Yet another exposition of

"ETH is false"

Ramprasad Saptharishi*

(supervisor)

Abstract

This is an exposition of the recent "ETH is false" result announced by both OpenAI [Ope26] and Anthropic [Cla26]. This exposition is almost entirely based on Ryan O'Donnell's fantastic video lectures [O'D26a, O'D26b] on this proof, and on his lecture notes [O'D26c].

The exposition in this write-up is aimed at a graduate student in complexity and I will mostly focus on what I believe are the most important aspects of the proof. I will be assuming lots of facts that I believe are standard, but add their proofs in the appendix.

AI methodology: This write-up was written two days prior to a talk I was to give at ITU Copenhagen on Ryan O'Donnell's exposition of the proof [O'D26a, O'D26b, O'D26c]. I decided to take this up as an experiment (my first such attempt) to let Claude write the whole exposition with just supervision from me after I explained my understanding of the proof.

The write-up was almost entirely written by Claude Sonnet 5.5 with supervision. I started a Claude Code session with a minimal LaTeX file and presented Ryan's lecture notes [O'D26c] (and also [Ope26, Cla26] but Claude only read the title to add the bibtex entry and relied only on Ryan's notes). I then proceeded to explain the entire proof as I understand to Claude, and the way I wanted the proof to be written (over multiple messages to break up the proof into blocks). Claude then proceeded to suggest a skeleton and outline of this write-up. The bare theorem / lemma statements were added, and I made minor adjustments and restructuring. Claude then proceeded to build sections one at a time. Although I provided the structure of the proof, notations I wanted to use (and avoid), the proofs were written by Claude (with some course-corrections whenever the proof written by Claude seemed more involved / detailed than necessary, or when I had a simpler argument in mind). In several other occasions, my proof sketch was sloppy and Claude corrected those. This write-up is the output of this effort.

The entire proof was written over a single session of about 3 hours.

*Tata Institute of Fundamental Research, Mumbai. Email ramprasad@tifr.res.in

1 Introduction

Theorem 1.1 (Main theorem: sparse permanents)**.** *Let A be an $n \times n$ integer matrix with at most $\Delta = O(1)$ nonzero entries in every row and column (and entries of bit-length $\mathrm{poly}(n)$). Then $\mathrm{Perm}(A)$ can be computed in time $2^{o(n)}$.*

Corollary 1.2 (ETH is false; see [Val79, BDH93, DHM+14, CIP06])**.** *There is an algorithm that solves #3SAT on n variables in time $2^{o(n)}$. In particular, the Exponential Time Hypothesis is false.*

This write-up focuses on Theorem 1.1. Corollary 1.2 follows from it via standard facts (the sparsification lemma and Valiant's reduction from #3SAT to the permanent), which we do not cover here.

1.1 Overview of the proof

Let A be an $n \times n$ matrix with $O(1)$ nonzero entries in every row and column. We want $\mathrm{Perm}(A)$ in time $2^{o(n)}$. Throughout, think of A as the biadjacency matrix of a bipartite graph of bounded degree.

Two easy regimes, and their combination. There are two situations in which the permanent is known to be easy (Section 2). If a matrix is *B-blocky*, that is, block diagonal with blocks of size at most B, then its permanent is a product of permanents of small blocks and takes time $2^{O(B)}\,\mathrm{poly}(n)$ by Ryser's formula. If a matrix has rank at most r then its permanent takes time $n^{O(r)}$. Moreover the two can be combined: if P is B-blocky and $\mathrm{rank}(E) \le r$ then $\mathrm{Perm}(P+E)$ takes time $2^{O(B)} \cdot n^{O(r)}$. This is subexponential as soon as $B = o(n)$ and $r = o(n/\log n)$.

The matrix A itself has no reason to be of this form. The plan is to *deform* A into one that is.

Step I: Hermite interpolation. We will construct (in $\mathrm{poly}(n)$ time) a matrix $A(t)$ whose entries are polynomials in t of degree at most d, such that

(1) $A(0) = A$, and

(2) for each $i = 1, \dots, \ell$, the matrix $A(i)$ is blocky plus low rank, $A(i) = P_i + E_i$.

Then $\mathrm{Perm}(A(t))$ is a univariate polynomial of degree at most nd, and we want its value at $t = 0$. A polynomial of degree less than ℓk is determined by its values and derivatives of order up to $k-1$ at ℓ distinct points. So if $k\ell > nd$ it suffices to compute $\mathrm{Perm}(A(i))$ and its derivatives of order less than k for $i = 1, \dots, \ell$, and then interpolate. By expanding along k rows and columns, a derivative of order k of $\mathrm{Perm}(A(t))$ at $t = i$ reduces to computing $(n/k)^{O(k)}$ permanents of submatrices of $A(i)$ (plus some small permanents of size less than k), and submatrices of a blocky-plus-low-rank matrix are again blocky plus low rank.

The parameters we will end up with are

$$d \approx \sqrt{\ell} \cdot \operatorname{polylog} \ell, \qquad \ell \approx \frac{\log n}{\operatorname{polylog} \log n}, \qquad k \approx \frac{nd}{\ell} \approx \frac{n}{\sqrt{\log n}} \cdot \operatorname{polylog} \log n.$$

With these, $(n/k)^{O(k)} = 2^{o(n)}$, so the derivatives are within budget. Section 3 carries out this step.

The tension. What makes this non-trivial is that the number ℓ of "easy" points is much larger than the degree d. A polynomial of degree d in t is determined by $d+1$ of its values, so $A(t)$ is completely determined by its values at $d+1$ of the points $1, \ldots, \ell$. We are asking that it nevertheless looks blocky plus low rank at each of $\ell \gg d$ points. This is the heart of the proof, and the rest of the exposition constructs such an $A(t)$.

Step II: Labelling the rows and columns. Choose $q = \operatorname{polylog} n$ and ℓ so that $q^\ell = \Theta(\sqrt{n})$. We label every row and column by a string in $[q]^\ell$, such that (i) all labels are used about equally often, (ii) distinct edges (r, c) get distinct label pairs $(\mathrm{lab}(r), \mathrm{lab}(c))$, and (iii) for every edge (r, c), the labels of r and c differ in *every* coordinate. A simple greedy algorithm finds such a labelling (Section 4).

Step III: Factoring A through label pairs. Insert a middle layer M consisting of the label pairs (σ, π) with $\sigma_i \neq \pi_i$ for all i. Connect each row r to the pairs $(\mathrm{lab}(r), \mathrm{lab}(c))$ for the neighbours c of r (with weight A_{rc}), and each column c to the corresponding pairs; this factors $A = RC$. We then add a twist: use two copies of M, and connect the first copy to the second by the matrix S that swaps $(\sigma, \pi) \mapsto (\pi, \sigma)$. Nothing has changed, and still $A = RSC'$. The point is that S is a product $S = X_1 X_2 \cdots X_\ell$ of commuting matrices, where X_i swaps only the i-th coordinate of σ and π (Section 5).

Step IV: The matrix polynomial. Replace each X_i by $Y_i(t) = (1 - t/i)X_i$, which equals X_i at $t = 0$ and *vanishes* at $t = i$. For a polynomial P, the matrix $R\,P(Y_1, \ldots, Y_\ell)\,C'$ at $t = i$ does not involve X_i, so it never touches the i-th coordinate. Every surviving edge (r, c) then has $\mathrm{lab}(r)_i = \mathrm{lab}(c)_i$, which splits the rows and columns into q groups of size about n/q: the matrix is blocky. Of course, $P(X_1, \ldots, X_\ell)$ will not equal $X_1 \cdots X_\ell$, so we correct for this with a t-independent term:

$$A(t) \;=\; R\Big(P(Y_1(t), \ldots, Y_\ell(t)) + \big(S - P(X_1, \ldots, X_\ell)\big)\Big)C'.$$

Now $A(0) = A$ and $A(i)$ is blocky plus $R\,(S - P(X))\,C'$, so it remains to choose P so that $S - P(X)$ has low rank.

Step V: Why the correction has low rank. The X_i commute and are involutions without fixed points (by property (iii) of the labelling), so they are simultaneously diagonalisable: in a common eigenbasis (Fourier basis on each orbit $\{0,1\}^\ell$), each X_i acts as a diagonal ± 1 matrix y_i. In this basis $S = X_1 \cdots X_\ell$ is the *parity* $\prod_i y_i$, and $P(X)$ is $P(y)$. So we need a low-degree polynomial P that agrees with parity on almost all of the cube $\{\pm 1\}^\ell$; every disagreement contributes to the rank of the correction. Parity cannot be computed exactly by a low-degree polynomial, but on the middle slices it can: a univariate polynomial p of degree about $\sqrt{\ell}$ can be interpolated to match parity at all sums $y_1 + \cdots + y_\ell \in [-d, d]$, and these slices contain all but a $1/\operatorname{poly}(\ell)$ fraction of the cube. Taking $P = p(y_1 + \cdots + y_\ell)$ makes the correction have rank $n/\operatorname{poly}(\ell)$, which is small enough, and also explains why the degree is $d \approx \sqrt{\ell}$ (Section 6).

Finally, Section 7 checks that all of the constraints can be met simultaneously with $B = O(n/q) = o(n)$ and $r = o(n/\log n)$, so that the interpolation algorithm of Step I runs in time $2^{o(n)}$.

1.2 Roadmap

The rest of the write-up is organised as follows.

- Section 2 recalls the standard facts about permanents that we use as black boxes: Ryser's formula, blocky matrices, low rank matrices, and the combination of blocky plus low rank. Their proofs are in Appendix A.
- Section 3 states the target, a matrix polynomial $A(t)$ that is easy at many points, and shows how Hermite interpolation turns it into a fast algorithm for $\operatorname{Perm}(A)$.
- Sections 4 to 6 construct $A(t)$. We first label the rows and columns (Section 4), then factor $A = R\,S\,C'$ through a middle layer of label pairs and break the swap S into commuting pieces $X_1, \ldots, X_\ell$ (Section 5), and finally define $A(t)$ by deforming each X_i so that it disappears at $t = i$, and choose the deformation to approximate parity (Section 6).
- Section 7 fixes the parameters and proves Theorem 1.1.

2 Standard facts about the permanent

Fact 2.1 (Ryser [Rys63])**.** *The permanent of an $n \times n$ matrix can be computed in time $2^n \cdot \operatorname{poly}(n)$.*

Since we will have to take square submatrices of our matrices, blocks need not be square.

Definition 2.2 (Blocky matrices)**.** *An $m \times m'$ matrix P is* B-blocky *if there are partitions of the rows $[m] = R_1 \sqcup \cdots \sqcup R_s$ and of the columns $[m'] = C_1 \sqcup \cdots \sqcup C_s$ (parts may be empty) with $|R_j|, |C_j| \le B$ for all j, such that $P[a,b] = 0$ unless $a \in R_j$ and $b \in C_j$ for some j. The blocks $P[R_j, C_j]$ may be rectangular.* $\diamondsuit$

Deleting rows and columns of a B-blocky matrix gives a B-blocky matrix. If P is square and B-blocky, then $\mathrm{Perm}(P) = \prod_j \mathrm{Perm}(P[R_j, C_j])$ when $|R_j| = |C_j|$ for all j, and $\mathrm{Perm}(P) = 0$ otherwise. This vanishing does *not* persist for $P + E$ with E of low rank, which is why rectangular blocks must be allowed in Fact 2.5.

Fact 2.3 (Blocky matrices)**.** *If P is a square $n \times n$ B-blocky matrix then* $\mathrm{Perm}(P)$ *can be computed in time* $2^{O(B)} \cdot \mathrm{poly}(n)$.

Fact 2.4 (Low-rank matrices; Barvinok [Bar96])**.** *If* $\mathrm{rank}(E) \le r$ *then* $\mathrm{Perm}(E)$ *can be computed in time* $n^{O(r)}$.

Fact 2.5 (Blocky plus low rank)**.** *Let P be a square B-blocky matrix of dimension at most n (with possibly rectangular blocks, Definition 2.2) and let E be a matrix of the same dimension with* $\mathrm{rank}(E) \le r$. *Then* $\mathrm{Perm}(P + E)$ *can be computed in time* $2^{O(B)} \cdot n^{O(r)} \cdot \mathrm{poly}(n)$. *In particular this applies to every square submatrix of an $n \times n$ matrix of the form $P + E$, since deleting rows and columns preserves both properties.*

Proofs of these facts are deferred to Appendix A.

3 The interpolation framework

In this section we show how a certain kind of matrix polynomial $A(t)$ leads to a fast algorithm for $\mathrm{Perm}(A)$. Constructing such an $A(t)$ is the subject of the remaining sections.

The target. Given a sparse matrix A, we would like to construct an $n \times n$ matrix $A(t)$ whose entries are polynomials in t, with the following properties, for suitable parameters d, ℓ, B, r.

(T1) Every entry of $A(t)$ has degree at most d in t.

(T2) $A(0) = A$.

(T3) For each $i = 1, \ldots, \ell$, we have $A(i) = P_i + E_i$, where P_i is B-blocky and $\mathrm{rank}(E_i) \le r$.

Such a matrix is "easy" at ℓ points although its entries have degree only d, which is much smaller than ℓ.

Theorem 3.1 (The target)**.** *Let A be an $n \times n$ matrix with at most $\Delta = O(1)$ nonzero entries in each row and column. Then one can compute in* $\mathrm{poly}(n)$ *time a matrix $A(t)$ satisfying (T1)–(T3) with*

$$d = \widetilde{O}\big(\sqrt{\log n}\big), \qquad \ell = \widetilde{\Theta}(\log n), \qquad B = O\Big(\frac{n}{\log n}\Big), \qquad r = \widetilde{O}\Big(\frac{n}{\log^2 n}\Big).$$

Here $\widetilde{O}$ and $\widetilde{\Theta}$ hide factors of $(\log \log n)^{O(1)}$.

The rest of the write-up proves Theorem 3.1. For now, we assume it and see why it implies Theorem 1.1.

From the target to the permanent. Since every entry of $A(t)$ has degree at most d, the polynomial $\mathrm{Perm}(A(t))$ has degree at most nd, and we want its value at $t = 0$. We can evaluate $\mathrm{Perm}(A(i))$ for each $i \in [\ell]$ efficiently by Fact 2.5, but since $\ell \ll nd$ this is far too few values to determine the polynomial. The way out is to also ask for derivatives, which is where Hermite interpolation comes in.

Lemma 3.2 (Hermite interpolation)**.** *Let $z_1, \dots, z_\ell \in \mathbb{Q}$ be distinct points and $k \geq 1$. A polynomial $f \in \mathbb{Q}[t]$ of degree less than ℓk is uniquely determined by the numbers $f^{(j)}(z_i)$ for $i \in [\ell]$ and $0 \leq j < k$. Moreover, for every $z \in \mathbb{Q}$ there are coefficients $c_{i,j}$, depending only on z, the points z_i and k (and not on f), such that*

$$f(z) = \sum_{i \in [\ell]} \sum_{0 \leq j < k} c_{i,j} \, f^{(j)}(z_i),$$

and these coefficients can be computed in $\mathrm{poly}(\ell k)$ arithmetic operations.

Proof. Consider the linear map sending f (of degree less than ℓk) to the vector $\left(f^{(j)}(z_i)\right)_{i,j} \in \mathbb{Q}^{\ell k}$. If f is in the kernel then each z_i is a root of f of multiplicity at least k, so $\prod_i (t - z_i)^k$, which has degree ℓk, divides f; hence $f = 0$. So the map is injective, and since both spaces have dimension ℓk it is a bijection. The coefficients $c_{i,j}$ are the entries of the row of the inverse matrix corresponding to evaluation at z. □

So if $k\ell > nd$, it suffices to know $\mathrm{Perm}(A(t))$ and its first $k - 1$ derivatives at $t = 1, \dots, \ell$. A high-order derivative sounds expensive, but it turns out to be only a sum over choices of k rows and k columns.

Lemma 3.3 (Derivatives from submatrix permanents)**.** *Let $A(t)$ be an $n \times n$ matrix polynomial, let $P(t) = \mathrm{Perm}(A(t))$, and fix a point t_0. Suppose that the permanent of every square submatrix of $A(t_0)$ can be computed in time T. Then for every $1 \leq k \leq n/2$, the values*

$$P(t_0), \; P'(t_0), \; \dots, \; P^{(k-1)}(t_0)$$

can be computed together in time $T \cdot \mathrm{poly}(n) \cdot (n/k)^{O(k)}$ (assuming the entries of $A(t)$ have polynomial-size descriptions). In other words, computing derivatives up to order k costs only $(n/k)^{O(k)}$ permanents of submatrices of $A(t_0)$ itself.

Proof. Let $A_0 = A(t_0)$ and let $\Delta(z) = A(t_0 + z) - A_0$, a matrix whose entries are polynomials in z with no constant term. Then $P(t_0 + z) = \mathrm{Perm}(A_0 + \Delta(z))$, and by Taylor expansion $P^{(j)}(t_0) = j! \cdot [z^j] \, P(t_0 + z)$. So we need the coefficients of $z^0, \dots, z^{k-1}$ in $\mathrm{Perm}(A_0 + \Delta(z))$.

Expanding $\prod_u (A_0 + \Delta)_{u,\pi(u)}$ and recording the set S of rows that use their Δ-entry and the set

$T = \pi(S)$ of columns they are matched to, we get the *permanent addition formula*

$$\operatorname{Perm}\big(A_0 + \Delta(z)\big) = \sum_{a=0}^{n} \sum_{\substack{S,T\subseteq[n]\\|S|=|T|=a}} \operatorname{Perm}\big(A_0[S^c, T^c]\big) \cdot \operatorname{Perm}\big(\Delta(z)[S,T]\big).$$

Every entry of $\Delta(z)$ is divisible by z, so $\operatorname{Perm}(\Delta(z)[S,T])$ is divisible by z^a. Hence the terms with $a \geq k$ do not contribute to the coefficients of $z^0, \ldots, z^{k-1}$, and for $0 \leq j < k$,

$$P^{(j)}(t_0) = j! \sum_{a<k} \sum_{|S|=|T|=a} \operatorname{Perm}\big(A_0[S^c, T^c]\big) \cdot [z^j] \operatorname{Perm}\big(\Delta(z)[S,T]\big).$$

We evaluate this as follows. For each $a < k$ and each pair (S,T) of a-subsets, the first factor is the permanent of a square submatrix of A_0, which takes time T by assumption. The second factor is the permanent of an $a \times a$ matrix over $\mathbb{Q}[z]/(z^k)$ (we only need its coefficients up to z^{k-1}), which Ryser's formula (Fact 2.1) computes in $2^{O(a)}\operatorname{poly}(n)$ operations, since each operation in $\mathbb{Q}[z]/(z^k)$ costs $\operatorname{poly}(k)$. Accumulating all contributions gives the values $P(t_0), \ldots, P^{(k-1)}(t_0)$ in total time

$$T \cdot \operatorname{poly}(n) \cdot \sum_{a<k} \binom{n}{a}^2 2^{O(a)} \leq T \cdot \operatorname{poly}(n) \cdot k \binom{n}{k}^2 2^{O(k)} \leq T \cdot \operatorname{poly}(n) \cdot (n/k)^{O(k)},$$

where we used that $\binom{n}{a}$ is increasing for $a \leq n/2$, that $\binom{n}{k} \leq (en/k)^k$, and that $n/k \geq 2$ since $k \leq n/2$. □

Proposition 3.4 (Target ⇒ fast permanent)**.** *Suppose $A(t)$ satisfies (T1)–(T3) and k is an integer with $1 \leq k \leq n/2$ and $k\ell > nd$. Then $\operatorname{Perm}(A)$ can be computed in time*

$$(n/k)^{O(k)} \cdot 2^{O(B)} \cdot n^{O(r)} \cdot \operatorname{poly}(n).$$

Proof. Let $f(t) = \operatorname{Perm}(A(t))$. Every entry of $A(t)$ has degree at most d, so f has degree at most $nd < k\ell$. By Lemma 3.2 (with the points $1, \ldots, \ell$), $f(0)$ is a fixed linear combination of the numbers $f^{(j)}(i)$ for $i \in [\ell]$ and $0 \leq j < k$, and the coefficients can be computed in $\operatorname{poly}(\ell k)$ operations. Since $A(0) = A$, we have $f(0) = \operatorname{Perm}(A)$. So it suffices to compute all the ℓk numbers $f^{(j)}(i)$.

Fix $i \in [\ell]$. By (T3), $A(i) = P_i + E_i$. Every square submatrix of $A(i)$ is the sum of the corresponding submatrices of P_i and E_i. The submatrix of P_i is again B-blocky (with possibly rectangular blocks, Definition 2.2) and the submatrix of E_i has rank at most r, so by Fact 2.5 its permanent can be computed in time $T = 2^{O(B)} n^{O(r)}$. Hence, by Lemma 3.3 applied at $t_0 = i$, the numbers $f(i), f'(i), \ldots, f^{(k-1)}(i)$ can all be computed in time $T \cdot \operatorname{poly}(n) \cdot (n/k)^{O(k)}$.

Doing this for each of the ℓ points and then interpolating takes time $\ell \cdot T \cdot \operatorname{poly}(n) \cdot (n/k)^{O(k)} + \operatorname{poly}(\ell k)$, which is the claimed bound as $\ell, d \leq \operatorname{poly}(n)$ in our application. □

Combining Proposition 3.4 with Theorem 3.1 and a suitable choice of k proves Theorem 1.1; we do this in Section 7.

4 Step 1: Labelling rows and columns

Think of A as the biadjacency matrix of a bipartite graph G with n vertices on each side, maximum degree Δ, and $O(n)$ edges.

Lemma 4.1 (Good labelling)**.** *Let q, ℓ be integers with $q^\ell \approx C\sqrt{n}$, where C is a large constant depending on Δ, and $q \gg \Delta \cdot \ell$. Then there are maps* $\mathrm{lab} : [n]_{rows} \sqcup [n]_{cols} \to [q]^\ell$ *such that*

(L1) (balance) *every label is used at most $2n/q^\ell$ times (on each side);*

(L2) (distinct edges) *distinct edges $(r, c) \neq (r', c')$ have $(\mathrm{lab}(r), \mathrm{lab}(c)) \neq (\mathrm{lab}(r'), \mathrm{lab}(c'))$;*

(L3) (coordinatewise different) *for every edge (r, c) and every $i \in [\ell]$, $\mathrm{lab}(r)_i \neq \mathrm{lab}(c)_i$.*

Proof. Write $Q = q^\ell$ and $L = 2n/Q$. We label the $2n$ vertices one at a time, in an arbitrary order, each time picking a label that keeps (L1)–(L3) true for the vertices labelled so far. Say v is the next vertex. It is enough to show that fewer than Q labels are *forbidden* for v. The forbidden labels come in four kinds.

(a) *Full labels* (those already used L times on v's side; avoiding them gives (L1)). There are at most $n/L = Q/2$ of them.

(b) *Coordinate clashes* (giving (L3)). A labelled neighbour u of v forbids every label that agrees with $\bar{u}$ in some coordinate, which is at most $\ell Q/q$ labels. Over at most Δ neighbours this is $\Delta \cdot \ell Q/q$ which is a negligible fraction of Q since $q \gg \Delta \cdot \ell$.

(c) *Vertices at distance two.* Avoid the labels of the (at most Δ^2) labelled vertices on v's side that share a neighbour with v. This is a constant number of labels, negligible compared to Q.

(d) *Repeated pairs.* Say $v = r$ is a row and c is a labelled neighbour. If some already-labelled edge (r', c') has $\bar{c}' = \bar{c}$, we must not choose $\bar{r} = \bar{r}'$, or else (r, c) and (r', c') get the same pair of labels. There are at most Δ choices of c, at most L columns c' with $\bar{c}' = \bar{c}$, and at most Δ neighbours r' of each. So at most $\Delta^2 L = 2\Delta^2 n/Q$ labels are forbidden.

Only (d) is a real constraint: we need $2\Delta^2 n/Q \leq Q/4$, that is, $Q^2 \geq 8\Delta^2 n$. This is the reason for taking $Q = q^\ell \approx C\sqrt{n}$ for a large enough constant C (say $C = 8\Delta^2$ is plenty). Then (a) is $Q/2$, (d) is at most $Q/4$, and (b), (c) are tiny (at most $Q/8$ each), so fewer than Q labels are forbidden and a valid label exists.

Finally, (L2) follows: if two distinct edges $(r, c) \neq (r', c')$ had the same pair of labels, they either share an endpoint, say $r = r'$, so that c, c' are at distance two with the same label (ruled out by (c)), or they are disjoint, which (d) rules out. □

Remark 4.2. *The constraint $q \gtrsim \ell$ together with $q^\ell \approx \sqrt{n}$ forces $\ell \log \ell \lesssim \log n$. This is why ℓ is at most about $\log n / \log\log n$ in the proof of Theorem 3.1.* ◊

5 Step 2: Factoring A through the label pairs

Fix a labelling as in Lemma 4.1.

Definition 5.1 (Middle set). *Let* $M = \{(\sigma, \pi) \in [q]^\ell \times [q]^\ell : \sigma_i \neq \pi_i \ \forall i \in [\ell]\}$. ◊

The middle set M will be the vertex set of an intermediate layer. It has $|M| = (q(q-1))^\ell \leq q^{2\ell} = \Theta(n)$ elements.

5.1 First attempt: a tripartite picture

For brevity, we write $\bar{r} = \mathrm{lab}(r)$ for the label of a row r and $\bar{c} = \mathrm{lab}(c)$ for the label of a column c. For every edge $e = (r, c)$ of the bipartite graph of A, write $m(e) = (\bar{r}, \bar{c})$. By property (L3) this is an element of M, and by property (L2) distinct edges give distinct elements of M. We build a tripartite graph with the rows $[n]$ on the left, the set M in the middle, and the columns $[n]$ on the right.

- Each row r is connected to $m(r, c)$ for every neighbour c of r, with weight A_{rc}.
- Each column c is connected to $m(r', c)$ for every neighbour r' of c, with weight 1.

Let $R \in \mathbb{Q}^{[n] \times M}$ and $C \in \mathbb{Q}^{M \times [n]}$ be the two biadjacency matrices. Note that the first coordinate of every middle vertex adjacent to r is always $\bar{r}$, and the second coordinate of every middle vertex adjacent to c is always $\bar{c}$.

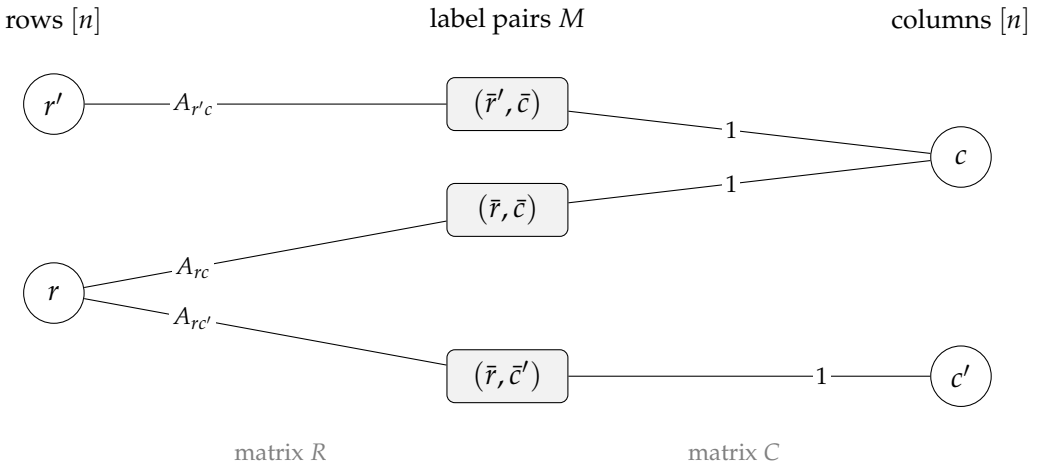

Figure 1: The tripartite graph for the edges (r, c), (r, c') and (r', c) of the bipartite graph of A. The middle vertex $(\bar{r}, \bar{c})$ is reached from r with weight A_{rc}, and leads to c with weight 1. The only path from r to c goes through this vertex.

Lemma 5.2. $A = R\,C$.

Proof. The entry $(RC)_{r,c}$ is the total weight of the paths from r to c. A path goes through a middle vertex (σ, π) that is $m(r, c_1)$ for a neighbour c_1 of r and also $m(r_1, c)$ for a neighbour r_1 of c. By (L2), $m(r, c_1) = m(r_1, c)$ forces $(r, c_1) = (r_1, c)$. Hence if (r, c) is not an edge there is no path, and otherwise the only path goes through $m(r, c)$ and has weight A_{rc}. □

Nothing interesting has happened so far: we have just written A as a product of two matrices by routing every edge through its own private middle vertex. The interesting part comes next.

5.2 The twist: a four-layer picture

Now we add a literal twist. We use *two* copies of the middle layer, and connect them by the *swap*: the vertex (σ, π) of the first copy is joined to the vertex (π, σ) of the second copy. The graph now has four layers.

- Row r is connected to $(\bar{r}, \bar{c})$ in the first copy, for every neighbour c of r, with weight A_{rc}. This is the same matrix R as before.
- The vertex (σ, π) of the first copy is connected to (π, σ) of the second copy with weight 1. Let $S \in \{0,1\}^{M \times M}$ be this matrix, that is, the permutation matrix of the map $(\sigma, \pi) \mapsto (\pi, \sigma)$.
- Column c is connected to the *swapped* vertices $(\bar{c}, \bar{r}')$ in the second copy, for every neighbour r' of c, with weight 1. Call this matrix C'.

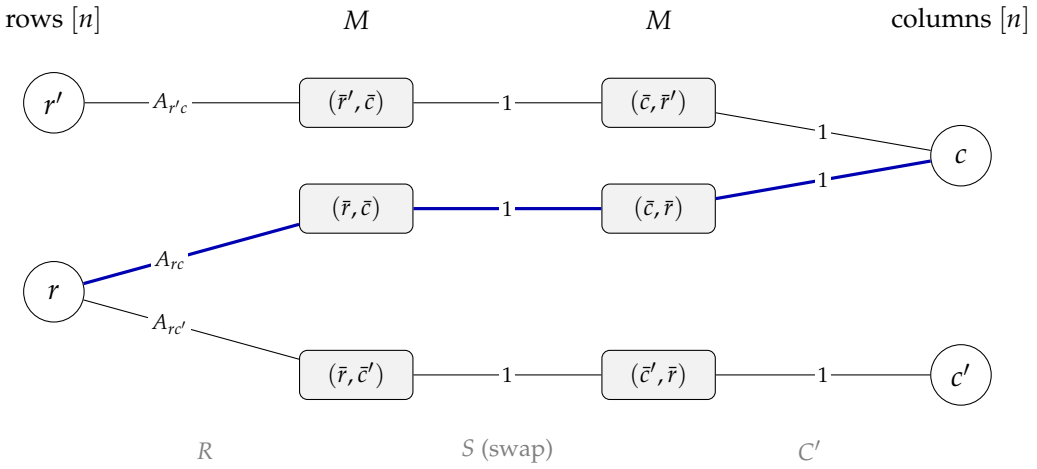

Figure 2: The four-layer graph for the same edges as in Figure 1. Compared to the tripartite graph, the second copy of M has been swapped: the edge (r, c) now goes through $(\bar{r}, \bar{c})$, then through its swap $(\bar{c}, \bar{r})$, and ends at c. The path from r to c is highlighted.

Lemma 5.3. $A = R\,S\,C'$.

Proof. By construction, $C' = SC$: the matrix C' is the matrix C with the two coordinates of each middle vertex swapped. Since S is the permutation matrix of an involution, $S^2 = I$, and so $R\,S\,C' = R\,S^2\,C = R\,C = A$ by Lemma 5.2. □

So, again, nothing has really happened: we have merely inserted $S \cdot S = I$ in the middle of the factorisation $A = RC$. What matters is that the swap S has a lot of structure, which we turn to next.

5.3 The coordinate swaps

The swap S exchanges the two strings $\sigma, \pi \in [q]^\ell$ of a middle vertex (σ, π), that is, it swaps all ℓ pairs of coordinates (σ_i, π_i) at once. We break it up into ℓ pieces, each of which swaps only one pair of coordinates.

Definition 5.4 (Coordinate swaps)**.** *For $i \in [\ell]$, let X_i be the permutation matrix of the map $s_i \colon M \to M$ that swaps σ_i and π_i and leaves all other coordinates unchanged,*

$$s_i\big((\sigma_1, \ldots, \sigma_i, \ldots, \sigma_\ell), (\pi_1, \ldots, \pi_i, \ldots, \pi_\ell)\big) = \big((\sigma_1, \ldots, \pi_i, \ldots, \sigma_\ell), (\pi_1, \ldots, \sigma_i, \ldots, \pi_\ell)\big). \qquad \Diamond$$

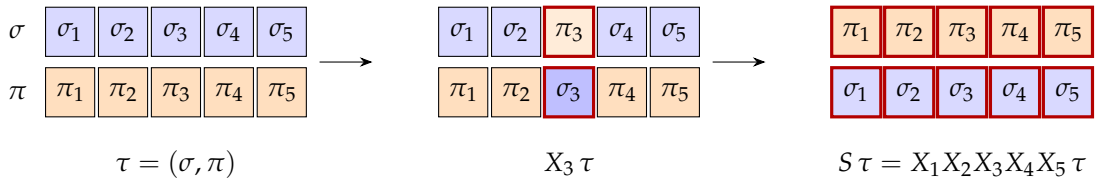

Figure 3: The effect of X_3 and of S on a middle vertex $\tau = (\sigma, \pi)$ (with $\ell = 5$). The top row is σ (blue) and the bottom row is π (orange). The matrix X_3 exchanges only the third column; S exchanges all columns.

Lemma 5.5. *(1) Each s_i maps M to M and $s_i \circ s_i = \mathrm{id}$, so $X_i^2 = I$.*

(2) The X_i commute with each other, and $S = X_1 X_2 \cdots X_\ell$.

(3) Fix $\tau = (\sigma, \pi) \in M$. The connected component of τ (where every vertex is joined to its images under $s_1, \ldots, s_\ell$) consists of the 2^ℓ vertices (σ', π') in which, for each $i \in [\ell]$, the pair (σ_i', π_i') is either (σ_i, π_i) or (π_i, σ_i). In particular M splits into $|M|/2^\ell$ components, each of size 2^ℓ.

Proof. (1) The definition of M is symmetric in σ_i and π_i, so swapping them keeps us inside M; swapping twice does nothing. (2) The maps s_i and s_j for $i \neq j$ act on different coordinates, so they commute; applying all of them swaps every coordinate, which is S.

(3) At each position i, the pair of values is one of two options: (σ_i, π_i) or (π_i, σ_i). These two options are different because $\sigma_i \neq \pi_i$. The map s_i toggles the choice at position i and does not touch

any other position. So starting from τ and applying the s_i we can reach every combination of the ℓ binary choices, and we can never leave this set of 2^ℓ combinations. □

Thus we can identify each component with $\{0,1\}^\ell$, where $a_i = 1$ means that the pair at position i is swapped compared to the base vertex τ. Under this identification X_i flips the bit a_i, and S flips all the bits.

It is here that property (L3) of the labelling is used: without it a middle vertex could have $\sigma_i = \pi_i$, and X_i would fix such a vertex.

6 Step 3: The matrix polynomial

We have factored $A = R\,S\,C'$ with $S = X_1 X_2 \cdots X_\ell$. The idea is to deform the middle matrix S.

6.1 Dropping one swap gives a blocky matrix

Suppose we replace the middle matrix $S = X_1 \cdots X_\ell$ by some other matrix Q that is built out of the X_j but *does not use* X_i for some fixed i, for example $Q = \prod_{j \neq i} X_j$, or any linear combination of products of the X_j with $j \neq i$. Then $R\,Q\,C'$ is a blocky matrix.

Lemma 6.1. *Let $i \in [\ell]$ and let Q be a linear combination of products of the matrices X_j with $j \neq i$. Then $R\,Q\,C'$ is B-blocky (Definition 2.2) with $B \leq 2n/q$. The blocks are indexed by $v \in [q]$: the v-th block consists of the rows r with $\bar{r}_i = v$ and the columns c with $\bar{c}_i = v$.*

Proof. Every product of the X_j with $j \neq i$ moves a middle vertex (σ, π) to a vertex (σ', π') with $\sigma'_i = \sigma_i$ (and $\pi'_i = \pi_i$), because only positions other than i are swapped. The same is therefore true of Q: it has a nonzero entry from (σ, π) to (σ', π') only if $\sigma'_i = \sigma_i$.

Now follow a path from a row r to a column c in the four-layer graph with Q in the middle (Figure 2). The row r is joined only to vertices (σ, π) with $\sigma = \bar{r}$, and the column c is joined only to vertices (σ', π') with $\sigma' = \bar{c}$. Hence a path exists only if $\bar{c}_i = \sigma'_i = \sigma_i = \bar{r}_i$. So the entry $(R\,Q\,C')_{r,c}$ is zero unless $\bar{r}_i = \bar{c}_i$, which is the claimed block structure. By property (L1), each label is used at most $2n/q^\ell$ times, and there are $q^{\ell-1}$ labels with a given value of the i-th coordinate, so each block has at most $2n/q$ rows and at most $2n/q$ columns. □

Two remarks. First, the blocks need not be square (the number of rows with $\bar{r}_i = v$ can differ from the number of columns with $\bar{c}_i = v$), which is why we allowed rectangular blocks in Definition 2.2. Second, q is large, so the blocks are small compared to n: $B \leq 2n/q = o(n)$.

Of course the original matrix $A = R\,S\,C'$ is not of this form, since S uses every X_i. But Lemma 6.1 says that for each i separately, we can get a blocky matrix by getting rid of X_i. This is the mechanism that we will exploit to make the matrix easy at the ℓ different points $t = 1, \ldots, \ell$: at the point $t = i$, we will arrange for X_i to disappear.

6.2 Making X_i disappear at $t = i$

Replace each X_i by $Y_i(t) = (1 - t/i)\, X_i$. At $t = 0$ this is X_i and at $t = i$ it is the zero matrix. Let $P(Y_1, \ldots, Y_\ell)$ be a polynomial of degree at most d in ℓ commuting variables, to be chosen later. Then $P(Y_1(t), \ldots, Y_\ell(t))$ has degree at most d in t, and at $t = i$ the variable Y_i is zero, so it is a linear combination of products of the X_j with $j \neq i$. By Lemma 6.1, the matrix $R\, P(Y(t))\, C'$ is blocky at $t = i$, for every $i \in [\ell]$.

The only problem is that at $t = 0$ this matrix is $R\, P(X_1, \ldots, X_\ell)\, C'$, and it is not A unless $P(X_1, \ldots, X_\ell) = S = X_1 \cdots X_\ell$. We fix this with a correction term that does not depend on t:

Definition 6.2. *Let $Y_i(t) = (1 - t/i)\, X_i$ and let P be a polynomial of degree at most d. Define*

$$A(t) \;=\; R\Big(P\big(Y_1(t), \ldots, Y_\ell(t)\big) + \big(S - P(X_1, \ldots, X_\ell)\big)\Big)C'. \qquad \lozenge$$

Lemma 6.3 (Properties of $A(t)$)**.** *(1) Entries of $A(t)$ have degree at most d in t.*

(2) $A(0) = A$.

(3) For $i \in [\ell]$, $A(i) = R\, P(Y(i))\, C' + R\,\big(S - P(X)\big)\, C'$, where the first matrix is B-blocky with $B \leq 2n/q$ and the second matrix does not depend on i.

Proof. (1) Each $Y_i(t)$ has degree 1 in t and P has degree d, so the entries of $P(Y(t))$, and hence of $A(t)$, have degree at most d in t. (The correction term does not depend on t.)

(2) At $t = 0$ we have $Y_i(0) = X_i$ for all i, so the correction term cancels $P(Y(0)) = P(X)$, and $A(0) = R\, S\, C' = A$ by Lemma 5.3.

(3) Setting $t = i$ sets $Y_i = 0$, so X_i does not appear in $P(Y(i))$, and Lemma 6.1 shows that $R\, P(Y(i))\, C'$ is B-blocky with $B \leq 2n/q$. The correction term $R\,(S - P(X))\, C'$ does not depend on t. □

So $A(t)$ is blocky plus E at each of the points $t = 1, \ldots, \ell$, where $E = R\,(S - P(X))\, C'$ is one fixed matrix. It remains to choose P so that E has *low rank*.

An exact choice, and why the degree is the issue. If we did not care about the degree, we could in fact make the correction vanish. Take the middle matrix to be

$$\prod_{i=1}^{\ell} \Big(\big(1 - \tfrac{t}{i}\big) X_i + \tfrac{t}{i}\, I \Big).$$

At $t = 0$ this is $X_1 \cdots X_\ell = S$, so we recover A exactly. At $t = i$, the i-th factor is the identity, so the whole product is built from the X_j with $j \neq i$ only, and by Lemma 6.1 the matrix $R\,(\cdots)\, C'$ is blocky, for every $i \in [\ell]$. So this gives a perfectly good $A(t)$ with *no* low rank error at all ($E = 0$), except that every entry of it has degree ℓ and not $d \approx \sqrt{\ell}$. Then $\mathrm{Perm}(A(t))$ has degree $n\ell$, and

since we only have ℓ easy points, Hermite interpolation would need derivatives of order $k > n$, for which the bound of Lemma 3.3 is no longer $2^{o(n)}$.

So the degree is the issue. The real task is to find a polynomial P of much lower degree, $d \approx \sqrt{\ell}$, for which E has low rank: we trade exactness for degree, and pay for it with a low rank error.

6.3 The correction has low rank when P approximates parity

We work inside a single connected component K of M. By Lemma 5.5, fixing a base vertex in K lets us identify K with $\{0,1\}^\ell$, where X_i flips the bit a_i. A vector $v \in \mathbb{R}^K$ is then just a function $v\colon \{0,1\}^\ell \to \mathbb{R}$, and X_i acts on it by

$$(X_i v)(a) = v(a \oplus e_i),$$

where e_i is the i-th unit vector.

Definition 6.4 (The vectors χ_T)**.** *For a subset $T \subseteq [\ell]$, let $\chi_T \in \mathbb{R}^K$ be the vector*

$$\chi_T(a) \;=\; (-1)^{\sum_{i\in T} a_i} \;=\; \prod_{i\in T}(-1)^{a_i}. \qquad \Diamond$$

Lemma 6.5 (Common eigenbasis)**.** *(1) For every $T \subseteq [\ell]$ and $i \in [\ell]$, $X_i\,\chi_T = (-1)^{[i\in T]}\,\chi_T$. That is, χ_T is an eigenvector of X_i with eigenvalue $+1$ if $i \notin T$ and -1 if $i \in T$.*

(2) The 2^ℓ vectors χ_T are pairwise orthogonal, and hence form a basis of $\mathbb{R}^K$.

Proof. (1) We have $(X_i\chi_T)(a) = \chi_T(a \oplus e_i)$. Flipping the bit a_i changes the sign of $(-1)^{\sum_{j\in T} a_j}$ exactly when $i \in T$, and does not change it otherwise. So $\chi_T(a \oplus e_i) = (-1)^{[i\in T]}\chi_T(a)$ for every a.

(2) For $T \neq T'$,

$$\langle \chi_T, \chi_{T'} \rangle = \sum_{a\in\{0,1\}^\ell} (-1)^{\sum_{i\in T\triangle T'} a_i} = \prod_{i\in T\triangle T'} \Big(\sum_{a_i\in\{0,1\}} (-1)^{a_i} \Big) \cdot \prod_{i\notin T\triangle T'} 2 = 0,$$

because $T\triangle T'$ is nonempty and each factor $\sum_{a_i}(-1)^{a_i}$ is 0. Orthogonal nonzero vectors are linearly independent, and there are $2^\ell = \dim \mathbb{R}^K$ of them. $\square$

In other words, in the basis $\{\chi_T\}_{T\subseteq[\ell]}$ of $\mathbb{R}^K$, each X_i is a *diagonal* matrix whose diagonal entry at χ_T is $(-1)^{[i\in T]} \in \{+1,-1\}$. Doing this in every component of M gives a basis of $\mathbb{R}^M$ in which all the $X_1, \ldots, X_\ell$ are simultaneously diagonal.

Reading off the diagonal. Let us think of one block of size 2^ℓ as having its rows and columns indexed by subsets $T \subseteq [\ell]$ (the index of χ_T). Then X_i is the diagonal matrix with entry -1 at every T that contains i and entry $+1$ at every T that does not. Consequently, at the entry indexed by T:

- $S = X_1 X_2 \cdots X_\ell$ has diagonal entry $\prod_i (-1)^{[i \in T]} = (-1)^{|T|}$, the *parity* of $|T|$;
- $X_1 + \cdots + X_\ell$ has diagonal entry $\sum_i (-1)^{[i \in T]} = \ell - 2|T|$, which measures how far T is from being balanced (the *bias*); this is the sum of the ℓ signs $y_i = (-1)^{[i \in T]}$;
- more generally, for any polynomial P, the matrix $P(X_1, \ldots, X_\ell)$ is diagonal with entry $P(y_1, \ldots, y_\ell)$ at T, where $y_i = (-1)^{[i \in T]}$.

In particular, $S - P(X_1, \ldots, X_\ell)$ is diagonal with entry $\prod_i y_i - P(y)$ at T. This is nonzero exactly at those T where P disagrees with parity. We therefore want to find a polynomial P of as low a degree as possible that nevertheless agrees with parity at many places.

Lemma 6.6 (Low rank)**.** *Let P be a polynomial, and let N_P be the number of $y \in \{\pm 1\}^\ell$ (equivalently, of subsets $T \subseteq [\ell]$) with $P(y) \neq \prod_i y_i$. Then*

$$\operatorname{rank}\left(R\left(S - P(X_1, \ldots, X_\ell)\right) C'\right) \;\leq\; \frac{|M|}{2^\ell} \cdot N_P.$$

Proof. By Lemma 6.5, in each of the $|M|/2^\ell$ components of M the matrix $S - P(X_1, \ldots, X_\ell)$ is diagonal in the basis $\{\chi_T\}_T$, with entry $\prod_i y_i - P(y)$ at χ_T, where $y_i = (-1)^{[i \in T]}$. This entry is nonzero for exactly N_P subsets T. Hence $S - P(X)$ has rank exactly N_P on each component, and, as it is block diagonal over the components, its total rank is $\frac{|M|}{2^\ell} \cdot N_P$. Finally, the rank of a product is at most the rank of any of its factors, so

$$\operatorname{rank}\left(R\left(S - P(X)\right) C'\right) \leq \operatorname{rank}\left(S - P(X)\right) = \frac{|M|}{2^\ell} \cdot N_P. \qquad \square$$

Here $|M|/2^\ell$ is the number of components of M, and each component contributes the N_P disagreements. Since $|M| = \Theta(n)$, we need N_P to be a tiny fraction of 2^ℓ for the rank to be small.

6.4 Choosing P

We need a polynomial of low degree that agrees with parity $\prod_i y_i$ at many points y of the cube $\{\pm 1\}^\ell$. Parity itself is as far from low degree as possible, but we do not need it everywhere. The key observation is that the cube is concentrated around its middle slices. Slice the cube according to the sum $s = y_1 + \cdots + y_\ell$, so that the number of -1's in y is $(\ell - s)/2$. The number of points in the slice is $\binom{\ell}{(\ell-s)/2}$, which is largest at $s = 0$ and decays like $e^{-s^2/2\ell}$. So almost all the points lie in the slices with $|s| \leq d$ for d a little larger than $\sqrt{\ell}$. On each slice, parity is a constant, namely $(-1)^{(\ell-s)/2}$, so we only need a polynomial in s that interpolates these $O(d)$ constants.

Lemma 6.7 (Approximating parity)**.** *For every constant c there is $d = O(\sqrt{\ell \log \ell})$ and a polynomial $P(Y_1, \ldots, Y_\ell)$ of degree at most d, of the form $P(Y_1, \ldots, Y_\ell) = p(Y_1 + \cdots + Y_\ell)$ for a univariate polynomial*

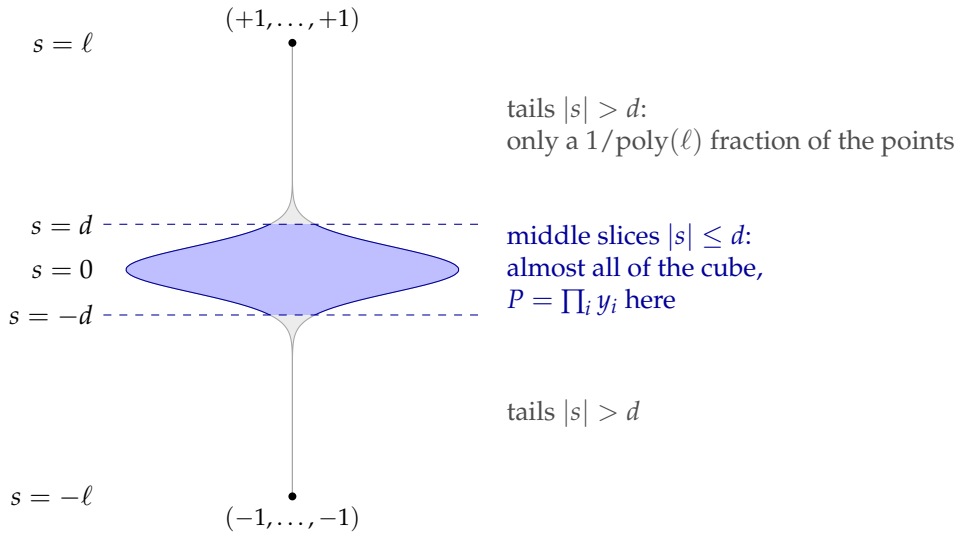

Figure 4: The cube $\{\pm 1\}^\ell$ drawn by slices $s = \sum_i y_i$, with the width of the picture at height s proportional to the number of points in that slice. Almost all of the mass is in the slices with $|s| \le d$, which is only a window of width about $d \approx \sqrt{\ell}$ out of the $\ell + 1$ slices.

p, such that

$$P(y) = \prod_{i=1}^{\ell} y_i \qquad \textit{for every } y \in \{\pm 1\}^\ell \textit{ with } \Big|\sum_i y_i\Big| \le d.$$

In particular, the number N_P of points $y \in \{\pm 1\}^\ell$ with $P(y) \ne \prod_i y_i$ satisfies $N_P \le 2^\ell/\ell^c$.

Proof. Let $y \in \{\pm 1\}^\ell$ and let w be the number of -1's in y. Then $s = \sum_i y_i = \ell - 2w$ and $\prod_i y_i = (-1)^w = (-1)^{(\ell - s)/2}$, which depends only on s. The integers $s \equiv \ell \pmod 2$ with $|s| \le d$ are at most $d + 1$ in number. By Lagrange interpolation, there is a polynomial p of degree at most d with $p(s) = (-1)^{(\ell-s)/2}$ for each of these s. For this p and for any y with $|\sum_i y_i| \le d$, we get $P(y) = p(s) = \prod_i y_i$.

It remains to see that the slices with $|s| \le d$ cover all but a $1/\mathrm{poly}(\ell)$ fraction of the cube. For a uniformly random y, the sum $\sum_i y_i$ is a sum of ℓ independent uniform signs, so by Hoeffding's inequality $\Pr[|\sum_i y_i| > d] \le 2e^{-d^2/2\ell}$. Taking $d = \lceil \sqrt{2(c+1)\,\ell \ln \ell}\,\rceil$ makes this at most $2\ell^{-(c+1)} \le \ell^{-c}$, so at most $2^\ell/\ell^c$ points y lie outside the middle slices. □

We use $P(Y_1, \ldots, Y_\ell) = p(Y_1 + \cdots + Y_\ell)$ in Definition 6.2. The degree is $d = O(\sqrt{\ell \log \ell})$, which is much smaller than ℓ.

Proof of Theorem 3.1. Let C be the constant of Lemma 4.1. Take $\ell = \lfloor \log_2 n/(2 \log_2 \log_2 n) \rfloor$ and $q = \lceil (C\sqrt{n})^{1/\ell} \rceil$. Then $q \ge 2^{\log_2 n/2\ell} \ge \log_2 n \gg \Delta\ell$ for large n, and $C\sqrt{n} \le q^\ell \le e\,C\sqrt{n}$ (for the upper bound, $q \le x + 1$ where $x = (C\sqrt{n})^{1/\ell} \ge \log_2 n \ge \ell$, so $q^\ell \le x^\ell(1 + 1/x)^\ell \le e\,C\sqrt{n}$),

so Lemma 4.1 applies, and $|M| \le q^{2\ell} = O(n)$. Let $P = p(Y_1 + \cdots + Y_\ell)$ be the polynomial of Lemma 6.7 with $c = 2$, so that its degree is $d = O(\sqrt{\ell \log \ell}) = O(\sqrt{\log n})$ and $N_P \le 2^\ell / \ell^2$.

Compute a labelling, build $R, S, C', X_1, \ldots, X_\ell$, and define $A(t)$ by Definition 6.2; this takes $\mathrm{poly}(n)$ time since all matrices have dimension $O(n)$. By Lemma 6.3, the entries of $A(t)$ have degree at most d and $A(0) = A$, which gives (T1) and (T2). For (T3), Lemma 6.3 also says that $A(i) = P_i + E$ for every $i \in [\ell]$, where P_i is B-blocky with $B \le 2n/q \le 3n/\log_2 n$, and $E = R\,(S - P(X))\,C'$ is the same for every i. By Lemma 6.6 and Lemma 6.7,

$$\mathrm{rank}(E) \le \frac{|M|}{2^\ell} \cdot N_P \le \frac{|M|}{\ell^2} = O\Big(\frac{n}{\ell^2}\Big) = \widetilde{O}\Big(\frac{n}{\log^2 n}\Big). \qquad \square$$

7 Putting it together: parameters

We now combine Theorem 3.1 and Proposition 3.4.

Proof of Theorem 1.1. By Theorem 3.1, we can compute in $\mathrm{poly}(n)$ time a matrix $A(t)$ satisfying (T1)–(T3) with $d = \widetilde{O}(\sqrt{\log n})$, $\ell = \widetilde{\Theta}(\log n)$, $B = O(n/\log n)$ and $r = \widetilde{O}(n/\log^2 n)$. Let $k = \lceil nd/\ell \rceil + 1$, so that $k\ell > nd$ and $k = \widetilde{O}(n/\sqrt{\log n}) = o(n)$. By Proposition 3.4, $\mathrm{Perm}(A)$ can be computed in time

$$(n/k)^{O(k)} \cdot 2^{O(B)} \cdot n^{O(r)} \cdot \mathrm{poly}(n),$$

and each factor is $2^{o(n)}$:

- $n/k \le \ell/d \le \ell$, so $(n/k)^{O(k)} \le 2^{O(k \log \ell)} = 2^{\widetilde{O}(n/\sqrt{\log n})}$;
- $2^{O(B)} = 2^{O(n/\log n)}$;
- $n^{O(r)} = 2^{O(r \log n)} = 2^{\widetilde{O}(n/\log n)}$.

Hence the running time is $2^{o(n)}$. $\square$

Remark 7.1. *Throughout we have counted arithmetic operations. The numbers involved (the entries of A, the coefficients of the polynomial p, the points $1, \ldots, \ell$, and the coefficients in the Hermite interpolation) are rational numbers of* $\mathrm{poly}(n)$ *bit-length, so the cost of bit-arithmetic is only a* $\mathrm{poly}(n)$ *factor, which is absorbed in* $2^{o(n)}$. $\diamondsuit$

Acknowledgements: Massive thanks to Ryan O'Donnell for his awesome videos, and for ITU Copenhagen for arranging a watch-party. I'm especially grateful Vishwas Bhargava for sharing his hand-written notes, and also to Nutan Limaye and Prahladh Harsha for patiently sitting through my understanding of the proof.

References

[Bar96] Alexander I. Barvinok. Two Algorithmic Results for the Traveling Salesman Problem. *Mathematics of Operations Research*, 21(1):65–84, 1996.

[BDH93] Amir Ben-Dor and Shai Halevi. Zero-One Permanent Is #P-Complete, A Simpler Proof. In *Proceedings of the 2nd Israel Symposium on Theory of Computing and Systems (ISTCS)*, pages 108–117, 1993.

[CIP06] Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. A Duality between Clause Width and Clause Density for SAT. In *Conference on Computational Complexity (CCC)*, pages 252–260, 2006.

[Cla26] Claude (Anthropic). A $2^{o(n)}$ Time Algorithm for 3-SAT, 2026. Preprint.

[DHM+14] Holger Dell, Thore Husfeldt, Dániel Marx, Nina Taslaman, and Martin Wahlén. Exponential Time Complexity of the Permanent and the Tutte Polynomial. *ACM Transactions on Algorithms*, 10(4):21:1–21:32, 2014.

[O'D26a] Ryan O'Donnell. ETH False, 2026. Video lecture.

[O'D26b] Ryan O'Donnell. ETH False II: completing the proof, 2026. Video lecture.

[O'D26c] Ryan O'Donnell. Some exposition of "The Exponential Time Hypothesis is False", 2026. Lecture notes.

[Ope26] OpenAI. The Exponential Time Hypothesis is False, 2026. Preprint, October 25, 2026 version.

[Rys63] Herbert J. Ryser. *Combinatorial Mathematics*. Number 14 in Carus Mathematical Monographs. Mathematical Association of America, 1963.

[Val79] Leslie G. Valiant. The Complexity of Computing the Permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

A Proofs of the standard facts

A.1 Ryser's formula (Fact 2.1)

Ryser's formula states that for an $n \times n$ matrix A,

$$\mathrm{Perm}(A) \;=\; \sum_{S \subseteq [n]} (-1)^{n-|S|} \prod_{u=1}^{n} \Big(\sum_{v \in S} A_{uv} \Big).$$

This is just inclusion–exclusion. The product $\prod_u \sum_{v \in S} A_{uv}$ is the sum of $\prod_u A_{u,f(u)}$ over all functions $f\colon [n] \to S$, so the alternating sum over S leaves only the functions f that are onto $[n]$, that is, the permutations. We leave the details as an exercise. Evaluating each of the 2^n terms takes $\mathrm{poly}(n)$ arithmetic operations, which gives Fact 2.1.

A.2 Blocky matrices (Fact 2.3)

As noted after Definition 2.2, $\mathrm{Perm}(P)$ is 0 if some block is not square, and is the product of the permanents of the blocks $P[R_j, C_j]$ otherwise. Each block has dimension at most B, so its permanent takes $2^B \,\mathrm{poly}(B)$ operations by Fact 2.1, and there are at most n nonempty blocks. Hence the total time is $2^{O(B)} \cdot \mathrm{poly}(n)$.

A.3 Low rank matrices (Fact 2.4)

We give a short, slick proof (which in some sense is a variant of Barvinok's algorithm). We may assume $r \geq 1$, since a matrix of rank 0 is zero. By Gaussian elimination, write $E = UV^{\mathsf{T}}$ with $U, V \in \mathbb{Q}^{n \times r}$. Let $u_i, v_j \in \mathbb{Q}^r$ be the i-th row of U and the j-th row of V respectively, so that $E_{ij} = \langle u_i, v_j \rangle$. Thus,

$$\mathrm{Perm}(E) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} E_{i,\sigma(i)} = \sum_{\sigma \in S_n} \prod_{i=1}^{n} \left\langle u_i, v_{\sigma(i)} \right\rangle .$$

Let $t_1, \ldots, t_r$ be variables, and write $t = (t_1, \ldots, t_r)$ and $\partial_t = (\partial_{t_1}, \ldots, \partial_{t_r})$. Define the linear forms and the first-order differential operators

$$\ell_j(t) = \langle v_j, t \rangle = \sum_{b=1}^{r} v_{j,b}\, t_b, \qquad D_i = \langle u_i, \partial_t \rangle = \sum_{a=1}^{r} u_{i,a}\, \partial_{t_a},$$

and let

$$Q(t) = \prod_{j=1}^{n} \ell_j(t), \qquad P(\partial_t) = \prod_{i=1}^{n} D_i.$$

So Q is a polynomial in the r variables $t_1, \ldots, t_r$, and P is a polynomial in the r variables $\partial_{t_1}, \ldots, \partial_{t_r}$, and both are homogeneous of degree n. Note that

$$D_i\, \ell_j = \sum_{a=1}^{r} u_{i,a}\, \partial_{t_a} \Big(\sum_{b=1}^{r} v_{j,b}\, t_b \Big) = \sum_{a=1}^{r} u_{i,a} v_{j,a} = \langle u_i, v_j \rangle .$$

Claim. $P(\partial_t)\, Q(t) = \mathrm{Perm}(E)$.

Proof. By the product rule (Leibniz rule), applying $D_1, \ldots, D_n$ to $\ell_1 \cdots \ell_n$ is a sum over all ways of choosing, for each D_i, which factor ℓ_j it differentiates. If some ℓ_j is differentiated twice, the term vanishes, since the second derivative of a linear form is 0. So only the terms in which each D_i is matched to a distinct factor survive, that is, the terms given by permutations $\sigma \in S_n$, and each such term equals $\prod_i D_i \ell_{\sigma(i)} = \prod_i \left\langle u_i, v_{\sigma(i)} \right\rangle$. Summing over σ gives $\mathrm{Perm}(E)$. □

Computing $P(\partial_t)\,Q(t)$. Write $Q(t) = \sum_{|\alpha|=n} q_\alpha\, t^\alpha$ and $P(\partial_t) = \sum_{|\alpha|=n} p_\alpha\, \partial_t^\alpha$, where $\alpha \in \mathbb{N}^r$. For $|\alpha| = |\beta| = n$ we have $\partial_t^\alpha t^\beta = \alpha!$ if $\alpha = \beta$ (with $\alpha! = \prod_a \alpha_a!$), and $\partial_t^\alpha t^\beta = 0$ otherwise, because some $\alpha_a > \beta_a$. Hence

$$\mathrm{Perm}(E) = \sum_{|\alpha|=n} \alpha!\; p_\alpha\, q_\alpha.$$

There are at most $(n+1)^r$ monomials of degree n in r variables. We can compute Q by multiplying out the n linear forms ℓ_j one at a time, and every intermediate polynomial has at most $(n+1)^r$ monomials, so this takes $O(nr(n+1)^r)$ operations. The same works for P, and the final sum has at most $(n+1)^r$ terms. Hence the total running time is $n^{O(r)}$.

A.4 Blocky plus low rank (Fact 2.5)

Let P be a square B-blocky matrix of dimension $m \le n$, with row groups $R_1, \dots, R_s$ and column groups $C_1, \dots, C_s$ (Definition 2.2), and let E be an $m \times m$ matrix of rank at most r, where we may assume $r \ge 1$. As in the proof of Fact 2.4, write $E = UV^{\mathsf{T}}$ and let $u_i, v_j \in \mathbb{Q}^r$ be the rows of U and V, so that $E_{ij} = \langle u_i, v_j \rangle$. We use the same linear forms, but now with two independent sets of variables $y = (y_1, \dots, y_r)$ and $t = (t_1, \dots, t_r)$, where y plays the role of ∂_t:

$$D_i(y) = \langle u_i, y \rangle \;\text{ for each row } i, \qquad \ell_j(t) = \langle v_j, t \rangle \;\text{ for each column } j.$$

Let L be the linear map on polynomials in (y, t) that sends $y^\alpha t^\beta$ to $\alpha!$ if $\alpha = \beta$, and to 0 otherwise. By the proof of Fact 2.4, applied to the submatrix $E[X, Y] = U[X]\, V[Y]^{\mathsf{T}}$, we have for every set X of rows and Y of columns with $|X| = |Y|$,

$$\mathrm{Perm}\big(E[X,Y]\big) \;=\; L\Big(\prod_{i\in X} D_i(y) \cdot \prod_{j\in Y} \ell_j(t)\Big). \tag{$*$}$$

(This is the identity $P(\partial_t)Q(t) = \mathrm{Perm}(E)$, written using L; for $X = Y = \emptyset$ both sides are 1.)

Splitting into P-edges and E-edges. Expanding $\prod_i (P+E)_{i,\pi(i)}$ and recording the set S of rows that use their P-entry and the set $T = \pi(S)$ of columns they are matched to, we get

$$\mathrm{Perm}(P+E) = \sum_{\substack{S,T\subseteq[m]\\ |S|=|T|}} \mathrm{Perm}\big(P[S,T]\big) \cdot \mathrm{Perm}\big(E[S^c, T^c]\big).$$

Since P is zero outside the blocks, $\mathrm{Perm}(P[S,T]) = 0$ unless $|S_b| = |T_b|$ for every block b, where $S_b = S \cap R_b$ and $T_b = T \cap C_b$, and in that case $\mathrm{Perm}(P[S,T]) = \prod_b \mathrm{Perm}(P[S_b, T_b])$. For each block

b, define the polynomial in (y, t)

$$\Theta_b(y,t) = \sum_{\substack{S_b \subseteq R_b,\ T_b \subseteq C_b \\ |S_b| = |T_b|}} \operatorname{Perm}\big(P[S_b, T_b]\big) \cdot \prod_{i \in R_b \setminus S_b} D_i(y) \cdot \prod_{j \in C_b \setminus T_b} \ell_j(t).$$

Multiplying these polynomials multiplies out the sum over all choices of $(S_b, T_b)_b$, so by linearity of L and $(*)$ (applied to $X = S^c$ and $Y = T^c$, which have equal sizes),

$$L\big(\Theta_1 \Theta_2 \cdots \Theta_s\big) = \sum_{S,T} \operatorname{Perm}\big(P[S,T]\big) \cdot L\Big(\prod_{i \notin S} D_i \prod_{j \notin T} \ell_j\Big) = \operatorname{Perm}(P + E).$$

Running time. Fix a block b. There are at most 2^{2B} pairs (S_b, T_b). For each pair, $\operatorname{Perm}(P[S_b, T_b])$ takes $2^B \operatorname{poly}(B)$ operations by Fact 2.1. The product of at most B linear forms in y and at most B linear forms in t is a polynomial with at most $(B+1)^{2r}$ monomials, and we can compute it by multiplying out one factor at a time in $(B+1)^{O(r)}$ operations. So each Θ_b is computed in time $2^{O(B)} \cdot n^{O(r)}$. Only blocks with a nonempty row or column group matter (the others have $\Theta_b = 1$), and there are at most $2m \le 2n$ of them. We multiply the Θ_b one at a time. Every partial product has degree at most m in y and in t, so it has at most $(n+1)^{2r}$ monomials, and each multiplication costs at most $(n+1)^{2r}(B+1)^{2r} \le (n+1)^{4r}$ operations. Finally, we apply L to the result, which has at most $(n+1)^{2r}$ monomials. Altogether, the running time is $2^{O(B)} \cdot n^{O(r)} \cdot \operatorname{poly}(n)$, which proves Fact 2.5.

Note that the blocks may be rectangular: y and t are independent variables, so each Θ_b can have different degrees in y and in t, and the two are only tied together at the very end by L.