.44}$$

$$4R-(R+1-S)^2 \le 9\varepsilon^2. \tag{7.45}$$

Hence

$$\begin{aligned}
4\widetilde{R}\widetilde{C}-\left(\widetilde{R}+\widetilde{C}-\widetilde{S}\right)^2 &= 4R-(R+1-S)^2 && (7.46)\\
&\quad +4\left[\left(\widetilde{R}-R\right)+\left(\widetilde{R}-R\right)\left(\widetilde{C}-1\right)+R\left(\widetilde{C}-1\right)\right]\\
&\quad -\left(\left(\widetilde{R}-R\right)+\left(\widetilde{C}-1\right)+\left(\widetilde{S}-S\right)\right)^2\\
&\quad -2(R+1-S)\left(\left(\widetilde{R}-R\right)+\left(\widetilde{C}-1\right)+\left(\widetilde{S}-S\right)\right)\\
&\le 9\varepsilon^2+4\varepsilon+4\varepsilon^2+4R\varepsilon+2(R+1)(3\varepsilon) && (7.47)\\
&\le 12\varepsilon(R+1) && (7.48)
\end{aligned}$$

where line (7.48) uses $\varepsilon \le \frac{1}{10}$. So

$$\begin{aligned}
\sin^2\left(\widetilde{\theta}-\widetilde{\tau}\right) &= 1-\cos^2\left(\widetilde{\theta}-\widetilde{\tau}\right) && (7.49)\\
&= 1-\frac{\left(\widetilde{R}+\widetilde{C}-\widetilde{S}\right)^2}{4\widetilde{R}\widetilde{C}} && (7.50)\\
&\le \frac{12\varepsilon(R+1)}{4\widetilde{R}\widetilde{C}} && (7.51)\\
&\le \frac{12\varepsilon(R+1)}{4(R-\varepsilon)(1-\varepsilon)} && (7.52)\\
&\le 4\varepsilon\left(1+\frac{1}{R}\right), && (7.53)
\end{aligned}$$

where line (7.53) uses $\varepsilon \leq \frac{1}{10}\min\{1,R\}$. So

$$\left|\widetilde{\theta}-\theta\right| - |\widetilde{\tau}| \leq |\theta| + \left|\widetilde{\theta}-\widetilde{\tau}\right| \tag{7.54}$$

$$\leq \arcsin\frac{3\varepsilon}{2\sqrt{R}} + \arcsin\sqrt{4\varepsilon\left(1+\frac{1}{R}\right)} \tag{7.55}$$

$$\leq 1.1\left(\frac{3\varepsilon}{2\sqrt{R}} + 2\sqrt{\varepsilon}\left(1+\frac{1}{\sqrt{R}}\right)\right) \tag{7.56}$$

$$\leq 3\sqrt{\varepsilon}\left(\frac{1}{\sqrt{R}}+1\right), \tag{7.57}$$

where line (7.56) uses the inequality $\arcsin x \leq 1.1x$ for $x \leq 1/2$ and line (7.57) uses $\varepsilon \leq 1/10$. Next suppose $|R+1-T| > 3\varepsilon$. Then by the triangle inequality,

$$\left|\left|\widetilde{R}+\widetilde{C}-\widetilde{T}\right| - |R+1-T|\right| \leq \left|\widetilde{R}-R\right| + \left|\widetilde{C}-1\right| + \left|\widetilde{T}-T\right| \leq 3\varepsilon, \tag{7.58}$$

which implies that

$$\operatorname{sgn}\left(\widetilde{R}+\widetilde{C}-\widetilde{T}\right) = \operatorname{sgn}(R+1-T) \tag{7.59}$$

and hence $\widetilde{b} = b$. So

$$\left|\widetilde{\theta}-\theta\right| - |\widetilde{\tau}| \leq \left|\arccos\left(\frac{\widetilde{R}+\widetilde{C}-\widetilde{S}}{2\sqrt{\widetilde{R}\widetilde{C}}}\right) - \arccos\left(\frac{R+1-S}{2\sqrt{R}}\right)\right| \tag{7.60}$$

$$\leq \arccos\left(\frac{R+1-S-3\varepsilon}{2\sqrt{\widetilde{R}\widetilde{C}}}\right) - \arccos\left(\frac{R+1-S}{2\sqrt{R}}\right) \tag{7.61}$$

$$\leq 2\left(\frac{3\varepsilon}{2\sqrt{\widetilde{R}\widetilde{C}}} + \frac{|R+1-S|}{2}\left|\frac{1}{\sqrt{R}} - \frac{1}{\sqrt{\widetilde{R}\widetilde{C}}}\right|\right) \tag{7.62}$$

$$\leq \frac{3\varepsilon}{\sqrt{(R-\varepsilon)(1-\varepsilon)}} + 2\sqrt{R}\left(\frac{1}{\sqrt{R}} - \frac{1}{\sqrt{(R+\varepsilon)(1+\varepsilon)}}\right) \tag{7.63}$$

$$\leq \frac{3\varepsilon}{\sqrt{(0.9R)(0.9)}} + 2\sqrt{R}\frac{\sqrt{(R+\varepsilon)(1+\varepsilon)}-\sqrt{R}}{\sqrt{R}\sqrt{(R+\varepsilon)(1+\varepsilon)}} \tag{7.64}$$

$$\leq \frac{3.4\varepsilon}{\sqrt{R}} + \frac{\varepsilon+\varepsilon R+\varepsilon^2}{\sqrt{R}\sqrt{R}} \tag{7.65}$$

$$\leq \frac{1.1\varepsilon}{R} + \frac{3.4\varepsilon}{\sqrt{R}} + \varepsilon \tag{7.66}$$

$$\leq \sqrt{\varepsilon}\left(\frac{1.1\sqrt{R/10}}{R} + \frac{3.4\sqrt{R/10}}{\sqrt{R}} + \sqrt{\frac{1}{10}}\right) \tag{7.67}$$

$$\leq \sqrt{\varepsilon}\left(\frac{0.35}{\sqrt{R}} + 1.4\right). \tag{7.68}$$

Here line (7.61) uses the monotonicity of the arccos function, line (7.62) uses the inequality

$$\arccos(x-\varepsilon) - \arccos x \leq 2\varepsilon \tag{7.69}$$

for $\varepsilon \leq 1/2$, line (7.63) uses the geometric inequality

$$\frac{R+1-S}{2\sqrt{R}} \leq 1\,, \tag{7.70}$$

which follows since the left-hand side represents a valid input to the arccos function, line (7.64) uses $\varepsilon \leq (1/10)\min\{1,R\}$, line (7.65) uses the inequality

$$\sqrt{x+\varepsilon} - \sqrt{x} \leq \frac{\varepsilon}{2\sqrt{x}}\,, \tag{7.71}$$

and lines (7.66), (7.67) and (7.68) again use $\varepsilon \leq (1/10)\min\{1,R\}$. Combining the two cases, when $C=1$ and $\tau=0$ we have

$$\begin{aligned}
\left|\widetilde{\theta}-\theta\right| &\leq |\widetilde{\tau}| + \max\left\{3\sqrt{\varepsilon}\left(\frac{1}{\sqrt{R}}+1\right), \sqrt{\varepsilon}\left(\frac{0.35}{\sqrt{R}}+1.4\right)\right\} && (7.72)\\
&= |\widetilde{\tau}| + 3\sqrt{\varepsilon}\left(\frac{1}{\sqrt{R}}+1\right) && (7.73)
\end{aligned}$$

as claimed. □

We will also need a lemma about the autocorrelation of the Gaussian distribution, which will be reused in Section 9.

Lemma 7.4 (Autocorrelation of Gaussian Distribution)**.** *Consider the distributions*

$$\mathcal{D}_1 = \mathcal{N}\left(0,(1-\varepsilon)^2\right)^N_{\mathbb{C}}, \tag{7.74}$$

$$\mathcal{D}_2 = \prod_{i=1}^{N} \mathcal{N}(v_i,1)_{\mathbb{C}} \tag{7.75}$$

for some vector $v \in \mathbb{C}^N$. We have

$$\left\|\mathcal{D}_1 - \mathcal{G}^N\right\| \leq 2N\varepsilon, \tag{7.76}$$

$$\left\|\mathcal{D}_2 - \mathcal{G}^N\right\| \leq \|v\|_2\,. \tag{7.77}$$

Proof. It will be helpful to think of each complex coordinate as two real coordinates, in which case $\mathcal{G}^N = \mathcal{N}(0,1/2)^{2N}_{\mathbb{R}}$ and v is a vector in $\mathbb{R}^{2N}$. For the first part, we have

$$\begin{aligned}
\left\|\mathcal{D}_1 - \mathcal{G}^N\right\| &\leq 2N\left\|\mathcal{N}\left(0,\frac{(1-\varepsilon)^2}{2}\right)_{\mathbb{R}} - \mathcal{N}\left(0,\frac{1}{2}\right)_{\mathbb{R}}\right\| && (7.78)\\
&= \frac{N}{\sqrt{\pi}}\int_{-\infty}^{\infty}\left|e^{-x^2/(1-\varepsilon)^2} - e^{-x^2}\right|dx && (7.79)\\
&\leq 2N\varepsilon && (7.80)
\end{aligned}$$

where line (7.78) follows from the triangle inequality and line (7.80) from straightforward estimates. For the second part, by the rotational invariance of the Gaussian distribution, the variation distance is unaffected if we replace v by any other vector with the same 2-norm. So let $v := (\ell, 0, \ldots, 0)$ where $\ell = \|v\|_2$. Then

$$\left\|\mathcal{D}_2 - \mathcal{G}^N\right\| = \frac{1}{2}\int_{x_1,\ldots,x_{2N}=-\infty}^{\infty} \left| \frac{e^{-(x_1-\ell)^2}}{\sqrt{\pi}} \frac{e^{-x_2^2}}{\sqrt{\pi}} \cdots \frac{e^{-x_{2N}^2}}{\sqrt{\pi}} - \frac{e^{-x_1^2}}{\sqrt{\pi}} \frac{e^{-x_2^2}}{\sqrt{\pi}} \cdots \frac{e^{-x_{2N}^2}}{\sqrt{\pi}} \right| dx_1 \cdots dx_{2N} \tag{7.81}$$

$$= \frac{1}{2\sqrt{\pi}} \int_{-\infty}^{\infty} \left| e^{-(x-\ell)^2} - e^{-x^2} \right| dx \tag{7.82}$$

$$\leq \ell, \tag{7.83}$$

where line (7.83) follows from straightforward estimates. □

Using Lemmas 7.3 and 7.4, we can now complete the proof of Theorem 1.7: that assuming Conjecture 1.6 (the Permanent Anti-Concentration Conjecture), the $\text{GPE}_\times$ and $|\text{GPE}|_\pm^2$ problems are polynomial-time equivalent under nonadaptive reductions.

Proof of Theorem 1.7. Lemma 7.1 already gave an unconditional reduction from $|\text{GPE}|_\pm^2$ to $\text{GPE}_\times$. So it suffices to reduce in the other direction—from $\text{GPE}_\times$ to $|\text{GPE}|_\pm^2$—assuming the Permanent Anti-Concentration Conjecture. Furthermore, since Lemma 7.2 already reduced $|\text{GPE}|_\times^2$ to $|\text{GPE}|_\pm^2$ assuming the PACC, it suffices for us to reduce $\text{GPE}_\times$ to $|\text{GPE}|_\times^2$ assuming the PACC. Throughout the proof, we will fix an $N \times N$ input matrix $X = (x_{ij}) \in \mathbb{C}^{N\times N}$ which we think of as sampled from the Gaussian distribution $\mathcal{G}^{N\times N}$. Probabilities will always be with respect to $X \sim \mathcal{G}^{N\times N}$. For convenience, we will often assume that "bad events"(i. e., estimates of various quantities outside the desired error bounds) simply do not occur; then, at the end, we will use the union bound to show that the assumption was justified. The $\text{GPE}_\times$ problem can be stated as follows. Given the input $\langle X, 0^{1/\varepsilon}, 0^{1/\delta}\rangle$ for some $\varepsilon, \delta > 0$ output a complex number $z \in \mathbb{C}$ such that

$$|z - \text{Per}(X)| \leq \varepsilon\, |\text{Per}(X)|\,, \tag{7.84}$$

with success probability at least $1-\delta$ over X in time $\text{poly}(N, 1/\varepsilon, 1/\delta)$. Let $\mathcal{O}$ be an oracle that solves $|\text{GPE}|_\times^2$. That is, given an input $\langle A, 0^{1/\varepsilon}, 0^{1/\Delta}\rangle$ where A is an $n \times n$ complex matrix, $\mathcal{O}$ outputs a nonnegative real number $\mathcal{O}\left(\langle A, 0^{1/\varepsilon}, 0^{1/\Delta}\rangle\right)$ such that

$$\Pr_{A\sim\mathcal{G}^{n\times n}}\left[\left|\mathcal{O}\left(\left\langle A, 0^{1/\varepsilon}, 0^{1/\Delta}\right\rangle\right) - |\text{Per}(A)|^2\right| \leq \varepsilon\, |\text{Per}(A)|^2\right] \geq 1 - \Delta. \tag{7.85}$$

Then assuming Conjecture 1.6, we will show how to solve the $\text{GPE}_\times$ instance $\langle X, 0^{1/\varepsilon}, 0^{1/\delta}\rangle$ in time $\text{poly}(N, 1/\varepsilon, 1/\delta)$ with the help of $3N$ nonadaptive queries to $\mathcal{O}$. Let $R = |\text{Per}(X)|^2$. Then by simply calling $\mathcal{O}$ on the input matrix X we can obtain a good approximation $\widetilde{R}$ to R such that (say)

$$\left|\widetilde{R} - R\right| \leq \frac{\varepsilon R}{10}.$$

Therefore, our problem reduces to estimating the *phase* $\theta = \text{Per}(X) / |\text{Per}(X)|$. In other words, we need to give a procedure that returns an approximation $\widetilde{\theta}$ to θ such that (say) $\left|\widetilde{\theta} - \theta\right| \leq 0.9\varepsilon$ and does so with high probability. (Here and throughout, it is understood that all differences between angles are mod 2π.) For all $n \in [N]$ let X_n be the bottom-right $n \times n$ submatrix of X (thus $X_N = X$). A crucial observation is that, since X is a sample from $\mathcal{G}^{N\times N}$ each X_n can be thought of as a sample from $\mathcal{G}^{n\times n}$. As in Theorem 4.3, given a complex number w and a matrix $A = (a_{ij})$ let $A^{[w]}$ be the matrix that is identical to A except that its top-left entry equals $a_{11} - w$ instead of a_{11}. Then for any n and w we can think of the matrix $X_n^{[w]}$ as having been drawn from a distribution $\mathcal{D}_n^{[w]}$ that is identical to $\mathcal{G}^{n\times n}$ except that the top-left entry is distributed according to $\mathcal{N}(-w, 1)_{\mathbb{C}}$ rather than $\mathcal{G}$. Recall that by Lemma 7.4, the variation distance between $\mathcal{D}_n^{[w]}$ and $\mathcal{G}^{n\times n}$ satisfies

$$\left\| \mathcal{D}_n^{[w]} - \mathcal{G}^{n\times n} \right\| \le |w| \,. \tag{7.86}$$

Let $\lambda > 0$ be a parameter to be determined later. Then for each $n \in [N]$ we will be interested in two specific $n \times n$ matrices besides X_n namely $X_n^{[\lambda]}$ and $X_n^{[i\lambda]}$. Similarly to Theorem 4.3, our reduction will be based on the identities

$$\operatorname{Per}\left(X_n^{[\lambda]}\right) = \operatorname{Per}(X_n) - \lambda \operatorname{Per}(X_{n-1}), \tag{7.87}$$

$$\operatorname{Per}\left(X_n^{[i\lambda]}\right) = \operatorname{Per}(X_n) - i\lambda \operatorname{Per}(X_{n-1}) \,. \tag{7.88}$$

More concretely, let

$$R_n := |\operatorname{Per}(X_n)|^2, \tag{7.89}$$

$$\theta_n := \frac{\operatorname{Per}(X_n)}{|\operatorname{Per}(X_n)|}, \tag{7.90}$$

$$S_n := \left|\operatorname{Per}\left(X_n^{[\lambda]}\right)\right|^2 = |\operatorname{Per}(X_n) - \lambda \operatorname{Per}(X_{n-1})|^2, \tag{7.91}$$

$$T_n := \left|\operatorname{Per}\left(X_n^{[i\lambda]}\right)\right|^2 = |\operatorname{Per}(X_n) - i\lambda \operatorname{Per}(X_{n-1})|^2 \,. \tag{7.92}$$

Then some simple algebra—identical to what appeared in Lemma 7.3—yields the identity

$$\theta_n = \theta_{n-1} + \operatorname{sgn}(R_n + R_{n-1} - T_n) \arccos\left(\frac{R_n + R_{n-1} - S_n}{2\sqrt{R_n R_{n-1}}}\right) \tag{7.93}$$

for all $n \ge 2$. "Unravelling" this recursive identity, we obtain a useful formula for $\theta = \theta_N = \frac{\operatorname{Per}(X)}{|\operatorname{Per}(X)|}$:

$$\theta = \frac{x_{NN}}{|x_{NN}|} + \sum_{n=2}^{N} \xi_n \tag{7.94}$$

where

$$\xi_n := \operatorname{sgn}(R_n + R_{n-1} - T_n) \arccos\left(\frac{R_n + R_{n-1} - S_n}{2\sqrt{R_n R_{n-1}}}\right) . \tag{7.95}$$

Our procedure to approximate θ will simply consist of evaluating the above expression for all $n \geq 2$ but using estimates $\widetilde{R}_n, \widetilde{S}_n, \widetilde{T}_n$ produced by the oracle $\mathcal{O}$ in place of the true values R_n, S_n, T_n. In more detail, let $\widetilde{R}_1 := |x_{NN}|^2$, and for all $n \geq 2$ let

$$\widetilde{R}_n := \mathcal{O}\left(\left\langle X_n, 0^{1/\varepsilon}, 0^{1/\Delta}\right\rangle\right), \tag{7.96}$$

$$\widetilde{S}_n := \mathcal{O}\left(\left\langle X_n^{[\lambda]}, 0^{1/\varepsilon}, 0^{1/\Delta}\right\rangle\right), \tag{7.97}$$

$$\widetilde{T}_n := \mathcal{O}\left(\left\langle X_n^{[i\lambda]}, 0^{1/\varepsilon}, 0^{1/\Delta}\right\rangle\right), \tag{7.98}$$

where $\varepsilon, \Delta > 1/\operatorname{poly}(N)$ are parameters to be determined later. Then our procedure for approximating θ is to return

$$\widetilde{\theta} := \frac{x_{NN}}{|x_{NN}|} + \sum_{n=2}^{N} \widetilde{\xi}_n, \tag{7.99}$$

where

$$\widetilde{\xi}_n := \operatorname{sgn}\left(\widetilde{R}_n + \widetilde{R}_{n-1} - \widetilde{T}_n\right) \arccos\left(\frac{\widetilde{R}_n + \widetilde{R}_{n-1} - \widetilde{S}_n}{2\sqrt{\widetilde{R}_n \widetilde{R}_{n-1}}}\right). \tag{7.100}$$

Clearly this procedure runs in polynomial time and makes at most $3N$ nonadaptive calls to $\mathcal{O}$. We now upper-bound the error $\left|\widetilde{\theta} - \theta\right|$ incurred in the approximation. Since

$$\left|\widetilde{\theta} - \theta\right| \leq \sum_{n=2}^{N} \left|\widetilde{\xi}_n - \xi_n\right|, \tag{7.101}$$

it suffices to upper-bound $\left|\widetilde{\xi}_n - \xi_n\right|$ for each n. By the definition of $\mathcal{O}$ for all $n \in [N]$ we have

$$\Pr\left[\left|\widetilde{R}_n - R_n\right| \leq \varepsilon R_n\right] \geq 1 - \Delta, \tag{7.102}$$

$$\Pr\left[\left|\widetilde{S}_n - S_n\right| \leq \varepsilon S_n\right] \geq 1 - \Delta - \left\|\mathcal{D}_n^{[\lambda]} - \mathcal{G}^{n\times n}\right\| \tag{7.103}$$

$$\geq 1 - \Delta - \lambda, \tag{7.104}$$

$$\Pr\left[\left|\widetilde{T}_n - T_n\right| \leq \varepsilon T_n\right] \geq 1 - \Delta - \left\|\mathcal{D}_n^{[i\lambda]} - \mathcal{G}^{n\times n}\right\| \tag{7.105}$$

$$\geq 1 - \Delta - \lambda. \tag{7.106}$$

Also, let $p(n, 1/\beta)$ be a polynomial such that

$$\Pr_{A\sim\mathcal{G}^{n\times n}}\left[|\operatorname{Per}(A)|^2 \geq \frac{n!}{p(n, 1/\beta)}\right] \geq 1 - \beta \tag{7.107}$$

for all n and $\beta > 0$; such a p is guaranteed to exist by Conjecture 1.6. It will later be convenient to assume p is monotone. Then

$$\Pr\left[R_n \geq \frac{n!}{p(n, 1/\beta)}\right] \geq 1 - \beta \tag{7.108}$$

In the other direction, it can be shown without much difficulty that $\mathrm{E}[R_n] = n!$ (see equations (8.5)-(8.8) in the next section). Hence, for all $0 < \kappa < 1$ Markov's inequality gives us

$$\Pr\left[R_n \leq \frac{n!}{\kappa}\right] \geq 1-\kappa\,, \tag{7.109}$$

$$\Pr\left[S_n \leq \frac{n!}{\kappa}\right] \geq 1-\kappa-\lambda\,, \tag{7.110}$$

$$\Pr\left[T_n \leq \frac{n!}{\kappa}\right] \geq 1-\kappa-\lambda\,, \tag{7.111}$$

where we have again used the fact that S_n, T_n are random variables with variation distance at most λ from R_n. Now think of $\beta, \kappa > 1/\operatorname{poly}(N)$ as parameters to be determined later, and suppose that *all seven* of the events listed above hold, for all $n \in [N]$. In that case,

$$\begin{aligned}
\left|\widetilde{R}_n - R_n\right| &\leq \varepsilon R_n && (7.112)\\
&\leq \varepsilon \frac{n!}{\kappa} && (7.113)\\
&= \varepsilon \frac{R_{n-1} n}{\kappa} \frac{(n-1)!}{R_{n-1}} && (7.114)\\
&\leq \varepsilon \frac{R_{n-1} n}{\kappa} p(n-1, 1/\beta) && (7.115)\\
&\leq \varepsilon \frac{R_{n-1} n}{\kappa} p(n, 1/\beta) && (7.116)\\
&= \frac{\varepsilon n \cdot p(n, 1/\beta)}{\kappa \lambda^2} \lambda^2 R_{n-1} && (7.117)
\end{aligned}$$

and likewise

$$\left|\widetilde{S}_n - S_n\right|,\ \left|\widetilde{T}_n - T_n\right| \leq \frac{\varepsilon n \cdot p(n, 1/\beta)}{\kappa \lambda^2} \lambda^2 R_{n-1}\,. \tag{7.118}$$

Plugging the above bounds into Lemma 7.3, we find that, if there are no "bad events," then noisy triangulation returns an estimate $\widetilde{\xi}_n$ of ξ_n such that

$$\begin{aligned}
\left|\widetilde{\xi}_n - \xi_n\right| &\leq 3\sqrt{\frac{\varepsilon n \cdot p(n, 1/\beta)}{\kappa \lambda^2}} \left(\lambda \sqrt{\frac{R_{n-1}}{R_n}} + 1\right) && (7.119)\\
&\leq 3\sqrt{\frac{\varepsilon n \cdot p(n, 1/\beta)}{\kappa \lambda^2}} \left(\lambda \sqrt{\frac{(n-1)!/\kappa}{n!/p(N, 1/\beta)}} + 1\right) && (7.120)\\
&\leq 3\sqrt{\varepsilon} \left(\frac{p(n, 1/\beta)}{\kappa} + \frac{\sqrt{n}\sqrt{p(n, 1/\beta)}}{\lambda \sqrt{\kappa}}\right) && (7.121)\\
&\leq 3\sqrt{\varepsilon} \left(\frac{p(N, 1/\beta)}{\kappa} + \frac{\sqrt{N}\sqrt{p(N, 1/\beta)}}{\lambda \sqrt{\kappa}}\right), && (7.122)
\end{aligned}$$

where line (7.122) uses $n \leq N$ together with the monotonicity of p. We now upper-bound the probability of a bad event. Taking the union bound over all $n \in [N]$ and all seven possible bad events, we find that the total probability that the procedure fails is at most

$$p_{\text{FAIL}} := (3\Delta + 3\kappa + 4\lambda + \beta)\, N\,. \tag{7.123}$$

Thus, let us now make the choices

$$\Delta, \kappa := \frac{\delta}{12N}\,, \quad \lambda := \frac{\delta}{16N}\,, \quad \text{and} \quad \beta := \frac{\delta}{4N}$$

so that $p_{\text{FAIL}} \leq \delta$ as desired. Let us also make the choice

$$\varepsilon := \frac{\epsilon^2 \delta^3}{60000 N^6 p\,(N, 4N/\delta)^2}\,. \tag{7.124}$$

Then

$$\begin{aligned}
\left|\widetilde{\theta} - \theta\right| &\leq \sum_{n=2}^{N} \left|\widetilde{\xi}_n - \xi_n\right| && (7.125)\\
&\leq 3\sqrt{\varepsilon}\left(\frac{12N \cdot p\,(N, 4N/\delta)}{\delta} + \frac{32\sqrt{3}N^2\sqrt{p\,(N, 4N/\delta)}}{\delta^{3/2}}\right) N && (7.126)\\
&\leq \frac{9\epsilon}{10} && (7.127)
\end{aligned}$$

as desired. Furthermore, if none of the bad events happen, then we get "for free" that

$$\left|\widetilde{R} - R\right| = \left|\widetilde{R}_N - R_N\right| \leq \varepsilon R_N \leq \frac{\epsilon R}{10}\,. \tag{7.128}$$

So letting $r := \sqrt{R}$ and $\widetilde{r} := \sqrt{\widetilde{R}}$ by the triangle inequality we have

$$\begin{aligned}
\left|\widetilde{r}e^{i\widetilde{\theta}} - re^{i\theta}\right| &\leq |\widetilde{r} - r| + r\sqrt{2 - 2\cos\left(\widetilde{\theta} - \theta\right)} && (7.129)\\
&\leq \frac{\left|\widetilde{R} - R\right|}{\widetilde{r} + r} + r\left|\widetilde{\theta} - \theta\right| && (7.130)\\
&\leq \frac{\epsilon R}{10r} + r\frac{9\epsilon}{10} && (7.131)\\
&= \epsilon r && (7.132)\\
&= \epsilon\,|\text{Per}\,(X)|\,, && (7.133)
\end{aligned}$$

and hence we have successfully approximated $\text{Per}\,(X) = re^{i\theta}$. □

When combined with Lemmas 7.1 and 7.2, Theorem 1.7 has the corollary that, assuming the Permanent Anti-Concentration Conjecture, the $\text{GPE}_\times$ $|\text{GPE}|^2_\times$ $\text{GPE}_\pm$ and $|\text{GPE}|^2_\pm$ problems are *all* polynomial-time equivalent.

8 The distribution of Gaussian permanents

In this section, we seek an understanding of the distribution over $\operatorname{Per}(X)$ where $X \sim \mathcal{G}^{n\times n}$ is a matrix of i. i. d. Gaussians. Here, recall that $\mathcal{G} = \mathcal{N}(0,1)_{\mathbb{C}}$ is the standard complex normal distribution, though one suspects that most issues would be similar with $\mathcal{N}(0,1)_{\mathbb{R}}$ or possibly even the uniform distribution over $\{-1,1\}$. As explained in Section 1.2.2, the reason why we focus on the complex Gaussian ensemble $\mathcal{G}^{n\times n}$ is simply that, as shown by Theorem 5.1, the Gaussian ensemble arises naturally when we consider truncations of Haar-random unitary matrices.

Our goal is to give evidence in favor of Conjecture 1.6, the *Permanent Anti-Concentration Conjecture* (PACC). This is the conjecture that, if $X \sim \mathcal{G}^{n\times n}$ is Gaussian, then $\operatorname{Per}(X)$ is "not *too* concentrated around 0": a $1-1/\operatorname{poly}(n)$ fraction of its probability mass is greater than $\sqrt{n!}/\operatorname{poly}(n)$ in absolute value, $\sqrt{n!}$ being the standard deviation. More formally, there exists a polynomial p such that for all n and $\delta > 0$,

$$\Pr_{X\sim\mathcal{G}^{n\times n}}\left[|\operatorname{Per}(X)| < \frac{\sqrt{n!}}{p(n,1/\delta)}\right] < \delta\,. \tag{8.1}$$

An equivalent formulation is that there exist constants C, D and $\beta > 0$ such that for all n and $\varepsilon > 0$,

$$\Pr_{X\sim\mathcal{G}^{n\times n}}\left[|\operatorname{Per}(X)| < \varepsilon\sqrt{n!}\right] < Cn^D\varepsilon^\beta\,. \tag{8.2}$$

Conjecture 1.6 has two applications to strengthening the conclusions of this paper. First, it lets us *multiplicatively* estimate $\operatorname{Per}(X)$ (that is, solve the $\mathrm{GPE}_\times$ problem), assuming only that we can *additively* estimate $\operatorname{Per}(X)$ (that is, solve the $\mathrm{GPE}_\pm$ problem). Indeed, if Conjecture 1.6 holds, then as pointed out in Lemma 7.1, additive and multiplicative estimation become equivalent for this problem. Second, as shown by Theorem 1.7, Conjecture 1.6 lets us estimate $\operatorname{Per}(X)$ itself, assuming we can estimate $|\operatorname{Per}(X)|^2$. The bottom line is that, if Conjecture 1.6 holds, then we can base our conclusions about the hardness of approximate BOSONSAMPLING on the natural conjecture that $\mathrm{GPE}_\times$ is #P-hard, rather than the relatively-contrived conjecture that $|\mathrm{GPE}|^2_\pm$ is #P-hard.

At a less formal level, we believe proving Conjecture 1.6 might also provide intuition essential to proving the "bigger" conjecture, that these problems are #P-hard in the first place.

The closest result to Conjecture 1.6 that we know of comes from a 2009 paper of Tao and Vu [65]. These authors show the following:

Theorem 8.1 (Tao-Vu [65])**.** *For all $\varepsilon > 0$ and sufficiently large n,*

$$\Pr_{X\in\{-1,1\}^{n\times n}}\left[|\operatorname{Per}(X)| < \frac{\sqrt{n!}}{n^{\varepsilon n}}\right] < \frac{1}{n^{0.1}}\,. \tag{8.3}$$

Alas, Theorem 8.1 falls short of what we need, since it only upper-bounds the probability that $|\operatorname{Per}(X)| < \sqrt{n!}/n^{\varepsilon n}$ whereas we need to upper-bound the probability that $|\operatorname{Per}(X)| < \sqrt{n!}/\operatorname{poly}(n)$. Two more minor differences between Theorem 8.1 and what we need are the following:

(1) The upper bound in Theorem 8.1 is $1/n^{0.1}$ whereas we need an upper bound of the form $1/p(n)$ for *any* polynomial p.

(2) Theorem 8.1 applies to Bernoulli random matrices, not Gaussian ones.

Fortunately, differences (1) and (2) seem to "cancel each other out": Tao[26] has reported that, if the proof techniques from [65] are applied to the Gaussian case, then one should be able not only to reprove Theorem 8.1, but also to replace the $1/n^{0.1}$ by $1/n^C$ for any constant C.

In the rest of the section, we will give three pieces of evidence for Conjecture 1.6. The first, in Section 8.1, is that it is supported numerically. The second, in Section 8.2, is that the analogous statement holds with the *determinant* instead of the permanent. The proof of this result makes essential use of geometric properties of the determinant, which is why we do not know how to extend it to the permanent. On the other hand, Godsil and Gutman [31] observed that, for all matrices $X = (x_{ij})$,

$$\operatorname{Per}(X) = \mathrm{E}\left[\operatorname{Det}\begin{pmatrix} \pm\sqrt{x_{11}} & \cdots & \pm\sqrt{x_{1n}} \\ \vdots & \ddots & \vdots \\ \pm\sqrt{x_{n1}} & \cdots & \pm\sqrt{x_{nn}} \end{pmatrix}^2\right], \tag{8.4}$$

where the expectation is over all 2^{n^2} ways of assigning $+$'s and $-$'s to the entries. Because of this fact, together with our numerical data, we suspect that the story for the permanent may be similar to that for the determinant. The third piece of evidence is that a weaker form of Conjecture 1.6 holds: basically, $|\operatorname{Per}(X)|$ has at least a $\Omega(1/n)$ probability of being $\Omega\left(\sqrt{n!}\right)$. We prove this by calculating the fourth moment of $\operatorname{Per}(X)$. Unfortunately, extending the calculation to higher moments seems difficult.

Before going further, let us make some elementary remarks about the distribution over $\operatorname{Per}(X)$ for $X \sim \mathcal{G}^{n\times n}$. By symmetry, clearly $\mathrm{E}[\operatorname{Per}(X)] = 0$. The second moment is also easy to calculate:

$$\mathrm{E}\left[|\operatorname{Per}(X)|^2\right] = \mathrm{E}\left[\sum_{\sigma,\tau\in S_n}\prod_{i=1}^n x_{i,\sigma(i)}\overline{x}_{i,\tau(i)}\right] \tag{8.5}$$

$$= \mathrm{E}\left[\sum_{\sigma\in S_n}\prod_{i=1}^n \left|x_{i,\sigma(i)}\right|^2\right] \tag{8.6}$$

$$= \sum_{\sigma\in S_n}\prod_{i=1}^n \mathrm{E}\left[\left|x_{i,\sigma(i)}\right|^2\right] \tag{8.7}$$

$$= n!. \tag{8.8}$$

We will often find it convenient to work with the normalized random variable

$$P_n := \frac{|\operatorname{Per}(X)|^2}{n!}, \tag{8.9}$$

so that $\mathrm{E}[P_n] = 1$.

8.1 Numerical data

Figure 5 shows the numerically computed probability density function of P_n when $n = 6$. For comparison, we have also plotted the pdf of $D_n := |\operatorname{Det}(X)|^2/n!$ which looks essentially indistinguishable.

[26]See http://mathoverflow.net/questions/45822/anti-concentration-bound-for-permanents-of-gaussian-matrices

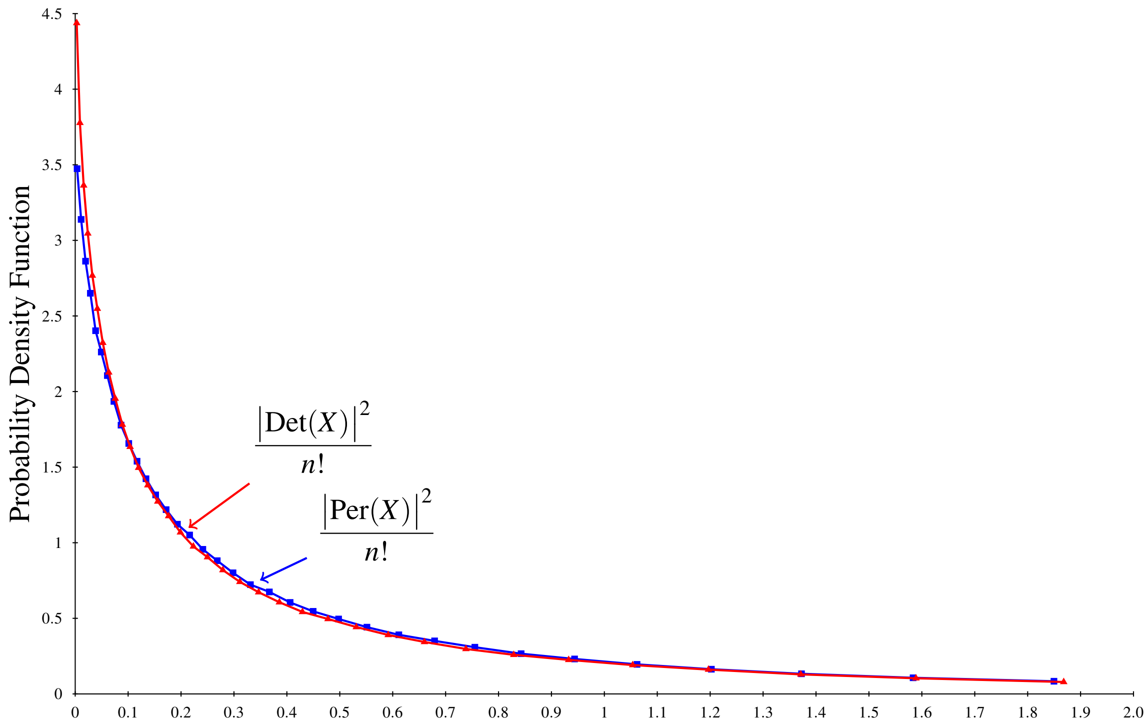

Figure 5: Probability density functions of the random variables $P_n = |\mathrm{Per}(X)|^2/n!$ (blue, squares) and $D_n = |\mathrm{Det}(X)|^2/n!$ (red, triangles) where $X \sim \mathcal{G}^{n\times n}$ is a complex Gaussian random matrix, in the case $n = 6$. Note that $\mathrm{E}[P_n] = \mathrm{E}[D_n] = 1$. As n increases, the bends on the left become steeper. We do not know exactly how the pdfs behave near the origin. This density plot estimate for each of the distributions is produced by generating 10^6 samples and sorting them into 40 equal buckets of $250,000$ points each. So, the first bucket contains the lowest 2.5^{th} percentile of samples, the next bucket the 2.5^{th} through 5^{th} percentile, and so on. The density is then estimated for each bucket as the fraction of points it contains (0.025) divided by its width, and plotted as a point at the bucket's center. In effect, the plot is a histogram, except rather than using equal intervals on the density axis, the buckets are chosen to contain equal numbers of sample points. As a result, near 0 where the distribution is concentrated, the buckets are narrower and more density estimate points are plotted, allowing the limit behavior for the PACC to be seen more clearly. The interested reader can find our raw data among the source files of this paper. Thanks to John Watrous for correcting an error in an earlier version of this figure.

The numerical evidence up to $n = 10$ is strongly consistent with Conjecture 1.6. Indeed, from the data it seems likely that for all $0 \le \beta < 2$ there exist constants C, D such that for all n and $\varepsilon > 0$,

$$\Pr_{X \sim \mathcal{G}^{n\times n}} \left[|\mathrm{Per}(X)| < \varepsilon\sqrt{n!} \right] < Cn^D \varepsilon^\beta , \tag{8.10}$$

and perhaps the above even holds when $\beta = 2$.

8.2 The analogue for determinants

We prove the following theorem, which at least settles Conjecture 1.6 with the determinant in place of the permanent:

Theorem 8.2 (Determinant Anti-Concentration Theorem)**.** *For all $0 \le \beta < 2$, there exists a constant C_β such that for all n and $\varepsilon > 0$,*

$$\Pr_{X \sim \mathcal{G}^{n\times n}} \left[|\mathrm{Det}(X)| < \varepsilon\sqrt{n!} \right] < C_\beta n^{\beta(\beta+2)/8} \varepsilon^\beta . \tag{8.11}$$

We leave as an open problem whether Theorem 8.2 holds when $\beta = 2$.

Compared to the permanent, a lot is known about the determinants of Gaussian matrices. In particular, Girko [30] (see also Costello and Vu [18, Appendix A]) have shown that

$$\frac{\ln|\mathrm{Det}(X)| - \ln\sqrt{(n-1)!}}{\sqrt{\frac{\ln n}{2}}} \tag{8.12}$$

converges weakly to the normal distribution $\mathcal{N}(0,1)_\mathbb{R}$. Unfortunately, weak convergence is not enough to imply Theorem 8.2, so we will have to do some more work. Indeed, we will find that the probability density function of $|\mathrm{Det}(X)|^2$ in the critical regime where $|\mathrm{Det}(X)|^2 \approx 0$ is *different* than one might guess from the above formula.

The key fact about $\mathrm{Det}(X)$ that we will use is that we can compute its moments *exactly*—even the fractional and inverse moments. To do so, we use the following beautiful characterization, which can be found (for example) in Costello and Vu [18].

Lemma 8.3 ([18])**.** *Let $X \sim \mathcal{G}^{n\times n}$ be a complex Gaussian random matrix. Then $|\mathrm{Det}(X)|^2$ has the same distribution as*

$$\prod_{i=1}^{n} \left(\sum_{j=1}^{i} |\xi_{ij}|^2 \right) \tag{8.13}$$

where the ξ_{ij}'s are independent $\mathcal{N}(0,1)_\mathbb{C}$ Gaussians. (In other words, $|\mathrm{Det}(X)|^2$ is distributed as $T_1 \cdots T_n$, where each T_k is an independent χ^2 random variable with k degrees of freedom.)

The proof of Lemma 8.3 (which we omit) uses the interpretation of the determinant as the volume of a parallelepiped, together with the spherical symmetry of the Gaussian distribution.

As with the permanent, it will be convenient to work with the normalized random variable

$$D_n := \frac{|\mathrm{Det}(X)|^2}{n!}, \tag{8.14}$$

so that $\mathrm{E}[D_n] = 1$. Using Lemma 8.3, we now calculate the moments of D_n.

Lemma 8.4. *For all real numbers $\alpha > -1$,*

$$\mathrm{E}[D_n^\alpha] = \frac{1}{(n!)^\alpha} \prod_{k=1}^{n} \frac{\Gamma(k+\alpha)}{\Gamma(k)}. \tag{8.15}$$

(If $\alpha \leq -1$ then $\mathrm{E}[D_n^\alpha] = \infty$.)

Proof. By Lemma 8.3,

$$\mathrm{E}[D_n^\alpha] = \frac{1}{(n!)^\alpha} \mathrm{E}[T_1^\alpha \cdots T_n^\alpha] \tag{8.16}$$

$$= \frac{1}{(n!)^\alpha} \prod_{k=1}^{n} \mathrm{E}[T_k^\alpha], \tag{8.17}$$

where each T_k is an independent χ^2 random variable with k degrees of freedom. Now, T_k has probability density function

$$f(x) = \frac{e^{-x} x^{k-1}}{\Gamma(k)} \tag{8.18}$$

for $x \geq 0$. So

$$\mathrm{E}[T_k^\alpha] = \frac{1}{\Gamma(k)} \int_0^\infty e^{-x} x^{k+\alpha-1} dx \tag{8.19}$$

$$= \frac{\Gamma(k+\alpha)}{\Gamma(k)} \tag{8.20}$$

as long as $k+\alpha > 0$. (If $k+\alpha \leq 0$ as can happen if $\alpha \leq -1$, then the above integral diverges.) □

As a sample application of Lemma 8.4, if α is a positive integer then we get

$$\mathrm{E}[D_n^\alpha] = \prod_{i=1}^{\alpha-1} \binom{n+i}{i} = \Theta\left(n^{\alpha(\alpha-1)/2}\right). \tag{8.21}$$

For our application, though, we are interested in the dependence of $\mathrm{E}[D_n^\alpha]$ on n when α is *not* necessarily a positive integer. The next lemma shows that the asymptotic behavior above generalizes to negative and fractional α.

Lemma 8.5. *For all real numbers $\alpha > -1$ there exists a positive constant C_α such that*

$$\lim_{n\to\infty} \frac{\mathrm{E}[D_n^\alpha]}{n^{\alpha(\alpha-1)/2}} = C_\alpha. \tag{8.22}$$

Proof. Let us write

$$\mathrm{E}[D_n^\alpha] = \frac{\Gamma(1+\alpha)}{n^\alpha} \prod_{k=1}^{n-1} \frac{\Gamma(k+\alpha+1)}{k^\alpha \Gamma(k+1)}. \tag{8.23}$$

Then by Stirling's approximation,

$$\ln \prod_{k=1}^{n-1} \frac{\Gamma(k+\alpha+1)}{k^{\alpha}\Gamma(k+1)} = \sum_{k=1}^{n-1} \left(\ln \frac{\Gamma(k+\alpha+1)}{\Gamma(k+1)} - \alpha \ln k \right) \tag{8.24}$$

$$= H_{\alpha} + o(1) + \sum_{k=1}^{n-1} \left(\ln \left(\frac{\sqrt{2\pi(k+\alpha)} \left(\frac{k+\alpha}{e}\right)^{k+\alpha}}{\sqrt{2\pi k} \left(\frac{k}{e}\right)^{k}} \right) - \alpha \ln k \right) \tag{8.25}$$

$$= H_{\alpha} + o(1) + \sum_{k=1}^{n-1} \left(\left(k + \alpha + \frac{1}{2} \right) \ln \left(\frac{k+\alpha}{k} \right) - \alpha \right) \tag{8.26}$$

$$= H_{\alpha} + J_{\alpha} + o(1) + \sum_{k=1}^{n-1} \left(\left(k + \alpha + \frac{1}{2} \right) \left(\frac{\alpha}{k} - \frac{\alpha^2}{2k^2} \right) - \alpha \right) \tag{8.27}$$

$$= H_{\alpha} + J_{\alpha} + o(1) + \sum_{k=1}^{n-1} \left(\frac{\alpha(\alpha+1)}{2k} - \frac{\alpha^2(2\alpha+1)}{4k^2} \right) \tag{8.28}$$

$$= H_{\alpha} + J_{\alpha} + L_{\alpha} + o(1) + \frac{\alpha(\alpha+1)}{2} \ln n \,. \tag{8.29}$$

In the above, H_{α} J_{α} and L_{α} are finite error terms that depend only on α (and not n):

$$H_{\alpha} = \sum_{k=1}^{\infty} \ln \left(\frac{\Gamma(k+\alpha+1)}{\Gamma(k+1)} \frac{\sqrt{k}\left(\frac{k}{e}\right)^{k}}{\sqrt{k+\alpha}\left(\frac{k+\alpha}{e}\right)^{k+\alpha}} \right), \tag{8.30}$$

$$J_{\alpha} = \sum_{k=1}^{\infty} \left(k + \alpha + \frac{1}{2} \right) \left(\ln \left(\frac{k+\alpha}{k} \right) - \left(\frac{\alpha}{k} - \frac{\alpha^2}{2k^2} \right) \right), \tag{8.31}$$

$$L_{\alpha} = \frac{\alpha(\alpha+1)}{2} \left(\lim_{n\to\infty} \sum_{k=1}^{\infty} \frac{1}{k} - \ln n \right) - \sum_{k=1}^{\infty} \frac{\alpha^2(2\alpha+1)}{4k^2} \tag{8.32}$$

$$= \frac{\alpha(\alpha+1)\gamma}{2} - \frac{\alpha^2(2\alpha+1)\pi^2}{24k^2}, \tag{8.33}$$

where $\gamma \approx 0.577$ is the Euler-Mascheroni constant. The $o(1)$'s represent additional error terms that go to 0 as $n \to \infty$. Hence

$$\prod_{k=1}^{n-1} \frac{\Gamma(k+\alpha+1)}{k^{\alpha}\Gamma(k+1)} = e^{H_{\alpha}+J_{\alpha}+L_{\alpha}+o(1)} n^{\alpha(\alpha+1)/2} \tag{8.34}$$

and

$$\lim_{n\to\infty} \frac{\mathrm{E}[D_n^{\alpha}]}{n^{\alpha(\alpha-1)/2}} = \lim_{n\to\infty} \left(\frac{1}{n^{\alpha(\alpha-1)/2}} \cdot \frac{\Gamma(1+\alpha)}{n^{\alpha}} e^{H_{\alpha}+J_{\alpha}+L_{\alpha}+o(1)} n^{\alpha(\alpha+1)/2} \right) \tag{8.35}$$

$$= \Gamma(1+\alpha) e^{H_{\alpha}+J_{\alpha}+L_{\alpha}}, \tag{8.36}$$

which is a positive constant C_{α} depending on α. □

We can now complete the proof of Theorem 8.2.

Proof of Theorem 8.2. Let $\alpha := -\beta/2$. Then by Markov's inequality, for all $\varepsilon > 0$ we have

$$\mathrm{E}\left[D_n^{\alpha}\right] = \mathrm{E}\left[\left(\frac{\sqrt{n!}}{|\mathrm{Det}(X)|}\right)^{\beta}\right] \tag{8.37}$$

$$\geq \Pr_{X\sim\mathcal{G}^{n\times n}}\left[|\mathrm{Det}(X)| < \varepsilon\sqrt{n!}\right]\cdot\frac{1}{\varepsilon^{\beta}}. \tag{8.38}$$

Hence

$$\Pr_{X\sim\mathcal{G}^{n\times n}}\left[|\mathrm{Det}(X)| < \varepsilon\sqrt{n!}\right] \leq \mathrm{E}\left[D_n^{\alpha}\right]\cdot\varepsilon^{\beta} \tag{8.39}$$

$$< C_{\alpha} n^{\alpha(\alpha-1)/2}\varepsilon^{\beta} \tag{8.40}$$

$$= C'_{\beta} n^{\beta(\beta+2)/8}\varepsilon^{\beta} \tag{8.41}$$

for some positive constants C_{α}, C'_{β} depending only on α and β respectively. □

8.3 Weak version of the PACC

We prove the following theorem about concentration of Gaussian permanents.

Theorem 8.6 (Weak Anti-Concentration of the Permanent)**.** *For all $\alpha < 1$,*

$$\Pr_{X\sim\mathcal{G}^{n\times n}}\left[|\mathrm{Per}(X)|^2 \geq \alpha\cdot n!\right] > \frac{(1-\alpha)^2}{n+1}. \tag{8.42}$$

While Theorem 8.6 falls short of proving Conjecture 1.6, it at least shows that $|\mathrm{Per}(X)|$ has a *non-negligible* probability of being large enough for our application when X is a Gaussian random matrix. In other words, it rules out the possibility that $|\mathrm{Per}(X)|$ is almost always tiny compared to its expected value, and that only for (say) a $1/\exp(n)$ fraction of matrices X does $|\mathrm{Per}(X)|$ become enormous.

Recall that P_n denotes the random variable $|\mathrm{Per}(X)|^2/n!$ and that $\mathrm{E}[P_n] = 1$. Our proof of Theorem 8.6 will proceed by showing that $\mathrm{E}\left[P_n^2\right] = n+1$. As we will see later, it is almost an "accident" that this is true—$\mathrm{E}\left[P_n^3\right]$ $\mathrm{E}\left[P_n^4\right]$ and so on all grow exponentially with n—but it is enough to imply Theorem 8.6.

To calculate $\mathrm{E}\left[P_n^2\right]$ we first need a proposition about the number of cycles in a random permutation, which can be found in Lange [45, p. 76] for example, though we prove it for completeness. Given a permutation $\sigma \in S_n$ let $\mathrm{cyc}(\sigma)$ be the number of cycles in σ.

Proposition 8.7. *For any constant $c \geq 1$,*

$$\mathop{\mathrm{E}}_{\sigma\in S_n}\left[c^{\mathrm{cyc}(\sigma)}\right] = \binom{n+c-1}{c-1}. \tag{8.43}$$

Proof. Assume for simplicity that c is a positive integer. Define a *c-colored permutation* (on n elements) to be a permutation $\sigma \in S_n$ in which every cycle is colored one of c possible colors. Then clearly the number of c-colored permutations equals

$$f(n) := \sum_{\sigma\in S_n} c^{\mathrm{cyc}(\sigma)}. \tag{8.44}$$

Now consider forming a c-colored permutation σ. There are n possible choices for $\sigma(1)$. If $\sigma(1) = 1$ then we have completed a cycle of length 1 and there are c possible colors for that cycle. Therefore the number of c-colored permutations σ such that $\sigma(1) = 1$ is $c \cdot f(n-1)$. On the other hand, if $\sigma(1) = b$ for some $b \neq 1$ then we can treat the pair $(1, b)$ as though it were a single element, with an incoming edge to 1 and an outgoing edge from b. Therefore the number of c-colored permutations σ such that $\sigma(1) = b$ is $f(n-1)$. Combining, we obtain the recurrence relation

$$f(n) = c \cdot f(n-1) + (n-1) f(n-1) \tag{8.45}$$

$$= (n+c-1) f(n-1)\,. \tag{8.46}$$

Together with the base case $f(0) = 1$ this implies that

$$f(n) = (n+c-1)(n+c-2) \cdot \cdots \cdot c \tag{8.47}$$

$$= \binom{n+c-1}{c-1} \cdot n!\,. \tag{8.48}$$

Hence

$$\operatorname*{E}_{\sigma \in S_n}\left[c^{\operatorname{cyc}(\sigma)}\right] = \frac{f(n)}{n!} = \binom{n+c-1}{c-1}. \tag{8.49}$$

The above argument can be generalized to non-integer c using standard tricks (though we will not need that in the paper). □

We can now compute $\operatorname{E}\left[P_n^2\right]$.

Lemma 8.8. $\operatorname{E}\left[P_n^2\right] = n+1$.

Proof. We have

$$\operatorname{E}\left[P_n^2\right] = \frac{1}{(n!)^2} \operatorname*{E}_{X \sim \mathcal{G}^{n \times n}}\left[\operatorname{Per}(X)^2\, \overline{\operatorname{Per}(X)}^2\right] \tag{8.50}$$

$$= \frac{1}{(n!)^2} \operatorname*{E}_{X \sim \mathcal{G}^{n \times n}}\left[\sum_{\sigma,\tau,\alpha,\beta \in S_n} \prod_{i=1}^{n} x_{i,\sigma(i)} x_{i,\tau(i)} \overline{x}_{i,\alpha(i)} \overline{x}_{i,\beta(i)}\right] \tag{8.51}$$

$$= \frac{1}{(n!)^2} \sum_{\sigma,\tau,\alpha,\beta \in S_n} M(\sigma,\tau,\alpha,\beta) \tag{8.52}$$

where

$$M(\sigma,\tau,\alpha,\beta) := \operatorname*{E}_{X \sim \mathcal{G}^{n \times n}}\left[\prod_{i=1}^{n} x_{i,\sigma(i)} x_{i,\tau(i)} \overline{x}_{i,\alpha(i)} \overline{x}_{i,\beta(i)}\right] \tag{8.53}$$

$$= \prod_{i=1}^{n} \operatorname*{E}_{X \sim \mathcal{G}^{n \times n}}\left[x_{i,\sigma(i)} x_{i,\tau(i)} \overline{x}_{i,\alpha(i)} \overline{x}_{i,\beta(i)}\right], \tag{8.54}$$

line (8.54) following from the independence of the Gaussian variables x_{ij}. We now evaluate $M(\sigma,\tau,\alpha,\beta)$. Write $\sigma\cup\tau=\alpha\cup\beta$ if

$$\{(1,\sigma(1)),(1,\tau(1)),\ldots,(n,\sigma(n)),(n,\tau(n))\}=\{(1,\alpha(1)),(1,\beta(1))\ldots,(n,\alpha(n)),(n,\beta(n))\}\,. \tag{8.55}$$

If $\sigma\cup\tau\neq\alpha\cup\beta$ then we claim that $M(\sigma,\tau,\alpha,\beta)=0$. This is because the Gaussian distribution is uniform over phases—so if there exists an x_{ij} that is not "paired" with its complex conjugate $\overline{x}_{ij}$ (or vice versa), then the variations in that x_{ij} will cause the entire product to equal 0. So suppose instead that $\sigma\cup\tau=\alpha\cup\beta$. Then for each $i\in[n]$ in the product, there are two cases. First, if $\sigma(i)\neq\tau(i)$ then

$$\begin{aligned}
\operatorname*{E}_{X\sim\mathcal{G}^{n\times n}}\left[x_{i,\sigma(i)}x_{i,\tau(i)}\overline{x}_{i,\alpha(i)}\overline{x}_{i,\beta(i)}\right] &= \operatorname*{E}_{X\sim\mathcal{G}^{n\times n}}\left[\left|x_{i,\sigma(i)}\right|^2\left|x_{i,\tau(i)}\right|^2\right] &(8.56)\\
&= \operatorname*{E}_{X\sim\mathcal{G}^{n\times n}}\left[\left|x_{i,\sigma(i)}\right|^2\right]\operatorname*{E}_{X\sim\mathcal{G}^{n\times n}}\left[\left|x_{i,\tau(i)}\right|^2\right] &(8.57)\\
&= 1\,. &(8.58)
\end{aligned}$$

Second, if $\sigma(i)=\tau(i)$ then

$$\operatorname*{E}_{X\sim\mathcal{G}^{n\times n}}\left[x_{i,\sigma(i)}x_{i,\tau(i)}\overline{x}_{i,\alpha(i)}\overline{x}_{i,\beta(i)}\right]=\operatorname*{E}_{X\sim\mathcal{G}^{n\times n}}\left[\left|x_{i,\sigma(i)}\right|^4\right]=2\,. \tag{8.59}$$

The result is that $M(\sigma,\tau,\alpha,\beta)=2^{K(\sigma,\tau)}$ where $K(\sigma,\tau)$ is the number of i's such that $\sigma(i)=\tau(i)$. Now let $N(\sigma,\tau)$ be the number of pairs $\alpha,\beta\in S_n$ such that $\sigma\cup\tau=\alpha\cup\beta$. Then

$$\begin{aligned}
\operatorname{E}\left[P_n^4\right] &= \frac{1}{(n!)^2}\sum_{\sigma,\tau,\alpha,\beta\in S_n}M(\sigma,\tau,\alpha,\beta) &(8.60)\\
&= \frac{1}{(n!)^2}\sum_{\sigma,\tau\in S_n}2^{K(\sigma,\tau)}N(\sigma,\tau) &(8.61)\\
&= \operatorname*{E}_{\sigma,\tau\in S_n}\left[2^{K(\sigma,\tau)}N(\sigma,\tau)\right] &(8.62)\\
&= \operatorname*{E}_{\sigma,\tau\in S_n}\left[2^{K(\sigma^{-1}\sigma,\sigma^{-1}\tau)}N\left(\sigma^{-1}\sigma,\sigma^{-1}\tau\right)\right] &(8.63)\\
&= \operatorname*{E}_{\xi\in S_n}\left[2^{K(e,\xi)}N(e,\xi)\right], &(8.64)
\end{aligned}$$

where e denotes the identity permutation. Here line (8.63) follows from symmetry—specifically, from the easily-checked identities $K(\sigma,\tau)=K(\alpha\sigma,\alpha\tau)$ and $N(\sigma,\tau)=N(\alpha\sigma,\alpha\tau)$. We claim that the quantity $2^{K(e,\xi)}N(e,\xi)$ has a simple combinatorial interpretation as $2^{\operatorname{cyc}(\xi)}$, where $\operatorname{cyc}(\xi)$ is the number of cycles in ξ. To see this, consider a bipartite multigraph G with n vertices on each side, and an edge from left-vertex i to right-vertex j if $i=j$ or $\xi(i)=j$ (or a double-edge from i to j if $i=j$ *and* $\xi(i)=j$). Then since e and ξ are both permutations, G is a disjoint union of cycles. By definition, $K(e,\xi)$ equals the number of indices i such that $\xi(i)=i$—which is simply the number of double-edges in G, or equivalently, the number of cycles in ξ of length 1. Also, $N(e,\xi)$ equals the number of ways to partition the edges of G into two perfect matchings, corresponding to α and β respectively. In partitioning G the only freedom we have is that each cycle in G of length at least 4 can be decomposed in two inequivalent ways. This implies that $N(e,\xi) = 2^{L(\xi)}$ where $L(\xi)$ is the number of cycles in ξ of length at least 2 (note that a cycle in ξ of length k gives rise to a cycle in G of length $2k$). Combining,

$$2^{K(e,\xi)} N(e,\xi) = 2^{K(e,\xi)+L(\xi)} = 2^{\mathrm{cyc}(\xi)}. \tag{8.65}$$

Hence

$$\mathrm{E}\left[P_n^2\right] = \mathop{\mathrm{E}}_{\xi \in S_n}\left[2^{\mathrm{cyc}(\xi)}\right] = n+1 \tag{8.66}$$

by Proposition 8.7. □

Using Lemma 8.8, we can now complete the proof of Theorem 8.6, that $\Pr[P_n \geq \alpha] > \frac{(1-\alpha)^2}{n+1}$.

Proof of Theorem 8.6. Let F denote the event that $P_n \geq \alpha$ and let $\delta := \Pr[F]$. Then

$$\begin{aligned} 1 &= \mathrm{E}[P_n] & (8.67)\\ &= \Pr[F]\,\mathrm{E}[P_n \mid F] + \Pr\left[\overline{F}\right]\mathrm{E}\left[P_n \mid \overline{F}\right] & (8.68)\\ &< \delta\,\mathrm{E}[P_n \mid F] + \alpha, & (8.69)\end{aligned}$$

so

$$\mathrm{E}[P_n \mid F] > \frac{1-\alpha}{\delta}. \tag{8.70}$$

By Cauchy-Schwarz, this implies

$$\mathrm{E}\left[P_n^2 \mid F\right] > \frac{(1-\alpha)^2}{\delta^2} \tag{8.71}$$

and hence

$$\begin{aligned} \mathrm{E}\left[P_n^2\right] &= \Pr[F]\,\mathrm{E}\left[P_n^2 \mid F\right] + \Pr\left[\overline{F}\right]\mathrm{E}\left[P_n^2 \mid \overline{F}\right] & (8.72)\\ &> \delta \cdot \frac{(1-\alpha)^2}{\delta^2} + 0 & (8.73)\\ &= \frac{(1-\alpha)^2}{\delta}. & (8.74)\end{aligned}$$

Now, we know from Lemma 8.8 that $\mathrm{E}\left[P_n^2\right] = n+1$. Rearranging, this means that

$$\delta > \frac{(1-\alpha)^2}{n+1} \tag{8.75}$$

which is what we wanted to show. □

A natural approach to proving Conjecture 1.6 would be to calculate the *higher* moments of P_n—$\mathrm{E}\left[P_n^3\right]$ $\mathrm{E}\left[P_n^4\right]$ and so on—by generalizing Lemma 8.8. In principle, these moments would determine the probability density function of P_n completely.

When we do so, here is what we find. Given a bipartite k-regular multigraph G with n vertices on each side, let $M(G)$ be the number of ways to decompose G into an ordered list of k disjoint perfect matchings. Also, let M_k be the expectation of $M(G)$ over a k-regular bipartite multigraph G chosen uniformly at random. Then the proof of Lemma 8.8 extends to show the following:

Theorem 8.9. $\mathrm{E}\left[P_n^k\right] = M_k$ *for all positive integers* k.

However, while $M_1 = 1$ and $M_2 = n+1$ it is also known that $M_k \sim (k/e)^n$ for all $k \geq 3$: this follows from the *van der Waerden conjecture*, which was proved by Falikman [23] and Egorychev [22] in 1981. In other words, the higher moments of P_n grow exponentially with n. Because of this, it seems one would need to know the higher moments extremely precisely in order to conclude anything about the quantities of interest, such as $\Pr[P_n < \alpha]$.

9 The hardness of Gaussian permanents

In this section, we move on to discuss Conjecture 1.5, which says that $\mathrm{GPE}_\times$—the problem of multiplicatively estimating $\mathrm{Per}(X)$ where $X \sim \mathcal{G}^{n\times n}$ is a Gaussian random matrix—is #P-hard. Proving Conjecture 1.5 is the central theoretical challenge that we leave.[27]

Intuitively, Conjecture 1.5 implies that if $\mathsf{P}^{\#\mathsf{P}} \neq \mathsf{BPP}$ then no algorithm for $\mathrm{GPE}_\times$ can run in time $\mathrm{poly}(n, 1/\varepsilon, 1/\delta)$. Though it will not be needed for this work, one could also consider a stronger conjecture, which would say that if $\mathsf{P}^{\#\mathsf{P}} \neq \mathsf{BPP}$ then no algorithm for $\mathrm{GPE}_\times$ can run in time $n^{f(\varepsilon,\delta)}$ for any function f.

In contrast to the case of the Permanent Anti-Concentration Conjecture, the question arises of why one should even expect Conjecture 1.5 to be true. Undoubtedly the main reason is that the analogous statement for *permanents over finite fields* is true: this is the *random self-reducibility of the permanent*, first proved by Lipton [47]. Thus, we are "merely" asking for the real or complex analogue of something already known in the finite field case.

A second piece of evidence for Conjecture 1.5 is that, if $X \sim \mathcal{G}^{n\times n}$ is a Gaussian matrix, then all known approximation algorithms fail to find any reasonable approximation to $\mathrm{Per}(X)$. If X were a *nonnegative* matrix, then we could use the celebrated approximation algorithm of Jerrum, Sinclair, and Vigoda [37]—but since X has negative and complex entries, it is not even clear how to estimate $\mathrm{Per}(X)$ in $\mathsf{BPP}^{\mathsf{NP}}$ let alone in BPP. Perhaps the most relevant approximation algorithms are those of Gurvits [33], which we discuss in Appendix B. In particular, Theorem B.4 will give a randomized algorithm due to Gurvits that approximates $\mathrm{Per}(X)$ to within an additive error $\pm\varepsilon \|X\|^n$ in $O\left(n^2/\varepsilon^2\right)$ time. For a Gaussian matrix $X \sim \mathcal{G}^{n\times n}$ it is known that $\|X\| \approx 2\sqrt{n}$ almost surely, as a consequence of the *Tracy-Widom law*.[28] So in $O\left(n^2/\varepsilon^2\right)$ time, we can approximate $\mathrm{Per}(X)$ to within additive error $\pm\varepsilon\,(2\sqrt{n})^n$. However, this is larger than what we need (namely $\pm\varepsilon\sqrt{n!}/\mathrm{poly}(n)$) by a $\sim (2\sqrt{e})^n$ factor.

A third piece of evidence for Conjecture 1.5 was recently provided by Arora et al. [8], motivated by an earlier version of this paper. They show that the $\mathrm{GPE}_\pm$ problem is *self-checkable*—in the sense that one can decide, in randomized polynomial time, whether a given oracle solves $\mathrm{GPE}_\pm$ or is far from solving it. Since #P-complete problems are well-known to be self-checkable, this provides a useful "sanity check," showing that $\mathrm{GPE}_\pm$ shares at least *one* important property with #P-complete problems. Of course, assuming the Permanent Anti-Concentration Conjecture, their result would apply to $\mathrm{GPE}_\times$ as well.

[27] Though note that, for our BOSONSAMPLING hardness argument to work, all we *really* need is that estimating $\mathrm{Per}(X)$ for Gaussian X is not in the class $\mathsf{BPP}^{\mathsf{NP}}$ and one could imagine giving evidence for this that fell short of #P-hardness.

[28] See http://terrytao.wordpress.com/2010/01/09/254a-notes-3-the-operator-norm-of-a-random-matrix/ for an accessible overview.

In the rest of this section, we discuss the prospects for proving Conjecture 1.5. First, in Section 9.1, we at least show that *exactly* computing $\operatorname{Per}(X)$ for a Gaussian random matrix $X \sim \mathcal{G}^{n\times n}$ is #P-hard. The proof is a simple extension of the classic result of Lipton [47], that the permanent over *finite fields* is "random self-reducible": that is, as hard to compute on average as it is in the worst case. As in Lipton's proof, we use the facts that (1) the permanent is a low-degree polynomial, and (2) low-degree polynomials constitute excellent error-correcting codes. However, in Section 9.2, we then explain why *any extension of this result to show average-case hardness of approximating* $\operatorname{Per}(X)$ *will require a fundamentally new approach.* In other words, the "polynomial reconstruction paradigm" cannot suffice, on its own, to prove Conjecture 1.5.

9.1 Evidence that $\mathrm{GPE}_\times$ is #P-hard

We already saw, in Theorem 4.3, that approximating the permanent (or even the magnitude of the permanent) of *all* matrices $X \in \mathbb{C}^{n\times n}$ is a #P-hard problem. But what about the "opposite" problem: *exactly* computing the permanent of *most* matrices $X \sim \mathcal{G}^{n\times n}$? In this section, we will show that the latter problem is #P-hard as well. This means that, if we want to prove the Permanent-of-Gaussians Conjecture, then the difficulty really is just to *combine* approximation with an average-case assumption.

Our result will be an adaptation of a famous result on the random-self-reducibility of the permanent over *finite* fields:

Theorem 9.1 (Random-Self-Reducibility of the PERMANENT [47, 28, 29, 74])**.** *For all $\alpha \geq 1/\operatorname{poly}(n)$ and primes $p > (3n/\alpha)^2$ the following problem is #P-hard: given a uniform random matrix $M \in \mathbb{F}_p^{n\times n}$ output $\operatorname{Per}(M)$ with probability at least α over M.*

The proof of Theorem 9.1 proceeds by reduction: suppose we had an oracle $\mathcal{O}$ such that

$$\Pr_{M\in\mathbb{F}_p^{n\times n}}[\mathcal{O}(M) = \operatorname{Per}(M)] \geq \alpha\,. \tag{9.1}$$

Using $\mathcal{O}$ we give a randomized algorithm that computes the permanent of an *arbitrary* matrix $X \in \mathbb{F}_p^{n\times n}$. The latter is certainly a #P-hard problem, which implies that computing $\operatorname{Per}(M)$ for even an α fraction of M's must have been #P-hard as well.

There are actually *four* variants of Theorem 9.1, which handle increasingly small values of α. All four are based on the same idea—namely, reconstructing a low-degree polynomial from noisy samples—but as α gets smaller, one has to use more and more sophisticated reconstruction methods. For convenience, we have summarized the variants in the table below.

Success probability α	Reconstruction method	Curve in $\mathbb{F}^{n\times n}$	Reference
$1-\frac{1}{3n}$	Lagrange interpolation	Linear	Lipton [47]
$\frac{3}{4}+\frac{1}{\operatorname{poly}(n)}$	Berlekamp-Welch	Linear	Gemmell et al. [28]
$\frac{1}{2}+\frac{1}{\operatorname{poly}(n)}$	Berlekamp-Welch	Polynomial	Gemmell-Sudan [29]
$\frac{1}{\operatorname{poly}(n)}$	Sudan's list decoding [64]	Polynomial	Cai et al. [74]

In adapting Theorem 9.1 to matrices over $\mathbb{C}$ we face a choice of which variant to prove. For simplicity, we have chosen to prove only the $\alpha = \frac{3}{4} + 1/\operatorname{poly}(n)$ variant in this paper. However, we believe that it should be possible to adapt the $\alpha = 1/2 + 1/\mathrm{poly}(n)$ and $\alpha = 1/\mathrm{poly}(n)$ variants to the complex case as well; we leave this as a problem for future work.

Let us start by explaining how the reduction works in the finite field case, when $\alpha = 3/4 + \delta$ for some $\delta = 1/\mathrm{poly}(n)$. Assume we are given as input a matrix $X \in \mathbb{F}_p^{n\times n}$ where $p \geq n/\delta$ is a prime. We are also given an oracle $\mathcal{O}$ such that

$$\Pr_{M \in \mathbb{F}_p^{n\times n}} [\mathcal{O}(M) = \mathrm{Per}(M)] \geq \frac{3}{4} + \delta\,. \tag{9.2}$$

Then using $\mathcal{O}$ our goal is to compute $\mathrm{Per}(X)$.

We do so using the following algorithm. First choose another matrix $Y \in \mathbb{F}_p^{n\times n}$ uniformly at random. Then set

$$X(t) := X + tY\,, \tag{9.3}$$

$$q(t) := \mathrm{Per}(X(t))\,. \tag{9.4}$$

Notice that $q(t)$ is a univariate polynomial in t of degree at most n. Furthermore, $q(0) = \mathrm{Per}(X(0)) = \mathrm{Per}(X)$ whereas for each $t \neq 0$ the matrix $X(t)$ is uniformly random. So by assumption, for each $t \neq 0$ we have

$$\Pr[\mathcal{O}(X(t)) = q(t)] \geq \frac{3}{4} + \delta\,. \tag{9.5}$$

Let S be the set of all nonzero t such that $\mathcal{O}(X(t)) = q(t)$. Then by Markov's inequality,

$$\Pr\left[|S| \geq \left(\frac{1}{2} + \delta\right)(p-1)\right] \geq 1 - \frac{\frac{1}{4} - \delta}{\frac{1}{2} - \delta} \geq \frac{1}{2} + \delta\,. \tag{9.6}$$

So if we can just compute $\mathrm{Per}(X)$ in the case where $|S| \geq (1/2 + \delta)(p-1)$ then all we need to do is run our algorithm $O(1/\delta^2)$ times (with different choices of the matrix Y), and output the majority result.

So the problem reduces to the following: reconstruct a univariate polynomial $q : \mathbb{F}_p \to \mathbb{F}_p$ of degree n given "sample data" $\mathcal{O}(X(1)), \ldots, \mathcal{O}(X(p-1))$ that satisfies $q(t) = \mathcal{O}(X(t))$ for at least a $1/2 + \delta$ fraction of t's. Fortunately, we can solve that problem efficiently using the well-known *Berlekamp-Welch algorithm*:

Theorem 9.2 (Berlekamp-Welch Algorithm)**.** *Let q be a univariate polynomial of degree d over any field $\mathbb{F}$. Suppose we are given m pairs of $\mathbb{F}$-elements $(x_1, y_1), \ldots, (x_m, y_m)$ (with the x_i's all distinct), and are promised that $y_i = q(x_i)$ for more than $(m+d)/2$ values of i. Then there is a deterministic algorithm to reconstruct q using* $\mathrm{poly}(n,d)$ *field operations.*

Theorem 9.2 applies to our scenario provided p is large enough (say, at least n/δ). Once we have the polynomial q we then simply evaluate it at 0 to obtain $q(0) = \mathrm{Per}(X)$.

The above argument shows that it is #P-hard to compute the permanent of a "random" matrix—but only over a sufficiently-large *finite* field $\mathbb{F}$ and with respect to the uniform distribution over matrices. By contrast, what if $\mathbb{F}$ is the field of complex numbers, and the distribution over matrices is the Gaussian distribution, $\mathcal{G}^{n\times n}$?

In that case, one can check that the entire argument still goes through, *except* for the part where we asserted that the matrix $X(t)$ was uniformly random. In the Gaussian case, it is easy enough to arrange that $X(t) \sim \mathcal{G}^{n\times n}$ for some *fixed* $t \neq 0$, but we can no longer ensure that $X(t) \sim \mathcal{G}^{n\times n}$ for *all* $t \neq 0$ simultaneously. Indeed, $X(t)$ becomes arbitrarily close to the input matrix $X(0) = X$ as $t \to 0$. Fortunately, we can deal with that problem by means of Lemma 7.4, which implies that, if the matrix $M \in \mathbb{C}^{n\times n}$ is sampled from $\mathcal{G}^{n\times n}$ and if E is a small shift, then $M+E$ is nearly indistinguishable from a sample from $\mathcal{G}^{n\times n}$. Using Lemma 7.4, we now adapt Theorem 9.1 to the complex case.

Theorem 9.3 (Random Self-Reducibility of Gaussian Permanent)**.** *For all $\delta \geq 1/\operatorname{poly}(n)$ the following problem is #P-hard. Given an $n\times n$ matrix M drawn from $\mathcal{G}^{n\times n}$ output* $\operatorname{Per}(M)$ *with probability at least $3/4+\delta$ over M.*

Proof. Let $X = (x_{ij}) \in \{0,1\}^{n\times n}$ be an arbitrary 0/1 matrix. We will show how to compute $\operatorname{Per}(X)$ in probabilistic polynomial time, given access to an oracle $\mathcal{O}$ such that

$$\Pr_{M\sim\mathcal{G}^{n\times n}}[\mathcal{O}(M) = \operatorname{Per}(M)] \geq \frac{3}{4}+\delta\,. \tag{9.7}$$

Clearly this suffices to prove the theorem. The first step is to choose a matrix $Y \in \mathbb{C}^{n\times n}$ from the Gaussian distribution $\mathcal{G}^{n\times n}$. Then define

$$X(t) := (1-t)Y + tX, \tag{9.8}$$

so that $X(0) = Y$ and $X(1) = X$. Next define

$$q(t) := \operatorname{Per}(X(t))\,, \tag{9.9}$$

so that $q(t)$ is a univariate polynomial in t of degree at most n and $q(1) = \operatorname{Per}(X(1)) = \operatorname{Per}(X)$. Now let

$$L := \left\lceil \frac{n}{\delta} \right\rceil \quad \text{and} \quad \varepsilon := \frac{\delta}{(4n^2+2n)L}\,.$$

For each $\ell \in [L]$ call the oracle $\mathcal{O}$ on input matrix $X(\varepsilon\ell)$. Then, using the Berlekamp-Welch algorithm (Theorem 9.2), attempt to find a degree-n polynomial $q' : \mathbb{C} \to \mathbb{C}$ such that

$$q'(\varepsilon\ell) = \mathcal{O}(X(\varepsilon\ell)) \tag{9.10}$$

for at least a $3/4+\delta$ fraction of $\ell \in [L]$. If no such q' is found, then fail; otherwise, output $q'(1)$ as the guessed value of $\operatorname{Per}(X)$. We claim that the above algorithm succeeds (that is, outputs $q'(1) = \operatorname{Per}(X)$) with probability at least $1/2+\delta/2$ over Y. Provided that holds, it is clear that the success probability can be boosted to (say) $2/3$ by simply repeating the algorithm $O(1/\delta^2)$ times with different choices of Y and then outputting the majority result. To prove the claim, note that for each $\ell \in [L]$ one can think of the matrix $X(\varepsilon\ell)$ as having been drawn from the distribution

$$\mathcal{D}_\ell := \prod_{i,j=1}^{n} \mathcal{N}\left(\varepsilon\ell a_{ij}, (1-\varepsilon\ell)^2\right)_{\mathbb{C}}\,. \tag{9.11}$$

Let

$$\mathcal{D}'_\ell := \prod_{i,j=1}^{n} \mathcal{N}(\varepsilon \ell a_{ij}, 1)_{\mathbb{C}} \,. \tag{9.12}$$

Then by the triangle inequality together with Lemma 7.4,

$$\begin{aligned} \left\| \mathcal{D}_\ell - \mathcal{G}^{n\times n} \right\| &\leq \left\| \mathcal{D}_\ell - \mathcal{D}'_\ell \right\| + \left\| \mathcal{D}'_\ell - \mathcal{G}^{n\times n} \right\| && (9.13)\\ &\leq 2n^2 \varepsilon \ell + \sqrt{n^2 (\varepsilon \ell)^2} && (9.14)\\ &\leq \left(2n^2 + n\right) \varepsilon L && (9.15)\\ &\leq \frac{\delta}{2} \,. && (9.16) \end{aligned}$$

Hence

$$\begin{aligned} \Pr\left[\mathcal{O}\left(X\left(\varepsilon\ell\right)\right) = q\left(\varepsilon\ell\right)\right] &\geq \frac{3}{4} + \delta - \left\| \mathcal{D}_\ell - \mathcal{N}(0,1)_{\mathbb{C}}^{n\times n} \right\| && (9.17)\\ &\geq \frac{3}{4} + \frac{\delta}{2} \,. && (9.18) \end{aligned}$$

Now let S be the set of all $\ell \in [L]$ such that $\mathcal{O}\left(X\left(\varepsilon\ell\right)\right) = q\left(\varepsilon\ell\right)$. Then by Markov's inequality,

$$\Pr\left[|S| \geq \left(\frac{1}{2} + \frac{\delta}{2}\right) L\right] \geq 1 - \frac{\frac{1}{4} - \frac{\delta}{2}}{\frac{1}{2} - \frac{\delta}{2}} \geq \frac{1}{2} + \frac{\delta}{2} \,. \tag{9.19}$$

Furthermore, suppose

$$|S| \geq \left(\frac{1}{2} + \frac{\delta}{2}\right) L \,.$$

Then by Theorem 9.2, the Berlekamp-Welch algorithm will succeed; that is, its output polynomial q' will be equal to q. This proves the claim and hence the lemma. □

As mentioned before, we conjecture that it is possible to improve Theorem 9.3, to show that it is #P-hard even to compute the permanent of an $\alpha = 1/\mathrm{poly}(n)$ fraction of matrices X drawn from the Gaussian distribution $\mathcal{G}^{n\times n}$.

Let us mention two other interesting improvements that one can make to Theorem 9.3. First, one can easily modify the proof to show that not just $\mathrm{Per}(X)$ but also $|\mathrm{Per}(X)|^2$, is as hard to compute for X drawn from the Gaussian distribution $\mathcal{G}^{n\times n}$ as it is in the worst case. For this, one simply needs to observe that, just as $\mathrm{Per}(X)$ is a degree-n polynomial in the entries of X so $|\mathrm{Per}(X)|^2$ is a degree-$2n$ polynomial in the entries of X together with their complex conjugates (or alternatively, in the real and imaginary parts of the entries). The rest of the proof goes through as before. Since $|\mathrm{Per}(X)|^2$ is #P-hard to compute in the worst case by Theorem 4.3, it follows that $|\mathrm{Per}(X)|^2$ is #P-hard to compute for X drawn from the Gaussian distribution as well.

Second, in the proof of Theorem 9.3, one can relax the requirement that the oracle $\mathcal{O}$ computes $\mathrm{Per}(X)$ *exactly* with high probability over $X \sim \mathcal{G}^{n\times n}$, and merely require that

$$\Pr_{X\sim\mathcal{G}^{n\times n}}\left[\left|\mathcal{O}(X) - \mathrm{Per}(X)\right| \leq 2^{-q(n)}\right] \geq \frac{3}{4} + \frac{1}{\mathrm{poly}(n)}, \tag{9.20}$$

for some sufficiently large polynomial q. To do so, one can appeal to the following lemma of Paturi.

Lemma 9.4 (Paturi [51]; see also Buhrman et al. [14])**.** *Let $p:\mathbb{R}\to\mathbb{R}$ be a real polynomial of degree d, and suppose $|p(x)|\le\delta$ for all $|x|\le\varepsilon$. Then $|p(1)|\le\delta e^{2d(1+1/\varepsilon)}$.*

From this perspective, the whole challenge in proving the Permanent-of-Gaussians Conjecture is to replace the $2^{-q(n)}$ approximation error with $1/q(n)$.

Combining, we obtain the following theorem, whose detailed proof we omit.

Theorem 9.5. *There exists a polynomial p for which the following problem is #P-hard, for all $\delta\ge 1/\operatorname{poly}(n)$. Given an $n\times n$ matrix X drawn from $\mathcal{G}^{n\times n}$ output a real number y such that*

$$\left|y-|\operatorname{Per}(X)|^2\right|\le 2^{-p(n,1/\delta)}$$

with probability at least $\frac{3}{4}+\delta$ over X.

As a final observation, it is easy to find *some* efficiently samplable distribution $\mathcal{D}$ over matrices $X\in\mathbb{C}^{n\times n}$ such that estimating $\operatorname{Per}(X)$ or $|\operatorname{Per}(X)|^2$ for most $X\sim\mathcal{D}$ is a #P-hard problem. To do so, simply start with any problem that is known to be #P-hard on average: for example, computing $\operatorname{Per}(M)$ for most matrices $M\in\mathbb{F}_p^{n\times n}$ over a *finite* field $\mathbb{F}_p$. Next, use Theorem 4.3 to reduce the computation of $\operatorname{Per}(M)$ (for a uniform random M) to the estimation of $|\operatorname{Per}(X_1)|^2,\ldots,|\operatorname{Per}(X_m)|^2$ for various matrices $X_1,\ldots,X_m\in\mathbb{C}^{n\times n}$. Finally, output a random X_i as one's sample from $\mathcal{D}$. Clearly, if one could estimate $|\operatorname{Per}(X)|^2$ for a $1-1/\operatorname{poly}(n)$ fraction of $X\sim\mathcal{D}$ one could also compute $\operatorname{Per}(M)$ for a $1-1/\operatorname{poly}(n)$ fraction of $M\in\mathbb{F}_p^{n\times n}$ and thereby solve a #P-hard problem. Because of this, we see that the challenge is "merely" how to prove average-case #P-hardness, in the specific case where the distribution $\mathcal{D}$ over matrices that interests us is the Gaussian distribution $\mathcal{G}^{n\times n}$ (or more generally, some other "nice" or "uniform-looking" distribution).

9.2 The barrier to proving the PGC

In this section, we identify a significant barrier to proving Conjecture 1.5, and explain why a new approach seems needed.

As Section 9.1 discussed, all existing proofs of the worst-case/average-case equivalence of the PERMANENT are based on *low-degree polynomial interpolation*. More concretely, given a matrix $X\in\mathbb{F}^{n\times n}$ for which we want to compute $\operatorname{Per}(X)$ we first choose a random low-degree curve $X(t)$ through $\mathbb{F}^{n\times n}$ satisfying $X(0)=X$. We then choose nonzero points $t_1,\ldots,t_m\in\mathbb{R}$ and compute or approximate $\operatorname{Per}(X(t_i))$ for all $i\in[m]$ using the assumption that the PERMANENT is easy on average. Finally, using the fact that $q(t):=\operatorname{Per}(X(t))$ is a low-degree polynomial in t we perform polynomial interpolation on the noisy estimates

$$y_1\approx q(t_1),\ldots,y_m\approx q(t_m),\tag{9.21}$$

in order to obtain an estimate of the worst-case permanent $q(0)=\operatorname{Per}(X(0))=\operatorname{Per}(X)$.

The above approach is a very general one, with different instantiations depending on the base field $\mathbb{F}$ the fraction of X's for which we can compute $\operatorname{Per}(X)$, and so forth. Nevertheless, we claim that, assuming the Permanent Anti-Concentration Conjecture, *the usual polynomial interpolation approach cannot possibly work to prove Conjecture 1.5.* Let us see why this is the case.

Let $X \in \mathbb{C}^{n\times n}$ be a matrix where every entry has absolute value at most 1. Then certainly it is a #P-hard problem to approximate $\operatorname{Per}(X)$ multiplicatively (as shown by Theorem 4.3, for example). Our goal is to reduce the approximation of $\operatorname{Per}(X)$ to the approximation of $\operatorname{Per}(X_1),\dots,\operatorname{Per}(X_m)$ for some matrices $X_1,\dots,X_m$ that are drawn from the Gaussian distribution $\mathcal{G}^{n\times n}$ or something close to it.

Recall from Section 8 that

$$\operatorname*{E}_{X\sim\mathcal{G}^{n\times n}}\left[|\operatorname{Per}(X)|^2\right] = n!\,, \tag{9.22}$$

which combined with Markov's inequality yields

$$\Pr_{X\sim\mathcal{G}^{n\times n}}\left[|\operatorname{Per}(X)| > k\sqrt{n!}\right] < \frac{1}{k^2} \tag{9.23}$$

for all $k > 1$. But this already points to a problem: $|\operatorname{Per}(X)|$ *could, in general, be larger than* $|\operatorname{Per}(X_1)|,\dots,|\operatorname{Per}(X_m)|$ *by an exponential factor.* Specifically, $|\operatorname{Per}(X)|$ could be as large as $n!$ (for example, if A is the all-1's matrix). By contrast, $|\operatorname{Per}(X_1)|,\dots,|\operatorname{Per}(X_m)|$ will typically be $O(\sqrt{n!})$ by equation (9.23). And yet, from constant-factor approximations to $\operatorname{Per}(X_1),\dots,\operatorname{Per}(X_m)$ we are supposed to recover a constant-factor approximation to $\operatorname{Per}(X)$, *even in the case that* $|\operatorname{Per}(X)|$ *is much smaller than* $n!$ (say, $|\operatorname{Per}(X)| \approx \sqrt{n!}$).

Why is this a problem? Because polynomial interpolation is linear with respect to additive errors. And therefore, even modest errors in estimating $\operatorname{Per}(X_1),\dots,\operatorname{Per}(X_m)$ could cause a large error in estimating $\operatorname{Per}(X)$.

To see this concretely, let X be the $n\times n$ all-1's matrix, and $X(t)$ be a randomly-chosen curve through $\mathbb{C}^{n\times n}$ that satisfies $X(0) = X$. Also, let $t_1,\dots,t_m \in \mathbb{R}$ be nonzero points such that, as we vary X each $X(t_i)$ is close to a Gaussian random matrix $X \sim \mathcal{G}^{n\times n}$. (We need not assume that the $X(t_i)$'s are independent.) Finally, let $q_0(t) := \operatorname{Per}(X(t))$. Then

(i) $|q_0(t_1)|,\dots,|q_0(t_m)|$ are each at most $n^{O(1)}\sqrt{n!}$ with high probability over the choice of X but

(ii) $|q_0(0)| = |\operatorname{Per}(X(0))| = |\operatorname{Per}(X)| = n!$.

Here (i) holds by our assumption that each $X(t_i)$ is close to Gaussian, together with equation (9.23).

All we need to retain from this is that a polynomial q_0 with properties (i) and (ii) *exists*, within whatever class of polynomials is relevant for our interpolation problem.

Now, suppose that instead of choosing X to be the all-1's matrix, we had chosen an$X$ such that $|\operatorname{Per}(A)| \le \sqrt{n!}$. Then as before, we could choose a random curve $X(t)$ such that $X(0) = X$ and $X(t_1),\dots,X(t_m)$ are approximately Gaussian, for some fixed interpolation points $t_1,\dots,t_m \in \mathbb{R}$. Then letting $q(t) := \operatorname{Per}(X(t))$ we would have

(i) $|q(t_1)|,\dots,|q(t_m)|$ are each at least $\sqrt{n!}/n^{O(1)}$ with high probability over the choice of X and

(ii) $|q(0)| = |\operatorname{Per}(X(0))| = |\operatorname{Per}(X)| \le \sqrt{n!}$.

Here (i) holds by our assumption that each $X(t_i)$ is close to Gaussian, together with Conjecture 1.6 (the Permanent Anti-Concentration Conjecture).

Now define a new polynomial

$$\widetilde{q}(t) := q(t) + \gamma q_0(t), \tag{9.24}$$

where, say, $|\gamma| = 2^{-n}$. Then for all $i \in [m]$ the difference

$$|\widetilde{q}(t_i) - q(t_i)| = |\gamma q_0(t_i)| \le \frac{n^{O(1)}}{2^n}\sqrt{n!}, \tag{9.25}$$

is negligible compared to $\sqrt{n!}$. This means that *it is impossible to distinguish the two polynomials $\widetilde{q}$ and q, given their approximate values at the points $t_1, \ldots, t_m$.* And yet the two polynomials have completely different behavior at the point 0: by assumption $|q(0)| \le \sqrt{n!}$ but

$$|\widetilde{q}(0)| \ge |\gamma q_0(0)| - |q(0)| \tag{9.26}$$

$$\ge \frac{n!}{2^n} - \sqrt{n!}. \tag{9.27}$$

We conclude that it is impossible, given only the approximate values of the polynomial $q(t) := \mathrm{Per}(X(t))$ at the points $t_1, \ldots, t_m$ to deduce its approximate value at 0. And therefore, assuming the PACC, the usual polynomial interpolation approach cannot suffice for proving Conjecture 1.5.

Nevertheless, we speculate that there *is* a worst-case/average-case reduction for approximating the permanents of Gaussian random matrices, and that the barrier we have identified merely represents a limitation of current techniques. So for example, perhaps one can do interpolation using a *restricted class* of low-degree polynomials, such as polynomials with an upper bound on their coefficients. To evade the barrier, what seems to be crucial is that the restricted class of polynomials one uses not be closed under addition.

Of course, the above argument relied on the Permanent Anti-Concentration Conjecture, so one conceivable way around the barrier would be if the PACC were false. However, in that case, the results of Section 7 would fail: that is, we would not know how to use the hardness of $\mathrm{GPE}_\times$ to deduce the hardness of $|\mathrm{GPE}|^2_\pm$ that we need for our application.

10 Open problems

The most exciting challenge we leave is to *do* the experiments discussed in Section 6, whether in linear optics or in other physical systems that contain excitations that behave as identical bosons. If successful, such experiments have 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 1.5 (the Permanent-of-Gaussians Conjecture): 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 $\mathsf{BPP}^{\mathsf{NP}}$?

(2) Another obvious problem is to prove Conjecture 1.6 (the Permanent Anti-Concentration Conjecture), that $|\mathrm{Per}(X)|$ almost always exceeds $\sqrt{n!}/\mathrm{poly}(n)$ for Gaussian random matrices $X \sim \mathcal{N}(0,1)_{\mathbb{C}}^{n\times n}$. Failing that, *any* progress on understanding the distribution of $\mathrm{Per}(X)$ for Gaussian X would be interesting.

(3) Can we reduce the number of modes needed for our linear-optics experiment, perhaps from $O(n^2)$ to $O(n)$?

(4) How far can we decrease the physical resource requirements for our experiment? For example, what happens if we have single-photon input states combined with Gaussian measurements? Also, if we have Gaussian input states combined with nonadaptive photon-number measurements, then Theorem 4.9 shows that our argument for the hardness of *exact* sampling goes through, but what about approximate sampling?

(5) 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?

(6) To what extent can one use quantum fault-tolerance techniques to decrease the effective error in our experiment? 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 $\mathrm{poly}(n)$ 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.

(7) 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 $\varepsilon > 1/\mathrm{poly}(n)$?

(8) 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 [12] 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?

(9) 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 $\mathsf{SampP} = \mathsf{SampBQP}$ or $\mathsf{FBPP} = \mathsf{FBQP}$. An obvious question that remains is, what about *decision* problems? Can we derive some unlikely collapse of classical complexity classes from the assumption that $\mathsf{P} = \mathsf{BQP}$ or $\mathsf{PromiseP} = \mathsf{PromiseBQP}$?

(10) 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 4.9, 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 4.9 is easily seen to relativize. However, we do not know the situation with our *approximate* result, Theorem 1.3. More precisely, does there exist an oracle A relative to which $\mathsf{FBPP} = \mathsf{FBQP}$ but PH is infinite? Such an oracle would show that Theorem 1.3 required the use of *some* nonrelativizing ingredient—for example, the #P-hardness of a concrete problem involving Gaussian permanents. Currently, the closest we have to this is a powerful result of Fortnow and Rogers [27], which gives an oracle A relative to which $\mathsf{P} = \mathsf{BQP}$ but PH is infinite. However, it is not even known how to extend the Fortnow-Rogers construction to get an oracle A relative to which $\mathsf{PromiseP} = \mathsf{PromiseBQP}$ but PH is infinite. The situation is summarized in the table below.

	Complexity consequence	
Assumption	PH collapses (relativizing)	PH collapses (in real world)
$\mathsf{P} = \mathsf{BQP}$	No [27]	?
$\mathsf{PromiseP} = \mathsf{PromiseBQP}$	?	?
$\mathsf{FBPP} = \mathsf{FBQP}$	?	Assuming our conjectures
Exact QSAMPLING easy	Yes	Yes

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

(12) As discussed in Section 6, 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 much can we do to remedy this? For example, can a prover with a BOSONSAMPLING oracle prove any nontrivial statements to a BPP verifier via an interactive protocol? Here we should mention the lovely recent result of Arora et al. [8], which was motivated by an earlier version of this paper. They show that the problem we call $\mathrm{GPE}_{\pm}$ is *self-checkable*. In other words, suppose we are given an oracle $\mathcal{O}$ which is claimed to output a good additive approximation to $\mathrm{Per}(X)$ for most Gaussian matrices $X \sim \mathcal{N}(0,1)_{\mathbb{C}}^{n\times n}$. Then Arora et al. show that it is possible to test, in randomized polynomial time, whether $\mathcal{O}$ satisfies the claim or is far from satisfying it. One consequence of their result is that, if we were given a purported classical randomized algorithm for BOSONSAMPLING—call it C—then we could check whether C worked in the class $\mathsf{BPP}^{\mathsf{NP}^C}$. Unfortunately, it remains unclear whether this result has any relevance for the verification of BOSONSAMPLING experiments in the lab.

(13) 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?

Appendix A The symmetric product

As mentioned in Section 1.1, one way to think about the Hilbert space of n identical photons, is as the so-called *symmetric product* of n single-photon Hilbert spaces. Although we never needed the formalism of symmetric products when proving our results, in this appendix we develop the formalism a bit, in order to explore what it says about the presence or absence of "entanglement" in linear-optical states.

Given an m-dimensional Hilbert space $\mathcal{H} = \mathbb{C}^m$ the symmetric product

$$\mathcal{H}^{\odot n} = \overbrace{\mathcal{H} \odot \cdots \odot \mathcal{H}}^{n \text{ factors}} \tag{A.1}$$

(also written $S^n(\mathbb{C}^m)$) is the $\binom{m+n-1}{n}$-dimensional space $H_{m,n}$ defined in Section 3. The $\odot$ operation also acts on states, in which case we can define $|\phi\rangle = |\psi_1\rangle \odot |\psi_2\rangle$ to be the unique state $|\phi\rangle$ such that $P_{|\phi\rangle} = P_{|\psi_1\rangle} \cdot P_{|\psi_2\rangle}$. Here $P_{|\phi\rangle}, P_{|\psi_2\rangle}, P_{|\psi_2\rangle}$ are the polynomials corresponding to $|\phi\rangle, |\psi_1\rangle, |\psi_2\rangle$ respectively as defined in Section 3.2 (see equation (3.37)), and $\cdot$ denotes ordinary polynomial multiplication. A crucial difference between the symmetric product $\odot$ and the ordinary tensor product $\otimes$ is that the symmetric product is commutative:

$$|A\rangle \odot |B\rangle = |B\rangle \odot |A\rangle\,. \tag{A.2}$$

The reason for this commutativity is the assumption that the bosons are identical, so that swapping them produces no physical change.

Because of the isomorphism between the symmetric product and polynomial multiplication, it follows immediately from Section 3.2 that we can represent any state $\varphi(U)|1_n\rangle$, obtained by taking the standard initial state and then applying a linear-optical transformation, as a symmetric product $|\psi_1\rangle \odot \cdots \odot |\psi_n\rangle$ of single-photon states.

But the symmetric product provides only one way to represent linear-optical states. A second representation is what we might call "naïve tensor-product form": this is like symmetric-product form, except that the commutativity (A.2) is made explicit by the symmetries of the state itself. (The analogue, in algebra, would be replacing every occurrence of xy by $(1/2)(xy+yx)$.) A third representation is the one we've been using for almost all of the paper: namely, a superposition over the occupation-number basis states $|s_1,\ldots,s_m\rangle$.

To illustrate these three representations, let $|\Lambda\rangle$ be the linear-optical state consisting of one photon in an equal superposition of two modes, let $|\Psi\rangle$ be the state consisting of two photons, one in each of two modes, and let $|\Phi\rangle$ be the two-photon output state produced in the Hong-Ou-Mandel experiment (see Section 6.1). Then the following table shows what $|\Lambda\rangle$ $|\Psi\rangle$ and $|\Phi\rangle$ look like in each of the representations.

	$\|\Lambda\rangle$	$\|\Psi\rangle$	$\|\Phi\rangle$
Occupation-Number	$\frac{\|1,0\rangle+\|0,1\rangle}{\sqrt{2}}$	$\|1,1\rangle$	$\frac{\|2,0\rangle-\|0,2\rangle}{\sqrt{2}}$
Naïve Tensor-Product	$\frac{\|1\rangle+\|2\rangle}{\sqrt{2}}$	$\frac{\|1\rangle\otimes\|2\rangle+\|2\rangle\otimes\|1\rangle}{\sqrt{2}}$	$\frac{\|1\rangle\otimes\|1\rangle-\|2\rangle\otimes\|2\rangle}{\sqrt{2}}$
Symmetric-Product	$\frac{\|1\rangle+\|2\rangle}{\sqrt{2}}$	$\|1\rangle\odot\|2\rangle$	$\frac{\|1\rangle+\|2\rangle}{\sqrt{2}}\odot\frac{\|1\rangle-\|2\rangle}{\sqrt{2}}$

Here the factorization in the bottom-right corner corresponds to the identity $x^2 - y^2 = (x+y)(x-y)$.

With these representations and examples in hand, we now pose the question: *are the states* $|\Lambda\rangle$*,* $|\Psi\rangle$*, and* $|\Phi\rangle$ *"entangled" or "unentangled"?* Alas, we see that the answers can *apparently* be different depending on which representation is used! In occupation-number form—treating basis states $|s,t\rangle$ as tensor products $|s\rangle \otimes |t\rangle$—the states $|\Lambda\rangle$ and $|\Phi\rangle$ both look entangled while $|\Psi\rangle$ looks unentangled. In naïve tensor-product form, $|\Psi\rangle$ and $|\Phi\rangle$ both look entangled while $|\Lambda\rangle$ looks unentangled. Finally, in symmetric-product form, all three states look unentangled.

Let us make three remarks that might help clarify the situation. First, *all* linear-optical states involving superpositions over different modes look entangled when written in occupation-number form. One can argue about whether such entanglement is "real" or "illusory"—but whichever the case, *this* is the reason why we generate entangled states when simulating a linear-optical quantum computer using a standard, qubit-based quantum computer in the manner of Theorem 3.12. As such, it is directly related to why our model seems intractable to simulate using a classical computer. Second, *all* multiple-photon states look entangled when written in naïve tensor-product form. In our view, however, any "entanglement" that appears only in the naïve tensor-product representation is just a mathematical artifact, since it arises solely from that representation's unphysical insistence on *labeling* identical photons. Third, by construction, all linear-optical states of the form $\varphi(U)|1_n\rangle$ look *un*entangled when written in symmetric-product form. This is the mathematical reflection of the physical fact that the photons never *interact* in our model. Indeed, this is arguably what justifies calling our model "noninteracting" in the first place.

Appendix B Positive results for simulation of linear optics

In this appendix, we present two results of Gurvits, both of which give surprising classical polynomial-time algorithms for computing certain properties of linear-optical networks. The first result, which appeared in [33], gives an efficient randomized algorithm to approximate the permanent of a (sub)unitary matrix with $\pm 1/\operatorname{poly}(n)$ additive error, and as a consequence, to estimate final amplitudes such as $\langle 1_n|\varphi(U)|1_n\rangle = \operatorname{Per}(U_{n,n})$ with $\pm 1/\operatorname{poly}(n)$ additive error, given any linear-optical network U. This ability is of limited use in practice, since $\langle 1_n|\varphi(U)|1_n\rangle$ will be exponentially small for most choices of U (in which case, 0 is also a good additive estimate!). On the other hand, we certainly do not know how to do anything similar for general, qubit-based quantum circuits—indeed, if we could, then BQP would equal BPP.

Gurvits's second result (unpublished) gives a way to compute the marginal distribution over photon numbers for any k modes, deterministically and in $n^{O(k)}$ time. Again, this is perfectly consistent with our hardness conjectures, since if one wanted to *sample* from the distribution over photon numbers (or compute a final probability such as $|\langle 1_n|\varphi(U)|1_n\rangle|^2$), one would need to take $k \geq n$.

To prove Gurvits's first result, our starting point will be the following identity of Ryser, which is also used for computing the permanent of an $n \times n$ matrix in $O(2^n n^2)$ time.

Lemma B.1 (Ryser's Formula)**.** *For all $V \in \mathbb{C}^{n\times n}$,*

$$\operatorname{Per}(V) = \operatorname*{E}_{x_1,\ldots,x_n \in \{-1,1\}}\left[x_1 \cdots x_n \prod_{i=1}^{n}(v_{i1}x_1 + \cdots + v_{in}x_n)\right]. \tag{B.1}$$

Proof. Let $p(x_1,\ldots,x_n)$ be the degree-n polynomial that corresponds to the product in the above expectation. Then the only monomial of p that can contribute to the expectation is $x_1 \cdots x_n$ since all the other monomials will be cancelled out by the multiplier of $x_1 \cdots x_n$ (which is equally likely to be 1 or -1). Furthermore, as in Lemma 3.8, the coefficient of $x_1 \cdots x_n$ is just

$$\sum_{\sigma \in S_n} \prod_{i=1}^{n} v_{i,\sigma(i)} = \operatorname{Per}(V). \tag{B.2}$$

Therefore the expectation equals

$$\operatorname{Per}(V) \mathop{\mathrm{E}}_{x_1,\ldots,x_n \in \{-1,1\}} \left[x_1^2 \cdots x_n^2\right] = \operatorname{Per}(V) . \tag{B.3}$$

(Indeed, all we needed about the random variables $x_1, \ldots, x_n$ was that they were independent and had mean 0 and variance 1.) □

Given $x = (x_1, \ldots, x_n) \in \{-1,1\}^n$, let

$$\operatorname{Rys}_x(V) := x_1 \cdots x_n \prod_{i=1}^{n} (v_{i1} x_1 + \cdots + v_{in} x_n) . \tag{B.4}$$

Then Lemma B.1 says that $\operatorname{Rys}_x(V)$ is an *unbiased estimator* for the permanent, in the sense that $\mathrm{E}_x[\operatorname{Rys}_x(V)] = \operatorname{Per}(V)$. Gurvits [33] observed the following key further fact about $\operatorname{Rys}_x(V)$.

Lemma B.2. $|\operatorname{Rys}_x(V)| \le \|V\|^n$ *for all* $x \in \{-1,1\}^n$ *and all* V.

Proof. Given a vector $x = (x_1, \ldots, x_n)$ all of whose entries are 1 or -1 let $y = Vx$ and let

$$y_i := v_{i1} x_1 + \cdots + v_{in} x_n \tag{B.5}$$

be the i^{th} component of y. Then $\|x\| = \sqrt{n}$, so $\|y\| \le \|V\| \|x\| = \|V\| \sqrt{n}$. Hence

$$\begin{aligned} |\operatorname{Rys}_x(V)| &= |x_1 \cdots x_n y_1 \cdots y_n| && \text{(B.6)}\\ &= |y_1 \cdots y_n| && \text{(B.7)}\\ &\le \left(\frac{|y_1| + \cdots + |y_n|}{n}\right)^n && \text{(B.8)}\\ &\le \left(\frac{\|y\|}{\sqrt{n}}\right)^n && \text{(B.9)}\\ &\le \|V\|^n , && \text{(B.10)} \end{aligned}$$

where line (B.8) follows from the arithmetic-geometric mean inequality, and line (B.9) follows from Cauchy-Schwarz. □

An immediate consequence of Lemma B.2 is the following:

Corollary B.3. $|\operatorname{Per}(V)| \le \|V\|^n$ *for all* V.

Another consequence is a fast additive approximation algorithm for $\operatorname{Per}(V)$ which works whenever $\|V\|$ is small.

Theorem B.4 (Gurvits's Permanent Approximation Algorithm [33])**.** *There exists a randomized (classical) algorithm that takes a matrix $V \in \mathbb{C}^{n \times n}$ as input, runs in $O\left(n^2/\varepsilon^2\right)$ time, and with high probability, approximates* $\operatorname{Per}(V)$ *to within an additive error* $\pm\varepsilon \|V\|^n$.

Proof. By Lemma B.1,

$$\operatorname{Per}(V) = \mathop{\mathrm{E}}_{x\in\{-1,1\}^n}\left[\operatorname{Rys}_x(V)\right]. \tag{B.11}$$

Furthermore, we know from Lemma B.2 that $|\operatorname{Rys}_x(V)| \leq \|V\|^n$ for every x. So our approximation algorithm is simply the following: for $T = O\left(1/\varepsilon^2\right)$ first choose T vectors $x(1),\ldots,x(T)$ uniformly at random from $\{-1,1\}^n$. Then output the empirical mean

$$\widetilde{p} := \frac{1}{T}\sum_{t=1}^{T}\operatorname{Rys}_{x(t)}(V) \tag{B.12}$$

as our estimate of $\operatorname{Per}(V)$. Since $\operatorname{Rys}_x(V)$ can be computed in $O\left(n^2\right)$ time, this algorithm takes $O\left(n^2/\varepsilon^2\right)$ time. The failure probability,

$$\Pr_{x(1),\ldots,x(T)}\left[\left|\widetilde{p} - \operatorname{Per}(V)\right| > \varepsilon\|V\|^n\right], \tag{B.13}$$

can be upper-bounded using a standard Chernoff bound. □

In particular, Theorem B.4 implies that, given an $n\times n$ unitary matrix U one can approximate $\operatorname{Per}(U)$ to within an additive error $\pm\varepsilon$ (with high probability) in $\operatorname{poly}(n,1/\varepsilon)$ time.

We now sketch a proof of Gurvits's second result, giving an $n^{O(k)}$-time algorithm to compute the marginal distribution over any k photon modes. We will assume the following lemma, whose proof will appear in a forthcoming paper of Gurvits.

Lemma B.5 (Gurvits)**.** *Let $V \in \mathbb{C}^{n\times n}$ be a matrix of rank k. Then $\operatorname{Per}(V+I)$ can be computed exactly in $n^{O(k)}$ time.*

We now show how to apply Lemma B.5 to the setting of linear optics.

Theorem B.6 (Gurvits's k-Photon Marginal Algorithm)**.** *There exists a deterministic classical algorithm that, given a unitary matrix $U\in\mathbb{C}^{m\times m}$ indices $i_1,\ldots,i_k\in[m]$, and occupation numbers $j_1,\ldots,j_k\in\{0,\ldots,n\}$, computes the joint probability*

$$\Pr_{S=(s_1,\ldots,s_m)\sim\mathcal{D}_U}\left[s_{i_1}=j_1\wedge\cdots\wedge s_{i_k}=j_k\right] \tag{B.14}$$

in $n^{O(k)}$ time.

Proof. By symmetry, we can assume without loss of generality that $(i_1,\ldots,i_k)=(1,\ldots,k)$. Let $c=(c_1,\ldots,c_k)$ be an arbitrary vector in $\mathbb{C}^k$. Then the crucial claim is that we can compute the expectation

$$\mathop{\mathrm{E}}_{S\sim\mathcal{D}_U}\left[|c_1|^{2s_1}\cdots|c_k|^{2s_k}\right] = \sum_{s_1,\ldots,s_k}\Pr\left[s_1,\ldots,s_k\right]|c_1|^{2s_1}\cdots|c_k|^{2s_k} \tag{B.15}$$

in $n^{O(k)}$ time. Given this claim, the theorem follows easily. We simply need to choose $(n+1)^k$ values for $|c_1|,\ldots,|c_k|$ compute

$$\mathop{\mathrm{E}}_{S\sim\mathcal{D}_U}\left[|c_1|^{2s_1}\cdots|c_k|^{2s_k}\right]$$

for each one, and then solve the resulting system of $(n+1)^k$ independent linear equations in $(n+1)^k$ unknowns to obtain the probabilities $\Pr[s_1,\dots,s_k]$ themselves. We now prove the claim. Let $I_c : \mathbb{C}^m \to \mathbb{C}^m$ be the diagonal linear transformation that maps the vector $(x_1,\dots,x_m)$ to $(c_1x_1,\dots,c_kx_k,x_{k+1},\dots,x_m)$ and let $I_{|c|^2} = I_c^\dagger I_c$ be the linear transformation that maps $(x_1,\dots,x_m)$ to $(|c_1|^2 x_1,\dots,|c_k|^2 x_k,x_{k+1},\dots,x_m)$. Also, let

$$U[J_{m,n}](x) = \sum_{S\in\Phi_{m,n}} a_S x^S. \tag{B.16}$$

Now define a polynomial q by

$$q(x) := I_c U[J_{m,n}](x), \tag{B.17}$$

and note that

$$q(x) = \sum_{S\in\Phi_{m,n}} a_S x^S c_1^{s_1}\cdots c_k^{s_k}. \tag{B.18}$$

Hence

$$\begin{aligned}
\operatorname*{E}_{S=(s_1,\dots,s_m)\sim\mathcal{D}_U}\left[|c_1|^{2s_1}\cdots|c_k|^{2s_k}\right] &= \sum_{S=(s_1,\dots,s_m)\in\Phi_{m,n}}\left(|a_S|^2 s_1!\cdots s_m!\right)|c_1|^{2s_1}\cdots|c_k|^{2s_k} && \text{(B.19)}\\
&= \sum_{S=(s_1,\dots,s_m)\in\Phi_{m,n}}\left(\overline{a}_S\overline{c}_1^{s_1}\cdots\overline{c}_k^{s_k}\right)\left(a_S c_1^{s_1}\cdots c_k^{s_k}\right)s_1!\cdots s_m! && \text{(B.20)}\\
&= \langle q,q\rangle. && \text{(B.21)}
\end{aligned}$$

Now,

$$\begin{aligned}
\langle q,q\rangle &= \langle I_c U[J_{m,n}], I_c U[J_{m,n}]\rangle && \text{(B.22)}\\
&= \left\langle U[J_{m,n}], I_{|c|^2} U[J_{m,n}]\right\rangle && \text{(B.23)}\\
&= \left\langle J_{m,n}, U^\dagger I_{|c|^2} U[J_{m,n}]\right\rangle && \text{(B.24)}\\
&= \operatorname{Per}\left(\left(U^\dagger I_{|c|^2} U\right)_{n,n}\right) && \text{(B.25)}
\end{aligned}$$

where lines (B.23) and (B.24) follow from Theorem 3.4, and line (B.25) follows from Lemma 3.8. Finally, let $\Lambda := I_{|c|^2} - I$. Then Λ is a diagonal matrix of rank at most k and

$$\begin{aligned}
\left(U^\dagger I_{|c|^2} U\right)_{n,n} &= \left(U^\dagger(\Lambda+I)U\right)_{n,n} && \text{(B.26)}\\
&= \left(U^\dagger \Lambda U + I\right)_{n,n} && \text{(B.27)}\\
&= V + I, && \text{(B.28)}
\end{aligned}$$

where $V := \left(U^\dagger \Lambda U\right)_{n,n}$ is an $n\times n$ matrix of rank at most k. So by Lemma B.5, we can compute

$$\operatorname{Per}(V+I) = \operatorname*{E}_{S=(s_1,\dots,s_m)\sim\mathcal{D}_U}\left[|c_1|^{2s_1}\cdots|c_k|^{2s_k}\right] \tag{B.29}$$

in $n^{O(k)}$ time. Furthermore, notice that we can compute V itself in $O\left(n^2k\right) = n^{O(1)}$ time, independent of m. Therefore the total time needed to compute the expectation is $n^{O(k)+O(1)} = n^{O(k)}$. This proves the claim. □

Appendix C The bosonic birthday paradox

By the *birthday paradox*, we mean the statement that, if n balls are thrown uniformly and independently into m bins, then with high probability we will see a collision (i. e., two or more balls in the same bin) if $m = O\left(n^2\right)$ but *not otherwise*.

In this appendix, we prove the useful fact that the birthday paradox still holds if the balls are identical bosons, and "throwing" the balls means applying a Haar-random unitary matrix. More precisely, suppose there are m modes, of which the first n initially contain n identical photons (with one photon in each mode) and the remaining $m-n$ are unoccupied. Suppose we mix the modes by applying an $m \times m$ unitary matrix U chosen uniformly at random from the Haar measure. Then if we measure the occupation number of each mode, we will observe a collision (i. e., two or more photons in the same mode) with probability bounded away from 0 if $m = O\left(n^2\right)$ but not otherwise.

It is well-known that identical bosons are "gregarious," in the sense of being *more* likely than classical particles to occur in the same state. For example, if we throw two balls uniformly and independently into two bins, then the probability of both balls landing in the same bin is only $1/2$ with classical balls, but $2/3$ if the balls are identical bosons.[29] So the interesting part of the bosonic birthday paradox is the "converse direction": when $m \gg n^2$ the probability of two or more bosons landing in the same mode is *not* too large. In other words, while bosons are "somewhat" more gregarious than classical particles, they are not *so* gregarious as to require a different asymptotic relation between m and n.

The proof of our main result, Theorem 1.3, implicitly used this fact: we needed that when $m \gg n^2$ the basis states with two or more photons in the same mode can safely be neglected. However, while in principle one could extract a proof of the bosonic birthday paradox from the proof of Theorem 1.3, we thought it would be illuminating to prove the bosonic birthday paradox directly.

The core of the proof is the following simple lemma about the transition probabilities induced by unitary matrices.

Lemma C.1 (Unitary Pigeonhole Principle). *Partition a finite set $[M]$ into a "good part" G and "bad part" $B = [M] \setminus G$. Also, let $U = (u_{xy})$ be any $M \times M$ unitary matrix. Suppose we choose an element $x \in G$ uniformly at random, apply U to $|x\rangle$ then measure $U|x\rangle$ in the standard basis. Then letting y be the measurement outcome, we have* $\Pr[y \in B] \leq |B| / |G|$.

Proof. Let R be an $M \times M$ doubly-stochastic matrix whose (x,y) entry is $r_{xy} := |u_{xy}|^2$. Then applying U to a computational basis state $|x\rangle$ and measuring immediately afterward is the same as applying R; in particular, $\Pr[y \in B] = r_{xy}$. Moreover,

$$\sum_{x,y \in G} r_{xy} = \sum_{x \in G, y \in [M]} r_{xy} + \sum_{x \in [M], y \in G} r_{xy} - \sum_{x,y \in [M]} r_{xy} + \sum_{x,y \in B} r_{xy} \tag{C.1}$$

$$= |G| + |G| - M + \sum_{x,y \in B} r_{xy} \tag{C.2}$$

$$\geq 2|G| - M\,, \tag{C.3}$$

[29]This is in stark contrast to the situation with identical fermions, no two of which ever occur in the same state by the Pauli exclusion principle.

where line (C.1) follows from simple rearrangements and line (C.2) follows from the double-stochasticity of R. Hence

$$\Pr[y \in G] = \operatorname*{E}_{x \in G}\left[\sum_{y \in G} r_{xy}\right] \geq \frac{2|G| - M}{|G|} = 1 - \frac{|B|}{|G|}, \tag{C.4}$$

and

$$\Pr[y \in B] = 1 - \Pr[y \in G] \leq \frac{|B|}{|G|}. \tag{C.5}$$

□

Lemma C.1 has the following important corollary. Suppose we draw the $M \times M$ unitary matrix U from a probability distribution $\mathcal{Z}$ where $\mathcal{Z}$ is *symmetric* with respect to some transitive group of permutations on the good set G. Then $\Pr[y \in B]$ is clearly independent of the choice of initial state $x \in G$. And therefore, in the statement of the lemma, we might as well *fix* $x \in G$ rather than choosing it randomly. The statement then becomes:

Corollary C.2. *Partition a finite set $[M]$ into a "good part" G and "bad part" $B = [M] \setminus G$. Also, let $\Gamma \leq S_M$ be a permutation group that is transitive with respect to G and let $\mathcal{Z}$ be a probability distribution over $M \times M$ unitary matrices that is symmetric with respect to Γ. Fix an element $x \in G$. Suppose we draw a unitary matrix U from $\mathcal{Z}$, apply U to $|x\rangle$ and measure $U|x\rangle$ in the standard basis. Then the measurement outcome will belong to B with probability at most $|B| / |G|$.*

Given positive integers $m \geq n$ recall that $\Phi_{m,n}$ is the set of lists of nonnegative integers $S = (s_1, \ldots, s_m)$ such that $s_1 + \cdots + s_m = n$. Also, recall from Theorem 1.3 that a basis state $S \in \Phi_{m,n}$ is called *collision-free* if each s_i is either 0 or 1. Let $G_{m,n}$ be the set of collision-free S's, and let $B_{m,n} = \Phi_{m,n} \setminus G_{m,n}$. Then we have the following simple estimate.

Proposition C.3.

$$\frac{|G_{m,n}|}{|\Phi_{m,n}|} > 1 - \frac{n^2}{m}. \tag{C.6}$$

Proof.

$$\frac{|G_{m,n}|}{|\Phi_{m,n}|} = \frac{\binom{m}{n}}{\binom{m+n-1}{n}} \tag{C.7}$$

$$= \frac{m!\,(m-1)!}{(m-n)!\,(m+n-1)!} \tag{C.8}$$

$$= \left(1 - \frac{n-1}{m}\right)\left(1 - \frac{n-1}{m+1}\right) \cdot \cdots \cdot \left(1 - \frac{n-1}{m+n-1}\right) \tag{C.9}$$

$$> 1 - \frac{n^2}{m}. \tag{C.10}$$

□

Now let U be an $m \times m$ unitary matrix, and recall from Section 3.1 that $\varphi(U)$ is the "lifting" of U to the n-photon Hilbert space of dimension $M = \binom{m+n-1}{n}$. Also, let $A = A(U,n)$ be the $m \times n$ matrix corresponding to the first n columns of U. Then recall that $\mathcal{D}_A$ is the probability distribution over $\Phi_{m,n}$ obtained by drawing each basis state $S \in \Phi_{m,n}$ with probability equal to $|\langle 1_n|\varphi(U)|S\rangle|^2$.

Using the previous results, we can upper-bound the probability that a Haar-random unitary maps the basis state $|1_n\rangle$ to a basis state containing two or more photons in the same mode.

Theorem C.4 (Boson Birthday Bound). *Recalling that $\mathcal{H}_{m,m}$ is the Haar measure over $m \times m$ unitary matrices,*

$$\operatorname*{E}_{U \in \mathcal{H}_{m,m}}\left[\Pr_{\mathcal{D}_{A(U,n)}}[S \in B_{m,n}]\right] < \frac{2n^2}{m}. \tag{C.11}$$

Proof. Given a permutation $\sigma \in S_m$ of single-photon states (or equivalently of modes), let $\varphi(\sigma)$ be the permutation on the set $\Phi_{m,n}$ of n-photon states that is induced by σ and let $\Gamma := \{\varphi(\sigma) : \sigma \in S_m\}$. Then Γ is a subgroup of S_M of order $m!$ (where as before, $M = \binom{m+n-1}{n}$). Furthermore, Γ is transitive with respect to the set $G_{m,n}$ since we can map any collision-free basis state $S \in G_{m,n}$ to any other collision-free basis state $S' \in G_{m,n}$ via a suitable permutation $\sigma \in S_m$ of the underlying modes. Now let $\mathcal{U}$ be the probability distribution over $M \times M$ unitary matrices V that is obtained by first drawing an $m \times m$ unitary matrix U from $\mathcal{H}_{m,m}$ and then setting $V := \varphi(U)$. Then since $\mathcal{H}_{m,m}$ is symmetric with respect to permutations $\sigma \in S_m$ it follows that $\mathcal{U}$ is symmetric with respect to permutations $\varphi(\sigma) \in \Gamma$. We want to upper-bound

$$\operatorname*{E}_{U \in \mathcal{H}_{m,m}}\left[\Pr_{\mathcal{D}_{A(U,n)}}[S \in B_{m,n}]\right].$$

This is simply the probability that, after choosing an $m \times m$ unitary U from $\mathcal{H}_{m,m}$ applying the $M \times M$ unitary $\varphi(U)$ to the basis state $|1_n\rangle$ and then measuring in the Fock basis, we obtain an outcome in $G_{m,n}$. So

$$\operatorname*{E}_{U \in \mathcal{H}_{m,m}}\left[\Pr_{\mathcal{D}_{A(U,n)}}[S \in B_{m,n}]\right] \leq \frac{|B_{m,n}|}{|G_{m,n}|} < \frac{n^2/m}{1 - n^2/m}. \tag{C.12}$$

Here the first inequality follows from Corollary C.2 together with the fact that $1_n \in G_{m,n}$ while the second inequality follows from Proposition C.3. Since the expectation is in any case at most 1 we therefore have an upper bound of

$$\min\left\{\frac{n^2/m}{1 - n^2/m}, 1\right\} \leq \frac{2n^2}{m}. \tag{C.13}$$

□

Acknowledgments

We thank Boris Alexeev, Sanjeev Arora, Carlo Beenakker, Justin Dove, 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 for allowing us to include several of his results in

Appendix B; Mick Bremner and Richard Jozsa for discussions of their work [12]; Joel Tropp for pointing us to the work of Jiang [38]; and John Watrous for correcting an error in Figure 5. Finally, we thank the editor, Oded Regev, as well as the anonymous reviewers, for their thorough and detailed comments and for catching several mistakes.

References

[1] SCOTT AARONSON: Algorithms for Boolean function query properties. *SIAM J. Comput.*, 32(5):1140–1157, 2003. [doi:10.1137/S0097539700379644] 173

[2] SCOTT AARONSON: Quantum computing, postselection, and probabilistic polynomial-time. *Proc. Roy. Soc. A*, 461(2063):3473–3482, 2005. Preprint at arXiv. [doi:10.1098/rspa.2005.1546] 150, 159, 162, 175, 179

[3] SCOTT AARONSON: BQP and the polynomial hierarchy. In *Proc. 42nd STOC*, pp. 141–150. ACM Press, 2010. Preprint at arXiv. [doi:10.1145/1806689.1806711] 150, 160, 161

[4] SCOTT AARONSON: The equivalence of sampling and searching. In *6th Internat. Comp. Sci. Symp. in Russia (CSR'11)*, pp. 1–14. Springer, 2011. Available at ECCC. Preprint at arXiv. [doi:10.1007/978-3-642-20712-9_1] 151, 163

[5] SCOTT AARONSON AND DANIEL GOTTESMAN: Improved simulation of stabilizer circuits. *Phys. Rev. A*, 70(5):052328, 2004. Preprint at arXiv. [doi:10.1103/PhysRevA.70.052328] 156, 158

[6] DANIEL S. ABRAMS AND SETH LLOYD: Simulation of many-body Fermi systems on a universal quantum computer. *Phys. Rev. Lett.*, 79(13):2586–2589, 1997. Preprint at arXiv. [doi:10.1103/PhysRevLett.79.2586] 174

[7] DORIT AHARONOV AND MICHAEL BEN-OR: Fault-tolerant quantum computation with constant error rate. *SIAM J. Comput.*, 38(4):1207–1282, 2008. Preliminary version in STOC'97. Preprint at arXiv. [doi:10.1137/S0097539799359385] 146, 202

[8] SANJEEV ARORA, ARNAB BHATTACHARYYA, RAJSEKAR MANOKARAN, AND SUSHANT SACHDEVA: Testing permanent oracles - revisited. In *Proc. 16th Internat. Workshop on Randomization and Computation (RANDOM'12)*, pp. 362–373, 2012. Available at ECCC. Preprint at arXiv. [doi:10.1007/978-3-642-32512-0_31] 205, 228, 237

[9] STEPHEN D. BARTLETT AND BARRY C. SANDERS: Requirement for quantum computation. *J. Modern Optics*, 50(15-17):2331–2340, 2003. Preprint at arXiv. [doi:10.1080/09500340308233564] 156, 157, 199

[10] CARLO W. J. BEENAKKER, DAVID P. DIVINCENZO, CLIVE EMARY, AND MARCO KINDERMANN: Charge detection enables free-electron quantum computation. *Phys. Rev. Lett.*, 93(2):020501, 2004. Preprint at arXiv. [doi:10.1103/PhysRevLett.93.020501] 159

[11] ETHAN BERNSTEIN AND UMESH VAZIRANI: Quantum complexity theory. *SIAM J. Comput.*, 26(5):1411–1473, 1997. Preliminary version in STOC'93. [doi:10.1137/S0097539796300921] 146, 159

[12] MICHAEL J. BREMNER, RICHARD JOZSA, AND DAN J. SHEPHERD: Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proc. Roy. Soc. A*, 467(2126):459–472, 2011. Preprint at arXiv. [doi:10.1098/rspa.2010.0301] 149, 150, 159, 160, 179, 236, 246

[13] 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*, 2012 (online). Preprint at arXiv. [doi:10.1126/science.1231440] 154, 200

[14] HARRY BUHRMAN, RICHARD CLEVE, RONALD DE WOLF, AND CHRISTOF ZALKA: Bounds for small-error and zero-error quantum algorithms. In *Proc. 40th FOCS*, pp. 358–368. IEEE Comp. Soc. Press, 1999. Preprint at arXiv. [doi:10.1109/SFFCS.1999.814607] 233

[15] EDUARDO R. CAIANIELLO: On quantum field theory, I: explicit solution of Dyson's equation in electrodynamics without use of Feynman graphs. *Nuovo Cimento*, 10(12):1634–1652, 1953. [doi:10.1007/BF02781659] 158

[16] DAVID M. CEPERLEY: An overview of quantum Monte Carlo methods. *Reviews in Mineralogy and Geochemistry*, 71(1):129–135, 2010. [doi:10.2138/rmg.2010.71.6] 159

[17] RICHARD CLEVE AND JOHN WATROUS: Fast parallel circuits for the quantum Fourier transform. In *Proc. 41st FOCS*, pp. 526–536. IEEE Comp. Soc. Press, 2000. Preprint at arXiv. [doi:10.1109/SFCS.2000.892140] 146

[18] KEVIN P. COSTELLO AND VAN H. VU: Concentration of random determinants and permanent estimators. *SIAM J. Discrete Math.*, 23(3):1356–1371, 2009. Preprint at arXiv. [doi:10.1137/080733784] 221

[19] ANDREA CRESPI, ROBERTO OSELLAME, ROBERTA RAMPONI, DANIEL J. BROD, ERNESTO F. GALVÃO, NICOLÒ SPAGNOLO, CHIARA VITELLI, ENRICO MAIORINO, PAOLO MATALONI, AND FABIO SCIARRINO: Experimental boson sampling in arbitrary integrated photonic circuits. 2012. [arXiv:1212.2783] 154, 200

[20] CONSTANTINOS DASKALAKIS, PAUL W. GOLDBERG, AND CHRISTOS H. PAPADIMITRIOU: The complexity of computing a Nash equilibrium. *Commun. ACM*, 52(2):89–97, 2009. [doi:10.1145/1461928.1461951] 163

[21] CONSTANTINOS DASKALAKIS, PAUL W. GOLDBERG, AND CHRISTOS H. PAPADIMITRIOU: The complexity of computing a Nash equilibrium. *SIAM J. Comput.*, 39(1):195–259, 2009. Preliminary version in STOC'06. [doi:10.1137/070699652] 163

[22] GEORGY P. EGORYCHEV: Proof of the van der Waerden conjecture for permanents. *Sibirsk. Mat. Zh.*, 22(6):65–71, 1981. English translation in Siberian Math. J. 22 (6), pp. 854-859, 1981. [doi:10.1007/BF00968054] 228

[23] DMITRY I. FALIKMAN: Proof of the van der Waerden conjecture regarding the permanent of a doubly stochastic matrix. *Mat. Zametki*, 29(6):931–938, 1981. English translation in Math. Notes 29 (6), pp. 475-479, 1981. [doi:10.1007/BF01163285] 228

[24] BILL FEFFERMAN AND CHRIS UMANS: Pseudorandom generators and the BQP vs. PH problem. 2010. [arXiv:1007.0305] 150

[25] STEPHEN FENNER, FREDERIC GREEN, STEVEN HOMER, AND RANDALL PRUIM: Determining acceptance possibility for a quantum computation is hard for the polynomial hierarchy. *Proc. Roy. Soc. A*, 455(1991):3953–3966, 1999. Preprint at arXiv. [doi:10.1098/rspa.1999.0485] 159

[26] RICHARD P. FEYNMAN: Simulating physics with computers. *Int. J. Theoretical Physics*, 21(6-7):467–488, 1982. [doi:10.1007/BF02650179] 174

[27] LANCE FORTNOW AND JOHN D. ROGERS: Complexity limitations on quantum computation. *J. Comput. System Sci.*, 59(2):240–252, 1999. Preliminary version in CCC'98. Preprint at arXiv. [doi:10.1006/jcss.1999.1651] 237

[28] PETER GEMMELL, RICHARD J. LIPTON, RONITT RUBINFELD, MADHU SUDAN, AND AVI WIGDERSON: Self-testing/correcting for polynomials and for approximate functions. In *Proc. 23rd STOC*, pp. 33–42. ACM Press, 1991. [doi:10.1145/103418.103429] 229

[29] PETER GEMMELL AND MADHU SUDAN: Highly resilient correctors for polynomials. *Inform. Process. Lett.*, 43(4):169–174, 1992. [doi:10.1016/0020-0190(92)90195-2] 229

[30] VIACHESLAV L. GIRKO: A refinement of the Central Limit Theorem for random determinants. *Teor. Veroyatnost. i Primenen*, 42(1):63–73, 1997. Translation in Theory Probab. Appl 42 (1) (1998), 121-129. [doi:10.1137/S0040585X97975939] 221

[31] CHRIS D. GODSIL AND IVAN GUTMAN: On the matching polynomial of a graph. In *Algebraic Methods in Graph Theory I*, pp. 241–249. North Holland, 1981. 219

[32] DANIEL GOTTESMAN, ALEXEI KITAEV, AND JOHN PRESKILL: Encoding a qubit in an oscillator. *Phys. Rev. A*, 64(1):012310, 2001. Preprint at arXiv. [doi:10.1103/PhysRevA.64.012310] 155, 156

[33] LEONID GURVITS: On the complexity of mixed discriminants and related problems. In *30th Internat. Symp. Mathematical Foundations of Computer Science (MFCS'05)*, pp. 447–458. Springer, 2005. [doi:10.1007/11549345_39] 153, 157, 164, 228, 239, 240

[34] YENJO HAN, LANE A. HEMASPAANDRA, AND THOMAS THIERAUF: Threshold computation and cryptographic security. *SIAM J. Comput.*, 26(1):59–78, 1997. Preliminary version in ISAAC'93. [doi:10.1137/S0097539792240467] 162

[35] JOHAN HÅSTAD: *Computational Limitations for Small Depth Circuits*. MIT Press, 1987. 160

[36] C. K. HONG, ZHE-YU OU, AND LEONARD MANDEL: Measurement of subpicosecond time intervals between two photons by interference. *Phys. Rev. Lett.*, 59(18):2044–2046, 1987. [doi:10.1103/PhysRevLett.59.2044] 158, 196, 200

[37] MARK JERRUM, ALISTAIR SINCLAIR, AND ERIC VIGODA: A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *J. ACM*, 51(4):671–697, 2004. Preliminary version in STOC'01. [doi:10.1145/1008731.1008738] 148, 175, 228

[38] TIEFENG JIANG: How many entries of a typical orthogonal matrix can be approximated by independent normals? *Ann. Probab.*, 34(4):1497–1529, 2006. [doi:10.1214/009117906000000205] 183, 200, 246

[39] STEPHEN P. JORDAN: Permutational quantum computing. *Quantum Information & Computation*, 10(5):470–497, 2010. Preprint at arXiv. [ACM:2011369] 157

[40] SUBHASH KHOT: On the Unique Games Conjecture. In *Proc. 25th IEEE Conf. on Computational Complexity (CCC'10)*, pp. 99–121. IEEE Comp. Soc. Press, 2010. [doi:10.1109/CCC.2010.19] 152

[41] EMANUEL KNILL: Fermionic linear optics and matchgates. 2001. [arXiv:quant-ph/0108033] 158

[42] EMANUEL KNILL AND RAYMOND LAFLAMME: Power of one bit of quantum information. *Phys. Rev. Lett.*, 81(25):5672–5675, 1998. Preprint at arXiv. [doi:10.1103/PhysRevLett.81.5672] 157

[43] EMANUEL KNILL, RAYMOND LAFLAMME, AND GERARD J. MILBURN: A scheme for efficient quantum computation with linear optics. *Nature*, 409(6816):46–52, 2001. See also arXiv:quant-ph/0006088. [doi:10.1038/35051009] 150, 155, 156, 157, 175, 179, 180, 181

[44] EMANUEL KNILL, RAYMOND LAFLAMME, AND WOJCIECH H. ZUREK: Resilient quantum computation. *Science*, 279(5349):342–345, 1998. Preprint at arXiv. [doi:10.1126/science.279.5349.342] 146, 202

[45] KENNETH LANGE: *Applied Probability*. Springer, 2003. 224

[46] YUAN LIANG LIM AND ALMUT BEIGE: Generalized Hong-Ou-Mandel experiments with bosons and fermions. *New J. Phys.*, 7(155):1–10, 2005. Preprint at arXiv. [doi:10.1088/1367-2630/7/1/155] 158

[47] RICHARD J. LIPTON: New directions in testing. In *Distributed Computing and Cryptography*, pp. 191–202. AMS, 1991. 153, 228, 229

[48] BRAHIM LOUNIS AND MICHEL ORRIT: Single-photon sources. *Reports on Progress in Physics*, 68(5):1129–1179, 2005. [doi:10.1088/0034-4885/68/5/R04] 199

[49] CHRISTIAN MASTRODONATO AND RODERICH TUMULKA: Elementary proof for asymptotics of large Haar-distributed unitary matrices. *Letters in Mathematical Physics*, 82(1):51–59, 2007. Preprint at arXiv. [doi:10.1007/s11005-007-0194-7] 184

[50] MICHAEL A. NIELSEN AND ISAAC L. CHUANG: *Quantum Computation and Quantum Information*. Cambridge University Press, 2000. 156, 174

[51] RAMAMOHAN PATURI: On the degree of polynomials that approximate symmetric Boolean functions. In *Proc. 24th STOC*, pp. 468–474. ACM Press, 1992. [doi:10.1145/129712.129758] 233

[52] DÉNES PETZ AND JÚLIA RÉFFY: On asymptotics of large Haar distributed unitary matrices. *Periodica Mathematica Hungarica*, 49(1):103–117, 2004. Preprint at arXiv. [doi:10.1023/B:MAHU.0000040542.56072.ab] 184

[53] DÉNÉS PETZ AND JÚLIA RÉFFY: Large deviation theorem for the empirical eigenvalue distribution of truncated Haar unitary matrices. *Prob. Theory and Related Fields*, 133(2):175–189, 2005. Preprint at arXiv. [doi:10.1007/s00440-004-0420-5] 184

[54] MICHAEL RECK, ANTON ZEILINGER, HERBERT J. BERNSTEIN, AND PHILIP BERTANI: Experimental realization of any discrete unitary operator. *Phys. Rev. Lett.*, 73(1):58–61, 1994. [doi:10.1103/PhysRevLett.73.58] 165, 201

[55] JÚLIA RÉFFY: *Asymptotics of random unitaries*. Ph. D. thesis, Budapest University of Technology and Economics, 2005. `http://www.math.bme.hu/~reffyj/disszer.pdf`. 184, 185

[56] TERRY RUDOLPH: A simple encoding of a quantum circuit amplitude as a matrix permanent. *Phys. Rev. A*, 80(5), 2009. [doi:10.1103/PhysRevA.80.054302, arXiv:0909.3005] 159

[57] STEFAN SCHEEL: Permanents in linear optical networks. 2004. [arXiv:quant-ph/0406127] 158

[58] DAN SHEPHERD AND MICHAEL J. BREMNER: Temporally unstructured quantum computation. *Proc. Roy. Soc. A*, 465(2105):1413–1439, 2009. Preprint at arXiv. [doi:10.1098/rspa.2008.0443] 160

[59] YAOYUN SHI: Both Toffoli and controlled-NOT need little help to do universal quantum computing. *Quantum Information & Computation*, 3(1):84–92, 2003. Preprint at arXiv. [ACM:2011515] 181

[60] PETER W. SHOR: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997. Preliminary version in FOCS'94. Preprint at arXiv. [doi:10.1137/S0097539795293172] 146, 161

[61] DANIEL R. SIMON: On the power of quantum computation. *SIAM J. Comput.*, 26(5):1474–1483, 1997. Preliminary version in FOCS'94. [doi:10.1137/S0097539796298637] 161

[62] JUSTIN B. SPRING, BENJAMIN J. METCALF, PETER C. HUMPHREYS, W. STEVEN KOLTHAMMER, XIAN-MIN JIN, MARCO BARBIERI, ANIMESH DATTA, NICHOLAS THOMAS-PETER, NATHAN K. LANGFORD, DMYTRO KUNDYS, JAMES C. GATES, BRIAN J. SMITH, PETER G. R. SMITH, AND IAN A. WALMSLEY: Boson sampling on a photonic chip. *Science*, 2012 (online). Preprint at arXiv. [doi:10.1126/science.1231692] 154, 200

[63] LARRY J. STOCKMEYER: On approximation algorithms for #P. *SIAM J. Comput.*, 14(4):849–861, 1985. Preliminary version in STOC'83. [doi:10.1137/0214060] 175

[64] MADHU SUDAN: Maximum likelihood decoding of Reed Solomon codes. In *Proc. 37th FOCS*, pp. 164–172. IEEE Comp. Soc. Press, 1996. [doi:10.1109/SFCS.1996.548475] 229

[65] TERENCE TAO AND VAN H. VU: On the permanent of random Bernoulli matrices. *Advances in Mathematics*, 220(3):657–669, 2009. Preprint at arXiv. [doi:10.1016/j.aim.2008.09.006] 153, 218, 219

[66] BARBARA M. TERHAL AND DAVID P. DIVINCENZO: Classical simulation of noninteracting-fermion quantum circuits. *Phys. Rev. A*, 65(3):032325, 2002. Preprint at arXiv. [doi:10.1103/PhysRevA.65.032325] 158

[67] BARBARA M. TERHAL AND DAVID P. DIVINCENZO: Adaptive quantum computation, constant-depth quantum circuits and Arthur-Merlin games. *Quantum Information & Computation*, 4(2):134–145, 2004. Preprint at arXiv. [ACM:2011582] 160, 180

[68] MAX TILLMANN, BORIVOJE DAKIĆ, RENÉ HEILMANN, STEFAN NOLTE, ALEXANDER SZAMEIT, AND PHILIP WALTHER: Experimental boson sampling. 2012. [arXiv:1212.2240] 154, 200

[69] SEINOSUKE TODA: PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991. Preliminary version in FOCS'89. [doi:10.1137/0220053] 150, 154, 179

[70] LIDROR TROYANSKY AND NAFTALI TISHBY: Permanent uncertainty: On the quantum evaluation of the determinant and the permanent of a matrix. In *Proc. 4th Workshop on Physics and Computation (PhysComp'96)*, 1996. Available here. 158

[71] LESLIE G. VALIANT: The complexity of computing the permanent. *Theoret. Comput. Sci.*, 8(2):189–201, 1979. [doi:10.1016/0304-3975(79)90044-6] 158, 175

[72] LESLIE G. VALIANT: Quantum circuits that can be simulated classically in polynomial time. *SIAM J. Comput.*, 31(4):1229–1254, 2002. Preliminary version in STOC'01. [doi:10.1137/S0097539700377025] 158

[73] LIEVEN M. K. VANDERSYPEN, MATTHIAS STEFFEN, GREGORY BREYTA, COSTANTINO S. YANNONI, MARK H. SHERWOOD, AND ISAAC L. CHUANG: Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414(6866):883–887, 2001. Preprint at arXiv. [doi:10.1038/414883a] 155

[74] JIN YI CAI, ADURI PAVAN, AND D. SIVAKUMAR: On the hardness of permanent. In *Proc. 16th Symp. Theoretical Aspects of Comp. Sci. (STACS'99)*, pp. 90–99, 1999. [doi:10.1007/3-540-49116-3_8] 229

AUTHORS

Scott Aaronson
associate professor
Massachusetts Institute of Technology, Cambridge, MA
aaronson@csail.mit.edu
http://www.scottaaronson.com

Alex Arkhipov
Ph. D. student
Massachusetts Institute of Technology, Cambridge, MA
arkhipov@mit.edu

ABOUT THE AUTHORS

SCOTT AARONSON received his bachelor's degree from Cornell University and his Ph. D. from UC Berkeley. He is known for his blog and for founding the Complexity Zoo. He publishes often in Theory of Computing.

ALEX ARKHIPOV is a Ph. D. student at MIT, where he also received a bachelor's degree in mathematics. His advisor is Scott Aaronson. His research interests are in quantum computing and complexity theory. In his spare time, Alex enjoys strategy board games, role-playing games, and juggling.