Generalizing and Derandomizing Gurvits's Approximation Algorithm for the Permanent

Scott Aaronson* Travis Hance†

Abstract

Around 2002, Leonid Gurvits gave a striking randomized algorithm to approximate the permanent of an $n \times n$ matrix A. The algorithm runs in $O\left(n^2/\varepsilon^2\right)$ time, and approximates $\mathrm{Per}\,(A)$ to within $\pm\varepsilon \|A\|^n$ additive error. A major advantage of Gurvits's algorithm is that it works for *arbitrary* matrices, not just for nonnegative matrices. This makes it highly relevant to *quantum optics*, where the permanents of bounded-norm complex matrices play a central role. Indeed, the existence of Gurvits's algorithm is why, in their recent work on the *hardness* of quantum optics, Aaronson and Arkhipov (AA) had to talk about sampling problems rather than estimation problems.

In this paper, we improve Gurvits's algorithm in two ways. First, using an idea from quantum optics, we generalize the algorithm so that it yields a better approximation when the matrix A has either repeated rows or repeated columns. Translating back to quantum optics, this lets us classically estimate the probability of *any* outcome of an AA-type experiment—even an outcome involving multiple photons "bunched" in the same mode—at least as well as that probability can be estimated by the experiment itself. (This does not, of course, let us solve the AA sampling problem.) It also yields a general upper bound on the probabilities of "bunched" outcomes, which resolves a conjecture of Gurvits and might be of independent physical interest.

Second, we use ε-biased sets to derandomize Gurvits's algorithm, in the special case where the matrix A is nonnegative. More interestingly, we generalize the notion of ε-biased sets to the complex numbers, construct "complex ε-biased sets," then use those sets to derandomize even our generalization of Gurvits's algorithm to the multirow/multicolumn case (again for nonnegative A). Whether Gurvits's algorithm can be derandomized for general A remains an outstanding problem.

1 Introduction

The *permanent* of an $n \times n$ matrix has played a major role in combinatorics and theoretical computer science for decades. Today this function also plays an increasing role in quantum computing, because of the fact (pointed out by Caianiello [5] and Troyansky and Tishby [16] among others) that the transition amplitudes for n identical, non-interacting bosons are given by $n \times n$ permanents.

Recall that the permanent of an $n \times n$ matrix $A = (a_{ij})$, with entries over any field, is defined as follows:

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

*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, an NSF STC grant, a TIBCO Chair, a Sloan Fellowship, and an Alan T. Waterman Award.

†MIT. Email: tjhance7@gmail.com

where S_n denotes the set of permutations of $\{1, \ldots, n\}$. A great deal is known about the complexity of computing $\mathrm{Per}(A)$, given A as input. Most famously, Valiant [17] showed in 1979 that computing $\mathrm{Per}(A)$ *exactly* is #P-complete, even if A is a matrix of nonnegative integers (or indeed, if all of its entries are either 0 or 1). On the other hand, in a celebrated application of the Markov-Chain Monte-Carlo method, Jerrum, Sinclair, and Vigoda [12] gave a randomized algorithm to *approximate* $\mathrm{Per}(A)$ to multiplicative error ε, in $\mathrm{poly}(n, 1/\varepsilon)$ time, in the special case where A is nonnegative. When A is an *arbitrary* matrix (which could include negative or complex entries), it is easy to show that even approximating $\mathrm{Per}(A)$ to polynomial multiplicative error remains a #P-hard problem; see Aaronson [1] for example.

We can, on the other hand, hope to estimate $\mathrm{Per}(A)$ to within nontrivial *additive* error for arbitrary $A \in \mathbb{C}^{n \times n}$. And that is exactly what a simple randomized algorithm due to Gurvits [8] does: it approximates $\mathrm{Per}(A)$ to within additive error $\pm\varepsilon \|A\|^n$ in $O\left(n^2/\varepsilon^2\right)$ time. Here $\|A\|$ represents the largest *singular value* of A. The appearance of this linear-algebraic quantity when discussing the permanent becomes a little less surprising once one knows the inequality

$$|\mathrm{Per}(A)| \leq \|A\|^n . \tag{2}$$

See Section 2 for Gurvits's algorithm, the proof of its correctness, and the closely-related proof of the inequality (2).

In particular, if U is a *unitary* matrix, or a submatrix of a unitary matrix, then $\|U\| \leq 1$, so Gurvits's algorithm approximates $\mathrm{Per}(U)$ to within $\pm\varepsilon$ additive error in $O\left(n^2/\varepsilon^2\right)$ time. Equivalently, the algorithm lets us detect when $\mathrm{Per}(U)$ is "anomalously close to 1" (note that, if U is a *random* unitary matrix, then $\mathrm{Per}(U)$ will usually be exponentially small, and 0 will accordingly be a good additive estimate to $\mathrm{Per}(U)$). This observation makes Gurvits's algorithm highly relevant to the field of quantum optics, where subunitary matrices play a central role.

1.1 Our Results

In this paper, motivated by the problem of simulating quantum optics on a classical computer, we study how far Gurvits's algorithm can be improved. We present two sets of results on this question.

First, we generalize Gurvits's algorithm, as well as the inequality (2), to perform better on matrices with repeated rows or repeated columns. Let $B \in \mathbb{C}^{n \times k}$ be an $n \times k$ complex matrix, and let $s_1, \ldots, s_k$ be positive integers summing to n. Also, let $A \in \mathbb{C}^{n \times n}$ be an $n \times n$ matrix consisting of s_i copies of the i^{th} column of B, for all $i \in [k]$. Then we give a randomized algorithm that takes $O\left(nk/\varepsilon^2\right)$ time, and that estimates $\mathrm{Per}(A)$ to within an additive error

$$\pm\varepsilon \cdot \frac{s_1! \cdots s_k!}{\sqrt{s_1^{s_1} \cdots s_k^{s_k}}} \|B\|^n . \tag{3}$$

Interestingly, we obtain this generalization by interpreting certain formal variables $x_1, \ldots, x_k$ arising in Gurvits's algorithm as what a physicist would call "bosonic creation operators," and then doing what would be natural for creation operators (e.g., replacing each x_i by $\sqrt{s_i} x_i$).

We also prove the general inequality

$$|\mathrm{Per}(A)| \leq \frac{s_1! \cdots s_k!}{\sqrt{s_1^{s_1} \cdots s_k^{s_k}}} \|B\|^n . \tag{4}$$

Notice that both of these results reduce to Gurvits's in the special case $k = n$ and $s_1 = \cdots = s_n = 1$, but in general they can be much better (for example, if we have information about $\|B\|$ but not $\|A\|$). We discuss the quantum-optics motivation for these improvements in Section 1.3.

Second, we show that both Gurvits's algorithm and our generalization of it can be *derandomized*, in the special case where all matrix entries are nonnegative. That is, for nonnegative $n \times n$ matrices $A \in \mathbb{R}^{n \times n}_{\geq 0}$, we show that we can approximate $\mathrm{Per}\,(A)$ to within additive error $\pm\varepsilon \cdot \|A\|^n$, deterministically and in $\mathrm{poly}\,(n, 1/\varepsilon)$ time. If $A \in \mathbb{R}^{n \times n}_{\geq 0}$ is obtained from an $n \times k$ matrix $B \in \mathbb{R}^{n \times k}_{\geq 0}$ as described above, then we also show that we can approximate $\mathrm{Per}\,(A)$ to within additive error

$$\pm\varepsilon \cdot \frac{s_1! \cdots s_k!}{\sqrt{s_1^{s_1} \cdots s_k^{s_k}}} \|B\|^n, \tag{5}$$

deterministically and in $\mathrm{poly}\,(n, 1/\varepsilon)$ time.

To derandomize Gurvits's original algorithm, the idea is simply to use ε-biased sets. To derandomize our generalization of Gurvits's algorithm is more interesting: there, we need to generalize the notion of ε-biased sets to complex roots of unity, then explicitly construct the "complex ε-biased sets" that we need.

Let us compare our algorithm to a previous deterministic approximation algorithm for the permanent. Gurvits and Samorodnitsky [9] gave a deterministic algorithm for estimating the *mixed discriminant* of a positive semidefinite matrix up to a multiplicative error of e^n. The permanent of a nonnegative matrix is a special case of this. Our algorithm improves over Gurvits and Samorodnitsky's [9] when $\mathrm{Per}\,(A)$ is large, i.e., close to $\|A\|^n$ or to $\frac{s_1! \cdots s_k!}{\sqrt{s_1^{s_1} \cdots s_k^{s_k}}} \|B\|^n$.

Of course, for nonnegative A, the algorithm of Jerrum, Sinclair, Vigoda [12] (JSV) gives a better approximation to $\mathrm{Per}\,(A)$ than any of the algorithms discussed above (including ours). Crucially, though, no one seems to have any idea how to derandomize JSV. One motivation for trying to derandomize Gurvits's algorithm is the hope that it *might* be a stepping-stone toward a derandomization of the much more complicated JSV algorithm.

1.2 Quantum Optics

This paper is about classical algorithms for a classical permanent-estimation problem. However, the motivation and even one of the algorithmic ideas came from quantum optics, so a brief discussion of the latter might be helpful. For details of quantum optics from a computer-science perspective, see for example Aaronson [1] or Aaronson and Arkhipov [2].

In *quantum (linear) optics*, one considers states involving identical, non-interacting photons (or other bosonic particles), which can move from one location (or "mode") to another, but are never created or destroyed. In more detail, the photons are in a superposition of basis states, each of the form $|s_1, \ldots, s_k\rangle$, like so:

$$|\Phi\rangle = \sum_{s_1, \ldots, s_k \geq 0 \,:\, s_1 + \cdots + s_k = n} \alpha_{s_1, \ldots, s_k} |s_1, \ldots, s_k\rangle\,, \tag{6}$$

for some complex numbers $\alpha_{s_1, \ldots, s_k}$ satisfying

$$\sum_{s_1, \ldots, s_k \geq 0 \,:\, s_1 + \cdots + s_k = n} |\alpha_{s_1, \ldots, s_k}|^2 = 1. \tag{7}$$

Here s_i is a nonnegative integer representing the number of photons in the i^{th} mode, and $s_1 + \cdots + s_k = n$ is the total number of photons. To modify the state $|\Phi\rangle$, one applies a network of "beamsplitters" and other optical elements, which induces some $k \times k$ unitary transformation U acting on the k modes. Via homomorphism, the $k \times k$ unitary U gives rise to a $\binom{k+n-1}{n} \times \binom{k+n-1}{n}$ unitary $\varphi(U)$ acting on the n-photon state $|\Phi\rangle$. Crucially, each entry of the "large" unitary $\varphi(U)$ is defined in terms of the *permanent* of a matrix formed from entries of the "small" unitary U. The formula is as follows:

$$\langle s_1, \ldots, s_k | \varphi(U) | t_1, \ldots, t_k \rangle = \frac{\mathrm{Per}\left(U_{s_1,\ldots,s_k,t_1,\ldots,t_k}\right)}{\sqrt{s_1! \cdots s_k! t_1! \cdots t_k!}}. \tag{8}$$

Here $U_{s_1,\ldots,s_k,t_1,\ldots,t_k}$ is the $n \times n$ matrix formed from U by taking s_i copies of the i^{th} row of U for all $i \in [k]$, and t_j copies of the j^{th} column of U for all $j \in [k]$. It can be checked [2, 1]—it is not obvious!—that φ is a homomorphism, and that $\varphi(U)$ is unitary for all U. (Indeed, the "reason" for the scaling term $\sqrt{s_1! \cdots s_k! t_1! \cdots t_k!}$ is to ensure these properties.)

Thus, in quantum optics, *calculating amplitudes reduces to calculating permanents of the above form.* By the standard rules of quantum mechanics, the probabilities of measurement outcomes can then be obtained by taking the absolute squares of the amplitudes.

Intuitively, the reason why the permanent arises here is simply that (by assumption) the n photons are identical, and therefore we need to sum over all $n!$ possible permutations of photons, each of which contributes a term to the final amplitude. Indeed, even in a classical situation, with n indistinguishable balls each thrown independently into one of k bins, the probability of a particular outcome (for example: 2 balls landing in the first bin, 0 balls landing in the second bin, etc.) could be expressed in terms of the permanent of a matrix A of transition probabilities. The difference is that, in the classical case, this A would be a *nonnegative* matrix. And therefore, $\mathrm{Per}(A)$ could be estimated in randomized polynomial time using the JSV algorithm [12]. In the quantum case, by contrast, we want to estimate $|\mathrm{Per}(A)|^2$ for a matrix A of *complex* numbers—and here is it known that multiplicative estimation is already a #P-hard problem [2]. That is why we need to settle for *additive* approximation—or equivalently, for approximation in the special case that $|\mathrm{Per}(A)|^2$ happens to be "anomalously large" (i.e., non-negligible compared to the general upper bound that we prove on $|\mathrm{Per}(A)|^2$).

1.3 Implications of Our Results for Quantum Optics

With that brief tour of quantum optics out of the way, we can now explain the implications for quantum optics of our first set of results (the ones generalizing Gurvits's algorithm to the multirow/multicolumn case).

Consider the "standard initial state" $|1, \ldots, 1, 0, \ldots, 0\rangle$, which consists of n identical photons, each in a separate mode, and the remaining $m - n$ modes unoccupied. Suppose we pass the n photons through a network of beamsplitters, then measure the numbers of photons in each of k output modes. Given nonnegative integers $s_1, \ldots, s_k$ summing to n, let p be the probability that exactly s_1 photons are found in the first output mode, exactly s_2 are found in the second output mode, and so on up to s_k. Then by (8), we have

$$p = \frac{|\mathrm{Per}(A)|^2}{s_1! \cdots s_k!}, \tag{9}$$

where A is the $n \times n$ matrix formed by taking s_i copies of the i^{th} row of the mode-mixing unitary U, for all $i \in [k]$. From this, together with the inequality (4), we have the upper bound

$$p \leq \frac{s_1! \cdots s_k!}{s_1^{s_1} \cdots s_k^{s_k}}, \tag{10}$$

and for the associated amplitude α,

$$|\alpha| \leq \sqrt{\frac{s_1! \cdots s_k!}{s_1^{s_1} \cdots s_k^{s_k}}}. \tag{11}$$

As a sample application of (10), suppose $s_1 = n$ and $s_2 = \cdots = s_k = 0$. Then

$$p \leq \frac{n!}{n^n} \approx \frac{1}{e^n}. \tag{12}$$

This says that, regardless of what unitary transformation U we apply, we can never get n identical photons, initially in n separate modes, to "congregate" into a single one of those modes with more than $\sim n/e^n$ probability (the factor of n arising from taking the union bound over the modes). Or in other words, for $n \geq 3$, there is no direct counterpart to the *Hong-Ou-Mandel* dip [10], the famous effect that causes $n = 2$ photons initially in separate modes to congregate into the same mode with probability 1.[1]

Let us remark that the bound (10) is tight. To saturate it, let the $n \times n$ unitary U be block-diagonal with k blocks, of sizes $s_1 \times s_1, \ldots, s_k \times s_k$. Also, let the i^{th} block consist of the s_i-dimensional Quantum Fourier Transform, or any other $s_i \times s_i$ unitary matrix whose top row equals $\left(1/\sqrt{s_i}, \ldots, 1/\sqrt{s_i}\right)$. Then one can calculate that the probability of observing s_1 photons in the first mode of the first block, s_2 photons in the first mode of the second block, and so on is precisely

$$\left|\langle 1, \ldots, 1 | \varphi(U) | s_1, 0, \ldots, 0, s_2, 0, \ldots, 0, \ldots \rangle\right|^2 = \frac{s_1! \cdots s_k!}{s_1^{s_1} \cdots s_k^{s_k}}. \tag{13}$$

Here is another implication of our results for quantum optics. Given a description of the unitary transformation U applied by the beamsplitter network, the bound (3) implies the existence of a randomized algorithm, taking $O\left(nk/\varepsilon^2\right)$ time, that estimates the amplitude $\alpha = \frac{\mathrm{Per}(A)}{\sqrt{s_1! \cdots s_k!}}$ to within an additive error of

$$\varepsilon \cdot \sqrt{\frac{s_1! \cdots s_k!}{s_1^{s_1} \cdots s_k^{s_k}}} \|B\|^n \leq \varepsilon \cdot \sqrt{\frac{s_1! \cdots s_k!}{s_1^{s_1} \cdots s_k^{s_k}}} \tag{14}$$

$$\leq \varepsilon, \tag{15}$$

and likewise estimates the probability $p = |\alpha|^2$ to within an additive error of

$$\varepsilon \cdot |\alpha| \sqrt{\frac{s_1! \cdots s_k!}{s_1^{s_1} \cdots s_k^{s_k}}} \|B\|^n \leq \varepsilon \cdot \frac{s_1! \cdots s_k!}{s_1^{s_1} \cdots s_k^{s_k}} \|B\|^n \tag{16}$$

$$\leq \varepsilon \cdot \frac{s_1! \cdots s_k!}{s_1^{s_1} \cdots s_k^{s_k}} \tag{17}$$

$$\leq \varepsilon. \tag{18}$$

[1] When $n = 2$, we get $p \leq 1/2$, which corresponds exactly to the Hong-Ou-Mandel dip: the 2 photons have probability 1/2 of both being found in the first mode, and probability 1/2 of both being found in the second mode.

Observe that the above guarantees match Gurvits's in the special case that all s_i's are equal to 1, but they become *better* than Gurvits's when $\sum_i |s_i - 1|$ is large. In the latter case, (10) says that the probability p is exponentially small, but (16) says that we can *nevertheless* get a decent estimate of p.

2 Gurvits's Randomized Algorithm for Permanent Estimation

The starting point for Gurvits's algorithm is the following well-known formula for the permanent of an $n \times n$ matrix, called *Ryser's formula* [15]:

$$\mathrm{Per}(A) = \sum_{x=(x_1,\ldots,x_n)\in\{0,1\}^n} (-1)^{x_1+\cdots+x_n} \prod_{i=1}^{n} (a_{i,1}x_1 + \cdots + a_{i,n}x_n). \tag{19}$$

Ryser's formula leads to an $O\left(2^n n^2\right)$ algorithm for computing the permanent exactly, which can be improved to $O\left(2^n n\right)$ by cleverly iterating through the x's in Gray code order. This remains the fastest-known exact algorithm for the permanent.[2]

To present Gurvits's randomized algorithm, we will use a similar formula due to Glynn [7], which pulls from the domain $\{-1,1\}^n$ rather than $\{0,1\}^n$. We state Glynn's formula in terms of the expectation of a random variable. For a given $x \in \{-1,1\}^n$, define the *Glynn estimator* of an $n \times n$ matrix A as

$$\mathrm{Gly}_x(A) := x_1 \cdots x_n \prod_{i=1}^{n} (a_{i,1}x_1 + \cdots + a_{i,n}x_n). \tag{20}$$

Then we have

$$\mathrm{Per}\,(A) = \mathop{\mathrm{E}}_{x\in\{-1,1\}^n} [\mathrm{Gly}_x(A)]. \tag{21}$$

To see why, we just need to expand the product:

$$\mathop{\mathrm{E}}_{x\in\{-1,1\}^n} [\mathrm{Gly}_x\,(A)] = \sum_{\sigma_1,\ldots,\sigma_n\in[n]} a_{1,\sigma_1}\cdots a_{n,\sigma_n} \mathop{\mathrm{E}}_{x\in\{-1,1\}^n} [(x_1\cdots x_n)(x_{\sigma_1}\cdots x_{\sigma_n})]. \tag{22}$$

Then, note that $\mathrm{E}_{x\in\{-1,1\}^n}[(x_1\cdots x_n)(x_{\sigma_1}\cdots x_{\sigma_n})]$ is 1 if the map $i \mapsto \sigma_i$ is a permutation, and 0 otherwise. Hence the above sum is simply $\mathrm{Per}(A)$.

As a special case of a more general algorithm for *mixed discriminants*, Gurvits [8] gave a polynomial-time randomized sampling algorithm for the permanent. His algorithm is simply the following: for some $m = O\left(1/\varepsilon^2\right)$, first choose n-bit strings $x_1, \ldots, x_m \in \{-1,1\}^n$ uniformly and independently at random. Then compute $\mathrm{Gly}_{x_j}(A)$ for all $j \in [m]$, and output

$$\frac{\mathrm{Gly}_{x_1}(A) + \cdots + \mathrm{Gly}_{x_m}(A)}{m} \tag{23}$$

as the estimate for $\mathrm{Per}\,(A)$.

[2] Recently Björklund [4] discovered a slightly faster algorithm, taking $2^{n-\Omega\left(\sqrt{n/\log n}\right)}$ time, for computing the permanent of a matrix of poly (n)-bit integers.

Since each $\mathrm{Gly}_{x_i}(A)$ can be computed in $O(n^2)$ time, the algorithm clearly takes $O(n^2/\varepsilon^2)$ time. The algorithm's correctness follows from a single (important) fact. Recall that $\|A\|$ denotes the largest singular value of A, or equivalently,

$$\|A\| := \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|} \tag{24}$$

where $\|x\|$ denotes the 2-norm of the vector x.

Proposition 1. *For any $n \times n$ complex matrix A, we have $|\mathrm{Gly}_x(A)| \leq \|A\|^n$.*

Proof. For any $x \in \{1,-1\}^n$, we have by the arithmetic-geometric mean inequality,

$$|\mathrm{Gly}_x(A)| \leq \left(\sqrt{\frac{\sum_{i=1}^n |a_{i,1}x_1 + \cdots + a_{i,n}x_n|^2}{n}} \right)^n \tag{25}$$

$$= \left(\frac{\|Ax\|}{\|x\|} \right)^n \tag{26}$$

$$\leq \|A\|^n . \tag{27}$$

□

To summarize, $\mathrm{Gly}_x(A)$ is a random variable that is bounded by $\|A\|^n$ in absolute value, and whose expectation is $\mathrm{Per}(A)$. This implies that

$$|\mathrm{Per}(A)| \leq \|A\|^n . \tag{28}$$

By a standard Chernoff bound, it also implies that we can estimate $\mathrm{Per}(A)$ to additive error $\pm\varepsilon\|A\|^n$, with high probability, by taking the empirical mean of $\mathrm{Gly}_x(A)$ for $O(1/\varepsilon^2)$ random x's.

3 Generalized Gurvits Algorithm

We now give our generalization of Gurvits's algorithm to the multirow/multicolumn case, as well as our generalized upper bound for the permanent. As in Section 1.1, given nonnegative integers $s_1, \ldots, s_k$ summing to n, we let $B \in \mathbb{C}^{n \times k}$ be an $n \times k$ matrix, and let $A \in \mathbb{C}^{n \times n}$ be the $n \times n$ matrix in which the i^{th} column of B is repeated s_i times. Also, let $\mathcal{R}[j]$ denote the set of the j^{th} roots of unity, and let $\mathcal{X} := \mathcal{R}[s_1+1] \times \cdots \times \mathcal{R}[s_k+1]$. Then for any $x \in \mathcal{X}$, we define the *generalized Glynn estimator* as follows. If $x = (x_1, \ldots, x_k)$, then let $y_i = \sqrt{s_i}x_i$ and

$$\mathrm{GenGly}_x(A) := \frac{s_1! \cdots s_k!}{s_1^{s_1} \cdots s_k^{s_k}} \overline{y_1}^{s_1} \cdots \overline{y_k}^{s_k} \prod_{i=1}^n (y_1 b_{i,1} + \cdots + y_k b_{i,k}) \tag{29}$$

where $b_{i,j}$ denotes the (i,j) entry of B. We will use this to estimate $\mathrm{Per}(A)$, as follows. First we will show that

$$\mathop{\mathrm{E}}_x [\mathrm{GenGly}_x(A)] = \mathrm{Per}(A) . \tag{30}$$

Then we will show that

$$|\mathrm{GenGly}_x(A)| \leq \frac{s_1! \cdots s_k!}{\sqrt{s_1^{s_1} \cdots s_k^{s_k}}} \|B\|^n . \tag{31}$$

The $\frac{s_1!\cdots s_k!}{s_1^{s_1}\cdots s_k^{s_k}}$ factor in the definition of GenGly_x is a normalization factor that we need for Lemma 2 below to hold. The value of the normalization factor depends on our choice of the scale factors when defining the y_i's. Here, we set $y_i = \sqrt{s_i}x_i$ in order to optimize the bound in Lemma 3 below.

Lemma 2. *For the uniform distribution of x over $\mathcal{X}$, the expected value of* $\mathrm{GenGly}_x(A)$ *is*

$$\mathop{\mathrm{E}}_{x\in\mathcal{X}}\left[\mathrm{GenGly}_x(A)\right] = \mathrm{Per}(A). \tag{32}$$

Proof. We have

$$\mathop{\mathrm{E}}_{x\in\mathcal{X}}\left[\frac{s_1!\cdots s_k!}{s_1^{s_1}\cdots s_k^{s_k}}\overline{y_1}^{s_1}\cdots\overline{y_k}^{s_k}\prod_{i=1}^{n}(y_1b_{i,1}+\cdots+y_kb_{i,k})\right] \tag{33}$$

$$= \frac{s_1!\cdots s_k!}{s_1^{s_1}\cdots s_k^{s_k}}\sum_{\sigma_1,\ldots,\sigma_n\in[k]} b_{1,\sigma_1}\cdots b_{n,\sigma_n}\mathop{\mathrm{E}}_{x\in\mathcal{X}}\left[(\overline{y_1}^{s_1}\cdots\overline{y_k}^{s_k})(y_{\sigma_1}\cdots y_{\sigma_n})\right]. \tag{34}$$

Now notice that by symmetry over the roots of unity,

$$\mathop{\mathrm{E}}_{x\in\mathcal{X}}\left[(\overline{y}_1^{s_1}\cdots\overline{y}_k^{s_k})(y_{t_1}\cdots y_{t_n})\right] = 0, \tag{35}$$

unless the product inside happens to evaluate to $|y_1|^{2s_1}\cdots|y_k|^{2s_k}$, in which case we have

$$\mathop{\mathrm{E}}_{x\in\mathcal{X}}\left[|y_1|^{2s_1}\cdots|y_k|^{2s_k}\right] = s_1^{s_1}\cdots s_k^{s_k}\mathop{\mathrm{E}}_{x\in\mathcal{X}}\left[|x_1|^{2s_1}\cdots|x_k|^{2s_k}\right] = s_1^{s_1}\cdots s_k^{s_k}. \tag{36}$$

Therefore,

$$\mathop{\mathrm{E}}_{x\in\mathcal{X}}\left[\mathrm{GenGly}_x(A)\right] = s_1!\cdots s_k!\sum_{t_1,\ldots,t_n\in[k]\,:\,|\{i:t_i=j\}|=s_j\ \forall j\in[k]}\prod_{i=1}^{n}b_{i,t_i} \tag{37}$$

$$= \mathrm{Per}(A) \tag{38}$$

where (38) follows from the fact that each product $\prod_{i=1}^{n}b_{i,t_i}$ appears $s_1!\cdots s_k!$ times in equation (1). □

Lemma 3. *For all $x\in\mathcal{X}$,*

$$\left|\mathrm{GenGly}_x(A)\right| \le \frac{s_1!\cdots s_k!}{\sqrt{s_1^{s_1}\cdots s_k^{s_k}}}\|B\|^n. \tag{39}$$

Proof. Let $y = (y_1,\ldots,y_k)$. Then

$$\|y\|_2 = \sqrt{|y_1|^2+\cdots+|y_k|^2} \tag{40}$$

$$= \sqrt{s_1+\cdots+s_k} \tag{41}$$

$$= \sqrt{n}. \tag{42}$$

Hence

$$\|By\| \le \|B\|\,\|y\| = \sqrt{n}\,\|B\|. \tag{43}$$

So letting $(By)_i$ be the i^{th} entry of By, we have

$$\left|\prod_{i=1}^{n}(By)_i\right| \leq \left(\sqrt{\frac{n\|B\|^2}{n}}\right)^n = \|B\|^n \tag{44}$$

by the arithmetic-geometric mean inequality. Therefore

$$|\mathrm{GenGly}_x(A)| = \left|\frac{s_1!\cdots s_k!}{s_1^{s_1}\cdots s_k^{s_k}}\overline{y}_1^{s_1}\cdots\overline{y}_k^{s_k}\prod_{i=1}^{n}(By)_i\right| \tag{45}$$

$$\leq \frac{s_1!\cdots s_k!}{s_1^{s_1}\cdots s_k^{s_k}}\sqrt{s_1^{s_1}\cdots s_k^{s_k}}\,\|B\|^n \tag{46}$$

$$= \frac{s_1!\cdots s_k!}{\sqrt{s_1^{s_1}\cdots s_k^{s_k}}}\|B\|^n. \tag{47}$$

□

To summarize, $\mathrm{GenGly}_x(A)$ is an unbiased estimator for $\mathrm{Per}(A)$ that is upper-bounded by $\frac{s_1!\cdots s_k!}{\sqrt{s_1^{s_1}\cdots s_k^{s_k}}}\|B\|^n$ in absolute value. This immediately implies that

$$|\mathrm{Per}(A)| \leq \frac{s_1!\cdots s_k!}{\sqrt{s_1^{s_1}\cdots s_k^{s_k}}}\|B\|^n. \tag{48}$$

It also implies that there exists a randomized algorithm, taking $O\left(nk/\varepsilon^2\right)$ time, that approximates $\mathrm{Per}(A)$ to within additive error

$$\pm\varepsilon\cdot\frac{s_1!\cdots s_k!}{\sqrt{s_1^{s_1}\cdots s_k^{s_k}}}\|B\|^n \tag{49}$$

with high probability. Just like in Section 2, that algorithm is simply to choose $m = O\left(1/\varepsilon^2\right)$ independent random samples $x_1,\ldots,x_m \in \mathcal{X}$, then output

$$\frac{\mathrm{GenGly}_{x_1}(A)+\cdots+\mathrm{GenGly}_{x_m}(A)}{m} \tag{50}$$

as the estimate for $\mathrm{Per}(A)$. The correctness of this algorithm follows from a standard Chernoff bound.

As discussed in Section 1.3, in a linear-optics experiment with the standard initial state, the above lets us estimate any output amplitude to within additive error $\pm\varepsilon\sqrt{\frac{s_1!\cdots s_k!}{s_1^{s_1}\cdots s_k^{s_k}}}$, and any output probability to within additive error $\pm\varepsilon\frac{s_1!\cdots s_k!}{s_1^{s_1}\cdots s_k^{s_k}}$. This is at least as good an approximation as is provided by running the experiment itself (and better, if some of the s_i's are large).

4 Derandomization

We now discuss the derandomization of Gurvits's algorithm (and its generalization) for nonnegative matrices.

4.1 Derandomizing Gurvits

We start by showing how Gurvits's algorithm can be derandomized in the case of computing the permanent of a $n \times n$ matrix with nonnegative real entries. That is, we give a deterministic algorithm that estimates $\mathrm{Per}\,(A)$ to within $\pm\varepsilon \|A\|^n$ additive error with certainty.

To do this, we need a pseudorandom generator with a specific guarantee.

Definition 4. *We say that a probability distribution $\mathcal{D}$ on $\{0,1\}^n$ is ε-**biased** if for any nonzero vector $a \in \{0,1\}^n$, we have*

$$\left| \mathop{\mathrm{E}}_{x \sim \mathcal{D}} [(-1)^{a \cdot x}] \right| \leq \varepsilon. \tag{51}$$

Furthermore, we say that a probabilistic algorithm G is ε-biased if it outputs an ε-biased distribution.

Note that if G outputs the uniform distribution over $\{0,1\}^n$, then G is 0-biased. Unfortunately, this requires n random bits, and so it takes exponential time to loop over all possible states. We will need an ε-biased generator with a much smaller seed. Such a generator was constructed by Naor and Naor in [14], who showed the following.

Theorem 5. *There exists an ε-biased generator which runs in $\mathrm{poly}(n, 1/\varepsilon)$ time and uses*

$$O\left(\log n + \log 1/\varepsilon\right) \tag{52}$$

bits of randomness.

This allows us to derandomize Gurvits's algorithm.

Theorem 6. *If $\mathcal{D}$ is an ε-biased distribution on $\{0,1\}^n$, then for any $n \times n$ matrix A with nonnegative real entries, we have*

$$\left| \mathrm{Per}(A) - \mathop{\mathrm{E}}_{x \sim \mathcal{D}} [\mathrm{Gly}_x(A)] \right| \leq \varepsilon \cdot \|A\|^n \tag{53}$$

where we define the vector $x \in \{-1, 1\}$ by $x_i = (-1)^{e_i}$.

Proof. Following equation (22) we get

$$\mathop{\mathrm{E}}_{e \sim \mathcal{D}} [\mathrm{Gly}_x(A)] = \sum_{\sigma_1, \ldots, \sigma_n \in [n]} a_{1,\sigma_1} \cdots a_{n,\sigma_n} \mathop{\mathrm{E}}_{x \sim \mathcal{D}} \left[(x_1 \cdots x_n)(x_{\sigma_1} \cdots x_{\sigma_n}) \right]. \tag{54}$$

If the map $i \mapsto \sigma_i$ is a permutation, then $\mathrm{E}_{e \sim \mathcal{D}} \left[(x_1 \cdots x_n)(x_{\sigma_1} \cdots x_{\sigma_n}) \right]$ will always be 1. We would like to bound the terms for which σ does not give a permutation. Given such a σ, define a vector $c \in \{0,1\}^n$ such that

$$c_i = 1 + (|\{j : \sigma_j = i\}|) \bmod 2. \tag{55}$$

Clearly, c is nonzero if σ does not give a permutation. Then

$$\left| \mathop{\mathrm{E}}_{e \sim \mathcal{D}} \left[(x_1 \cdots x_n)(x_{\sigma_1} \cdots x_{\sigma_n}) \right] \right| = \left| \mathop{\mathrm{E}}_{e \sim \mathcal{D}} [(-1)^{c \cdot e}] \right| \tag{56}$$

$$\leq \varepsilon \tag{57}$$

since $\mathcal{D}$ is ε-biased by assumption. Then,

$$\left|\mathrm{Per}(A) - \mathop{\mathrm{E}}_{e\sim\mathcal{D}}[\mathrm{Gly}_x(A)]\right| = \left|\sum_{\substack{\sigma_1,\ldots,\sigma_n\in[n]\\ i\mapsto\sigma_i \text{ not a permutation}}} a_{1,\sigma_1}\cdots a_{n,\sigma_n} \mathop{\mathrm{E}}_{e\sim\mathcal{D}}[(x_1\cdots x_n)(x_{\sigma_1}\cdots x_{\sigma_n})]\right| \tag{58}$$

$$\leq \sum_{\substack{\sigma_1,\ldots,\sigma_n\in[n]\\ i\mapsto\sigma_i \text{ not a permutation}}} a_{1,\sigma_1}\cdots a_{n,\sigma_n}\left|\mathop{\mathrm{E}}_{e\sim\mathcal{D}}[(x_1\cdots x_n)(x_{\sigma_1}\cdots x_{\sigma_n})]\right| \tag{59}$$

$$\leq \varepsilon\cdot \sum_{\substack{\sigma_1,\ldots,\sigma_n\in[n]\\ i\mapsto\sigma_i \text{ not a permutation}}} a_{1,\sigma_1}\cdots a_{n,\sigma_n} \tag{60}$$

$$\leq \varepsilon\cdot \sum_{\sigma_1,\ldots,\sigma_n\in[n]} a_{1,\sigma_1}\cdots a_{n,\sigma_n} \tag{61}$$

$$= \varepsilon\cdot\prod_{i=1}^{n}(a_{1,i}+\cdots+a_{n,i}) \tag{62}$$

$$= \varepsilon\cdot\mathrm{Gly}_{(1,\ldots,1)}(A) \tag{63}$$

$$\leq \varepsilon\cdot\|A\|^n \tag{64}$$

where for (59) we use the fact that the $a_{i,j}$ are nonnegative, and for (64) we appeal to Proposition 1. □

Corollary 7. *There exists an algorithm to deterministically approximate the permanent of an $n\times n$ matrix A with nonnegative entries to within error $\varepsilon\cdot\|A\|^n$ which runs in time* $\mathrm{poly}\,(n, 1/\varepsilon)$.

Proof. By Theorem 5 there is a polynomial time algorithm which outputs an ε-biased distribution $\mathcal{D}$ on $\{0,1\}^n$. Since the seed for this algorithm is only $O(\log n + \log 1/\varepsilon)$ bits, we can compute $\mathrm{E}_{e\sim\mathcal{D}}[\mathrm{Gly}_x(A)]$ by looping over all possible seeds. By Theorem 6, this suffices as our desired approximation. □

4.2 Derandomizing the Multicolumn Case

To derandomize the algorithm of Section 3, we need to generalize the concept of ε-bias to arbitrary roots of unity.

Definition 8. *Given $s_1,\ldots,s_k$ with $n = s_1+\cdots+s_k$, recall that we let $\mathcal{X} := \mathcal{R}[s_1+1]\times\cdots\times\mathcal{R}[s_k+1]$. Then we say that a probability distribution $\mathcal{D}$ on $\mathcal{X}$ is **complex-ε-biased** if for any $e_1,\ldots,e_k$, with $e_i\in\{0,\ldots,s_i\}$ where not all e_i are 0, we have*

$$\left|\mathop{\mathrm{E}}_{x\sim\mathcal{D}}\left[x_1^{e_i}\cdots x_k^{e_k}\right]\right| \leq \varepsilon. \tag{65}$$

Also, we say that a probabilistic algorithm G is complex-ε-biased if it outputs a complex-ε-biased distribution.

In Section 4.3 we will show the following:

Theorem 9. *There exists a complex-ε-biased generator which runs in* $\mathrm{poly}\,(n, 1/\varepsilon)$ *time and uses $O(\log n + \log 1/\varepsilon)$ bits of randomness.*

These complex-ε-biased distributions are useful because of the following property, which is analogous to Theorem 6:

Theorem 10. *If $\mathcal{D}$ is a complex-ε-biased distribution on $\mathcal{X}$, then for any $n \times k$ matrix B with nonnegative real entries, let A be an $n \times n$ matrix with s_i copies of the i^{th} column of B. Then we have*

$$\left|\mathrm{Per}(A) - \mathop{\mathrm{E}}_{x\sim\mathcal{D}}\left[\mathrm{GenGly}_x(A)\right]\right| \leq \varepsilon \cdot \frac{s_1! \cdots s_k!}{\sqrt{s_1^{s_1} \cdots s_k^{s_k}}} \|B\|^n . \tag{66}$$

Proof. We begin by expanding the product:

$$\mathop{\mathrm{E}}_{x\sim\mathcal{D}}\left[\mathrm{GenGly}_x(A)\right] = \mathop{\mathrm{E}}_{x\sim\mathcal{D}}\left[\frac{s_1! \cdots s_k!}{s_1^{s_1} \cdots s_k^{s_k}} \overline{y_1}^{s_1} \cdots \overline{y_k}^{s_k} \prod_{i=1}^{n} \left(y_1 b_{i,1} + \cdots + y_k b_{i,k}\right)\right] \tag{67}$$

$$= \frac{s_1! \cdots s_k!}{s_1^{s_1} \cdots s_k^{s_k}} \sum_{\sigma_1,\ldots,\sigma_n \in [k]} b_{1,\sigma_1} \cdots b_{n,\sigma_n} \mathop{\mathrm{E}}_{x\sim\mathcal{D}}\left[\left(\overline{y_1}^{s_1} \cdots \overline{y_k}^{s_k}\right)\left(y_{\sigma_1} \cdots y_{\sigma_n}\right)\right] \tag{68}$$

$$= s_1! \cdots s_k! \sum_{\sigma_1,\ldots,\sigma_n \in [k]} b_{1,\sigma_1} \cdots b_{n,\sigma_n} \sqrt{\frac{s_{\sigma_1} \cdots s_{\sigma_n}}{s_1^{s_1} \cdots s_k^{s_k}}} \mathop{\mathrm{E}}_{x\sim\mathcal{D}}\left[\left(\overline{x_1}^{s_1} \cdots \overline{x_k}^{s_k}\right)\left(x_{\sigma_1} \cdots x_{\sigma_n}\right)\right]. \tag{69}$$

Say that a sequence $\sigma = (\sigma_1, \ldots, \sigma_n)$ is a *desired sequence* if for all $i \in \{1, \ldots, k\}$, we have $|\{j : \sigma_j = i\}| = s_i$. Let

$$e_i = (-s_i + |\{j : \sigma_j = i\}|) \bmod (s_k + 1). \tag{70}$$

Then we have

$$\left(\overline{x_1}^{s_1} \cdots \overline{x_k}^{s_k}\right)\left(x_{\sigma_1} \cdots x_{\sigma_n}\right) = x_1^{e_1} \cdots x_n^{e_n}. \tag{71}$$

Note that $e_i = 0$ for all i if and only if σ is desired. Thus, in the case that σ is desired, the expected value of (71) is 1. Otherwise, since $\mathcal{D}$ is complex-ε-biased, we get

$$\left|\mathop{\mathrm{E}}_{x\sim\mathcal{D}}\left[\left(\overline{x_1}^{s_1} \cdots \overline{x_k}^{s_k}\right)\left(x_{\sigma_1} \cdots x_{\sigma_n}\right)\right]\right| \leq \varepsilon.$$

Therefore,

$$\left|\mathrm{Per}(A) - \mathop{\mathrm{E}}_{x\sim\mathcal{D}}\left[\mathrm{GenGly}_x(A)\right]\right| \tag{72}$$

$$= s_1!\cdots s_k!\left|\sum_{\substack{\sigma_1,\ldots,\sigma_n\in[k]\\ \sigma \text{ not desired}}} (b_{1,\sigma_1}\cdots b_{n,\sigma_n})\sqrt{\frac{s_{\sigma_1}\cdots s_{\sigma_n}}{s_1^{s_k}\cdots s_k^{s_k}}}\mathop{\mathrm{E}}_{x\sim\mathcal{D}}\left[(\overline{x_1}^{s_1}\cdots\overline{x_k}^{s_k})\,(x_{\sigma_1}\cdots x_{\sigma_n})\right]\right| \tag{73}$$

$$\leq \frac{s_1!\cdots s_k!}{\sqrt{s_1^{s_1}\cdots s_k^{s_k}}}\sum_{\substack{\sigma_1,\ldots,\sigma_n\in[k]\\ \sigma \text{ not desired}}} (b_{1,\sigma_1}\cdots b_{n,\sigma_n})\sqrt{s_{\sigma_1}\cdots s_{\sigma_n}}\left|\mathop{\mathrm{E}}_{x\sim\mathcal{D}}\left[(\overline{x_1}^{s_1}\cdots\overline{x_k}^{s_k})\,(x_{\sigma_1}\cdots x_{\sigma_n})\right]\right| \tag{74}$$

$$\leq \varepsilon\cdot\frac{s_1!\cdots s_k!}{\sqrt{s_1^{s_1}\cdots s_k^{s_k}}}\sum_{\substack{\sigma_1,\ldots,\sigma_n\in[k]\\ \sigma \text{ not desired}}} (b_{1,\sigma_1}\cdots b_{n,\sigma_n})\sqrt{s_{\sigma_1}\cdots s_{\sigma_n}} \tag{75}$$

$$\leq \varepsilon\cdot\frac{s_1!\cdots s_k!}{\sqrt{s_1^{s_1}\cdots s_k^{s_k}}}\sum_{\sigma_1,\ldots,\sigma_n\in[k]} (b_{1,\sigma_1}\cdots b_{n,\sigma_n})\sqrt{s_{\sigma_1}\cdots s_{\sigma_n}} \tag{76}$$

$$= \varepsilon\cdot\frac{s_1!\cdots s_k!}{\sqrt{s_1^{s_1}\cdots s_k^{s_k}}}\prod_{i=1}^{n}(\sqrt{s_1}b_{i,1}+\cdots+\sqrt{s_k}b_{i,k}) \tag{77}$$

$$= \varepsilon\cdot\mathrm{GenGly}_{(1,\ldots,1)}(A) \tag{78}$$

$$\leq \varepsilon\cdot\frac{s_1!\cdots s_k!}{\sqrt{s_1^{s_1}\cdots s_k^{s_k}}}\|B\|^n \tag{79}$$

where the last inequality follows from Lemma 3. □

Theorems 9 and 10 immediately imply the following.

Corollary 11. *For any $n\times k$ matrix B with nonnegative real entries, let A be an $n\times n$ matrix with s_i copies of the i^{th} column of B. Then we can deterministically approximate* $\mathrm{Per}(A)$ *to within additive error*

$$\varepsilon\cdot\frac{s_1!\cdots s_k!}{\sqrt{s_1^{s_1}\cdots s_k^{s_k}}}\|B\|^n \tag{80}$$

in $\mathrm{poly}\,(n,1/\varepsilon)$ *time.*

4.3 Constructing Complex-ε-Biased Generators

Our proof of Theorem 9 will be modeled after the proof of Theorem 5 given by Naor and Naor [14]. We just need to generalize each step to deal with the domain of complex roots of unity rather than the binary domain $\{1,-1\}$.

Recall that our goal is to generate, with as few bits of randomness as possible, complex numbers $(\zeta_1,\ldots,\zeta_k)\in\mathcal{X}$, such that for any exponents $e_1,\ldots,e_k$, we have

$$\left|\mathrm{E}\left[\zeta_1^{e_1}\cdots\zeta_k^{e_k}\right]\right|\leq\varepsilon. \tag{81}$$

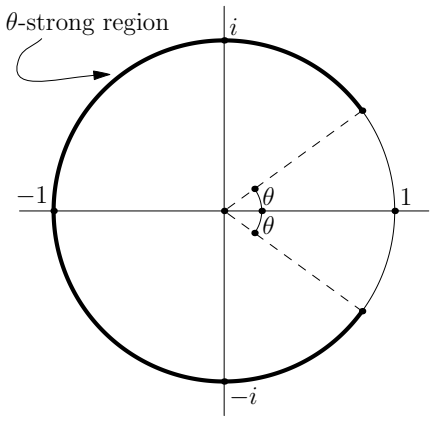

Figure 1: The θ-strong region of the unit circle in the complex plane.

In our construction, it will be simpler to adopt an alternate perspective, which we now describe. If we let f_j be such that $\zeta_j = e^{2\pi i f_j/(s_j+1)}$, and let $\xi_j = e^{2\pi i e_j/(s_j+1)}$, then

$$\zeta_j^{e_j} = \left(e^{2\pi i/(s_j+1)}\right)^{e_j f_j} = \xi_j^{f_j}. \tag{82}$$

Hence our task is equivalent to outputting a random k-tuple $(f_1, \ldots, f_k)$ such that for any choice of complex numbers $\xi_j = e^{2\pi i e_j/(s_j+1)}$, not all 1, we have

$$\left|\mathrm{E}\left[\xi_1^{f_1} \cdots \xi_k^{f_k}\right]\right| \leq \varepsilon. \tag{83}$$

The proof of Theorem 5 given by Naor and Naor [14] revolves around the ability to find vectors $r \in \{0,1\}^n$ such that $r \cdot x = 1 \bmod 2$ with constant probability for any nonzero $x \in \{0,1\}^n$. A uniformly random $r \in \{0,1\}^n$ will satisfy $r \cdot x = 1 \bmod 2$ with probability $1/2$, and in fact it is also possible to generate such an r using only a small amount of randomness (possibly allowing for a weaker constant).

In our setting, this would translate to generating $(f_1, \ldots, f_k)$ such that $\xi_1^{f_1} \cdots \xi_k^{f_k} = -1$ with constant probability. Unfortunately, this will not be possible since the ξ_i do not come from the binary domain $\{1, -1\}$. Instead, we will have to replace the concept of "being equal to -1" with what we call "θ-strongness."

Definition 12. *For a unit-norm complex number λ, we say that λ is θ-**strong** if $|\arg \lambda| \geq \theta$, where $\arg \lambda$ is the phase $\phi \in [-\pi, \pi)$ such that $\lambda = e^{i\phi}$.*

Lemma 13. *We can generate $f_1, \ldots, f_k$ such that for any $\xi_1, \ldots, \xi_k$, not all 1, we have*

$$\Pr[\xi_1^{f_1} \cdots \xi_k^{f_k} \text{ is } \pi/8\text{-strong}] \geq \frac{1}{16} \tag{84}$$

using $O(\log n)$ bits of randomness.

The proof of this will require the following fact.

Proposition 14. *We can generate k random variables $(g_1, \ldots, g_k)$ such that we have $g_i \in \{0, \ldots, s_i\}$ (nearly) uniformly, and c-wise independence between the g_i, using $O(c \log n)$ bits of randomness.*

This is well-known. See, e.g., Luby and Wigderson [13]. It is possible to generate k random variables $t_1, \ldots, t_k$ in $\mathbb{F}_p$, where p is a prime of size $\mathrm{poly}(k)$, such that each one is uniformly distributed in $\mathbb{F}_p$, and any c of them are independent, using only $c \log k$ random bits. Since our desired domains are $\{0, \ldots, s_i\}$, to get the above result, we have to take the $t_i \bmod s_i + 1$. There is a small error since p is not divisible by $s_i + 1$, but this error is small enough that it does not matter for our purposes.

Another fact we will need was proved by Naor and Naor [14]:

Proposition 15. *There is a constant c such that the following holds: Let ℓ and m be integers such that $m \leq \ell \leq 2m$, and let S is a fixed subset of $[k]$ with $|S| = \ell$. If T is a randomly chosen subset of $[k]$ such that each element $i \in [k]$ is in T with probability $2/m$, and the elements are chosen with pairwise independence, then*

$$\Pr\left[|S \cup T| \leq c\right] \leq \frac{1}{2}. \tag{85}$$

In particular, they show that we can take $c = 7$. These facts are enough for the following proof:

Proof of Lemma 13. Suppose that $\ell \geq 1$ of the ξ_i are not equal to 1, and let h be the integer such that $2^h \leq \ell < 2^{h+1}$. For now, suppose that we know h; we will relax this assumption later. Do the following (using Proposition 14):

- Generate $u = (u_1, \ldots, u_k)$ whose entries are c-wise independent and such that u_i is (nearly) uniformly distributed in $\{0, \ldots, s_i\}$, in the manner stated above. Here, c is the constant from Proposition 15.
- Generate $w = (w_1, \ldots, w_k)$ whose entries are pairwise independent and such that $w_i \in \{0, 1\}$ and $\Pr[w_i = 1] = \min(1/2^{h-1}, 1)$.

Finally, we define $f = (f_1, \ldots, f_k)$ by

$$f_i = \begin{cases} 0 & \text{if } w_i = 0 \\ u_i & \text{if } w_i = 1 \end{cases} \tag{86}$$

Let $V := \{i : \xi_i \neq 1\}$ and $W := \{i : w_i = 1\}$. Now we need that with probability at least $1/2$, we will have $1 \leq |V \cap W| \leq c$. This follows from Proposition 15. Now let us examine $\lambda := \xi_1^{f_1} \cdots \xi_k^{f_k}$. With probability $1/2$, we have $1 \leq |V \cap W| \leq c$. If this is the case, write $\lambda = \xi_{i_1}^{f_{i_1}} \cdots \xi_{i_d}^{f_{i_d}}$ where $V \cap W = \{i_1, \ldots, i_d\}$, and $1 \leq d \leq c$. Then, the values of f_{i_j} will be (nearly) uniform in $\{0, \ldots, s_{i_j}\}$ and independent. Because of the c-independence, for any fixed values of $f_{i_2}, \ldots, f_{i_d}$, we can condition on those choices and still get that f_{i_1} is uniform in $\{0, \ldots, s_{i_j}\}$. Now, we know that ξ_{i_1} is an $s_{i_1}^{\text{th}}$ root of unity other than 1 (but still not necessarily primitive). Therefore as $\xi_{i_1}^{f_{i_1}}$ ranges around the unit circle, we can see that

$$\xi_{i_1}^{f_{i_1}} \left(\xi_{i_1}^{f_{i_1}} \cdots \xi_{i_d}^{f_{i_d}} \right) = \xi_{i_1}^{f_{i_1}} \text{ (fixed unit-norm complex number)} \tag{87}$$

will have a probability of at least $1/4$ of being $\pi/4$-strong (being conservative with constants here). Putting it all together, we see that λ has a $1/8$ probability of being $\pi/4$-strong, and

we have generated λ using only $O(\log n)$ bits. However, this has all assumed that we know the value of h. Suppose we do not know h. Then, generate $O(\log n)$ random bits, and for every $h \in \{0, \ldots, \lfloor \log_2 k \rfloor\}$ use these bits to construct $\lambda^{(h)}$ using the above method. Let h_{actual} be the value such that $2^{h_{\text{actual}}} \leq \ell < 2^{h_{\text{actual}}+1}$. Now, randomly and indepently choose bits $b_0, \ldots, b_{\log_2 k}$ and let $\lambda := \prod_h \left(\lambda^{(h)}\right)^{b_h}$. There is a $1/8$ chance that $\lambda^{(h_{\text{actual}})}$ is $\pi/4$-strong. If this happens to be the case, then write

$$\lambda = \left(\lambda^{(h_{\text{actual}})}\right)^{b_{h_{\text{actual}}}} \prod_{h \neq h_{\text{actual}}} \left(\lambda^{(h)}\right)^{b_h} \tag{88}$$

If we fix the values of b_h for $h \neq h_{\text{actual}}$, then the factor on the right becomes fixed, call it Λ. If Λ is $\pi/8$-strong, then λ will be $\pi/8$-strong if $b_{h_{\text{actual}}} = 0$. If Λ is not $\pi/8$-strong, then λ will be $\pi/8$-strong if $b_{h_{\text{actual}}} = 1$ (here, using the fact that $\lambda^{(h_{\text{actual}})}$ is $\pi/4$-strong). In either case, there is a $1/2$ chance that λ is $\pi/8$-strong. Combining with the $1/8$ probabililty from earlier, we get that in total, there is at least a $1/16$ chance that λ is $\pi/8$-strong. □

However, we need a better guarantee than just constant probability given by Lemma 13. In particular, we will need to construct a set of λ which has high probability (at least $1 - \varepsilon/2$) of containing many $\pi/8$-strong elements. We could sample $O(\log(1/\varepsilon))$ samples independently, but that would require too many random bits.

To reduce the number of random bits, we use *deterministic amplification*. The following result was shown independently by Cohen and Wigderson [6] and by Impagliazzo and Zuckerman [11], using ideas from Ajtai, Komlós, and Szemerédi [3]. The original motivation for these results was to amplify error probabilities in BPP algorithms using few bits of randomness. Here is a formulation which is useful to us:

Theorem 16. *Let $\mathcal{T} = \{0,1\}^r$. Then with $r + O(\ell)$ bits of randomness we can efficiently generate a set $y_1, \ldots, y_\ell \in \mathcal{T}$ such that for any $\mathcal{S} \subseteq \mathcal{T}$ with $|\mathcal{S}| \geq 2|\mathcal{T}|/3$, we have*

$$\Pr[y_i \in \mathcal{S} \text{ for at least } \ell/2 \text{ values of } i] \geq 1 - 2^{-\Omega(\ell)}. \tag{89}$$

This gives the following lemma:

Lemma 17. *There are constants $p, q > 0$ such that using $O(\log n + \ell)$ bits of randomness, we can generate $\lambda_1, \ldots, \lambda_\ell$ such that with probability $1 - 2^{q\ell}$, there are at least $p\ell$ values of i such that λ_i is $\pi/8$-strong.*

Proof. We start with a small amount of initial amplification: let t be the constant such that if we generate $\lambda_1, \ldots, \lambda_t$ independently as in Lemma 13, then with probability at least $2/3$, one of the λ_i is $\pi/8$-strong, and let r be the number of random bits needed to generate these $\lambda_1, \ldots, \lambda_t$. If we want to generate $\lambda_1, \ldots, \lambda_\ell$ for some ℓ, then generate groups $\lambda_{j,1}, \ldots, \lambda_{j,t}$ for $j \in [\ell/t]$ according to Theorem 16. Then with probability at least $1 - 2^{-\Omega(\ell/t)} = 1 - 2^{-\Omega(\ell)}$, at least half of the groups will have at least one $\pi/8$-strong member, i.e., at least a $1/2t$ fraction of the λ's will be $\pi/8$-strong. The total number of random bits used is $r + O(\ell) = O(\log n + \ell)$. □

It is worth noting that Lemma 17 is a slight difference between our construction of complex-ε-biased distributions and the construction of ε-biased distributions by Naor and Naor [14]. Naor and Naor constructed many values, needing only one to be -1, while we construct many values needing *many* of them to be $\pi/8$-strong, for reasons evident in the proof of Theorem 9. Luckily, this

was not a problem, since as we saw, the same class of deterministic amplification results applied, allowing us to prove Lemma 17.

Proof of Theorem 9. Generate $\lambda_1, \ldots, \lambda_\ell$ as in Lemma 17, choosing

$$\ell = \left\lceil \max\left(\frac{\log_{1/2}(\varepsilon/2)}{q}, \frac{\log_\beta(\varepsilon/2)}{p} \right) \right\rceil. \tag{90}$$

where $\beta := \frac{1}{2}\left|1 + e^{\pi i/8}\right|$. Randomly and independently choose $d_1, \ldots, d_\ell$, each uniformly from $\{0,1\}$. Then let

$$\mu := \prod_{j=1}^{\ell} \lambda_i^{d_i}. \tag{91}$$

With probability at least $1 - 2^{-q\ell} \geq 1 - \varepsilon/2$, at least $p\ell$ of the λ_j will be $\pi/8$-strong. Suppose that this is the case. Then

$$\left| \mathop{\mathrm{E}}_{d \sim \{0,1\}^\ell} [\mu] \right| = \prod_{j=1}^{\ell} \left| \frac{1+\lambda_j}{2} \right| \tag{92}$$

$$\leq \beta^{p\ell} \tag{93}$$

$$\leq \varepsilon/2. \tag{94}$$

Thus, this shows that we get that with probability at least $1 - \varepsilon/2$, the norm of the expected value of μ is at most $\varepsilon/2$. Hence, the total expected value of μ, over the random choices of $\lambda_1, \ldots, \lambda_\ell$ and $d_1, \ldots, d_\ell$ is at most

$$\left(1 - \frac{\varepsilon}{2}\right)\frac{\varepsilon}{2} + \frac{\varepsilon}{2}(1) \leq \varepsilon. \tag{95}$$

Now, since we have to choose ℓ values d_j, each in $\{0,1\}$, this takes $O(\ell) = O(\log 1/\varepsilon)$ bits of randomness. Generating the $\lambda_1, \ldots, \lambda_\ell$ takes $O(\log n + \ell) = O(\log n + \log 1/\varepsilon)$ bits of randomness. Therefore, the entire procedure then takes $O(\log n + \log 1/\varepsilon)$ bits of randomness. □

5 Open Problems

There are many interesting open problems about Gurvits's algorithm, and the classical simulability of quantum optics more generally. Firstly, can we improve the error bound even further in the case that the matrix A has both repeated rows *and* repeated columns? More specifically, can we estimate any linear-optical amplitude

$$\langle s_1, \ldots, s_k | \varphi(U) | t_1, \ldots, t_k \rangle = \frac{\mathrm{Per}\left(U_{s_1,\ldots,s_k,t_1,\ldots,t_k}\right)}{\sqrt{s_1! \cdots s_k! t_1! \cdots t_k!}} \tag{96}$$

to $\pm 1/\operatorname{poly}(n)$ additive error (or better) in polynomial time? This is related to a question raised by Aaronson and Arkhipov [2], of whether quantum optics can be used to solve any *decision* problems that are classically intractable (rather than just sampling and search problems).

An even more obvious problem is to generalize our derandomization of Gurvits's algorithm so that it works for *arbitrary* matrices, not only matrices with nonnegative entries. Unfortunately, our current technique appears unable to handle arbitrary matrices, as it relies too heavily on the expansions of $\mathrm{Gly}_{(1,\ldots,1)}(A)$ and $\mathrm{GenGly}_{(1,\ldots,1)}(A)$ having no negative terms.

An alternative approach to derandomizing Gurvits's algorithm would be to prove a general "structure theorem," showing that if U is a unitary or subunitary matrix, then $\mathrm{Per}\,(U)$ can only be non-negligibly large if U has some special form: for example, if U is close (in a suitable sense) to a product of permutation and diagonal matrices. A deterministic algorithm to estimate $\mathrm{Per}\,(U)$ could then follow, by simply checking whether the structure is present or not. We conjecture that such a structure theorem holds, but do not even know how to formulate the conjecture precisely.

Even in the realm of nonnegative entries, there is much room for improvement. Among other things, it would be extremely interesting to derandomize a Jerrum-Sinclair-Vigoda-type algorithm [12], enabling us deterministically to approximate the permanent of nonnegative matrices to within small *multiplicative* error.

6 Acknowledgments

We thank Alex Arkhipov, Michael Forbes, Leonid Gurvits, and Shravas Rao for helpful discussions.

References

[1] S. Aaronson. A linear-optical proof that the permanent is #P-hard. *Proc. Roy. Soc. London*, A467(2088):3393–3405, 2011. arXiv:1109.1674.

[2] S. Aaronson and A. Arkhipov. The computational complexity of linear optics. In *Proc. ACM STOC*, 2011. ECCC TR10-170, arXiv:1011.3245.

[3] M. Ajtai, J. Komlós, and E. Szemerédi. Deterministic simulation in LOGSPACE. In *Proc. ACM STOC*, pages 132–140, 1987.

[4] A. Björklund. Below all subsets for permutational counting problems. arXiv:1211.0391, 2012.

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

[6] A. Cohen and A. Wigderson. Dispersers, deterministic amplification and weak random sources. In *Proc. IEEE FOCS*, pages 14–19, 1989.

[7] D. G. Glynn. The permanent of a square matrix. *European Journal of Combinatorics*, 31(7):1887–1891, 2010.

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

[9] L. Gurvits and A. Samorodnitsky. A deterministic algorithm for approximating the mixed discriminant and mixed volume, and a combinatorial corollary. *Discrete and Computational Geometry*, 27(4):531–550, 2002.

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

[11] R. Impagliazzo and D. Zuckerman. How to recycle random bits. In *Proc. IEEE FOCS*, pages 248–253, 1989.

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

[13] M. Luby and A. Wigderson. Pairwise independence and derandomization. *Foundations and Trends in Theoretical Computer Science*, 1(4):237–301, 2005.

[14] J. Naor and M. Naor. Small-bias probability spaces: efficient constructions and applications. *SIAM J. Comput.*, 22(4):838–856, 1993.

[15] H. Ryser. *Combinatorial Mathematics*. Mathematical Association of America, 1963.

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

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