

Some exposition of “The Exponential Time Hypothesis Is False”*

October 27, 2026

1 The theorem and the interpolation plan

Let $A \in \{0, 1\}^{N \times N}$ have at most three ones in each row and column. Equivalently, A is the adjacency matrix of a bipartite graph with N vertices on each side and maximum degree three. We will count its perfect matchings.

Theorem 1.1. *The permanent of such a matrix A can be computed exactly and deterministically in*

$$2^{O(N(\log \log N)^2 / \sqrt{\log N})} \quad (1.1)$$

bit operations.

This is already enough for the complexity consequence of the supplied manuscript [OAI26]. Under #ETH, counting perfect matchings in bipartite graphs of maximum degree three requires exponential time [Cur18, Corollary 4.1(1)]. The theorem contradicts #ETH, and hence ETH, which implies #ETH.

Recover the hard value from easier ones. We will construct a polynomial matrix $A(\theta)$ with $A(0) = A$. At selected nonzero points $t = 1, \dots, n$, the matrices $A(t)$ will have a structure that makes their permanents easier to compute. Set

$$H(\theta) = \text{per } A(\theta).$$

Our answer is $H(0)$. If the entries of $A(\theta)$ have degree at most b , then $\deg H \leq Nb$. The n values $H(t)$ alone will not determine this polynomial. Instead, at each point we will compute

$$H(t), H'(t), \dots, H^{(q)}(t), \quad q = \left\lfloor \frac{Nb}{n} \right\rfloor.$$

Since $n(q+1) > Nb$, standard Hermite interpolation recovers $H(0)$.

The challenge is to obtain these derivatives without losing the structure available at t . The permanent-of-a-sum identity does exactly this: an order- j derivative uses at most j perturbed entries, leaving a large complementary submatrix evaluated at the unchanged point t .

*Produced by `odonnell@cs.cmu.edu` through interaction with LLMs.

What makes the selected points easier? At each point we will have

$$A(t) = D_t + R, \tag{1.2}$$

where D_t is supported on small corresponding row and column blocks and R has small rank. The blocks may be rectangular. A theorem proved in Section 4 computes such a permanent, and every square submatrix permanent, in

$$2^{O(s)} N^{O(r)}$$

arithmetic operations, where s bounds the block sizes and r bounds the rank. Its proof records local partial matchings in polynomials and uses the r directions of a rank factorization to join the remaining endpoints.

Where the deformation comes from. We begin with the graph. A balanced coloring by strings gives a factorization

$$A = PJQ,$$

where the intermediate coordinates form a disjoint union of Boolean n -cubes, and J sends each vertex to its antipode. We then use a degree- $O(\sqrt{n \log n})$ family of functions

$$g_\theta(x) = p\left(\sum_{i=1}^n (1 - \theta/i)x_i\right), \quad x \in \{-1, 1\}^n.$$

The univariate polynomial p is chosen so that g_0 agrees with parity except on an n^{-10} fraction of the cube. At $\theta = t$, the function g_t is independent of x_t .

Two standard Fourier facts translate these properties into matrices. The XOR matrix

$$\mathcal{F}_g[S, T] = \widehat{g}(S \oplus T)$$

is close to J in rank when g agrees with parity on most inputs; it preserves coordinate t when g is independent of x_t . Applying the same matrix on every cube component, define

$$G(\theta) = J - \mathcal{F}_{g_0} + \mathcal{F}_{g_\theta}, \quad A(\theta) = PG(\theta)Q. \tag{1.3}$$

Then $G(0) = J$, so $A(0) = A$. The coloring converts coordinate preservation into small blocks, and multiplication by P, Q cannot increase the correction rank.

This orders the proof: first color the graph and specify the factorization, then construct the Fourier matrices, then prove the permanent-evaluation and derivative-recovery theorems. The final parameter choice makes all three costs—block size, rank, and derivative order—sublinear in N .

Notation and implementation. The input matrix has order N . The smaller parameter n is both the Boolean dimension and the number of interpolation points. We use d for the color alphabet size and M for the intermediate dimension. All logarithms are base two unless written $\ln$. Boolean Fourier notation follows [OD21]: $x^S = \prod_{i \in S} x_i$, and $S \oplus T$ is symmetric difference. A submatrix $C[S, T]$ has the indicated rows and columns; the permanent of the empty matrix is one. All derivatives are ordinary derivatives over $\mathbb{Q}$.

The main evaluation algorithm stores coefficient tables. Appendix A gives a polynomial-space replacement, and Appendix B proves polynomial bit cost per arithmetic operation. We use Ryser's formula and the permanent-of-a-sum identity as standard identities [Rys63, Min78].

2 Coloring the graph and factoring through antipodes

We want to put the vertices into balanced color classes so that every pair of classes contains at most one edge. The colors will already be strings in $[d]^n$. We also require the two colors on each edge to disagree in every coordinate. This last condition will make the intermediate set a union of full Boolean cubes, with no missing directions.

For orientation, our eventual parameters will be

$$d = \Theta(\log N), \quad n = \Theta\left(\frac{\log N}{\log \log N}\right), \quad K = d^n = \tilde{\Theta}(\sqrt{N}).$$

Their precise choice is deferred to Section 6.

2.1 A balanced coloring by strings

Lemma 2.1 (Balanced harmonious coloring). *Let $G = (V_L, V_R, E)$ be bipartite, with N vertices on each side and maximum degree three. Suppose*

$$K = d^n, \quad 16\sqrt{N} \leq K \leq N, \quad d \geq 12n.$$

There are deterministically computable maps $\alpha : V_L \rightarrow [d]^n$ and $\beta : V_R \rightarrow [d]^n$ such that:

- (i) *each string colors at most $\lceil 2N/K \rceil$ vertices on either side;*
- (ii) *distinct edges have distinct ordered pairs of endpoint colors;*
- (iii) *if $(u, v) \in E$, then $\alpha(u)_i \neq \beta(v)_i$ for every $i \in [n]$.*

The construction takes polynomial time in N .

Proof. This is an exercise in the greedy algorithm; we give the details. Put $L_0 = \lceil 2N/K \rceil$. A color is full on one side when already used L_0 times there. At any stage at most $N/L_0 \leq K/2$ colors are full, and the hypotheses imply $K \geq 256$.

First color the left vertices, requiring different colors for vertices with a common right neighbor. Each left vertex has at most six such conflicts. In a greedy pass, full colors and previously colored conflicting vertices forbid at most $K/2 + 6 < K$ colors. Thus every right vertex now has neighbors of distinct colors.

Next color the right vertices. For a vertex v , avoid full colors and colors that would repeat a previously used pair on an edge. A fixed left color occurs on at most L_0 vertices, incident to at most $3L_0$ edges. Since v has at most three neighbors, pair reuse forbids at most $9L_0$ colors.

Also avoid equality with a neighboring color in any coordinate. A prescribed symbol in one coordinate excludes K/d strings. There are at most $3n$ such restrictions, so they exclude at most $3nK/d \leq K/4$ strings. Altogether, the number of forbidden colors is at most

$$\frac{K}{2} + 9L_0 + \frac{K}{4} \leq \frac{3K}{4} + \frac{18N}{K} + 9 \leq \frac{3K}{4} + \frac{27K}{256} < K.$$

Choose the first available string. The pairs within this column are distinct because its neighboring left colors are distinct, and the choice prevents every earlier pair from recurring. Keeping color counts and a table of used pairs implements both passes in polynomial time. $\square$

Property (ii) is the harmonious-coloring condition, with separate left and right palettes. Balanced harmonious colorings of bounded-degree graphs are classical; see [ALSZ17, Section 3.2]. The extra coordinatewise restriction costs only the last term in the greedy estimate.

For each coordinate t and symbol a , form the groups

$$V_{L,t,a} = \{u : \alpha(u)_t = a\}, \quad V_{R,t,a} = \{v : \beta(v)_t = a\}. \quad (2.1)$$

Exactly K/d strings have a prescribed symbol at position t . Thus every group has size at most

$$\frac{K}{d} \left\lceil \frac{2N}{K} \right\rceil \leq \frac{2N + K}{d} \leq \frac{3N}{d}. \quad (2.2)$$

One balanced coloring has therefore supplied n different pairs of small partitions.

2.2 The color pairs are Boolean cubes

Let

$$\Omega = \{(\alpha, \beta) \in [d]^n \times [d]^n : \alpha_i \neq \beta_i \text{ for all } i\}, \quad M = |\Omega| = (d(d-1))^n. \quad (2.3)$$

The coordinate- i matching interchanges the two symbols at position i . It has no fixed points. These coordinate matchings give a simple description of the entire intermediate set.

Lemma 2.2 (Full Boolean-cube components). *The coordinate matchings partition Ω into $\binom{d}{2}^n$ copies of $\{-1, 1\}^n$. On each component, interchanging all color pairs is the antipodal permutation.*

Proof. Fix the unordered pair $\{\alpha_i, \beta_i\}$ at every coordinate. The two symbols are distinct, so each pair has two orientations, chosen independently. Encode the orientation by $+1$ if the smaller symbol comes first and by -1 otherwise. This identifies the component with the Boolean cube, and a coordinate interchange flips the corresponding sign. Reversing every pair sends x to $-x$. $\square$

Write J for the permutation matrix on Ω defined by

$$J_{\omega, \eta} = \mathbf{1}\{\eta = \bar{\omega}\}, \quad \overline{(\alpha, \beta)} = (\beta, \alpha).$$

It is the direct sum of the antipodal matrices on these cubes. This is the fixed matrix that the Fourier construction will approximate.

2.3 The colored factorization

Every input edge $e = (u, v)$ has its distinct position

$$\omega_e = (\alpha(u), \beta(v)) \in \Omega.$$

Use ω_e at its left endpoint and the antipodal position $\bar{\omega}_e = (\beta(v), \alpha(u))$ at its right endpoint. In both positions, the first word is the color of that endpoint.

Define zero-one matrices

$$P \in \{0, 1\}^{N \times M}, \quad Q \in \{0, 1\}^{M \times N}, \quad P_{u, \omega_e} = Q_{\bar{\omega}_e, v} = 1 \quad (e = (u, v) \in E), \quad (2.4)$$

with all other entries zero.

Proposition 2.3 (Factorization and coordinate preservation). *The matrices in (2.4) satisfy*

$$A = PJQ. \quad (2.5)$$

Fix $t \in [n]$. Suppose C is a rational matrix on Ω that does not connect different cube components and preserves the t th Boolean coordinate within each component. Then PCQ is supported on the corresponding groups in (2.1), each of size at most $3N/d$.

Proof. A term in $(PJQ)_{uv}$ comes from an edge e incident to u and an edge f incident to v . It contributes exactly when $J_{\omega_e, \bar{\omega}_f} = 1$, equivalently when $\omega_e = \omega_f$. The color-pair map is injective on edges, so this happens exactly when $e = f = (u, v)$. Thus the product equals A .

A term in $(PCQ)_{uv}$ passes between positions of the form

$$(\alpha(u), \beta(v')) \quad \text{and} \quad (\beta(v), \alpha(u')).$$

If its matrix entry is nonzero, these positions have the same unordered pair of symbols at coordinate t and the same orientation there. Their first symbols must therefore agree: $\alpha(u)_t = \beta(v)_t$. Thus $(PCQ)_{uv} = 0$ between different groups. Their sizes were bounded in (2.2). $\square$

We have reduced the construction to a problem on Boolean cubes. We seek a degree- $O(\sqrt{n \log n})$ family $G(\theta)$ satisfying $G(0) = J$ and

$$G(t) = C_t + R_0 \quad (t \in [n]),$$

where C_t preserves coordinate t and R_0 has small rank. Proposition 2.3 will turn PC_tQ into small blocks, and $\text{rank}(PR_0Q) \leq \text{rank } R_0$. The permanent algorithm only needs these conclusions after multiplication by P, Q ; the cube construction will supply them uniformly on the whole intermediate space.

3 A Boolean-Fourier deformation of the antipodal matrix

The construction on one cube has two ingredients. A degree- $O(\sqrt{n \log n})$ function agrees with parity on almost all inputs. Varying its coefficients makes it independent of a different coordinate at each selected parameter value. Standard Fourier matrices convert these two properties into rank and coordinate-preservation statements.

3.1 Reminder: the standard XOR matrix

For $g : \{-1, 1\}^n \rightarrow \mathbb{Q}$, use the Fourier expansion

$$g(x) = \sum_{S \subseteq [n]} \hat{g}(S) x^S, \quad \hat{g}(S) = \mathbb{E}_{\mathbf{x}}[g(\mathbf{x}) x^S], \quad x^S = \prod_{i \in S} x_i.$$

The expectation is uniform. Identify a subset with the cube vertex whose negative coordinates form that subset. Define the $2^n \times 2^n$ matrix

$$\mathcal{F}_g[S, T] = \hat{g}(S \oplus T). \quad (3.1)$$

Lemma 3.1 (Standard Fourier-matrix facts). *Let $J[S, T] = \mathbf{1}\{T = [n] \setminus S\}$ be the antipodal matrix. Then*

$$\mathcal{F}_{\chi_{[n]}} = J, \quad \text{rank}(J - \mathcal{F}_g) = 2^n \mathbb{P}_{\mathbf{x}}[g(\mathbf{x}) \neq \mathbf{x}^{[n]}]. \quad (3.2)$$

Moreover, g is independent of x_t if and only if

$$\mathcal{F}_g[S, T] = 0 \quad \text{whenever } t \in S \oplus T. \quad (3.3)$$

Thus $\mathcal{F}_g$ preserves coordinate t whenever g is independent of that coordinate.

Proof. These are the usual XOR-matrix rank identity and Fourier-support criterion for coordinate independence; we recall the matrix form. Let H be the unnormalized Walsh–Hadamard matrix,

$$H_{S,T} = (-1)^{|S \cap T|}, \quad H^2 = 2^n I,$$

and let D_g be diagonal with entries $g(x)$ in the same subset indexing. Then

$$\mathcal{F}_g = \frac{1}{2^n} H D_g H. \quad (3.4)$$

The (S, T) entry is $\mathbb{E}_{\mathbf{x}}[g(\mathbf{x}) x^S x^T] = \widehat{g}(S \oplus T)$. Conjugation by $2^{-n/2} H$ shows that $\text{rank}(\mathcal{F}_g - \mathcal{F}_h)$ is the number of inputs where g and h differ. Parity has its only Fourier coefficient at $[n]$, giving (3.2).

Finally, g is independent of x_t exactly when $\widehat{g}(S) = 0$ for all $S \ni t$. Substitution into (3.1) gives (3.3), in both directions. $\square$

We use these as standard Boolean Fourier facts [OD21, Chapter 1]. The construction problem is now about functions: arrange agreement with parity at zero, and independence of x_t at $\theta = t$.

3.2 Agree with parity on typical sums

Fix the degree at the scale used by the algorithm. For sufficiently large n , put

$$b = \left\lceil 10\sqrt{n \log n} \right\rceil. \quad (3.5)$$

Thus $b = O(\sqrt{n \log n})$ and $b = o(n)$.

Lemma 3.2 (Parity agreement). *For sufficiently large n , there is an explicitly constructible polynomial $p \in \mathbb{Q}[u]$ of degree at most b such that*

$$\mathbb{P}_{\mathbf{x} \sim \{-1, 1\}^n} \left[p \left(\sum_{i=1}^n x_i \right) \neq \mathbf{x}^{[n]} \right] \leq n^{-10}. \quad (3.6)$$

Its coefficients have polynomial bit length in n .

Proof. If x has j negative coordinates, then $\sum_i x_i = n - 2j$ and $x^{[n]} = (-1)^j$. Choose $b + 1$ consecutive integer weights j centered near $n/2$, and interpolate

$$p(n - 2j) = (-1)^j$$

on those weights. For large n , the chosen interval lies in $[0, n]$ and contains every integer within $b/4$ of $n/2$. Since the number of negative coordinates of uniform $\mathbf{x}$ has distribution $\text{Bin}(n, 1/2)$, Hoeffding's bound gives

$$\mathbb{P} \left[p \left(\sum_i \mathbf{x}_i \right) \neq \mathbf{x}^{[n]} \right] \leq 2 \exp \left(-\frac{b^2}{8n} \right) \leq 2n^{-25/2} \leq n^{-10}.$$

Here $\log n \geq \ln n$. The interpolation nodes are distinct integers of magnitude at most n and the prescribed values are ± 1 , so rational interpolation constructs p in polynomial time with polynomial-bit coefficients. Appendix B records explicit bounds. $\square$

This is agreement in Hamming distance: no bound is imposed on the exceptional values of p . The relevant approximation principle appears in work of Bhatnagar–Gopalan–Lipton, Wigderson (as presented by Viola), Srinivasan, and Alman–Williams [BGL06, Vio09, Sri13, AW15]. The rational case used here is just the interpolation argument above.

3.3 Drop one coordinate at each point

The function at zero is now fixed. To remove coordinate t at parameter value t , give x_i the coefficient $1 - \theta/i$.

Proposition 3.3 (A polynomial family with coordinate independence). *With p from Lemma 3.2, define*

$$g_\theta(x) = p \left(\sum_{i=1}^n \left(1 - \frac{\theta}{i} \right) x_i \right). \quad (3.7)$$

For every fixed x , this is a polynomial of degree at most $b = O(\sqrt{n \log n})$ in θ . At zero, g_0 agrees with parity on all but an n^{-10} fraction of the cube. At $\theta = t$, the function g_t is independent of x_t .

Proof. The argument of p is affine in θ , proving the degree bound. At zero it is $\sum_i x_i$, so Lemma 3.2 gives the agreement. At $\theta = t$, the coefficient of x_t is zero, proving independence. $\square$

3.4 Deform the antipodal matrix

On one cube, Fourier transform the family in (3.7) and add back its error at zero:

$$G(\theta) = J - \mathcal{F}_{g_0} + \mathcal{F}_{g_\theta}. \quad (3.8)$$

For a disjoint union of cubes, use the same construction separately on each component. We continue to write J , $\mathcal{F}_{g_\theta}$, and $G(\theta)$ for these block-diagonal matrices on Ω .

Theorem 3.4 (Deformation of the antipodal matrix). *Let Ω be a disjoint union of Boolean n -cubes with specified cube coordinates, of total size M , with n sufficiently large. The matrix polynomial (3.8) satisfies*

$$G(0) = J, \quad \deg_\theta G_{\omega, \eta} \leq b = O(\sqrt{n \log n}).$$

At every $t \in [n]$,

$$G(t) = \mathcal{F}_{g_t} + R_0, \quad R_0 = J - \mathcal{F}_{g_0}, \quad \text{rank } R_0 \leq M/n^{10}. \quad (3.9)$$

The matrix $\mathcal{F}_{g_t}$ does not connect different cube components and preserves coordinate t within each component. All coefficients can be constructed in polynomial time in M, n .

Proof. At zero the two Fourier matrices cancel. Fourier transformation is linear, so it preserves the degree bound in θ . Lemma 3.1 and Proposition 3.3 give rank at most $2^n/n^{10}$ on each component and preservation of coordinate t . Ranks add over the components, giving (3.9).

For constructibility, evaluate the polynomial $g_\theta(x)$ at all 2^n Boolean inputs, keeping its coefficients in θ , and compute their Fourier transforms. This gives the matrix (3.1) on a cube. Repeat it on every component. Since $2^n \leq M$ and $b = O(n)$, even direct evaluation and direct Fourier summation take polynomially many operations in M, n . The rational bit bounds are proved in Appendix B. $\square$

The only rank estimate used here is the standard XOR-matrix identity. Independence of x_t supplies the required support through the equally standard Fourier-support criterion. There is no approximation of the final matrix entries or of the permanent.

Corollary 3.5 (The input deformation). *Let the sparse input be colored as in Lemma 2.1, with n sufficiently large, and let P, J, Q be as in Section 2. Define*

$$A(\theta) = P(J - \mathcal{F}_{g_0} + \mathcal{F}_{g_\theta})Q. \quad (3.10)$$

Then $A(0) = A$, its entries have degree at most b , and at every $t \in [n]$,

$$A(t) = D_t + R, \quad D_t = P\mathcal{F}_{g_t}Q, \quad R = P(J - \mathcal{F}_{g_0})Q, \quad (3.11)$$

where D_t has blocks of size at most $3N/d$ and rank $R \leq M/n^{10}$. All data are explicitly constructible in polynomial time in N, M, n .

Proof. Use $A = PJQ$ at zero and the degree bound of Theorem 3.4. Proposition 2.3 gives the blocks because $\mathcal{F}_{g_t}$ preserves coordinate t . Multiplication by P, Q cannot increase the correction rank. $\square$

The construction is complete. We next prove the two algorithmic facts promised in the introduction: how to evaluate a permanent with small blocks plus low rank, and how to recover derivatives from those evaluations.

4 Permanents of small blocks plus low rank

A matrix supported on small blocks can be handled block by block. A low-rank addition may connect every pair of blocks, but its edges have only a small number of types. Recording their type multiplicities is enough to count the ways of joining the blocks.

Partition the row indices into groups $I_1, \dots, I_m$ and the column indices into groups $J_1, \dots, J_m$. We say that D is supported on these corresponding blocks if

$$D_{uv} = 0 \quad (u \in I_a, v \in J_c, a \neq c).$$

The groups on the two sides of one block need not have equal sizes.

Theorem 4.1 (Permanent of small blocks plus low rank). *Given rational $N \times N$ matrices D, R and corresponding block partitions for D , suppose each group has size at most s and rank $R \leq r$ for an integer $r \geq 1$. Then $\text{per}(D + R)$ can be computed in*

$$2^{O(s)} N^{O(r)} \quad (4.1)$$

arithmetic operations. The same bound, with the original N, s, r , applies to every square submatrix. For polynomial-bit input entries, it also bounds the number of bit operations.

We first explain the counting identity, then its implementation. Compute a rank factorization

$$R = XY^\top, \quad X, Y \in \mathbb{Q}^{N \times r},$$

using the actual rank if it is smaller than r . The entry $R_{uv} = \sum_{a=1}^r X_{ua}Y_{va}$ is a sum of weights of r edge types. The weights may be signed.

4.1 Pair the unmatched endpoints by type

For an unmatched row u , choosing type a contributes X_{ua} ; for an unmatched column v , it contributes Y_{va} . If there are ν_a row endpoints and ξ_a column endpoints of type a , they can be paired only if $\nu_a = \xi_a$, in which case there are $\nu_a!$ bijections. We encode precisely this rule.

Lemma 4.2 (Factorial coefficient pairing). *Introduce $\mathbf{z} = (z_1, \dots, z_r)$ and $\mathbf{w} = (w_1, \dots, w_r)$, and let*

$$X_u(\mathbf{z}) = \sum_{a=1}^r X_{ua}z_a, \quad Y_v(\mathbf{w}) = \sum_{a=1}^r Y_{va}w_a.$$

Define the linear functional

$$\mathcal{L}(\mathbf{z}^\nu \mathbf{w}^\xi) = \begin{cases} \nu!, & \nu = \xi, \\ 0, & \nu \neq \xi, \end{cases} \quad \nu! = \prod_{a=1}^r \nu_a!. \quad (4.2)$$

For equally sized row and column sets S, T ,

$$\text{per } R[S, T] = \mathcal{L} \left(\prod_{u \in S} X_u(\mathbf{z}) \prod_{v \in T} Y_v(\mathbf{w}) \right). \quad (4.3)$$

Proof. Expanding the linear forms assigns a type to each endpoint, with the stated weight. Applying $\mathcal{L}$ counts all bijections between row and column endpoints of each type. Conversely, a term in the expansion of the permanent chooses a bijection and a type for each matched pair. These are the same weighted choices. $\square$

This is the factorial pairing in Barvinok's bounded-rank permanent algorithm [Bar96, Algorithm 3.2 and Theorem 3.3]. We now combine it with the matchings available inside the blocks.

4.2 Combine the block polynomials

We use the permanent-of-a-sum identity [Min78]:

$$\text{per}(B + E) = \sum_{\substack{S, T \subseteq [N] \\ |S|=|T|}} \text{per } E[S, T] \text{ per } B[S^c, T^c]. \quad (4.4)$$

The indices S, T are matched using entries from E , and the complementary indices are matched using entries from B .

Lemma 4.3 (Product of the block polynomials). *For each block $I_a \times J_a$, define*

$$F_a(\mathbf{z}, \mathbf{w}) = \sum_{\substack{S \subseteq I_a, T \subseteq J_a \\ |S|=|T|}} \text{per } D[S, T] \prod_{u \in I_a \setminus S} X_u(\mathbf{z}) \prod_{v \in J_a \setminus T} Y_v(\mathbf{w}). \quad (4.5)$$

Then, with $F = \prod_a F_a$,

$$\text{per}(D + R) = \mathcal{L}(F). \quad (4.6)$$

Proof. Apply (4.4) with $E = D$ and $B = R$. For chosen sets S, T , put $S_a = S \cap I_a$ and $T_a = T \cap J_a$. The permanent of $D[S, T]$ vanishes unless $|S_a| = |T_a|$ for every block. When these equalities hold, it factors as $\prod_a \text{per } D[S_a, T_a]$.

Apply Lemma 4.2 to the complementary permanent of R and bring the sums inside $\mathcal{L}$. Choices in different blocks are independent, so the resulting polynomial is $\prod_a F_a$. Rectangular blocks cause no difficulty: equality of the numbers of selected rows and columns is imposed separately in each block. $\square$

The polynomial F_a records a local partial matching: matched edges have weights D_{uv} , unmatched rows have weights X_u , and unmatched columns have weights Y_v . The product records all local choices; the pairing joins the remaining endpoints globally.

4.3 Compute the coefficient tables

Proof of Theorem 4.1. Compute the polynomials F_a explicitly and multiply them. If

$$F(\mathbf{z}, \mathbf{w}) = \sum_{\nu, \xi} c_{\nu, \xi} \mathbf{z}^\nu \mathbf{w}^\xi,$$

then the answer is

$$\text{per}(D + R) = \sum_{\nu} \nu! c_{\nu, \nu}. \quad (4.7)$$

Each row contributes at most one linear form in $\mathbf{z}$, and each column at most one in $\mathbf{w}$. Thus F , every partial product of its block polynomials, and every intermediate term used to form a block polynomial have degree at most N in each individual variable. A coefficient table has at most

$$C = (N + 1)^{2r}$$

entries.

For one block, enumerate the at most 2^{2s} pairs S, T in (4.5). Ryser's formula computes each scalar permanent in 2^s times polynomial work. Multiply the remaining linear forms into a table one at a time and add the scaled result to F_a . Multiplication by a linear form uses at most rC coefficient updates. There are at most $2N$ nonempty blocks, so forming all block polynomials costs $2^{O(s)}C$ times polynomial work.

Multiply their tables successively by ordinary coefficient convolution. Even naive multiplication uses at most C^2 times polynomial work. Since

$$C^2 = (N + 1)^{4r} = N^{O(r)},$$

this gives (4.1). At rank zero all tables are scalars. No fast polynomial-multiplication algorithm is needed.

Deleting rows and columns only shrinks the groups and restricts X, Y to the retained indices. The same calculation therefore handles every square submatrix without increasing the parameters. Appendix B shows polynomial bit cost per operation, completing the bit-operation claim. $\square$

The coefficient tables may require superpolynomial space. This is immaterial for the time bound. Appendix A evaluates the same pairing from one numerical evaluation at a time, giving polynomial space with the same form of running time.

5 Recover the permanent using derivatives

We can now evaluate every square submatrix at each selected point. This is exactly what is needed for derivatives: a low-order derivative only sees terms with a few perturbed entries. The remaining entries are evaluated in a large complementary submatrix at the original point.

Theorem 5.1 (Recovery from a polynomial deformation). *Let $A(\theta)$ be a rational polynomial $N \times N$ matrix, with $A(0) = A$ and entry degrees at most b . At each $t \in [n]$, suppose an explicit decomposition $A(t) = D_t + R_t$ is given, with block sizes at most s and $\text{rank } R_t \leq r$ for some integer $r \geq 1$. Put*

$$q = \left\lfloor \frac{Nb}{n} \right\rfloor, \quad (5.1)$$

and assume $1 \leq q \leq N/2$. If n, b and the bit lengths of the supplied coefficients are polynomially bounded in N , then $\text{per } A$ can be recovered in

$$2^{O(s+q \log(N/q))} N^{O(r)} \quad (5.2)$$

bit operations.

The correction and the partitions may depend on t . Our construction happens to use the same correction at every point. In either case, one rank factorization per point is enough: every submatrix call uses the restriction of that factorization.

5.1 How many derivatives are enough?

Lemma 5.2 (Hermite interpolation). *Let $t_1, \dots, t_m$ be distinct rational points and let $a_1, \dots, a_m$ be positive integers. A polynomial of degree less than $D = \sum_i a_i$ is determined by*

$$H(t_i), H'(t_i), \dots, H^{(a_i-1)}(t_i) \quad (i \in [m]).$$

Its coefficients can be recovered with polynomially many rational operations in D .

Proof. If two candidate polynomials have these derivatives, their difference has a zero of multiplicity at least a_i at each t_i . It is divisible by $\prod_i (\theta - t_i)^{a_i}$, whose degree is D , and hence is zero. The corresponding $D \times D$ linear system is therefore nonsingular and gives a polynomial-time reconstruction. This is standard Hermite interpolation; see, for example, [BGKM23, Lemma 4.10]. $\square$

For our matrix polynomial, set $H(\theta) = \text{per } A(\theta)$. Each permanent term uses N entries, so $\deg H \leq Nb$. Since $n(q+1) > Nb$, it suffices to compute the value and the first q derivatives of H at each $t \in [n]$. We must do this without assuming that the nearby matrices $A(\theta)$ remain structured.

5.2 An order- j derivative uses at most j perturbed entries

Lemma 5.3 (Derivatives at one structured point). *Fix t and suppose $A(t) = D_t + R_t$ has block sizes at most s and correction rank at most r . Define the polynomial matrix*

$$\Delta_t(\theta) = A(\theta) - A(t).$$

For $0 \leq j \leq q$,

$$H^{(j)}(t) = \sum_{k=0}^j \sum_{\substack{S, T \subseteq [N] \\ |S|=|T|=k}} \frac{d^j}{d\theta^j} \text{per } \Delta_t(\theta)[S, T] \Big|_{\theta=t} \text{per } A(t)[S^c, T^c]. \quad (5.3)$$

All these derivatives can be computed together in

$$2^{O(s+q \log(N/q))} N^{O(r)}$$

arithmetic operations, for polynomial-size coefficient data.

Proof. Apply the permanent-of-a-sum identity (4.4) to $A(\theta) = A(t) + \Delta_t(\theta)$. Every entry of Δ_t vanishes at t . A permanent of order k in these entries therefore has a zero of multiplicity at least k at t : every product in it contains k factors vanishing there. Differentiating j times leaves only $k \leq j$, proving (5.3).

For fixed S, T of size $k \leq q$, compute the large complementary permanent by Theorem 4.1, at cost

$$\mathcal{T} = 2^{O(s)} N^{O(r)}.$$

It is evaluated at t , where the structure is available. Compute the small permanent's derivatives through order q by Ryser's formula, carrying derivative lists through its additions and multiplications. For scalar functions u, v , the multiplication rule is simply

$$(uv)^{(j)}(t) = \sum_{a=0}^j \binom{j}{a} u^{(a)}(t) v^{(j-a)}(t). \quad (5.4)$$

Thus each multiplication of two lists costs $O(q^2)$ scalar operations. The lists for the polynomial entries of Δ_t are obtained by direct differentiation. Ryser's computation on a $k \times k$ matrix uses 2^k times polynomial work, now with polynomial overhead for the derivative lists. It computes all required orders at once. At $k = 0$, the small permanent is the constant one.

There are $\binom{N}{k}^2$ pairs of index sets. Up to polynomial factors, the total work is at most

$$2\mathcal{T} \sum_{k=0}^q 2^k \binom{N}{k}^2.$$

For $1 \leq q \leq N/2$,

$$\sum_{k=0}^q 2^k \binom{N}{k}^2 \leq (q+1) 2^q \left(\frac{eN}{q} \right)^{2q} = 2^{O(q \log(N/q))}. \quad (5.5)$$

Accumulate the derivative lists using (5.3). These bounds prove the lemma. $\square$

The perturbation need not have any block structure or low rank. Its permanents are inexpensive because their order is at most q . Every large permanent remains a submatrix of the structured matrix $A(t)$.

Proof of Theorem 5.1. At every $t \in [n]$, apply Lemma 5.3 with the value of q in (5.1). Lemma 5.2 reconstructs H from these values and derivatives, and hence gives $H(0) = \text{per } A$. The number of points and the interpolation system size are polynomial in N . Since $q \log(N/q) \geq \log N$ in the stated range, this polynomial work and the extra $\log N$ in the evaluation cost are absorbed in (5.2). Appendix B proves polynomial bit overhead for all the rational operations. $\square$

6 Parameters and conclusion

All the requirements are met by a routine choice of scales.

Lemma 6.1 (Parameter choice). *For sufficiently large N , take*

$$d = 16\lceil \log N \rceil, \quad n = \min\{k \geq 1 : d^{2k} \geq 256N\}, \quad b = \lceil 10\sqrt{n \log n} \rceil. \quad (6.1)$$

Let $K = d^n$, $M = (d(d-1))^n$, and $q = \lfloor Nb/n \rfloor$. The coloring and deformation constructions apply, and the permanent algorithm runs in

$$2^{O(N(\log \log N)^2 / \sqrt{\log N})}$$

arithmetic operations.

Proof. We have $d = \Theta(\log N)$ and, by minimality of n ,

$$16\sqrt{N} \leq K < 16d\sqrt{N} \leq N.$$

Thus $n = \Theta(\log N / \log \log N)$, $d \geq 12n$, and $M \leq K^2 = O(N \log^2 N)$. In particular, the coloring lemma applies. Also $b = \Theta(\sqrt{\log N}) = o(n)$, as required by the parity construction, and

$$q = \Theta\left(\frac{N \log \log N}{\sqrt{\log N}}\right) \quad \text{lies in } [1, N/2].$$

The exponent in the recovery bound is

$$O\left(\frac{N}{d} + \frac{M \log N}{n^{10}} + q \log \frac{N}{q}\right) = O\left(\frac{N(\log \log N)^2}{\sqrt{\log N}}\right).$$

Indeed, the first term is $O(N/\log N)$, the second is $O(N(\log \log N)^{10}/(\log N)^7)$, and the third gives the displayed bound. The intermediate matrices and polynomial data have polynomial size. $\square$

Proof of Theorem 1.1. Choose the parameters of Lemma 6.1. Color the graph and form P, J, Q using Section 2. Construct p and g_θ , form their standard Fourier matrices on the cube components, and set

$$A(\theta) = P(J - \mathcal{F}_{g_\theta} + \mathcal{F}_{g_\theta})Q.$$

Corollary 3.5 supplies the block and rank bounds. Apply Theorem 5.1, using Theorem 4.1 for its complementary permanents. Hermite interpolation returns $H(0) = \text{per } A$ exactly. Lemma 6.1 bounds the running time by (1.1); the bit estimates in Appendix B apply to every step. Direct enumeration handles the finitely many smaller input sizes. Every step is deterministic. $\square$

The degree- $O(\sqrt{n \log n})$ approximation is used only to make $J - \mathcal{F}_{g_0}$ have small rank. That entire correction is retained and evaluated exactly. The graph, the Fourier transforms, the permanent calculations, and the final interpolation introduce no error in the answer.

A A polynomial-space implementation

The main evaluation algorithm explicitly stores $F = \prod_a F_a$, whose coefficient table can be large. We can instead compute the same functional $\mathcal{L}(F)$ as a weighted sum of numerical evaluations. This changes only the implementation of Theorem 4.1, not the deformation or the derivative argument.

A.1 Evaluate the block polynomials one point at a time

Fix rational values for $\mathbf{z}, \mathbf{w}$. Evaluate each F_a from (4.5) by enumerating S, T , using Ryser for $\text{per } D[S, T]$, and multiplying by the numerical values of the endpoint forms. There are at most 2^{2s} pairs and each permanent has order at most s . One evaluation of F , obtained by multiplying the block evaluations, costs $2^{O(s)}$ times polynomial work. The subsets and the Ryser terms can be processed sequentially in polynomial space.

Every variable of F has degree at most N . Hence its values on $\{0, \dots, N\}^{2r}$ determine it. We only need one linear functional of F , so we apply that functional to the interpolation formula rather than storing its coefficients.

A.2 The numerical pairing formula

Let

$$\Lambda_a(u) = \prod_{\substack{0 \leq c \leq N \\ c \neq a}} \frac{u - c}{a - c} \quad (a \in \{0, \dots, N\})$$

be the univariate Lagrange basis. Precompute the $(N+1)^2$ rational numbers

$$\kappa_{ab} = \sum_{k=0}^N k! [u^k] \Lambda_a(u) [u^k] \Lambda_b(u). \quad (\text{A.1})$$

The tensor-product interpolation formula for F and the definition of $\mathcal{L}$ give

$$\mathcal{L}(F) = \sum_{\mathbf{a}, \mathbf{b} \in \{0, \dots, N\}^r} \left(\prod_{c=1}^r \kappa_{a_c b_c} \right) F(\mathbf{a}, \mathbf{b}). \quad (\text{A.2})$$

Indeed, the pairing factors over types, and the value on $\Lambda_a(z_c) \Lambda_b(w_c)$ is κ_{ab} .

There are $(N+1)^{2r}$ grid points. At each point, compute its weight and its evaluation, add their product to one accumulator, and discard them. This takes

$$2^{O(s)} N^{O(r)}$$

arithmetic operations and polynomial space. At rank zero the sum has one term. Appendix B bounds the bit lengths of the weights, evaluations, and partial sums.

A.3 Space for the whole algorithm

Use this routine for each complementary permanent in (5.3). Enumerate the points t and sets S, T sequentially. Ryser's algorithm for a small perturbation permanent keeps only derivative lists of length $q + 1$, with multiplication by (5.4). There are only $n(q + 1)$ derivatives to retain for the final interpolation, and its linear system has polynomial size. The matrices defining the deformation and their rank factorizations also have polynomial size. Thus the full algorithm has a deterministic polynomial-space implementation with the time bound in Theorem 1.1.

B Exact arithmetic and bit complexity

We verify the bit-operation claims for both implementations. The point is not merely that the input coefficients are small: each intermediate sum must remain small before any cancellation. Here every expanded term is a product of only polynomially many bounded rational numbers, and the number of terms is at most exponential in a polynomial. This gives polynomial bit length.

Throughout, n, b, M are polynomially bounded in N . We also cover the general recovery theorem, whose supplied polynomial coefficients and structured decompositions have polynomial bit length.

B.1 Constructing the Fourier deformation

Write $\|p\|_1$ for the sum of absolute coefficients. The interpolation nodes for p are $b + 1$ consecutive attainable sums, spaced by two, of magnitude at most n . The denominator of the k th Lagrange basis polynomial has magnitude $2^b k!(b - k)!$, which divides $2^{bb!}$. Since the interpolation values are ± 1 ,

$$2^b b! p \in \mathbb{Z}[u], \quad \|p\|_1 \leq (b + 1)(n + 1)^b. \quad (\text{B.1})$$

These bounds hold for the intermediate Lagrange products and sums as well.

For a fixed Boolean input x , the affine argument

$$a_x(\theta) = \sum_{i=1}^n (1 - \theta/i) x_i$$

has coefficient norm at most $2n$ and denominator dividing $n!$. Hence every coefficient of $g_\theta(x) = p(a_x(\theta))$ has absolute value at most $\|p\|_1 (2n)^b$ and denominator dividing $2^b b! (n!)^b$. Fourier transformation takes signed averages over 2^n inputs. Thus every coefficient of $\mathcal{F}_{g_\theta}$ has denominator dividing

$$\Delta_0 = 2^{n+bb!} (n!)^b, \quad \log \Delta_0 = O(n + bn \log n), \quad (\text{B.2})$$

and absolute value at most $\|p\|_1 (2n)^b$. Before division by 2^n , a partial Fourier sum is bounded by 2^n times this quantity. All these numbers have polynomial bit length.

Copying the matrices to the cube components changes no entries. Adding J and multiplying by P, Q introduces only polynomially many summands, so the coefficients of $G(\theta)$, $A(\theta)$, and the correction R also have polynomial bit length. In fact, each entry after multiplication by P, Q sums at most nine auxiliary entries.

Differentiating a degree- b polynomial and evaluating at $t \in [n]$ multiplies its coefficient bound by at most $b!(n + 1)^b$ and introduces no denominator. This controls all entry derivatives of A and Δ_t used by Lemma 5.3.

The parameters themselves require no real-number oracle. For example, the integer in (3.5) is the least b such that $2^{b^2} \geq n^{100n}$, an integer comparison with polynomially many bits in n . The parameters in (6.1) are likewise computable by integer comparisons and integer powers.

B.2 Rational linear algebra and numerical pairing weights

Clear the denominators of the polynomially many entries in an input to Gaussian elimination. The resulting common denominator still has polynomial bit length. If an integer matrix has order at most m and entries bounded by 2^L , every minor has magnitude at most $m!2^{mL}$, and hence bit length $O(m(L + \log m))$.

Elimination can be implemented with polynomial-bit entries: with a nonsingular pivot block C , each remaining Schur-complement entry is a bordered minor divided by $\det C$, by

$$\det \begin{pmatrix} C & u \\ w^\top & a \end{pmatrix} = \det C (a - w^\top C^{-1} u).$$

Reducing fractions after each operation, or using fraction-free elimination, therefore keeps all pivots and intermediate entries polynomial in bit length. The same minor bounds control solutions and back substitution.

For a concrete rank factorization, choose row and column sets S, T of size $r = \text{rank } R$ such that $C = R[S, T]$ is nonsingular, and set

$$X = R[:, T], \quad Y^\top = C^{-1} R[S, :].$$

The chosen columns span the column space, so $XY^\top = R$. The construction is polynomial time with polynomial-bit coefficients. Rank zero requires no solve. Submatrix evaluations only restrict X, Y and need no further factorizations.

The Lagrange basis in Appendix A satisfies

$$N! \Lambda_a \in \mathbb{Z}[u], \quad \|\Lambda_a\|_1 \leq (N+1)^N.$$

It follows that

$$(N!)^2 \kappa_{ab} \in \mathbb{Z}, \quad |\kappa_{ab}| \leq (N+1)N!(N+1)^{2N}. \quad (\text{B.3})$$

Thus all pairing weights can be precomputed in polynomial time and have $O(N \log N)$ bits.

B.3 Bound the summations before cancellation

Collect the rational scalar data used by the evaluation and differentiation stages: block entries, rank-factorization entries, derivatives of the perturbation entries at each selected point, and, for the space-efficient version, the κ_{ab} . This is a polynomial-size list of polynomial-bit rationals. Choose a common denominator Δ and a bound $L = N^{O(1)}$ such that

$$\log \Delta \leq L, \quad |c| \leq 2^L$$

for every entry c in this list.

In a block with a rows and c columns, a term using k internal matching edges contains k block entries, $a - k$ row-form coefficients, and $c - k$ column-form coefficients. That is $a + c - k$ scalar factors. Across disjoint blocks there are at most $2N$ such factors. This bounds every coefficient-table entry and every partial product. The final factorial weight is at most $N!$.

For a term in (5.3), the complementary permanent has order $N - k$, and the differentiated perturbation permanent has order k . Their expanded products contain at most $2(N - k) + k \leq 2N$ listed rational factors. Derivatives only introduce integer multinomial weights. Since their order is at most $q \leq N/2$, these weights are bounded by $N^{O(N)}$. For the streamed calculation, grid nodes add integer factors bounded by N , while the interpolation weight adds at most $r \leq N$ listed factors κ_{ab} . Thus every contribution in either implementation uses at most $3N$ listed rational factors.

There are at most $N^{O(N)}$ expanded contributions. Subset choices, type assignments, expansions of Ryser row sums, and the interpolation grid all have this bound. The distribution of derivative orders among at most N factors does too. Coefficient convolutions have the same bound because degrees are polynomially bounded. Integer factorial and multinomial weights and grid-node powers multiply the magnitude by at most another $N^{O(N)}$. All estimates also hold for partial sums and partial products by taking absolute values before cancellation.

Consequently every number in these summation stages, in reduced form, has denominator dividing Δ^{3N} and absolute value at most

$$2^{3NL} N^{O(N)}. \quad (\text{B.4})$$

Its numerator and denominator therefore have $O(NL + N \log N)$ bits. Temporary addition and multiplication at most double polynomial bit lengths; reducing fractions restores the bound. This justifies polynomial bit cost even though the number of stored coefficients or summands may be exponential.

B.4 Exact Hermite interpolation from ordinary derivatives

Let $m = n(q + 1)$. Recover $H(\theta)$ in the space of polynomials of degree less than m . The Hermite system has rows (t, j) , with $t \in [n]$ and $0 \leq j \leq q$, and columns $0 \leq k < m$. Its entries are

$$V_{(t,j),k} = \begin{cases} \frac{k!}{(k-j)!} t^{k-j}, & k \geq j, \\ 0, & k < j. \end{cases} \quad (\text{B.5})$$

These are integer derivatives of monomials. Their absolute values are bounded by $(mn)^m$, so their bit lengths are polynomial in N . The system is nonsingular by Lemma 5.2.

Multiply the derivative-data vector by Δ^{3N} . It becomes an integer vector of polynomial bit length by (B.4). The minor bounds above give polynomial-bit exact elimination and back substitution for the coefficients of $\Delta^{3N}H$. Divide the recovered constant coefficient by Δ^{3N} to obtain $H(0)$.

This completes the bit-complexity analysis. It also bounds the accumulator sizes used in Appendix A, proving its polynomial-space assertion.

References

- [ALSZ17] Akanksha Agrawal, Daniel Lokshantov, Saket Saurabh, and Meirav Zehavi. Split Contraction: The Untold Story. In *STACS, LIPIcs* 66, 5:1–5:14, 2017. Section 3.2 and references therein on balanced harmonious coloring. <https://doi.org/10.4230/LIPIcs.STACS.2017.5>.
- [AW15] Josh Alman and Ryan Williams. Probabilistic Polynomials and Hamming Nearest Neighbors. In *FOCS*, 2015. Lemma 3.1. <https://arxiv.org/abs/1507.05106>.
- [Bar96] Alexander I. Barvinok. Two Algorithmic Results for the Traveling Salesman Problem. *Mathematics of Operations Research* 21(1):65–84, 1996. Section 3, especially Algorithm 3.2 and Theorem 3.3. <https://doi.org/10.1287/moor.21.1.65>.

- [BGL06] Nayantara Bhatnagar, Parikshit Gopalan, and Richard J. Lipton. Symmetric Polynomials over $\mathbb{Z}_m$ and Simultaneous Communication Protocols. *Journal of Computer and System Sciences* 72(2):252–285, 2006. <https://doi.org/10.1016/j.jcss.2005.06.007>.
- [BGKM23] Vishwas Bhargava, Sumanta Ghosh, Mrinal Kumar, and Chandra Kanta Mohapatra. Fast, Algebraic Multivariate Multipoint Evaluation in Small Characteristic and Applications. *Journal of the ACM* 70(6), Article 42, 2023. Hermite interpolation is Lemma 4.10 in arXiv:2111.07572v3. <https://doi.org/10.1145/3625226>.
- [Cur18] Radu Curticapean. Block Interpolation: A Framework for Tight Exponential-Time Counting Complexity. *Information and Computation* 261:265–280, 2018. Corollary 4.1(1) in the full version, arXiv:1511.02910v2. <https://arxiv.org/abs/1511.02910>.
- [Min78] Henryk Minc. *Permanents*. Encyclopedia of Mathematics and its Applications, volume 6. Addison-Wesley, 1978; reprinted by Cambridge University Press, 1984.
- [OAI26] OpenAI. *The Exponential Time Hypothesis Is False*. Supplied manuscript dated October 17, 2026; revised October 25, 2026.
- [OD21] Ryan O’Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014; corrected arXiv edition, May 2021. <https://arxiv.org/abs/2105.10386>.
- [Rys63] Herbert John Ryser. *Combinatorial Mathematics*. Carus Mathematical Monographs, volume 14. Mathematical Association of America, 1963.
- [Sri13] Srikanth Srinivasan. On Improved Degree Lower Bounds for Polynomial Approximation. In *FSTTCS*, LIPIcs 24, pages 201–212, 2013. Lemma 14 and the preceding attribution. <https://doi.org/10.4230/LIPIcs.FSTTCS.2013.201>.
- [Vio09] Emanuele Viola. On the Power of Small-Depth Computation. *Foundations and Trends in Theoretical Computer Science* 5(1):1–72, 2009. Section 1.2.3, including an argument attributed to Avi Wigderson. <https://doi.org/10.1561/04000000033>.