Electronic Colloquium on Computational Complexity, Report No. 94 (2012)

Testing Permanent Oracles – Revisited

Sanjeev Arora* Arnab Bhattacharyya† Rajsekar Manokaran* Sushant Sachdeva*

Abstract

Suppose we are given an oracle that claims to approximate the permanent for most matrices X, where X is chosen from the *Gaussian ensemble* (the matrix entries are i.i.d. univariate complex Gaussians). Can we test that the oracle satisfies this claim? This paper gives a polynomial-time algorithm for the task.

The oracle-testing problem is of interest because a recent paper of Aaronson and Arkhipov showed that if there is a polynomial-time algorithm for simulating boson-boson interactions in quantum mechanics, then an approximation oracle for the permanent (of the type described above) exists in $\mathsf{BPP}^{\mathsf{NP}}$. Since computing the permanent of even 0/1 matrices is #P-complete, this seems to demonstrate more computational power in quantum mechanics than Shor's factoring algorithm does. However, unlike factoring, which is in NP, it was unclear previously how to test the correctness of an approximation oracle for the permanent, and this is the contribution of the paper.

The technical difficulty overcome here is that univariate polynomial self-correction, which underlies similar oracle-testing algorithms for permanent over *finite fields* —and whose discovery led to a revolution in complexity theory—does not seem to generalize to complex (or even, real) numbers. We believe that this tester will motivate further progress on understanding the permanent of Gaussian matrices.

1 Introduction

The permanent of an n-by-n matrix $X = (x_{i,j})$ is defined as

$$\mathsf{Per}(X) = \sum_{\pi} \prod_{i=1}^{n} x_{i,\pi(i)},$$

where π ranges over all permutations from $[n]$ to $[n]$. A recent paper of Aaronson and Arkhipov [AA11] (henceforth referred to as AA) introduced a surprising connection between quantum computing and the complexity of computing the permanent (which is well-known to be #P-complete to compute in the worst case [Val79]). They define and study a formal model of quantum computation with non-interacting bosons in which n bosons pass through a "circuit" consisting of optical elements. Each boson starts out in one of m different phases and, at the end of the experiment,

*Princeton University, Computer Science Department and Center for Computational Intractability. This work is supported by the NSF grants CCF-0832797 and CCF-1117309. Email: {arora,rajsekar,sachdeva}@cs.princeton.edu.

†Princeton University, Computer Science Department and Center for Computational Intractability. This work is supported by NSF Grants CCF-0832797, 0830673, and 0528414. Email: arnabb@cs.princeton.edu.

ISSN 1433-8092

the system is in a superposition of the basis states—one for each possible partition of the n bosons into m phases.

AA proceed to show that if there is an efficient classical randomized algorithm $\mathcal{A}$ that simulates the experiment, in the sense of being able to output random samples from the final distribution (up to a small error in total variation distance) of the Bosonic states at the end of the experiment, then there is a way to design an *approximation* algorithm $\mathcal{B}$ in $\mathsf{BPP}^{\mathsf{NP}}$ for the permanent problem for an interesting family of random matrices. The random matrices are drawn from the Gaussian ensemble—each entry is an independent standard Gaussian complex number—and the algorithm computes an additive approximation, in the sense that,

$$|\mathcal{B}(X) - \mathsf{Per}(X)|^2 \leqslant \delta^2 n!\,, \tag{1}$$

for at least a fraction $1-\eta$ of the input matrices X. (Note that the *variance* of $\mathsf{Per}(X)$ is $n!$ for Gaussian ensembles, so this approximation is nontrivial.) The running time of $\mathcal{B}$ is $\text{poly}(n, 1/\delta, 1/\eta)$ with access to an oracle in $\mathsf{NP}^{\mathcal{A}}$. In other words, $\mathcal{B} \in \mathsf{BPP}^{\mathsf{NP}^{\mathcal{A}}}$ for $\eta, \delta = \Omega\left(1/\text{poly}(n)\right)$ (refer to Problem 2 and Theorem 3 in [AA11]). The authors go on to conjecture that obtaining an additive approximation as in eq. (1) is #P-hard (this follows from Conjectures 5 and 6, and Theorem 7 in [AA11]). If true, this conjecture has surprising implications for the computational power of quantum systems. By contrast, the crown jewel of quantum computing, Shor's algorithm [Sho94], implies that the ability to simulate quantum systems would allow us to factor integers in polynomial time, but factoring (as well as other problems known to be in BQP) is not even known to be NP-Hard.

As evidence for their conjecture, Arkhipov and Aaronson point to related facts about the permanent problem for matrices over integers and finite fields. It is known that that if there is a constant factor approximation algorithm for computing $\mathsf{Per}(X)$ where X is an arbitrary matrix of integers, then one can solve #P problems in polynomial time. Thus, approximation on *all* inputs seems difficult[1]. Likewise, starting with a paper of Lipton, researchers have studied the complexity of computing the permanent (exactly) for *many matrices*. For example, given an algorithm that computes the permanent exactly for $1/\text{poly}(n)$ fraction of all matrices X over a finite field $GF(p)$ (where p is a sufficiently large prime), one can use self-correction procedures for univariate polynomials [GLR+91, GS92, CPS99] to again obtain efficient randomized algorithms for #P-hard problems.

Thus, either restriction —approximation on all matrices, or the ability to compute exactly on a significant fraction of matrices— individually results in a #P-hard problem. What makes the AA conjecture interesting is that it involves the conjunction of the two restrictions: the oracle in question *approximates* the value of the permanent for *most* matrices.

The focus of the current paper is the following question: *given an additive approximation oracle for permanents of Gaussian matrices ($\mathcal{B}$ in eq.* (1) *above), how can we test that the oracle is correct?* We want a tester that accepts with high probability when $\mathcal{B}$ satisfies the condition in eq. (1) and rejects with high probability when $\mathcal{B}$ does not approximate well on a substantial fraction of inputs. Note that the testing problem is a non-issue for previous quantum algorithms such as Shor's algorithm, since the correctness of a factoring algorithm is easy to test.

The testing question has been studied for the permanent problem over finite fields. Given an oracle that supposedly computes $\mathsf{Per}(\cdot)$ for even, say, $3/4^{\text{th}}$ of the matrices over $GF(p)$, one can verify this claim using self-correction for polynomials over finite fields and the *downward self-reducibility*

[1] Note that approximating the permanent is known to be feasible for the special case of non-negative real matrices [Bro86, JS89, JSV04].

of $\mathsf{Per}(\cdot)$, as described below in more detail in Section 1.1. (In fact, if the oracle satisfies the claim, then one can compute $\mathsf{Per}(\cdot)$ on all matrices with high probability.) However, as noted in AA, these techniques that work over finite fields fail badly over the complex numbers. The authors in AA also seem to suggest that techniques analogous to self-correction and downward self-reducibility can be generalized to complex numbers in some way, but this remains open.

In this paper, we solve the testing problem using downward self-reducibility alone. Perhaps this gives some weak evidence for the truth of the AA conjecture. Note that since we lack self-correction techniques, we do not get an oracle at the end that computes the permanent for all matrices as in the finite field case. Incidentally, an argument similar to the one presented in this paper works in the finite field case also, giving an alternate tester for the permanent that does not use self-correction of polynomials over finite fields.

1.1 Related Work

As mentioned above, testing an oracle for the permanent over finite fields has been extensively studied. The approach, basically arising from [LFKN92], uses self-correction of polynomials over finite fields and downward self-reducibility of the permanent. Let us revisit the argument.

Suppose we are given a sequence of oracles $\{\mathcal{O}_k\}_k$, where for each k, $\mathcal{O}_k$ allegedly computes the permanent for a 9/10 fraction of all k-by-k matrices over the field. The argument proceeds by first applying a self-correction procedure for low-degree polynomials (see [GS92]), noting that the permanent is a k-degree multilinear polynomial in the k^2 entries of the matrix, treated as variables.

The correction procedure, on input X, queries $\mathcal{O}_k$ at $\mathrm{poly}(n)$ points, and outputs the correct value of $\mathsf{Per}(X)$ with $1 - \exp(-n)$ probability (over the coin tosses of the procedure). Thus, the procedure acts as a proxy for the oracle, providing $\{\mathcal{O}^{\star}_k\}_k$ which can now be tested for mutual consistency using the downward self-reducibility of the permanent:

$$\mathsf{Per}(X) = \textstyle\sum_j x_{1,j} \cdot \mathsf{Per}(X_j). \tag{2}$$

Here, X_j is the submatrix formed by removing the first row and j^{th} column. Finally, since $\mathcal{O}_1$ can be verified by direct computation, this procedure tests and accepts sequences where $\mathcal{O}_k$ computes the permanent of a fraction 9/10 of all $k \times k$ matrices; while rejecting sequences of oracles where for some k, $\mathcal{O}_k(X) \neq \mathsf{Per}_k(X)$ on more than, say a fraction 3/10, of the inputs.

A natural attempt to port this argument to real/complex gaussian matrices runs into fatal issues with the self-correction procedures: since the oracles are only required to *approximate* the value of the permanent, a polynomial interpolation procedure incurs an exponential (in the degree) blow-up in the error at the point of interest (see [AK03]). In our work, we circumvent polynomial interpolation and only deal with self-reducibility, noting that eq. (2) expresses the permanent as a *linear* function of permanent of smaller matrices.

1.2 Overview of the Tester

We work with the following notion of quality of an oracle, naturally inspired by the AA conjecture: the approximation guarantee achieved by the oracle on all but a small fraction of the inputs.

Definition 1.1. *For an integer n, an oracle $\mathcal{O}_n : \mathbb{C}^{n \times n} \to \mathbb{C}$, is said to be (δ, η)-good, if, an $n \times n$ matrix X sampled from the Gaussian ensemble satisfies $|\mathcal{O}_n(X) - \mathsf{Per}_n(X)|^2 \leqslant \delta^2 n!$, with probability at least $1 - \eta$ over the sample.*

Note that since the tester is required to be efficient, we (necessarily) allow even good oracles to answer arbitrarily on a small fraction of inputs, because the tester will not encounter these bad inputs with high probability. As an aside, there is also the issue of *additive* vs *multiplicative* approximation, which AA conjecture have similar complexity. In this paper, we stick with additive approximation as defined above.

Our main result is stated informally below (see Theorem 3.1 for a precise statement).

Theorem 1.2 (Main theorem – informal)**.** *There exists an algorithm $\mathcal{A}$ that, given a positive integer n, an error parameter*[2] $\delta \geqslant {}^1/_{\mathrm{poly}(n)}$*, and access to oracles* $\{\mathcal{O}_k\}_{1\leqslant k\leqslant n}$ *such that* $\mathcal{O}_k : \mathbb{C}^{k^2} \to \mathbb{C}$*, has the following behavior:*

- *If for every $k \leqslant n$, the oracle $\mathcal{O}_k$ is* $(\delta, {}^1/_{\mathrm{poly}(n)})$*-good, then $\mathcal{A}$ accepts with probability at least* $1 - {}^1/_{\mathrm{poly}(n)}$*.*
- *If there exists a $k \leqslant n$ such that the oracle $\mathcal{O}_k$ is not even* $(\mathrm{poly}(n) \cdot \delta, {}^1/_{\mathrm{poly}(n)})$*-good, then $\mathcal{A}$ rejects with probability at least* $1 - {}^1/_{\mathrm{poly}(n)}$*.*
- *The query complexity as well as the time complexity of $\mathcal{A}$ is* $\mathrm{poly}(n/\delta)$*.*

We conduct the test in n stages, one stage for each submatrix size. Let $k \leqslant n$ denote a fixed stage, and let $X \in \mathbb{C}^{k^2}$. Now, using downward self-reducibility (eq. (2)), we have,

$$|\mathcal{O}_k(X) - \mathsf{Per}_k(X)| \leqslant \underbrace{\left|\mathcal{O}_k(X) - \textstyle\sum_j x_j \mathcal{O}_{k-1}(X_j)\right|}_{\text{(A)}} + \underbrace{\left|\textstyle\sum_j x_j \left[\mathcal{O}_{k-1}(X_j) - \mathsf{Per}_{k-1}(X_j)\right]\right|}_{\text{(B)}}. \tag{3}$$

Recall that X_j is the submatrix formed by removing the first row and j^{th} column (often referred to as a minor).

We bound term (A) above, by checking if $\mathcal{O}_k$ is a linear function in the variables along the first row (x_j in above), when the rest of the entries of the matrix are fixed; the coefficients of the linear function are determined by querying $\mathcal{O}_{k-1}$ on the k minors along the first row. The tolerance needed in the test is estimated as follows: a good collection of oracles estimates Per_{k-1} up to $\delta\sqrt{(k-1)!}$, and Per_k up to $\delta\sqrt{k!}$ additive error. Further, since the expression is identically zero for the permanent function, we have:

$$\begin{aligned}
\text{(A)} &\leqslant |\mathcal{O}_k(X) - \mathsf{Per}_k(X)| + \left|\textstyle\sum_j x_j \left(\mathcal{O}_{k-1}(X_j) - \mathsf{Per}_{k-1}(X_k)\right)\right| \\
&\leqslant \delta\sqrt{k!} + \left|\textstyle\sum_j x_j \delta\sqrt{(k-1)!}\right| \leqslant \delta\sqrt{k!} \cdot (1 + O(\sqrt{\log n})),
\end{aligned}$$

where the last inequality follows from standard Gaussian tail bounds.

We test this by simply querying the oracles for random X and the minors obtained thereof and checking if the downward self-reducibility condition is approximately met.

The second term, term (B), is linear in the error $\mathcal{O}_{k-1}$ makes on the minors, say $\varepsilon_{k-1}\sqrt{(k-1)!}$ on each minor. A naive argument as above says term (B) is at most $\varepsilon_{k-1}\sqrt{k!} \cdot \Theta(\sqrt{\log n})$. From this and eq. (3), the error in $\mathcal{O}_k$ is at most a $\Theta(\sqrt{\log n})$ factor times the error in $\mathcal{O}_{k-1}$. However, this bound is too weak to conclude anything useful about $\mathcal{O}_n$.

[2] All of the $\mathrm{poly}(\cdot)$ are fixed polynomials, hidden for clarity

We overcome this issue by measuring the error in a root-mean-square (RMS or ℓ_2) sense as follows:

$$\mathsf{err}_2(\mathcal{O}_k) = \sqrt{\mathop{\mathbf{E}}_{X}\left[\mathcal{O}_k(X) - \mathsf{Per}_k(X)\right]^2} = \|\mathcal{O}_k - \mathsf{Per}_k\|_2.$$

Now,

$$\|\mathcal{O}_k - \mathsf{Per}_k\|_2 \leqslant \|\mathcal{O}_k - \textstyle\sum_j x_j \mathcal{O}_{k-1}(X_j)\|_2 + \sqrt{\mathbf{E}\left[\textstyle\sum_j x_j(\mathcal{O}_{k-1} - \mathsf{Per}_{k-1})\right]^2}.$$

The first term is still $\delta\sqrt{k!}\cdot O(\sqrt{\log n})$ assuming the linearity test passes. Since each x_i is an independent standard Gaussian, the second term is at most $\sqrt{k}\cdot \mathsf{err}_2(\mathcal{O}_{k-1}) = \varepsilon_{k-1}\cdot\sqrt{k!}$. Then, $\mathsf{err}_2(\mathcal{O}_k) \leqslant (\delta\sqrt{\log n} + \varepsilon_{k-1})\cdot\sqrt{k!}$, and thus $\mathsf{err}_2(\mathcal{O}_n)$ is at most $\mathrm{poly}(n)\delta\sqrt{n!}$ as we set out to prove! The caveat however is that err_2 as defined cannot be bounded precisely because we necessarily need to discount a small fraction of the inputs: the oracles could be returning arbitrary values on a small fraction, outside the purview of any efficient tester. We deal with this by using a more sophisticated RMS error that discounts an η-fraction of the input:

$$\mathsf{err}_{2,\eta}(\mathcal{O}_k) = \inf_{S:\mu(S)\leqslant\eta}\sqrt{\mathop{\mathbf{E}}_{X}\left[1_s(\mathcal{O}_k(X) - \mathsf{Per}_k(X))\right]^2},$$

where 1_S denotes the indicator function of the set S. We then use a tail inequality on the permanent based on its fourth moment to carry through the inductive argument set up above. This requires a *Tail Test* on the oracles to check that the oracles have a tail similar to the permanent. Our analysis shows that the Linearity and Tail test we design are sufficient and efficient, proving Theorem 1.2.

Organization. In the next section, we set up the notation. Section 3 describes the test we design and follows it up with its analysis.

2 Preliminaries

Notation and Setup. We deal with complex valued functions on the space of square matrices over the complex numbers, $\mathbb{C}^{k\times k}$ for some integer k. We assume $\mathbb{C}^{k\times k}$ is endowed with the standard Gaussian measure $\mathcal{N}(0,1)_{\mathbb{C}}^{k\times k}$. We use the notation $\mathbf{P}_X[E]$ to denote the probability of an event E, when $X\sim\mathcal{N}(0,1)_{\mathbb{C}}^{k\times k}$. We denote by $\mathbf{E}_X[Y]$ to denote the expectation of the random variable Y, when $X\sim\mathcal{N}(0,1)_{\mathbb{C}}^{k\times k}$.

Functions from $\mathbb{C}^d$ to $\{0,1\}$ are called indicator functions (since they indicate inclusion in the set of points where the function's value is 1). We denote the indicator function for a predicate $q(X)$ by $\mathbf{I}[q(X)]$ and define it to be 1 when $q(X)$ is true and 0 otherwise. For example, $\mathbf{I}[|x|\geqslant 2]$ is 1 for all x whose magnitude is at least 2, and 0 otherwise.

Error and ℓ_2 norm of Oracles. The (standard) ℓ_2 norm of a square-integrable function $f:\mathbb{C}^d\to\mathbb{C}$ is denoted by $\|f\|_2$ and is equal to $\mathbf{E}_X[|f|^2]$, where $X\sim\mathcal{N}(0,1)_{\mathbb{C}}^{d}$. An oracle for the permanent is simply a function $\mathcal{O}_k:\mathbb{C}^{k\times k}\to\mathbb{C}$ that can be queried in a single time unit. We will work with a sequence of oracles $\{\mathcal{O}_k\}_{\{k\leqslant n\}}$, one for every dimension k less than n.

Moments of Permanents. The first and the second moments of the permanent under the Gaussian distribution on $k\times k$ matrices are easy to compute: $\mathbf{E}_X[\mathsf{Per}_k(X)] = 0,\ \mathbf{E}_X[|\mathsf{Per}_k(X)|^2] = k!$. We also know the fourth moment of the permanent function for Gaussian matrices, $\mathbf{E}_X[|\mathsf{Per}_k(X)|^4] = (k+1)(k!)^2$ (Lemma 56, [AA11]). This fact and Markov's inequality immediately imply:

Lemma 2.1 (Tail Bound for Permanent)**.** *For every positive integer k, the permanent satisfies* $\mathbf{P}_X[|\mathsf{Per}_k(X)| > T\sqrt{k!}] \leqslant {}^{(k+1)}/_{T^4}$.

3 Testing Approximate Permanent Oracles

Our testing procedure, PTest, has three parameters: a positive integer n, the dimension of the matrices being tested; $\delta \in (0,1]$, the amount of error allowed; and $c \in (0,1]$, a completeness parameter[3]. In addition, it has query access to the sequence of oracles, $\{\mathcal{O}_k\}_{\{k\leqslant n\}}$ being tested. In the following, for a matrix X, we denote the entries in the first row of X by $x_{11}, \ldots, x_{1k}$, and by X_i the minor obtained by removing the first row and the i^{th} column from X. (There will be no confusion since we will only be working with expansion along the first row.)

The guarantees of the tester are twofold: it accepts with probability at least $1-c$, if, for every k, and every $X \in \mathbb{C}^{k\times k}$, we have $|\mathcal{O}_k(X) - \mathsf{Per}_k(X)|^2 \leqslant \delta^2 k!$; on the other hand, the tester almost always rejects if for some $k \leqslant n$, $\mathcal{O}_k(X)$ is not $\mathrm{poly}(n)\delta \cdot \sqrt{k!}$ close to $\mathsf{Per}_k(X)$ with probability $1 - \frac{1}{\mathrm{poly}(n)}$ over X (see below for precise theorems). The query complexity of PTest is bounded by $\mathrm{poly}(n, {}^1/_\delta, {}^1/_c)$. Assuming that each oracle query takes constant time, the time complexity of PTest is also bounded by $\mathrm{poly}(n, {}^1/_\delta, {}^1/_c)$ (see below for precise bounds).

The test consists of two parts: The first is a *linearity* test, that tests that the oracles $\{\mathcal{O}_k\}_{\{k\leqslant n\}}$ satisfy $\mathcal{O}_k(X) \approx \sum_i x_{1i}\mathcal{O}_{k-1}(X_i)$ (observe that the permanent satisfies this exactly). The second part is a *tail* test, that tests that the function does not take large values too often (the permanent satisfies this property too, as shown by Lemma 2.1).

> $\mathsf{LinearityTest}(n,k,\delta)$: Sample a $k \times k$ matrix $X \sim \mathcal{N}(0,1)^{k\times k}_{\mathbb{C}}$. If $k = 1$, output `Reject` unless $|\mathcal{O}_k(X) - X|^2 \leqslant n^2 \cdot \delta^2$. Else, test if:
>
> $$\left|\mathcal{O}_k(X) - \sum_{i=1}^{k} x_{1i}\mathcal{O}_{k-1}(X_i)\right|^2 \leqslant n^2\delta^2 \cdot k!\,.$$
>
> Output `Reject` if it does not hold.
>
> $\mathsf{TailTest}(k,T)$: Sample a $k \times k$ matrix X. Test that $|f_k(X)|^2 \leqslant T^2 k!$. Output `Reject` if it does not hold.

The procedure PTest is formally defined in Figure 1. In the rest of the paper, we prove the following theorem about PTest.

Theorem 3.1 (Main Theorem)**.** *For all $n \in \mathbb{N}, \delta \in (0,1]$, and $c \in (0,1]$, satisfying $n = \Omega\left(\sqrt{\log \frac{1}{c\delta}}\right)$, given oracle access to $\{\mathcal{O}_k\}_{\{k\leqslant n\}}$, where $\mathcal{O}_k : \mathbb{C}^{k\times k} \to \mathbb{C}$, the procedure* PTest *satisfies the following:*

1. **(Completeness)** *If, for every $k \leqslant n$, and every $X \in \mathbb{C}^{k\times k}$, $|\mathcal{O}_k(X) - \mathsf{Per}_k(X)|^2 \leqslant \delta^2 k!$, then* PTest *accepts with probability at least $1-c$.*
2. **(Soundness)** *For every $1 \leqslant k \leqslant n$, either*

[3] We require the mild condition that $n = \Omega\left(\sqrt{\log \frac{1}{\delta c}}\right)$, which is satisfied for large enough n when $c, \delta = \frac{1}{\mathrm{poly}(n)}$.

```
Parameters: A positive integer $n \in \mathbb{N}$, error parameter $\delta \in (0,1]$, and completeness parameter $c \in (0,1]$.
Requires: Oracle access to $\{\mathcal{O}_k\}_{\{k \leqslant n\}}$, where $\mathcal{O}_k : \mathbb{C}^{k \times k} \to \mathbb{C}$.
  1. Set the following variables: $T = {}^{4n}/_{\delta\sqrt{c}}$, $d = {}^{192n^2}/_{\delta^4 c}$.
  2. For each $1 \leqslant k \leqslant n$,
      (a) Run LinearityTest$(n, k, \delta)$ $d$ times.
      (b) Run TailTest$(k, T)$ $d$ times.
  3. If none of the above tests output Reject, output Accept.
```

Figure 1: The tester PTest

> *There exists an indicator function* $1_k : \mathbb{C}^{k\times k} \to \{0,1\}$ *satisfying* $\mathbf{E}_X[1_k(X)] \geqslant 1 - \frac{\delta^4 c}{64n}$, *such that,* $\mathbf{E}_X[1_k(X) \cdot |\mathcal{O}_k(X) - \mathsf{Per}_k(X)|^2] \leqslant (2nk\delta)^2 k!\,.$

or else,

> PTest *outputs* `Reject` *with probability at least* $1 - e^{-n}$.

3. **(Complexity)** *The total number of queries made by* PTest *is* $O(n^4\delta^{-4}c^{-1})$. *Moreover, assuming that each oracle query takes constant time, the time required by* PTest *is also* $O(n^4\delta^{-4}c^{-1})$.

The three parts of the theorem are proved separately in Theorem 3.4, Theorem 3.6 and Theorem 3.12 in Sections 3.1, 3.2 and 3.3 respectively.

Remark 3.2. *Observe that, assuming both* ${}^1/_c$ *and* ${}^1/_\delta$ *are polynomial in* n, *the query complexity is* $\mathrm{poly}(n)$, *and hence, even if the oracles* $\{\mathcal{O}_k\}_{k\leqslant n}$ *satisfy* $|\mathcal{O}_k(X) - \mathsf{Per}_k(X)|^2 \leqslant \delta^2 k!$ *only with probability* $1 - \frac{1}{\mathrm{poly}(n)}$, PTest *would still accept with probability* $1 - c - \frac{1}{\mathrm{poly}(n)}$.

Remark 3.3. *Observe that the (informal) main theorem (Theorem 1.2) stated in the introduction follows from Theorem 3.1 from a simple Markov argument. Given* $\delta = \Omega({}^1/_{\mathrm{poly}(n)})$, *set* $c = \frac{1}{\mathrm{poly}(n)}$ *and note that the completeness follows directly from Theorem 3.1 and the previous remark. Further, from the* Soundness *claim of Theorem 3.1, we have an indicator function* $1_k : \mathbb{C}^{k\times k} \to \{0,1\}$ *satisfying* $\mathbf{E}_X[1_k(X)] \geqslant 1 - \frac{\delta^4 c}{64n} \geqslant 1 - \frac{1}{\mathrm{poly}(n)}$, *such that,* $\mathbf{E}_X[1_k(X) \cdot |\mathcal{O}_k(X) - \mathsf{Per}_k(X)|^2] \leqslant (2nk\delta)^2 k! \leqslant \mathrm{poly}(n)\cdot\delta^2 k!\,.$ *Applying Markov's inequality, we have that* $\mathbf{P}\left[1_k(X)\cdot|\mathcal{O}_k(X) - \mathsf{Per}_k(X)|^2 \geqslant \mathrm{poly}(n)\delta^2 k!\right] \leqslant {}^1/_{\mathrm{poly}(n)}$. *Now, note that* 1_k *is an indicator function, and* $\mathbf{P}[1_k(X) = 0]$ *is at most* $1/\mathrm{poly}(n)$. *This, along with the previous expression gives that the tester outputs* `Reject` *if the sequence of oracles is not even* $(\mathrm{poly}(n)\cdot\delta, {}^1/_{\mathrm{poly}(n)})$*-good.*

3.1 Completeness

We first prove the completeness of PTest: that a $(\delta, 0)$-good sequence of oracles is accepted with probability at least $1 - c$.

Theorem 3.4 (Completeness)**.** *If, for every $k \leqslant n$, and every $X \in \mathbb{C}^{k\times k}$, $|\mathcal{O}_k(X) - \mathsf{Per}_k(X)|^2 \leqslant \delta^2 k!$, then the procedure* PTest *accepts with probability at least $1-c$.*

Proof. Suppose we are given a sequence of oracles $\{\mathcal{O}_k\}_{k\leqslant n}$ such that for all $k \leqslant n$, we have that $|\mathcal{O}_k(X) - \mathsf{Per}_k(X)|^2 \leqslant \delta^2 \cdot k!$. Let X denote a randomly sampled $k \times k$ matrix.

We first bound the probability that the oracles $\{\mathcal{O}_k\}_{\{k\leqslant n\}}$ fail a linearity test. For $k = 1$, it is easy to see that $\mathsf{LinearityTest}(n, 1, \delta)$ never outputs `Reject` upon querying $\mathcal{O}_1$. For larger k, we have the following lemma that shows that $\mathcal{O}_k(X) \approx \sum_i x_{1i}\mathcal{O}_{k-1}(X_i)$, and hence $\mathsf{LinearityTest}$ outputs `Reject` only with small probability.

Lemma 3.5 (Completeness for $\mathsf{LinearityTest}$)**.** *For every $2 \leqslant k \leqslant n$, the oracles $\{\mathcal{O}_k\}_{\{k\leqslant n\}}$ satisfy*

$$\mathop{\mathbf{P}}_X[|\mathcal{O}_k(X) - \sum_i x_{1i}\mathcal{O}_{k-1}(X_i)|^2 > n^2\delta^2 k!\,] \leqslant 2e^{-\frac{(n-1)^2}{2}}.$$

We first complete a proof of the theorem assuming this lemma. This lemma implies that every call to $\mathsf{LinearityTest}(n, k, \delta)$ outputs `Reject` with probability at most $2e^{-\frac{(n-1)^2}{2}}$.

Next, we bound the probability that the oracles $\{\mathcal{O}_k\}_{\{k\leqslant n\}}$ fail a $\mathsf{TailTest}$. Using the tail bound for the permanent given by Lemma 2.1, we get, $\mathbf{P}_X[|\mathsf{Per}_k(X)| > (T-\delta)\sqrt{k!}] \leqslant (k+1)/(T-\delta)^4$. Since $|\mathcal{O}_k(X) - \mathsf{Per}_k(X)| \leqslant \delta \cdot \sqrt{k!}$, we use it in the above bound to get $\mathbf{P}_X[|\mathcal{O}_k(X)| > T\sqrt{k!}] \leqslant (k+1)/(T-\delta)^4$. Thus, every call to $\mathsf{TailTest}$ fails with probability at most $\frac{(n+1)}{(T-\delta)^4}$.

Now applying a union bound, we get that for n that is $\Omega\left(\sqrt{\log\frac{1}{\delta c}}\right)$, PTest outputs `Reject` with probability at most

$$\left(2e^{-\frac{(n-1)^2}{2}} + \frac{(n+1)}{(T-\delta)^4}\right) dn \leqslant 384\frac{n^3}{\delta^4 c} \cdot e^{-(n-1)^2/2} + \frac{192(n+1)n^3 c}{(4n - \delta^2\sqrt{c})^4} \leqslant c.$$

□

We now give a proof of Lemma 3.5.

Proof. (*of Lemma 3.5*). We have,

$$\begin{aligned}\left|\mathcal{O}_k(X) - \sum_i x_{1i}\mathcal{O}_{k-1}(X_i)\right| &\leqslant |\mathcal{O}_k(X) - \mathsf{Per}_k(X)| + \left|\sum_i x_{1i}\mathsf{Per}_{k-1}(X_i) - \sum_i x_{1i}\mathcal{O}_{k-1}(X_i)\right| \\ &\leqslant \delta\sqrt{k!} + \left|\sum_i x_{1i}(\mathsf{Per}_{k-1}(X_i) - \mathcal{O}_{k-1}(X_i))\right|. \qquad (4)\end{aligned}$$

Now, since $x_{11}, \ldots, x_{1k}$ are independent Gaussians with unit variance, $\sum_i x_{1i}(\mathsf{Per}_{k-1}(X_i) - \mathcal{O}_{k-1}(X_i))$ is a Gaussian with variance $\sum_i |\mathsf{Per}_{k-1}(X_i) - \mathcal{O}_{k-1}(X_i)|^2 \leqslant k \cdot \delta^2 \cdot (k-1)! = \delta^2 \cdot k!$. Thus, the second term in Equation (4) is bounded by $(n-1)\delta \cdot \sqrt{k!}$, except with probability at most $2e^{-\frac{(n-1)^2}{2}}$. Thus, $|\mathcal{O}_k(X) - \sum_i x_{1i}\mathcal{O}_{k-1}(X_i)| \leqslant n\delta \cdot \sqrt{k!}$, except with probability at most $2e^{-\frac{(n-1)^2}{2}}$. □

3.2 Soundness

The interesting part of the analysis is the soundness for PTest, which we prove in this section. Given $\{\mathcal{O}_k\}_{\{k\leqslant n\}}$, we need to define the following indicator functions to aid our analysis:

$$\begin{aligned}
1_k^{LIN}(X) &= \begin{cases} \mathbf{I}[(\mathcal{O}_k(X)-X)^2 \leqslant n^2\delta^2], & \text{if } k=1 \\ \mathbf{I}[(\mathcal{O}_k(X)-\sum_i x_{1i}\mathcal{O}_{k-1}(X_i))^2 \leqslant n^2\delta^2 k!\,], & \text{if } 2\leqslant k\leqslant n \end{cases} \\
1_k^{TAIL}(X) &= \mathbf{I}[\mathcal{O}_k(X)^2 \leqslant T^2\cdot k!\,], \\
1_k^{PERM}(X) &= \mathbf{I}[\mathsf{Per}_k(X)^2 \leqslant T^2\cdot k!\,], \\
1_k(X) &= 1_k^{LIN}(X)\wedge 1_k^{TAIL}(X)\wedge 1_k^{PERM}(X).
\end{aligned} \tag{5}$$

We now prove the following theorem.

Theorem 3.6 (Soundness)**.** *Let the indicator function 1_k be as defined by Equation (5). For every $k\leqslant n$, either both of the following two conditions hold:*

1. *The indicator 1_k satisfies $\mathbf{E}_X[1_k(X)] \geqslant 1-\frac{\delta^4 c}{64n}$.*
2. *The oracle $\mathcal{O}_k$ and the indicator 1_k satisfy $\mathbf{E}_X[1_k(X)\cdot|\mathcal{O}_k(X)-\mathsf{Per}_k(X)|^2] \leqslant (2nk\delta)^2 k!\,$,*

or else, PTest *outputs* `Reject` *with probability at least $1-e^{-n}$.*

Proof. We first prove the following lemma that shows that for all $k\leqslant n$, the expectation of 1_k is large.

Lemma 3.7 (Large Expectation of 1_k)**.** *Either, for every k, the indicator function 1_k satisfies $\mathbf{E}_X[1_k(X)] \geqslant 1-\frac{\delta^4 c}{64n}$, or else,* PTest *outputs* `Reject` *with probability at least $1-e^{-n}$.*

The first part of the theorem follows immediately from this lemma. The proof of this lemma is given later in this section.

For the second part of the theorem, we prove the following inductive claim about the oracles $\{\mathcal{O}_k\}$.

Lemma 3.8. *(Main Induction Lemma) Suppose that for some $2\leqslant k\leqslant n$, we have,*

$$\mathop{\mathbf{E}}_{X\,\in\mathbb{C}^{(k-1)\times(k-1)}}[1_{k-1}(X)\cdot|\mathcal{O}_{k-1}(X)-\mathsf{Per}_{k-1}(X)|^2)] \leqslant \varepsilon_{k-1}^2(k-1)!\,,$$

then, either we have,

$$\mathop{\mathbf{E}}_{X\,\in\mathbb{C}^{k\times k}}[1_k(X)\cdot|\mathcal{O}_k(X)-\mathsf{Per}_k(X)|^2)] \leqslant (\varepsilon_{k-1}+2n\delta)^2 k!\,,$$

or else, PTest *outputs* `Reject` *with probability at least $1-e^{-n}$.*

The proof of this lemma is also presented later in this section. Assuming this lemma, we can complete the proof of soundness for PTest.

For the second part of the theorem, we first show that the required bound holds for $k=1$. We know that for any $X\in\mathbb{C}$, whenever $1_1(X)=1$, we have $|\mathcal{O}_1(X)-X|^2\leqslant n^2\delta^2$. Thus,

$$\mathop{\mathbf{E}}_X[1_1(X)\cdot|\mathcal{O}_1(X)-\mathsf{Per}_1(X)|^2] \leqslant \mathop{\mathbf{E}}_X[1_1^{LIN}(X)\cdot|\mathcal{O}_1(X)-X|^2] \leqslant n^2\delta^2 < (2n\delta)^2\cdot 1!\,.$$

This gives us our base case. Assume that there is a $2 \leqslant j \leqslant n$ such that,

$$\mathop{\mathbf{E}}_{X \in \mathbb{C}^{(j-1)\times(j-1)}}[1_{j-1}(X) \cdot |\mathcal{O}_{j-1}(X) - \mathsf{Per}_{j-1}(X)|^2] \leqslant (2n(j-1)\delta)^2 \cdot (j-1)!\,.$$

Now, we use Lemma 3.8 to deduce that either,

$$\mathop{\mathbf{E}}_{X \in \mathbb{C}^{j\times j}}[1_j(X) \cdot |\mathcal{O}_j(X) - \mathsf{Per}_j(X)|^2] \leqslant (2nj\delta)^2 \cdot j!\,,$$

or else, PTest outputs `Reject` with probability at least $1-e^{-n}$. Thus, by induction, either for every $k \leqslant n$,

$$\mathop{\mathbf{E}}_{X}[1_k(X) \cdot |\mathcal{O}_k(X) - \mathsf{Per}_k(X)|^2] \leqslant (2nk\delta)^2 \cdot k!\,,$$

or else, PTest outputs `Reject` with probability at least $1-e^{-n}$. This completes the proof of the theorem. □

Large expectation of 1_k. We now prove Lemma 3.7 that states that the expectation of 1_k is large.

Proof. (*of Lemma 3.7*). We begin by making several claims about the structure the oracles $\{\mathcal{O}_k\}_{\{k \leqslant n\}}$ must have with high probability, assuming that PTest accepts. First, we claim that $\mathcal{O}_1$ must be close to the identity function.

Claim 3.9 (Soundness of $\mathsf{LinearityTest}$ for $\mathcal{O}_1$)**.** *Either the oracle $\mathcal{O}_1$ satisfies that*

$$\mathop{\mathbf{P}}_{X}\left[|\mathcal{O}_1(X) - X|^2 > n^2\delta^2\right] \leqslant \frac{n}{d}, \tag{6}$$

or else, PTest *outputs* `Reject` *with probability at least* $1-e^{-n}$.

A proof of this claim is included later in the section for completeness. We also need the following two claims stating that for every $2 \leqslant k \leqslant n$, $\mathcal{O}_k(X) \approx \sum_i x_{1i}\mathcal{O}_{k-1}(X_i)$ very often, and that $\mathcal{O}_k(X)$ does not take large values too often.

Claim 3.10 (Soundness of $\mathsf{LinearityTest}$)**.** *Either the oracles $\{\mathcal{O}_k\}$ satisfy the following for every $2 \leqslant k \leqslant n$,*

$$\mathop{\mathbf{P}}_{X}\left[|\mathcal{O}_k(X) - \textstyle\sum_i x_{1i}\mathcal{O}_{k-1}(X)|^2 > n^2\delta^2 k!\right] \leqslant \frac{n}{d},$$

or else, PTest *outputs* `Reject` *with probability at least* $1-e^{-n}$.

Claim 3.11 (Soundness of $\mathsf{TailTest}$)**.** *Either the oracles $\{\mathcal{O}_k\}$ satisfy the following for every $k \leqslant n$,*

$$\mathop{\mathbf{P}}_{X}\left[|\mathcal{O}_k(X)|^2 > T^2 \cdot k!\,\right] \leqslant \frac{n}{d},$$

or else, PTest *outputs* `Reject` *with probability at least* $1-e^{-n}$.

The proofs of these claims are very similar to that of Claim 3.9 and we skip them. We can restate the above claims in terms of 1_k^{LIN} and 1_k^{TAIL} defined in Equation (5) as follows: Either, for every $k \leqslant n$,

$$\mathop{\mathbf{E}}_X[1_k^{LIN}(X)] \geqslant 1 - \frac{n}{d},\ \mathop{\mathbf{E}}_X[1_k^{TAIL}(X)] \geqslant 1 - \frac{n}{d}, \tag{7}$$

or else, PTest will output $\texttt{Reject}$ with probability at least $1 - e^{-n}$.

From Lemma 2.1, we know that the permanent does not take large values too often. To be precise,

$$\mathop{\mathbf{P}}_X[|\mathsf{Per}_k(X)|^2 > T^2 \cdot k!\,] \leqslant \frac{(k+1)}{T^4}.$$

Again, this implies that $\mathbf{E}_X[1_k^{PERM}] \geqslant 1 - \frac{(k+1)}{T^4}$. Combining these three claims, we can now prove our lemma.

We know that $1_k = 1_k^{LIN} \wedge 1_k^{TAIL} \wedge 1_k^{PERM}$. We know that if either of the claims in Equation (7) does not hold, PTest outputs $\texttt{Reject}$ with probability at least $1 - e^{-n}$. Thus, we assume that both the claims in Equation (7) hold and apply the union bound to get,

$$\begin{aligned}\mathop{\mathbf{E}}_X[1_k(X)] &\geqslant 1 - \mathop{\mathbf{E}}_X[1 - 1_k^{LIN}(X)] - \mathop{\mathbf{E}}_X[1 - 1_k^{TAIL}(X)] - \mathop{\mathbf{E}}_X[1 - 1_k^{PERM}(X)] \\ &\geqslant 1 - \frac{n}{d} - \frac{n}{d} - \frac{k+1}{T^4} \geqslant 1 - \frac{\delta^4 c}{96n} - \frac{(n+1)\delta^4 c^2}{256 n^4} \geqslant 1 - \frac{\delta^4 c}{64n},\end{aligned}$$

for large enough n. □

Main Induction Lemma. We now give a proof of the main induction lemma.

Proof. (*of Lemma 3.8*). Recall that X_i is the minor obtained by deleting the first row and the i^{th} column from X. We first split the probability space for $X \in \mathbb{C}^{k \times k}$ according to whether all of its minors X_i satisfy $1_{k-1}(X_i) = 1$ or not.

$$\begin{aligned}\|1_k(X)(\mathcal{O}_k(X) - \mathsf{Per}_k(X))\|^2 = &\overbrace{\|1_k(X)\prod_i 1_{k-1}(X_i)(\mathcal{O}_k(X) - \mathsf{Per}_k(X))\|^2}^{\text{(C)}} \\ &+ \underbrace{\|1_k(X)(1 - \prod_i 1_{k-1}(X_i))(\mathcal{O}_k(X) - \mathsf{Per}_k(X))\|^2}_{\text{(D)}}\end{aligned} \tag{8}$$

Let $\tilde{1}_k(X) = 1_k(X)\prod_i 1_{k-1}(X_i)$. Term (C), above, is bounded by adding and subtracting the expression $\sum_i x_{1i}\mathcal{O}_{k-1}(X_i)$ and then expanding the permanent along the first row.

$$\begin{aligned}\|\tilde{1}_k(X)(\mathcal{O}_k(X) - \mathsf{Per}_k(X))\| \leqslant &\underbrace{\|\tilde{1}_k(X)[\mathcal{O}_k(X) - \sum_i x_{1i}\mathcal{O}_{k-1}(X_i)]\|}_{\text{(E)}} \\ &+ \underbrace{\|\tilde{1}_k(X)[\sum_i x_{1i}\mathcal{O}_{k-1}(X_i) - \sum_i x_{1i}\mathsf{Per}_{k-1}(X_i)]\|}_{\text{(F)}}\end{aligned} \tag{9}$$

We know that for all X such that $1_k(X) = 1$, $|\mathcal{O}_k(X) - \sum_i x_{1i}\mathcal{O}_{k-1}(X_i)|^2$ is bounded by $n^2\delta^2 k!$. Thus, term (E) in eq. (9) is at most $n\delta\sqrt{k!}$.

$$\text{(E)} \leqslant \|1_k(X)(\mathcal{O}_k(X) - \sum_i x_{1i}\mathcal{O}_{k-1}(X_i))\| \leqslant n\delta\sqrt{k!} \tag{10}$$

Term (F) is bounded by using the induction assumption:

$$\begin{aligned}
\text{(F)}^2 &= \Big\|1_k(X)\prod_i 1_{k-1}(X_i)\Big[\sum_i x_{1i}\mathcal{O}_{k-1}(X_i) - \sum_i x_{1i}\mathsf{Per}_{k-1}(X_i)\Big]\Big\|^2 \\
&\leqslant \mathop{\mathbf{E}}_{X_1,\dots X_k}\mathop{\mathbf{E}}_{x_{11},\dots,x_{1k}}\left[\prod_i 1_{k-1}(X_i)\cdot\left|\sum_i x_{1i}\mathcal{O}_{k-1}(X_i) - \sum_i x_{1i}\mathsf{Per}_{k-1}(X_i)\right|^2\right] \\
&\leqslant \mathop{\mathbf{E}}_{X_1,\dots X_k}\left[\prod_i 1_{k-1}(X_i)\cdot\sum_i|\mathcal{O}_{k-1}(X_i) - \mathsf{Per}_{k-1}(X_i)|^2\right] \\
&\leqslant \sum_i \mathop{\mathbf{E}}_{X_i}\left[1_{k-1}(X_i)\cdot|\mathcal{O}_{k-1}(X_i) - \mathsf{Per}_{k-1}(X_i)|^2\right] \\
&\leqslant k\varepsilon_{k-1}^2(k-1)! = \varepsilon_{k-1}^2 k!
\end{aligned} \tag{11}$$

Combining eqs. (9), (10), and (11), we get,

$$\text{(C)} = \mathop{\mathbf{E}}_X\left[1_k(X)\prod_i 1_{k-1}(X_i)\cdot|\mathcal{O}_k(X) - \mathsf{Per}_k(X)|^2\right] \leqslant (\varepsilon_{k-1} + n\delta)^2\cdot k!\,. \tag{12}$$

Next, we bound term (D) as follows. First use lemma 3.7 to deduce $\mathbf{P}_X[1_{k-1}(X_i) = 0] \leqslant \frac{\delta^4 c}{64n}$ (If it does not hold, we know that PTest outputs `Reject` with probability at least $1 - e^{-n}$). Whenever $1_k(X) = 1$, we have $|\mathcal{O}_k(X)|\leqslant T\sqrt{k!}$ and $|\mathsf{Per}_k(X)|\leqslant T\sqrt{k!}$. This implies that $1_k(X)\cdot|\mathcal{O}_k(X) - \mathsf{Per}_k(X)|^2\leqslant 4T^2k!$ everywhere. Thus, we have,

$$\begin{aligned}
\text{(D)} &= \|1_k(X)(1 - \prod_i 1_{k-1}(X_i))(\mathcal{O}_k(X) - \mathsf{Per}_k(X))\|^2 \leqslant 4T^2k!\mathop{\mathbf{E}}_X\left[1 - \prod_i 1_{k-1}(X_i)\right] \\
&\leqslant 4T^2k!\mathop{\mathbf{E}}_X\left[\sum_i(1 - 1_{k-1}(X_i))\right] \\
&\leqslant 4T^2k!\cdot k\cdot\frac{\delta^4 c}{64n} \leqslant n^2\delta^2\cdot k!\,.
\end{aligned} \tag{13}$$

Combining eqs. (8), (12), and (13) completes the proof:

$$\mathbf{E}\left[1_k(X)\cdot|\mathcal{O}_k(X) - \mathsf{Per}_k(X)|^2\right] \leqslant \Big((\varepsilon_{k-1} + n\delta)^2 + n^2\delta^2\Big)\cdot k! \leqslant (\varepsilon_{k-1} + 2n\delta)^2\cdot k!\,.$$ □

Proof of Claim 3.9 For completeness, we include a proof of Claim 3.9.

Proof. Assume that the oracle $\mathcal{O}_1$ does not satisfy Equation (6). We know that $|\mathcal{O}_1(X) - X|^2 > n^2\delta^2$ iff $\mathsf{LinearityTest}(n, 1, \delta)$ outputs `Reject` when X is sampled by the procedure. Thus, the measure of points $X \in \mathbb{C}$ that would fail the test $\mathsf{LinearityTest}(n, 1, \delta)$ is at least n/d. This implies that the probability that none of the d calls to $\mathsf{LinearityTest}(n, 1, \delta)$ made by PTest output `Reject` is at most $(1 - n/d)^d \leqslant e^{-n}$. □

3.3 Complexity

We finally note that the complexity of PTest is polynomially bounded in the input parameters.

Theorem 3.12 (Query and Time Complexity)**.** *The total number of queries made by* PTest *to all the oracles is $O(n^2 d) = O(n^4 \delta^{-4} c^{-1})$. Moreover, assuming that each oracle query takes constant time, the time required by* PTest *is also $O(n^2 d) = O(n^4 \delta^{-4} c^{-1})$.*

Proof. By the definition of PTest, it makes dn calls to LinearityTest and dn calls to TailTest. Each call to LinearityTest with parameters n, k, δ, makes at most $k+1$ queries to the oracles (for $k = 1$, it makes only one query), and requires $O(k)$ time. Each call to TailTest makes 1 query and requires $O(1)$ time. Thus, the total number of queries made is $O(dn^2) = O(n^4 \delta^{-4} c^{-1})$, and the total time required is also $O(dn^2) = O(n^4 \delta^{-4} c^{-1})$. □

Thus, if $1/\delta$ and $1/c$ are poly(n), the query complexity of PTest is also poly(n).

4 Acknowledgments

The authors would like to thank Madhur Tulsiani and Rishi Saket for extensive discussions during early stages of this work. We would also like to thank Scott Aaronson, Alex Arkhipov, Swastik Kopparty and Srikanth Srinivasan for helpful discussions.

References

[AA11] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In Lance Fortnow and Salil P. Vadhan, editors, *STOC*, pages 333–342. ACM, 2011. 1, 2, 5

[AK03] Sanjeev Arora and Subhash Khot. Fitting algebraic curves to noisy data. *J. Comput. Syst. Sci.*, 67(2):325–340, September 2003. 3

[Bro86] Andrei Z. Broder. How hard is it to marry at random? (on the approximation of the permanent). In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, STOC '86, pages 50–58, New York, NY, USA, 1986. ACM. 2

[CPS99] Jin-Yi Cai, Aduri Pavan, and D. Sivakumar. On the hardness of permanent. In *Proceedings of the 16th annual conference on Theoretical aspects of computer science*, STACS'99, pages 90–99, Berlin, Heidelberg, 1999. Springer-Verlag. 2

[GLR$^+$91] Peter Gemmell, Richard Lipton, Ronitt Rubinfeld, Madhu Sudan, and Avi Wigderson. Self-testing/correcting for polynomials and for approximate functions. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, STOC '91, pages 33–42, New York, NY, USA, 1991. ACM. 2

[GS92] Peter Gemmell and Madhu Sudan. Highly resilient correctors for polynomials. *Inf. Process. Lett.*, 43(4):169–174, September 1992. 2, 3

[JS89] Mark Jerrum and Alistair Sinclair. Approximating the permanent. *SIAM J. Comput.*, 18(6):1149–1178, December 1989. 2

[JSV04] 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, July 2004. 2

[LFKN92] Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, October 1992. 3

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *FOCS*, pages 124–134. IEEE Computer Society, 1994. 2

[Val79] Leslie G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979. 1

A Remaining proofs

Claim A.1 (Soundness of LinearityTest)**.** *Either the oracles* $\{\mathcal{O}_k\}$ *satisfy the following for every* $2 \leqslant k \leqslant n$,

$$\mathop{\mathbf{P}}_{X}\left[\left|\mathcal{O}_k(X) - \textstyle\sum_i x_{1i}\mathcal{O}_{k-1}(X)\right|^2 > n^2\delta^2 k!\right] \leqslant \frac{n}{d}, \tag{14}$$

or else, PTest *outputs* `Reject` *with probability at least* $1 - e^{-n}$.

Proof. Assume that there exists a k, such that $2 \leqslant k \leqslant n$ and the oracle $\mathcal{O}_k$ does not satisfy Equation (14). We recall that LinearityTest(n,k,δ) outputs `Reject` iff the sampled $X \in \mathbb{C}^{k\times k}$ satisfies $\left|\mathcal{O}_k(X) - \sum_i x_{1i}\mathcal{O}_{k-1}(X)\right|^2 > n^2\delta^2 k!$. Thus, a randomly sampled X will fail LinearityTest(n,k,δ) with probability at least n/d. This implies that the probability that none of the d calls made by PTest to LinearityTest(n,k,δ) output `Reject` is at most $(1 - n/d)^n \leqslant e^{-n}$. □

Claim A.2 (Soundness of TailTest)**.** *Either the oracles* $\{\mathcal{O}_k\}$ *satisfy the following for every* $k \leqslant n$,

$$\mathop{\mathbf{P}}_{X}\left[\left|\mathcal{O}_k(X)\right|^2 > T^2 \cdot k!\,\right] \leqslant \frac{n}{d}, \tag{15}$$

or else, PTest *outputs* `Reject` *with probability at least* $1 - e^{-n}$.

Proof. Suppose for some $k \leqslant n$, the oracle $\mathcal{O}_k$ does not satisfy Equation 15. Thus, for this choice of k, the test TailTest(k,T) fails with probability at least n/d. This implies that the probability that at least one of the d calls by PTest to TailTest(k,T) outputs `Reject` with probability at least $1 - (1 - n/d)^d \geqslant 1 - e^{-n}$. □