

A simplified proof of **ETH** is false

Shiva Kintali*

October 26, 2026

Abstract

For every fixed $\varepsilon > 0$, we give a deterministic 3-SAT algorithm on n variables and m clauses using $C_\varepsilon(n + m + 1)^{c_\varepsilon} 2^{\varepsilon n}$ bit operations, refuting ETH. Our simplified exposition of OpenAI's proof [1] combines sparsification, bounded occurrences, spectral replacement, and rational interpolation.

1 The goal and the proof strategy

A Boolean variable takes a value in $\{0, 1\}$, where zero means false and one means true. A literal is a variable or its negation. A clause is satisfied when at least one of its literals is true. A 3-CNF is a conjunction of clauses of *at most* three literals. Given such a formula on n declared variables with m clauses, 3-SAT asks whether some assignment satisfies every clause. An assignment specifies all declared variables, including any that occur in no clause. We allow $n = 0$ and $m = 0$, with the conventions below. Our goal is an algorithm with an arbitrarily small exponential rate.

Theorem 1.1. *For every fixed $\varepsilon > 0$, there are positive constants $C_\varepsilon, c_\varepsilon$ and a deterministic algorithm deciding 3-SAT in at most*

$$C_\varepsilon(n + m + 1)^{c_\varepsilon} 2^{\varepsilon n}$$

bit operations.

We use ETH in its standard positive-rate formulation [2, p. 368]: there is a constant $\delta > 0$ for which no deterministic 3-SAT algorithm runs in $2^{\delta n}(n + m + 1)^{O(1)}$ time. Theorem 1.1 contradicts this assertion by taking, for example, $\varepsilon = \delta/2$. The theorem asserts a family of algorithms, one for each fixed rate parameter ε . This parameter is fixed before the input is given; the algorithm and its polynomial time factor may depend on it.

The main idea. If the variables could be divided into small groups so that every clause used only one group, we could count satisfying assignments separately in each group and multiply the answers. The difficulty is that clauses link different groups.

We remove these links algebraically. First express each clause by a polynomial equal to its truth value on Boolean inputs. In these polynomials, replace each variable occurrence by a linear expression whose coefficients depend polynomially on a parameter T . At $T = 0$ the expression is the original variable. At selected positive integers $T = t$, each deformed clause factor depends on variables in just one group and on a small collection of integer linear measurements (linear sums with integer coefficients). Each measurement uses variables from a single group. Once their values are fixed, both the factors and the constraints realizing those values separate by group, so the group

*Email: shiva.kintali@gmail.com Homepage: shivakintali.github.io

sums can be multiplied. Summing the product of the deformed clause polynomials over all original Boolean assignments defines a polynomial $H(T)$. At zero it equals the desired count. At the other integer nodes its values need not be positive or integral; we use them as interpolation data to recover $H(0)$. Specifically, we compute Taylor coefficients up to an order that is a small fraction of the variable count, at each selected integer, and interpolate.

There are two main tasks. Section 2 shows how such a replacement gives a counting algorithm; its key observation is that a monomial in a low-order Taylor coefficient uses only a few formal variables. Section 3 constructs the replacement using commuting swap matrices and a polynomial that corrects their action on most basis directions. Section 4 chooses the parameters and accounts for exact arithmetic. After the two reductions below, we first deduce the main theorem from the counting proposition. The remaining sections then prove that proposition. We use elementary linear algebra, polynomial interpolation, and basic probability; the Boolean coefficient-extraction identity is derived when it is used. The only result about SAT complexity that we invoke without proof is the sparsification lemma stated below.

Conventions. All logarithms have base two except $\ln$, which is natural. For an integer $k \geq 0$, write $[k] = \{1, \dots, k\}$. Repeated input clauses count separately. An empty conjunction is true; an empty disjunction, including an empty clause, is false. A clause is tautological if it contains both a variable and its negation. After deleting tautological clauses and repeated literals within a clause, the number of occurrences of a variable is the number of clauses containing it. Variables are indexed by $[n]$, and input clauses are listed explicitly using binary variable indices and clause delimiters. Running-time bounds count bit operations and include reading this encoding; intermediate arithmetic-operation counts are explicitly identified. We write $\text{poly}_\theta(u)$ for a polynomial upper bound in u whose constants and degree may depend on the fixed parameter θ ; we use this notation with $\theta = s$ or $\theta = \varepsilon$. Empty products equal one and empty sums equal zero. The coefficient ℓ_1 norm is the sum of the absolute values of the coefficients. This norm will bound the range of our integer measurements and the size of the numbers used in exact arithmetic.

1.1 Reducing to bounded occurrences

The first reduction makes the number of clauses linear in the number of variables. It produces several formulas, and the original formula is satisfiable exactly when at least one of them is satisfiable.

Lemma 1.2 (Decision sparsification [3, Corollary 1, p. 523]). *For every fixed $\eta > 0$, a deterministic algorithm expresses a 3-CNF F on n declared variables as*

$$F \equiv \bigvee_{i=1}^L F_i, \quad 0 \leq L \leq 2^m,$$

where each F_i is a 3-CNF on the same declared variables with at most $C_\eta n$ clauses. The total construction time is $2^m(n + m + 1)^{O_\eta(1)}$. Here C_η depends only on η .

To apply the cited result, first remove tautological clauses, repeated literals within a clause, and duplicate clauses. These operations preserve the satisfying assignments and take polynomial bit time in $n + m + 1$. All declared variables are retained even if they no longer appear. If an empty clause is present, use $L = 0$; if no clauses remain, use $L = 1$ with F_1 the empty conjunction. These conventions also cover $n = 0$. For every other input, the cited construction applies to clauses of width at most three; its polynomial work, including binary encoding costs, is absorbed by the

displayed time bound. The equivalence is an equality of Boolean functions. Disjointness of the solution sets is unnecessary: we only ask whether at least one F_i is satisfiable.

The second reduction limits how many clauses can use the same variable. It preserves satisfiability; two formulas with this relationship are called *equisatisfiable*. We give each occurrence its own copy and enforce agreement with a cycle of implications. Bounded occurrences will later let us balance the formula's incidences among an auxiliary array's positions.

Lemma 1.3 (Bound the occurrences). *A 3-CNF with n declared variables and m clauses can be transformed using polynomially many bit operations in $n + m + 1$ into an equisatisfiable 3-CNF with $N \leq n + 3m$ declared variables in which each variable occurs in at most three clauses. The output has at most $4m$ clauses.*

Proof. First remove tautological clauses and repeated literals within a clause. If an empty clause occurs, return the single empty-clause formula on the same declared variables; all claimed bounds hold because then $m \geq 1$. Otherwise, let m' be the number of remaining clauses, and let d_x count the occurrences of variable x in them. For $d = d_x \geq 2$, replace its occurrences by $x_1 = x, x_2, \dots, x_d$, preserving their signs. All newly introduced copies are fresh variables, distinct across all original variables. Add the cycle

$$\bigwedge_{i=1}^d (\neg x_i \vee x_{i+1}), \quad x_{d+1} := x_1.$$

The cycle is satisfied exactly when all copies agree. If any copy is one, the implications force every copy around the cycle to be one; the all-zero assignment also satisfies the cycle. This includes $d = 2$, where the two clauses give the two directions of an equivalence.

Every satisfying assignment of the original formula extends to a satisfying assignment of the output by giving each copy its original variable's value. Conversely, any satisfying assignment of the output has equal copies, and reading x_1 for each copied variable satisfies every original clause. Thus the formulas are equisatisfiable. The extension is in fact unique. Variables with zero or one occurrence are retained.

Each copy appears in one rewritten original clause and two cycle clauses, so it occurs at most three times. Since $\sum_x d_x \leq 3m' \leq 3m$, the exact variable and clause counts obey

$$N = n + \sum_{x:d_x \geq 2} (d_x - 1) \leq n + 3m, \quad m_{\text{out}} = m' + \sum_{x:d_x \geq 2} d_x \leq 4m.$$

The new clauses have width two, so the width bound remains three. Scanning the clauses, assigning fresh indices, and writing the output all take polynomially many bit operations in $n + m + 1$. $\square$

It remains to solve these bounded-occurrence formulas at arbitrarily small exponential rates. The following statement will be proved without assuming any of the new lemmas of [1].

Proposition 1.4. *For every integer $s \geq 64$, there is a deterministic algorithm that exactly counts the satisfying assignments of a 3-CNF on N declared variables and m input clauses, each variable occurring at most three times after deleting tautological clauses and repeated literals within each clause, in*

$$\text{poly}_s(N + m + 1) 2^{a_s N} \quad \text{bit operations,}$$

where $a_s > 0$ and $a_s = O(\log s/s) \rightarrow 0$. An explicit choice of a_s is given in (4.2).

The parameter s is a fixed integer controlling the exponential rate; it is never the input length. All constants depending on s may be large.

1.2 How the counting proposition implies the main theorem

We allocate half of the allowed exponential rate to sparsification and half to solving its output formulas. The bounded-occurrence reduction enlarges the variable count by a constant factor. We therefore choose the counting parameter s only after that factor is known.

Deduction of Theorem 1.1 from Proposition 1.4. It suffices to consider rational $\varepsilon > 0$: a smaller positive rational target implies the bound for any prescribed real $\varepsilon > 0$. Fix such an ε , and put $\eta = \varepsilon/2$. The case $n = 0$ is decided directly in $O(m + 1)$ bit operations, so assume $n \geq 1$. Apply Lemma 1.2; every resulting formula has $m_i \leq C_\eta n$ clauses. By Lemma 1.3 it has an equisatisfiable bounded-occurrence formula with

$$N_i \leq (1 + 3C_\eta)n =: B_\eta n.$$

The reduced formula has at most $4m_i \leq 4C_\eta n$ clauses. The constant B_η is fixed before selecting the parameter s . Choose $s \geq 64$ sufficiently large that $a_s B_\eta \leq \varepsilon/2$, which is possible because $a_s \rightarrow 0$. Run Proposition 1.4 on each reduced formula, and accept if any count is positive. If there are no output formulas, reject. Including the copying step, each output formula costs at most

$$\text{poly}_\varepsilon(n + 1) 2^{a_s N_i} \leq \text{poly}_\varepsilon(n + 1) 2^{\varepsilon n/2}.$$

There are at most $2^{\eta n} = 2^{\varepsilon n/2}$ output formulas. The total time is therefore bounded by

$$2^{\eta n}(n + m + 1)^{O_\eta(1)} + 2^{\eta n} \text{poly}_\varepsilon(n + 1) 2^{\varepsilon n/2}.$$

Since $\eta = \varepsilon/2$ is fixed, this is at most $C_\varepsilon(n + m + 1)^{c_\varepsilon} 2^{\varepsilon n}$ for suitable positive constants $C_\varepsilon, c_\varepsilon$. All choices and all subsequent computations are deterministic. $\square$

The rest of the paper proves Proposition 1.4.

2 How a suitable deformation gives an algorithm

In this section we assume the replacement forms exist and explain how to use them. We call the replacement a *deformation* because it depends on T : it agrees with the original formula at zero and has a form that we can sum efficiently at the selected positive integers. Section 2.4 follows the entire recovery procedure on a four-variable formula.

2.1 The counting polynomial and the required structure

Let F be a 3-CNF on N declared variables and m input clauses, satisfying the bounded-occurrence promise of Proposition 1.4. Remove tautological clauses and repeated literals within a clause, return zero on an empty clause, and retain all N declared variables. No further variable copying is performed here. If $N = 0$, the remaining formula is the empty conjunction: return one, for its unique empty assignment. This preprocessing takes polynomial bit time in $N + m + 1$. For the rest of the section assume $N \geq 1$. For every unused variable add a dummy factor equal to one with that variable as its scope. A scope records which variable slots belong to a factor, even when the factor is constant. Let $\mathcal{F}$ index the clauses and dummy factors, and let I_e be the scope of factor e . An *incidence* is a pair (e, v) with $v \in I_e$; it is one variable slot in one factor. Write M for the total number of incidences. Then

$$|I_e| \leq 3, \quad |\{e : v \in I_e\}| \leq 3, \quad N \leq M := \sum_{e \in \mathcal{F}} |I_e| \leq 3N. \quad (2.1)$$

The dummy factors leave the product unchanged and give each unused variable one incidence. Thus every variable has between one and three incidences, which proves the bounds on M . All factors have nonempty scopes, so $|\mathcal{F}| \leq M \leq 3N$. For a real clause define its satisfaction polynomial

$$\Phi_e((X_v)_{v \in I_e}) = 1 - \prod_{v \in I_e} (1 - \ell_{e,v}(X_v)),$$

where $\ell_{e,v}(X_v)$ is X_v or $1 - X_v$ according to the sign of the literal. For a dummy factor put $\Phi_e = 1$. On Boolean inputs these polynomials are the clause indicators. For example, the clause $(x_1 \vee \neg x_2)$ has polynomial $1 - (1 - X_1)X_2$, which is zero exactly on its sole falsifying pair $(x_1, x_2) = (0, 1)$. The product of the indicators is one exactly on satisfying assignments, so summing this product counts them:

$$\#F = \sum_{x \in \{0,1\}^N} \prod_{e \in \mathcal{F}} \Phi_e((x_v)_{v \in I_e}). \quad (2.2)$$

Choose an integer $r \geq 2$, positive integers Q, R, D , and a positive rational τ ; their values will be fixed in Section 4. Here r is the number of groups, Q the number of interpolation nodes, and R a degree bound. The parameters τ and D control how many measurements are needed and how many values each can take. For now they are fixed constants; only N grows. For each incidence (e, v) we seek a form homogeneous and linear in X , $L_{e,v}(T; X) = \sum_{i=1}^N p_{e,v,i}(T)X_i$ with $p_{e,v,i} \in \mathbb{Q}[T]$, satisfying

$$L_{e,v}(0; X) = X_v, \quad \deg_T L_{e,v} \leq R.$$

For each integer $t \in [Q]$ we require the following additional structure:

- (i) Disjoint groups $[N] = S_{t,1} \sqcup \cdots \sqcup S_{t,r}$, possibly empty, with $|S_{t,g}| \leq 2N/r$.
- (ii) One assigned group $\sigma_t(e)$ for each factor e .
- (iii) Integer linear measurements on the separate groups, with at most $2\tau N$ measurements in total and coefficient ℓ_1 norm at most D for each measurement.
- (iv) At $T = t$, every $L_{e,v}$ is a rational linear combination of variables in $S_{t,\sigma_t(e)}$ and the measurements. Its coefficients are explicitly available.

The same assigned group must work for *all* inputs of a factor. Measurements are allowed to be redundant or identically zero. A measurement is simply an integer linear sum of bits. Its coefficients can be negative. The point is to enumerate its possible values while leaving the bits that realize each value to be summed within their own group.

Proposition 2.1 (Recovery from a deformation). *Fix r, Q, R, D, τ as above. Suppose the normalized factor system satisfies (2.1). Suppose the coefficient arrays of the rational forms $L_{e,v}$, the groups, the factor assignments, the measurements, and the rational node representations are supplied and satisfy the zero-value and degree conditions and requirements (i)–(iv). Put $q = \lfloor 3NR/Q \rfloor$. Then its exact satisfying-assignment count can be recovered deterministically using at most*

$$\text{poly}(N+1) \left(\sum_{j=0}^{\min(q,N)} \binom{N}{j} 2^j \right) (2D+1)^{2\tau N} 2^{2N/r} \quad (2.3)$$

arithmetic operations over $\mathbb{Q}$. The polynomial's constants and degree may depend on the fixed parameters.

This is a recovery statement from supplied data. For fixed parameters, these arrays have polynomially many entries in N : there are at most $3N$ forms, each with N coefficient polynomials of degree at most R . At each of the fixed number of nodes there are $O(N)$ measurements, and each node representation uses $O(N)$ rational coefficients. Constructing the data is the task of Section 3. The next two subsections prove this proposition. Its bound counts rational operations; coefficient lengths must also be controlled to obtain a bit-operation bound. For the explicit construction in Section 3, this is done in Section 4.1 and Appendix B.

2.2 Summing after the measurements are fixed

Fix t and abbreviate $S_g = S_{t,g}$. Let $x_g = (x_i)_{i \in S_g}$ be the bits in group g , and write $C_g x_g$ for its list of measurements, where C_g is an integer matrix with c_g rows. Every measurement of a Boolean assignment is an integer in $[-D, D]$. Enumerate all

$$y = (y_1, \dots, y_r), \quad y_g \in \{-D, \dots, D\}^{c_g}.$$

A group with no measurements has exactly one empty measurement tuple. An empty group has one empty assignment, and all its measurements are zero. For $g = \sigma_t(e)$, property (iv) supplies rational coefficients such that

$$L_{e,v}(t; x) = \sum_{i \in S_g} \alpha_{e,v,i} x_i + \sum_{h=1}^r \gamma_{e,v,h} \cdot (C_h x_h).$$

Here $\gamma_{e,v,h}$ has c_h coordinates; dependence of these coefficients on the fixed node t is suppressed. Write $\lambda_{e,v}(x_g; y)$ for the right side with $C_h x_h$ replaced by y_h . Thus fixing y leaves a factor assigned to g dependent only on x_g .

This is the reason for enumerating measurements: once y is fixed, there is no remaining variable dependence between different groups. We can sum all factors assigned to one group over that group's assignments, then multiply these group sums.

Work in $\mathbb{Q}[z]/(z^J)$ for an integer $J \geq 1$. Fix corrections $b_{e,v}(z) \in \mathbb{Q}[z]/(z^J)$ that are independent of the assignment x being summed over. Also fix a subset $E \subseteq [N]$ of the bits to one; $E = \emptyset$ imposes no such restriction. Working modulo z^J means keeping only coefficients of $1, z, \dots, z^{J-1}$ after every operation. The corrections and fixed bits anticipate the Taylor calculation below; taking all corrections to be zero and $E = \emptyset$ gives the basic grouped sum. Define

$$A_g(x_g; y, z) = \prod_{e: \sigma_t(e)=g} \Phi_e((\lambda_{e,v}(x_g; y) + b_{e,v}(z))_{v \in I_e}).$$

Distributivity gives the identity

$$\sum_{\substack{x \in \{0,1\}^N \\ x_i=1 \ (i \in E)}} \prod_e \Phi_e((L_{e,v}(t; x) + b_{e,v}(z))_{v \in I_e}) = \sum_y \prod_{g=1}^r \sum_{\substack{x_g \in \{0,1\}^{S_g} \\ C_g x_g = y_g \\ x_i=1 \ (i \in E \cap S_g)}} A_g(x_g; y, z). \quad (2.4)$$

Indeed each full Boolean assignment has a unique measurement tuple, even when measurements are redundant. For a fixed tuple y , the measurement equations and the fixed-bit constraints each involve only one group, so their solution set is a Cartesian product of group solution sets. This proves the identity by finite distributivity in the indicated ring. Infeasible tuples contribute zero; no positivity of the summands is required. The number of tuples is at most $(2D+1)^{2\tau N}$. For each tuple, we compute the group sums separately and then multiply them, enumerating at most

$$\sum_{g=1}^r 2^{|S_g|} \leq r 2^{2N/r}$$

group assignments. Testing the constraints and evaluating the assigned factors use only polynomial work per group assignment: there are $O(N)$ factors and measurements with explicitly supplied coefficients. Suppressing polynomial factors in N and J for fixed r, Q, R, D, τ , the ring-operation count is

$$(2D + 1)^{2\tau N} 2^{2N/r}. \quad (2.5)$$

2.3 Recovering Taylor data without losing the separation

The grouped sum makes the deformed sum accessible at each chosen node. Its degree can be much larger than the number of nodes, so values alone do not suffice in general. We therefore compute several Taylor coefficients at each node. The challenge is to do this while keeping the groups separate.

Which Taylor coefficients suffice? Define the univariate polynomial

$$H(T) = \sum_{x \in \{0,1\}^N} \prod_e \Phi_e((L_{e,v}(T; x))_{v \in I_e}). \quad (2.6)$$

Then $H(0) = \#F$. A clause factor with $|I_e|$ inputs has T -degree at most $|I_e|R$, and a dummy factor has degree zero. Products add these degree bounds, and summing over x cannot increase them. Hence $\deg H \leq MR \leq 3NR$. Put

$$q = \left\lfloor \frac{3NR}{Q} \right\rfloor, \quad J = q + 1. \quad (2.7)$$

The polynomial $H(t + z) \bmod z^J$ records the first J Taylor coefficients at t : the coefficient of z^j is $H^{(j)}(t)/j!$, for $0 \leq j < J$. These records, for $t = 1, \dots, Q$, determine H uniquely. Indeed, if two polynomials of degree less than QJ have the same records, their difference is divisible by $(T - t)^J$ for each node t . The nodes are distinct, so their difference is divisible by $\prod_{t=1}^Q (T - t)^J$. This product has degree QJ , forcing the difference to be zero. Since $\deg H \leq 3NR < QJ$, this applies to H . There are QJ coefficient unknowns and QJ linear conditions, so this uniqueness proves that the square interpolation matrix is invertible. Whenever a sharper degree bound is available, any positive integer J with $QJ > \deg H$ suffices by the same argument. Solving for a polynomial from these records is called *Hermite interpolation*; Appendix B gives its exact implementation.

Why low Taylor order gives small supports. Fix a node t and a Boolean assignment x . We keep the base inputs $L_{e,v}(t; x)$ fixed and put their changes into a *separate* formal vector X :

$$G_x(X, z) = \prod_e \Phi_e((L_{e,v}(t; x) + L_{e,v}(t + z; X) - L_{e,v}(t; X))_{v \in I_e}).$$

The base inputs retain the structure required by the grouped sum. Each correction $L_{e,v}(t + z; X) - L_{e,v}(t; X)$ is linear in X and divisible by z . Substituting $X = x$ restores the full shifted inputs, so $H(t + z) = \sum_x G_x(x, z)$.

In an expanded term, choosing d formal-variable factors from the corrections supplies at least z^d . Thus a coefficient of z^j cannot have total X -degree greater than j :

$$\deg_X[z^j]G_x(X, z) \leq j. \quad (2.8)$$

Here $[z^j]$ means the coefficient of z^j . Reduce G_x modulo z^J and replace every positive power of each X_i by X_i . For example, $X_1^3 X_2$ becomes $X_1 X_2$; the two expressions agree on every Boolean

vector. This operation is called *Boolean multilinearization*. It preserves Boolean evaluations and cannot increase the number of distinct variables in a monomial. The set of indices of those variables is the monomial's *support*. Since we keep only powers of z below $J = q + 1$, (2.8) shows that the multilinearized polynomial has the unique expansion

$$\sum_{\substack{E \subseteq [N] \\ |E| \leq q}} a_E(x, z) \prod_{i \in E} X_i.$$

Here $a_E(x, z) \in \mathbb{Q}[z]/(z^J)$. These coefficients may still depend on the fixed assignment x ; only the separate formal variables X have been multilinearized. We need coefficients indexed only by supports of size at most q .

Extracting a coefficient from Boolean values. For a two-variable multilinear polynomial $p(X_1, X_2) = b_0 + b_1 X_1 + b_2 X_2 + b_{12} X_1 X_2$, we have

$$b_{12} = p(1, 1) - p(1, 0) - p(0, 1) + p(0, 0).$$

The constant and the two linear terms cancel. The same cancellation works for any number of variables, including when the coefficients are themselves truncated polynomials in z . For any $U \subseteq [N]$, let $\mathbf{1}_U$ be its Boolean indicator vector. Boolean inclusion–exclusion extracts each coefficient:

$$a_E(x, z) = \sum_{U \subseteq E} (-1)^{|E| - |U|} G_x(\mathbf{1}_U, z) \pmod{z^J}.$$

To see this, Boolean evaluation agrees with the multilinearized polynomial, so $G_x(\mathbf{1}_U, z) \equiv \sum_{A \subseteq U} a_A(x, z)$ modulo z^J . If $A \subseteq E$, its coefficient in the displayed alternating sum is $\sum_{A \subseteq U \subseteq E} (-1)^{|E| - |U|}$, equal to one for $A = E$ and zero otherwise. If $A \not\subseteq E$, it never appears. This also covers $E = \emptyset$.

Returning to the count. For Boolean x , the monomial $\prod_{i \in E} x_i$ is one exactly when all bits indexed by E are one. Substitute $X = x$ in the multilinear expansion and use the extracted coefficients. Summing over x gives

$$H(t + z) \equiv \sum_{\substack{E \subseteq [N] \\ |E| \leq q}} \sum_{U \subseteq E} (-1)^{|E| - |U|} \sum_{\substack{x \in \{0,1\}^N \\ x_i = 1 \ (i \in E)}} G_x(\mathbf{1}_U, z) \pmod{z^J}. \quad (2.9)$$

The inner sum has exactly the form (2.4), with fixed corrections $b_{e,v}(z) = L_{e,v}(t + z; \mathbf{1}_U) - L_{e,v}(t; \mathbf{1}_U)$. The set U fixes the formal correction inputs; the set E fixes bits of the assignment being summed over. Their roles are distinct. For fixed t and U , these corrections are the same for every assignment x , which is precisely what the grouped sum requires.

Conclusion of the proof of Proposition 2.1. The algorithm never constructs the multivariate expansion above. It enumerates (E, U) and performs truncated *univariate* arithmetic using (2.4). There are $\sum_{j=0}^{\min(q, N)} \binom{N}{j} 2^j$ pairs (E, U) , and each contributes the cost in (2.5). This formula also covers $q = 0$, with just the pair $(\emptyset, \emptyset)$, and $q \geq N$, when all supports are enumerated. For each pair the correction polynomials can be computed from the supplied arrays using polynomially many rational operations. There are a fixed number Q of nodes, and $J = O(N)$ coefficients per truncated polynomial. Each truncated operation therefore costs polynomially many rational operations. Hermite interpolation uses a linear system of order $QJ = O(N)$, so it also adds only a polynomial number of rational operations. These observations give (2.3) and recover $H(0) = \#F$. $\square$

The three nonpolynomial factors have separate origins: the choices of (E, U) , the measurement tuples, and the assignments within one group. In Section 4 we make all three exponential rates small at the same time.

2.4 A complete four-variable example

We now follow the same formula through the whole recovery procedure:

$$F = (x_1 \vee x_3) \wedge (x_2 \vee \neg x_4).$$

Its clause indicators are

$$\Phi_1(X_1, X_3) = X_1 + (1 - X_1)X_3, \quad \Phi_2(X_2, X_4) = 1 - (1 - X_2)X_4.$$

Each clause has three satisfying assignments on its own two variables, so the answer is $3 \cdot 3 = 9$. The example illustrates recovery, without claiming an asymptotic saving. Use groups $S_1 = \{1, 2\}$, $S_2 = \{3, 4\}$, the single measurement $y = x_3 + x_4$ on S_2 , and forms

$$\begin{aligned} L_1(T; X) &= X_1, & L_2(T; X) &= X_2, \\ L_3(T; X) &= X_3 + TX_4, & L_4(T; X) &= X_4 + TX_3. \end{aligned}$$

The incidence labels are suppressed because each variable occurs once. At $T = 0$ these forms give the original variables. At $t = 1$ both clauses are assigned to S_1 , because their inputs depend only on x_1, x_2, y . For fixed y , put

$$A = x_1 + (1 - x_1)y, \quad B = 1 - (1 - x_2)y.$$

Here y replaces the measurement $x_3 + x_4$; when we later fix a full assignment x , it always has that assignment's measurement value. The group-1 sum is $\sum_{x_1, x_2} AB = (1 + y)(2 - y)$. Group 2 has no assigned factors, so its sum just counts assignments with $x_3 + x_4 = y$. Only $y = 0, 1, 2$ have feasible assignments:

y	0	1	2	$H(1) = 2 + 4 + 0 = 6.$
group-1 sum	2	2	0	
group-2 sum	1	2	1	
product	2	4	0	

For instance, at $y = 2$ the factor B takes values -1 and 1 ; these weighted summands can cancel.

Each deformed clause factor is affine in T , so their product and $H(T)$ have degree at most two. Thus a single node $t = 1$ and the three Taylor coefficients modulo z^3 suffice here; we use this exact degree bound instead of the general upper bound $3NR$. Consequently only supports of size at most two are needed in this example. Keep the assignment x in the base inputs and the formal vector X in the corrections. The shifted product is

$$\begin{aligned} G_x(X, z) &= (A + z(1 - x_1)X_4)(B - z(1 - x_2)X_3) \\ &= AB - zA(1 - x_2)X_3 + z(1 - x_1)BX_4 - z^2(1 - x_1)(1 - x_2)X_3X_4. \end{aligned}$$

It is already multilinear in X . For example, inclusion-exclusion recovers its last coefficient from four fixed correction inputs:

$$a_{\{3,4\}}(x, z) = G_x(\mathbf{1}_{\{3,4\}}, z) - G_x(\mathbf{1}_{\{3\}}, z) - G_x(\mathbf{1}_{\{4\}}, z) + G_x(0, z) = -z^2(1 - x_1)(1 - x_2).$$

Substituting $X = x$ requires this coefficient to be summed only over assignments with $x_3 = x_4 = 1$. For those assignments $y = 2$, and its sum over x_1, x_2 is $-z^2$. The other supports are handled in the same way. Their contributions, split by measurement value, are

E	$y = 0$	$y = 1$	$y = 2$	$\sum_{x: x_i=1} a_E(x, z)$
$\emptyset$	2	4	0	6
$\{3\}$	0	$-2z$	$-3z$	$-5z$
$\{4\}$	0	z	0	z
$\{3, 4\}$	0	0	$-z^2$	$-z^2$

For example, when $E = \{3\}$, exactly one group-2 assignment is feasible at each of $y = 1, 2$, and the group-1 sum is $-z(1 + y)$. All other supports have zero coefficient. Adding the four rows gives $H(1 + z) = 6 - 4z - z^2$. Because $\deg H \leq 2$, no higher terms have been discarded: this is an exact polynomial identity. We may therefore substitute $z = -1$ to recover $H(0) = 6 + 4 - 1 = 9$, the satisfying-assignment count.

3 Constructing the deformation

This section supplies the structure used above. The construction has a constant-size algebraic part, depending on the chosen parameters, and a polynomial-time placement of the actual variables.

Proposition 3.1 (Construction of a suitable deformation). *Let $r \geq 2$, $R \geq 32$, and $Q \geq 6R$ be integers, let $\tau > 0$ be rational, and put $D = r^{2Q}$. Consider the normalized factor system of Section 2, with nonempty scopes satisfying (2.1). Suppose*

$$12D^2 \leq M, \quad 2Qr^2 \leq N,$$

and

$$2e^{-R^2/(8Q)} \leq \frac{\tau}{3r}. \quad (3.1)$$

Then one can deterministically construct forms $L_{e,v}$ homogeneous and linear in X , with rational polynomial coefficients satisfying $L_{e,v}(0; X) = X_v$ and $\deg_T L_{e,v} \leq R - 1$, together with groups, factor assignments, integer measurements, and rational node representations satisfying requirements (i)–(iv) of Section 2. For fixed r, Q, R, τ , the construction uses polynomially many bit operations in $N + 1$.

The proof has three stages. First we construct an operator whose error can be described by a few measurements. Next we turn its limited pattern of disagreements into low-degree forms. Finally we place the formula's incidences in balanced copies of the auxiliary array and restrict every measurement to individual groups.

3.1 An auxiliary array and a low-rank correction

We first build replacement forms on an auxiliary array, without reference to the input formula. At each node, every array position has two labels: the group containing its variable and the group assigned to its form. Swapping the labels gives perfect group matching. Our construction modifies this swap so that it also preserves most linear directions; the few remaining directions will be supplied by measurements.

Let

$$\mathcal{B} = ([r] \times [r])^Q, \quad D = |\mathcal{B}| = r^{2Q}.$$

Write $u \in \mathcal{B}$ as $((a_u(t), d_u(t)))_{t=1}^Q$. At node t , a coordinate w belongs to group $a_w(t)$, whereas the form associated with u is assigned to group $d_u(t)$. Let $\zeta = (\zeta_w)_{w \in \mathcal{B}}$ be formal coordinates.

For $t \in [Q]$ let $s_t(u)$ be the label obtained from u by swapping the two entries in pair t . Define the permutation matrix Π_t by $(\Pi_t \zeta)_u = \zeta_{s_t(u)}$. These symmetric matrices commute and satisfy $\Pi_t^2 = I$. Define

$$V = \prod_{t=1}^Q \Pi_t, \quad W = \frac{1}{2} \sum_{t=1}^Q (I - \Pi_t).$$

The full swap V has the desired group matching: a nonzero entry $V[u, w]$ requires $a_w(t) = d_u(t)$ at every node. However, $V\zeta$ need not equal ζ , so it cannot by itself recover the original variables at $T = 0$. We will use $A = Vf(W)$ for a low-degree polynomial f . Two properties will make this useful: $I - A$ can be recovered from a few measurements, and $A[u, w] \neq 0$ allows disagreement at only a few nodes. The first property comes from eigenvalues; the second will come from the degree of f .

Let e_{ab} denote the coordinate unit vector indexed by $(a, b) \in [r]^2$. For a single pair, an orthogonal eigenbasis for the swap consists of

$$e_{aa} \ (a \in [r]), \quad e_{ab} + e_{ba} \ (a < b), \quad e_{ab} - e_{ba} \ (a < b).$$

The first two types give $r(r+1)/2$ vectors of eigenvalue $+1$; the third gives $r(r-1)/2$ vectors of eigenvalue -1 . Choose a one-pair basis vector v_t independently for each node t . The corresponding product vector has coordinates

$$\psi_u = \prod_{t=1}^Q v_t(a_u(t), d_u(t)).$$

Inner products of these vectors factor into the one-pair inner products, so the D choices give an orthogonal basis Ψ of $\mathbb{Q}^{\mathcal{B}}$. This is the tensor-product basis; its entries all lie in $\{0, 1, -1\}$. We use the usual coordinate inner product throughout. If ψ has $j(\psi)$ negative factors, then

$$W\psi = j(\psi)\psi, \quad V\psi = (-1)^{j(\psi)}\psi.$$

Indeed each summand $(I - \Pi_t)/2$ contributes one on a negative factor and zero on a positive factor, while V multiplies all the signs. For a uniformly chosen basis vector, $j(\psi)$ has distribution $\text{Bin}(Q, p)$ with $p = (r-1)/(2r)$. This probability statement just counts basis vectors: choose one of the r^2 one-pair basis vectors independently at each of the Q positions. The fraction with a given number of negative factors is the corresponding binomial probability.

To make A act as the identity on a vector with j negative factors, we need $f(j) = (-1)^j$. Interpolating at every $j \in \{0, \dots, Q\}$ would require too much degree. Instead, we interpolate only at R consecutive integers near the mean. A binomial tail bound will tell us when this interval contains most basis vectors. Choosing R, Q accordingly will keep the degree below R while leaving few exceptional vectors.

Assume $R \geq 32$ and $6R \leq Q$. Put $\mu = Qp$, and let

$$b = \left\lfloor \mu - \frac{R-1}{2} \right\rfloor, \quad I_0 = \{b, b+1, \dots, b+R-1\}.$$

Since $r \geq 2$, we have $Q/4 \leq \mu \leq Q/2$. The inequalities $6R \leq Q$ and $R \geq 32$ imply $b \geq 0$ and $b+R-1 \leq Q$: indeed, $\mu - (R-1)/2 \geq R+1/2$ and $\mu + (R-1)/2 \leq Q/2 + (R-1)/2 < Q$. Moreover,

$$b \leq \mu - (R-1)/2 < \mu - R/4, \quad b+R-1 > \mu + (R-3)/2 > \mu + R/4.$$

Thus I_0 contains every integer at distance less than $R/4$ from μ . Let $f \in \mathbb{Q}[Y]$ be the unique polynomial of degree at most $R - 1$ with

$$f(j) = (-1)^j \quad (j \in I_0).$$

For example, its coefficients are obtained explicitly from

$$f(Y) = \sum_{j \in I_0} (-1)^j \prod_{\substack{h \in I_0 \\ h \neq j}} \frac{Y - h}{j - h}.$$

The matrix $A = Vf(W)$ therefore fixes every ψ with $j(\psi) \in I_0$.

Let $\Psi_{\text{bad}} = \{\psi \in \Psi : j(\psi) \notin I_0\}$, let $c = |\Psi_{\text{bad}}|$, and form a matrix C whose rows are $\psi^\top$ for $\psi \in \Psi_{\text{bad}}$.

Lemma 3.2 (Exceptional spectral directions). *The exceptional set satisfies*

$$c \leq 2D \exp(-R^2/(8Q)). \quad (3.2)$$

The correction $(I - A)\zeta$ is determined by the c measurements $C\zeta$, with the explicit coefficients in (3.3) below. Each measurement has integer coefficients and coefficient ℓ_1 norm at most D .

Proof. The exceptional indices lie outside I_0 , hence at distance at least $R/4$ from the mean. Their fraction is bounded by the binomial estimate $\Pr(|Y - \mathbb{E}Y| \geq h) \leq 2e^{-2h^2/Q}$ with $h = R/4$; its elementary derivation is in Appendix A.

For an orthogonal basis, the expansion of a vector is

$$\zeta = \sum_{\psi \in \Psi} \frac{\langle \psi, \zeta \rangle}{\langle \psi, \psi \rangle} \psi.$$

The operator $I - A$ vanishes on all the nonexceptional basis vectors. Applying it to this expansion therefore leaves only the c exceptional terms. In particular, $\text{rank}(I - A) \leq c$, and the correction can be specified by the c scalar measurements $\langle \psi, \zeta \rangle$ with $\psi \in \Psi_{\text{bad}}$. Coordinate by coordinate, the resulting identity is

$$((I - A)\zeta)_u = \sum_{\psi \in \Psi_{\text{bad}}} \beta_{u,\psi} \langle \psi, \zeta \rangle, \quad \beta_{u,\psi} = \frac{(1 - (-1)^{j(\psi)} f(j(\psi))) \psi_u}{\langle \psi, \psi \rangle}. \quad (3.3)$$

The denominator $\langle \psi, \psi \rangle$ is a positive integer; in fact it is a power of two, since the one-pair basis vectors have squared norm one or two. Thus every $\beta_{u,\psi}$ is an explicitly computable rational number. No row-space computation is required. Each measurement $\langle \psi, \zeta \rangle$ has integer coefficients and coefficient ℓ_1 norm at most D , since ψ has D entries in $\{0, 1, -1\}$. $\square$

3.2 Turn sparse disagreement into low degree

We now use the support of A to build forms that recover the original variables at zero and become local at each node. Each disagreement will cost one linear factor in T , so a small number of disagreements gives a small degree.

For $u, w \in \mathcal{B}$ let $E_{u,w} = \{t \in [Q] : a_w(t) \neq d_u(t)\}$.

Lemma 3.3 (Sparse disagreement). *Every nonzero entry of A disagrees with its assigned group at fewer than R nodes:*

$$A[u, w] \neq 0 \implies |E_{u,w}| \leq R - 1. \quad (3.4)$$

Proof. Since f has degree at most $R - 1$, expansion of $Vf(W)$ writes A as a rational linear combination of matrices

$$\prod_{t \notin S} \Pi_t, \quad |S| \leq R - 1.$$

Indeed a degree- j term in W is a sum of products of at most j swaps, with repeated swaps canceling. Multiplication by V leaves every pair outside some such S swapped. A nonzero entry of this permutation obeys $a_w(t) = d_u(t)$ for $t \notin S$. If $A[u, w] \neq 0$, at least one permutation in this expansion has a nonzero (u, w) entry, so its set S contains every disagreement. This proves (3.4). $\square$

The basic cancellation is visible in $(\Pi_1 \Pi_2 \Pi_3) \Pi_2 = \Pi_1 \Pi_3$: only the second of these three nodes loses its swap. Applying this cancellation to V after multiplying it by at most j swaps can lose group matching at at most j nodes.

For each nonzero entry define

$$\phi_{u,w}(T) = \prod_{t \in E_{u,w}} (1 - T/t),$$

so that $\phi_{u,w}(0) = 1$ and $\phi_{u,w}(t) = 0$ at every disagreement node. For instance, disagreement at nodes 2 and 5 gives $(1 - T/2)(1 - T/5)$, a degree-two polynomial that preserves the coefficient at zero and removes it at those two nodes. Define an auxiliary form

$$P_u(T; \zeta) = \sum_{\psi \in \Psi_{\text{bad}}} \beta_{u,\psi} \langle \psi, \zeta \rangle + \sum_{\substack{w \in \mathcal{B} \\ A[u,w] \neq 0}} A[u, w] \phi_{u,w}(T) \zeta_w. \quad (3.5)$$

At $T = 0$ the two sums are the u th coordinates of $(I - A)\zeta$ and $A\zeta$, so $P_u(0; \zeta) = \zeta_u$. Its degree is at most $R - 1$. At a node t , the second sum uses only coordinates w with $a_w(t) = d_u(t)$: all other coefficients contain the vanishing factor $1 - T/t$. Thus $P_u(t; \zeta)$ can be evaluated from the variables in its assigned group and the c measurements $C\zeta$.

3.3 Place the formula in balanced copies of the array

It remains to put the actual variable occurrences into the auxiliary array. At a node t , the variable's schedule will specify its group, and the factor's schedule will specify which group is responsible for that factor. The two schedules together give exactly the pair labels used above.

Choose a schedule $a_v \in [r]^Q$ for each variable and a schedule $d_e \in [r]^Q$ for each factor. Give incidence (e, v) the label $u(e, v) = (a_v, d_e) \in \mathcal{B}$. For $u \in \mathcal{B}$ let c_u be the number of incidences with this label, and set $n_{t,g} = |\{v : a_v(t) = g\}|$. All incidences of a variable share its schedule a_v , and all inputs of a factor share its schedule d_e . These two consistencies will preserve the original variables and the common assigned group of each factor.

We need two kinds of balance. No variable group should be too large, because we enumerate its assignments. No array label should receive too many incidences, because each extra copy of the array needs another set of exceptional measurements. The next lemma achieves both at once.

Lemma 3.4. *If $12D^2 \leq M$ and $2Qr^2 \leq N$, the schedules can be constructed deterministically in polynomial time, for fixed r, Q , so that*

$$c_u \leq 2M/D \quad (u \in \mathcal{B}), \quad n_{t,g} \leq 2N/r \quad (t \in [Q], g \in [r]).$$

Proof. Initially let all schedules be independent and uniform. Define

$$\Phi = \sum_u (c_u - M/D)^2, \quad \Xi = \sum_{t=1}^Q \sum_{g=1}^r (n_{t,g} - N/r)^2.$$

These are sums of squared deviations from the two desired averages. We first bound their expectations and then fix every schedule while preserving a bound on their normalized sum. Writing $\mathcal{A} = \{(e, v) : v \in I_e\}$, expansion gives

$$\Phi = \sum_{(e,v),(e',v') \in \mathcal{A}} \mathbf{1}\{(a_v, d_e) = (a_{v'}, d_{e'})\} - M^2/D.$$

If two incidences share neither a variable nor a factor, their labels are independent and their equality probability is $1/D$. For a fixed incidence, at most three incidences share its variable and at most three share its factor. Thus at most six incidences, including itself, can violate this independence condition. Bound the equality probabilities of these pairs by one; the other pairs cancel the corresponding terms in M^2/D . This gives $\mathbb{E}\Phi \leq 6M$.

For fixed t, g , the count $n_{t,g}$ is a sum of N independent indicators, each with probability $1/r$. It has mean N/r and variance $N(1/r)(1 - 1/r)$. Summing this variance over all Qr choices of t, g gives $\mathbb{E}\Xi = QN(1 - 1/r) \leq QN$. For the exact computation below, the analogous collision expansion is

$$\Xi = \sum_{t=1}^Q \sum_{v,v' \in [N]} \mathbf{1}\{a_v(t) = a_{v'}(t)\} - QN^2/r.$$

Both collision sums use ordered pairs, including equal pairs.

Fix the schedules in a fixed order, choosing the lexicographically first value that does not increase the conditional expectation of $\Phi/(6M) + \Xi/(QN)$. Such a value exists because the current expectation is the average over all values of the next schedule. The final sum is at most two. Both potentials are nonnegative, so $\Phi \leq 12M$ and $\Xi \leq 2QN$. Each individual squared deviation is bounded by its whole potential; taking square roots and using the lemma's two size assumptions gives

$$c_u \leq M/D + \sqrt{12M} \leq 2M/D, \quad n_{t,g} \leq N/r + \sqrt{2QN} \leq 2N/r.$$

Thus the random choice is used only to prove that a good next value exists; the schedule is actually constructed by exact comparisons of conditional expectations. This is an effective polynomial-time construction. There are $K = r^Q$ possible values for each schedule, a constant here. Each indicator in the collision expansion of Φ depends on at most four distinct schedules, and each in that of Ξ on at most two. Conditioned on the schedules already fixed, the remaining schedules are still independent and uniform. For each indicator, enumerate the choices of its distinct unassigned schedules: at most K^4 choices suffice to compute its exact expectation. Summing $M^2 + QN^2$ terms, trying all K values, and repeating for $N + |\mathcal{F}| \leq N + M = O(N)$ schedules takes polynomially many operations on rationals with $O_{r,Q}(\log(N+1))$ bits. To see the bit bound, every unnormalized indicator expectation has denominator dividing the fixed integer K^4 . After adding the constant terms and normalizing the potentials, a common denominator divides $6MQNK^4$. Using this common denominator keeps all numerators and denominators to $O_{r,Q}(\log(N+1))$ bits, since $M \leq 3N$. $\square$

Think of $h = \max_u c_u$ copies of the array as h layers, with one slot per label in each layer. There are enough layers to give every incidence with the same label a distinct slot. Take these h copies of $\mathcal{B}$. Place each incidence (e, v) at coordinate $u(e, v)$ in a different copy from the other incidences

with that label. Put the original variable X_v at that position and zero at unused positions. Write $\zeta^{(j)}(X)$ for copy j . Let $O_j \subseteq \mathcal{B}$ be its occupied coordinates, and let $\nu_j(w)$ be the variable index stored at $w \in O_j$, so $\zeta_w^{(j)}(X) = X_{\nu_j(w)}$. There are at most $hD \leq 2M \leq 6N$ positions in total. These arrays introduce *no new Boolean variables*: repeated copies of a symbol X_v are evaluated using the same bit.

If incidence (e, v) was placed in copy $j(e, v)$, set

$$L_{e,v}(T; X) = P_{u(e,v)}(T; \zeta^{(j(e,v))}(X)).$$

It has degree at most $R - 1$ and equals X_v at zero.

Figure 1 follows one incidence through this substitution. The slot labels organize appearances of variables; the vector X still contains exactly the original N variables.

$$\boxed{\text{incidence } (e, v)} \xrightarrow{\text{label and layer}} \boxed{u = (a_v, d_e) \text{ in layer } j(e, v)}$$

$$\zeta_u^{(j(e,v))}(X) = X_v \implies L_{e,v}(T; X) = P_u(T; \zeta^{(j(e,v))}(X)).$$

Another occupied slot	What its stored symbol means
$w = (a_{v'}, d_{e'})$ in the same layer $j = j(e, v)$	Its entry is $\zeta_w^{(j)}(X) = X_{v'}$. It can contribute to the local part of $P_u(t; \zeta^{(j)}(X))$ only if $a_{v'}(t) = d_e(t)$.
$u' = (a_v, d_{e''})$ in a layer k	Another occurrence of v stores $\zeta_{u'}^{(k)}(X) = X_v$. Both entries use the same bit when $X = x$.

Figure 1: From an occurrence to its replacement form. Array positions store original symbols. Locality is determined by the first schedule of the stored variable and the second schedule of the receiving factor.

At node t use the actual groups $S_{t,g} = \{v : a_v(t) = g\}$ and assign factor e to $d_e(t)$. Every coordinate in the local part of its auxiliary form contains either zero or a variable in that assigned group: an occupied coordinate w contains a variable $X_{v'}$ with first schedule $a_{v'} = a_w$, and locality at node t requires $a_w(t) = d_e(t)$.

There are hc auxiliary measurements $\langle \psi, \zeta^{(j)}(X) \rangle$, indexed by $j \in [h]$ and $\psi \in \Psi_{\text{bad}}$. At each fixed node t , a measurement may involve several actual groups, so split it into its r restrictions:

$$m_{t,j,\psi,g}(X) = \sum_{\substack{w \in O_j \\ a_w(t)=g}} \psi_w X_{\nu_j(w)} \quad (g \in [r]).$$

Each summand uses a variable in $S_{t,g}$, because an occupied coordinate satisfies $a_w = a_{\nu_j(w)}$. Moreover, $\langle \psi, \zeta^{(j)}(X) \rangle = \sum_{g=1}^r m_{t,j,\psi,g}(X)$. This splitting is essential: fixing a sum involving two groups would otherwise leave those groups coupled. For example, if x_1 and x_2 are in different groups, fixing $x_1 + x_2 = 1$ leaves the two possibilities $(0, 1)$ and $(1, 0)$; the choices are not independent. Fixing each group's contribution separately restores the product structure used in (2.4). Each restricted measurement still has integer coefficients and coefficient ℓ_1 norm at most D . Indeed, repeated occurrences of the same original variable add their coefficients, and the triangle inequality shows that this aggregation cannot increase the sum of absolute values. The hypothesis (3.1), together with (3.2), gives $c \leq D\tau/(3r)$. At each node, the number of restricted measurements, including any zero or redundant ones, therefore satisfies

$$rhc \leq rh \frac{D\tau}{3r} = \frac{hD\tau}{3} \leq 2\tau N.$$

For an incidence (e, v) , abbreviate $u = u(e, v)$ and $j = j(e, v)$. Its explicit node representation is

$$L_{e,v}(t; X) = \sum_{\psi \in \Psi_{\text{bad}}} \beta_{u,\psi} \sum_{g=1}^r m_{t,j,\psi,g}(X) \\ + \sum_{\substack{w \in O_j, A[u,w] \neq 0 \\ a_w(t) = d_e(t)}} A[u, w] \phi_{u,w}(t) X_{\nu_j(w)}.$$

The first sum uses the supplied measurements; the second uses only variables in the factor's assigned group. Every restriction of a given measurement has the same coefficient $\beta_{u,\psi}$. Together with the group-size bound, the measurement count, and the coefficient-norm bound, this verifies all four requirements in Section 2.

All auxiliary arrays are fixed rational data generated by the displayed formulas. Lemma 3.4 supplies the schedules in polynomial bit time. For each original variable, substitution into an auxiliary form sums at most D fixed coefficient polynomials. Their coefficient lengths therefore remain bounded in terms of the parameters, as do the node coefficients; measurement coefficients are integers of absolute value at most D . There are $O(N)$ forms and $O(N)$ measurements per node, each using at most N variables. Their dense coefficient arrays and node representations have polynomial size and can be constructed in polynomial bit time, including the $O(\log(N+1))$ -bit variable indices. This proves Proposition 3.1.

4 Fixed parameters and the complete running time

We work with the normalized factor system of Section 2. Its empty-clause and $N = 0$ cases have already been settled, so assume $N \geq 1$. The parameters must do two jobs at once. The Taylor cutoff satisfies $q/N \leq 3R/Q$, so R/Q should be small. The exceptional fraction is at most $2e^{-R^2/(8Q)}$, so R^2/Q should be large. Taking $Q = s^4$ and $R = s^3$ achieves both: the two ratios are $1/s$ and s^2 . We also increase the number of groups and decrease the measurement budget as s grows.

Fix an integer $s \geq 64$, and choose

$$r = s, \quad Q = s^4, \quad R = s^3, \quad \tau = s^{-6}, \quad D = r^{2Q} = s^{2s^4}. \quad (4.1)$$

These are constants throughout the algorithm on N variables. The group-enumeration rate is $2/r = 2/s$, and the measurement-enumeration rate will be $2\tau \log_2(2D+1) = O(\log s/s^2)$. We have $R \geq 32$, $6R \leq Q$, and

$$2e^{-R^2/(8Q)} = 2e^{-s^2/8} \leq \frac{1}{3s^7} = \frac{\tau}{3r}.$$

Indeed, one term of the exponential series gives

$$e^{s^2/8} \geq \frac{(s^2/8)^7}{7!} \geq 6s^7,$$

where the last inequality follows from $s^7 \geq 64^7 > 6 \cdot 8^7 \cdot 7!$. Taking reciprocals proves the claim for every $s \geq 64$. The construction is therefore valid whenever

$$N \geq N_0(s) := \max\{12D^2, 2Qr^2\},$$

because $M \geq N$. For $N < N_0(s)$ we count directly by exhaustive search. Its cost is at most $2^{N_0(s)} \text{poly}(N+m+1)$, so this fallback can be absorbed into a multiplicative constant depending on s .

For $N \geq N_0(s)$, (2.7) gives $q = \lfloor 3N/s \rfloor$. Thus the sets E that occur in Taylor recovery contain at most a $3/s$ fraction of the variables. Binary entropy bounds the number of such sets. Write

$$H_2(p) = -p \log_2 p - (1-p) \log_2(1-p) \quad (0 < p < 1), \quad H_2(0) = H_2(1) = 0.$$

The endpoint values are the continuous extensions. For $0 < j < N$, put $p_j = j/N$. The j th term in the binomial expansion of $1 = (p_j + (1-p_j))^N$ is at most one, so

$$\binom{N}{j} p_j^j (1-p_j)^{N-j} \leq 1, \quad \binom{N}{j} \leq 2^{NH_2(j/N)}.$$

The endpoint cases $j = 0, N$ are equalities. Since $H_2'(p) = \log_2((1-p)/p) \geq 0$ for $0 < p \leq 1/2$, it follows that

$$\sum_{j=0}^{\lfloor pN \rfloor} \binom{N}{j} \leq (N+1) 2^{NH_2(p)} \quad (0 < p \leq 1/2).$$

Apply this with $p = 3/s < 1/2$. Since $q = \lfloor pN \rfloor \leq N$ and $2^j \leq 2^{pN}$ for $j \leq q$, we obtain

$$\sum_{j=0}^q \binom{N}{j} 2^j \leq (N+1) 2^{N(H_2(3/s) + 3/s)}.$$

Table 1 collects the three costs in Proposition 2.1. These costs count rational arithmetic operations. A rate a means a contribution of at most $\text{poly}_s(N+1)2^{aN}$ to that count. The first rate is controlled by R/Q , the second by $\tau \log_2(2D+1)$, and the third by $1/r$.

What is enumerated	Purpose and controlling quantity	Rate for (4.1)
Supports E and subsets $U \subseteq E$	Recover Taylor coefficients; $q/N \leq 3R/Q = 3/s$	$H_2(3/s) + 3/s$
Measurement tuples y	Separate groups; at most $2\tau N$ measurements, each in $[-D, D]$	$2\tau \log_2(2D+1) = O(\log s/s^2)$
Assignments in one group	Evaluate each group sum; $ S_{t,g} \leq 2N/r$	$2/r = 2/s$

Table 1: The three exponential costs. Their rates add because the enumerations are nested. The number of nodes and all remaining work contribute only polynomial factors for fixed s .

Adding the three rates in (2.3) gives the explicit choice promised in Proposition 1.4:

$$a_s = H_2(3/s) + \frac{5}{s} + 2s^{-6} \log_2(2s^{2s^4} + 1). \quad (4.2)$$

This number is positive, since its summands are nonnegative and $5/s > 0$. The elementary inequality $-(1-p) \ln(1-p) \leq p$ gives $H_2(p) \leq p \log_2(e/p)$ for $0 < p < 1$. Also, $2D+1 \leq 4D$ yields the explicit estimate

$$2s^{-6} \log_2(2s^{2s^4} + 1) \leq \frac{4}{s^6} + \frac{4 \log_2 s}{s^2} = O(\log s/s^2).$$

Applying the entropy bound with $p = 3/s$ now gives $a_s = O(\log s/s) \rightarrow 0$. It remains to justify that exact arithmetic adds only a polynomial factor.

4.1 Exact implementation and bit complexity

So far we have counted arithmetic operations. To obtain a bit-operation bound, we must also control the size of the integers and rationals in those operations. There are two stages: evaluating the grouped sums and then solving the interpolation system. The first uses numbers with $O_s(N)$ bits; the second may use larger numbers, but still only polynomially many bits.

Lemma 4.1 (Exact arithmetic). *For the construction of Section 3 with fixed s , the grouped sums and Taylor coefficients can be computed using rationals whose numerators and denominators have $O_s(N)$ bits. The Hermite system has $O_s(N)$ rows and can be solved using integers with $O_s(N^2 + N \log(N+1))$ bits. Consequently exact arithmetic adds only a polynomial factor to the operation count.*

Appendix B proves the lemma, including bounds for intermediate values before cancellations. Every input coefficient has a fixed common denominator depending only on s . Products use at most $3N$ inputs, and sums contain at most $2^{O_s(N)}$ terms. A common denominator raised to the power $3N$ therefore keeps coefficient lengths linear in N . For interpolation, determinant bounds control the larger integers created by elimination and exact back substitution.

Completion of the proof of Proposition 1.4. Preprocessing takes polynomial bit time in $N + m + 1$ and handles the direct-return cases of Section 2. For $N \geq N_0(s)$, construct the data and apply Propositions 3.1 and 2.1 with (4.1). The estimates above bound the arithmetic-operation count by $\text{poly}_s(N + m + 1)2^{a_s N}$, and Lemma 4.1 gives the same bound in bit operations after enlarging the polynomial. The polynomial construction cost is absorbed as well. For $N < N_0(s)$, the exhaustive-search bound is absorbed into the constant of this polynomial, since s is fixed and $2^{a_s N} \geq 1$. Thus the claimed bound covers every input size. $\square$

Execution order. For a fixed s , perform the preprocessing of Section 2, returning immediately in its terminal cases, and use exhaustive search below $N_0(s)$. Otherwise generate the fixed auxiliary data, construct the schedules by conditional expectations, and place the incidences in copies. Write the replacement forms, restricted measurements, and node representations. For every node t , enumerate the pairs (E, U) in (2.9). For each pair enumerate the measurement tuples and the assignments within each group, evaluating (2.4) modulo z^J . Accumulate the resulting Taylor coefficients with the signs $(-1)^{|E|-|U|}$ from (2.9), and solve the Hermite system to obtain $H(0)$. Every enumeration and arithmetic step has been accounted for in (2.3) and the bit analysis above.

What was simplified. The proof retains the spectral replacement and Taylor recovery ideas of [1, Sections 2–4], but uses only width-three, three-occurrence formulas. The single fixed integer s replaces the input-dependent optimization of several logarithmic parameters. The explicit orthogonal expansion (3.3) replaces row-space solving. Because the auxiliary data have fixed size, their coefficient bounds may depend arbitrarily on s ; Appendix B tracks the remaining growth with N . Ordinary decision sparsification suffices. Exact counting remains an internal device because interpolation uses rational addition and subtraction, but the final conclusion is only the decision theorem. No counting-preserving sparsification, sparse-polynomial-system solver, or tensor-contraction theorem is used.

A The binomial tail estimate

We prove the estimate used in (3.2). Let $Q \geq 1$ be an integer, let $B_1, \dots, B_Q$ be independent Bernoulli variables with success probability $p \in [0, 1]$, and put $Y = \sum_{i=1}^Q B_i$ and $\mu = Qp$. We will show that

$$\Pr(|Y - \mu| \geq h) \leq 2e^{-2h^2/Q} \quad (h > 0).$$

For one Bernoulli variable B with success probability p , define

$$g(\lambda) = \ln \mathbb{E} e^{\lambda(B-p)} = \ln(1 - p + pe^\lambda) - p\lambda \quad (\lambda \in \mathbb{R}).$$

We have $g(0) = g'(0) = 0$, and direct differentiation gives

$$g''(\lambda) = \frac{p(1-p)e^\lambda}{(1-p+pe^\lambda)^2} \leq \frac{1}{4}.$$

The last inequality is $4ab \leq (a+b)^2$ with $a = 1-p$ and $b = pe^\lambda$. The denominator is positive even when $p = 0$ or $p = 1$; in those cases g is identically zero. Taylor's formula with integral remainder gives

$$g(\lambda) = \lambda^2 \int_0^1 (1-u)g''(u\lambda) du \leq \frac{\lambda^2}{8}.$$

This formula applies to both signs of λ . Independence now yields

$$\mathbb{E} e^{\lambda(Y-\mu)} = \prod_{i=1}^Q \mathbb{E} e^{\lambda(B_i-p)} \leq e^{Q\lambda^2/8} \quad (\lambda \in \mathbb{R}).$$

For $h > 0$ and $\lambda > 0$, Markov's inequality now yields

$$\Pr(Y - \mu \geq h) \leq e^{-\lambda h} \mathbb{E} e^{\lambda(Y-\mu)} \leq \exp(Q\lambda^2/8 - \lambda h).$$

Taking $\lambda = 4h/Q$ makes the right side $e^{-2h^2/Q}$. Using the exponential-moment bound at $-\lambda$ gives the same estimate for $\Pr(\mu - Y \geq h)$. Adding the two tail bounds proves the claim.

For its application in Lemma 3.2, take $p = (r-1)/(2r)$. As shown there, the number $j(\psi)$ of negative factors in a uniformly chosen basis vector has the distribution of Y , and every integer at distance less than $R/4$ from μ belongs to I_0 . Hence

$$\frac{c}{D} = \Pr(Y \notin I_0) \leq \Pr(|Y - \mu| \geq R/4) \leq 2e^{-R^2/(8Q)}.$$

Multiplying by D proves (3.2).

B Exact rational arithmetic

This appendix proves Lemma 4.1. Throughout, s is fixed and $N \geq 1$.

Fixed auxiliary data. The index set $\mathcal{B}$ and the rational coefficient arrays in $\Psi, f, A, \beta, \phi, P$ are finite and depend only on s . They can be generated by the explicit formulas above. Their construction cost is a constant depending on s . There is an integer $B_s \geq 1$ divisible by all their coefficient denominators, including coefficients after shifts $T = t + z$ for $t \in [Q]$.

An actual form is obtained by replacing at most D auxiliary coordinates by original variables or zero. This substitution only merges or deletes monomials, so it cannot increase the coefficient ℓ_1 norm and preserves the common denominator B_s . Each form uses the measurements of only its own array copy. After the measurements have been split, it uses at most rc measurement values, with the fixed coefficients (3.3). All enumerated values lie in $[-D, D]$, whether or not the tuple is feasible. For the corrections $b_{e,v}(z) = L_{e,v}(t + z; \mathbf{1}_U) - L_{e,v}(t; \mathbf{1}_U)$ used by the Taylor algorithm, each substituted input in (2.4) is a polynomial in z of bounded degree, with a coefficient-norm bound depending only on s and coefficient denominators dividing B_s after enlarging this constant if necessary. Let $K'_s \geq 1$ bound these input norms. Enlarge it, if necessary, to bound the norms of the original inputs $L_{e,v}(T; x)$ for Boolean x as well; their coefficient denominators also divide B_s .

Products and sums. Preprocessing ensures that every factor has a nonempty scope, of size $1 \leq d \leq 3$, and uses at most d such inputs. Put $C_s = 2 + K'_s$. Coefficient convolution and the triangle inequality give $\|uv\|_1 \leq \|u\|_1 \|v\|_1$ for polynomials u, v . A clause factor therefore has norm at most $1 + (1 + K'_s)^d \leq C_s^d$. A dummy factor has norm one and satisfies the same bound. Each factor's coefficient denominators divide B_s^d , since it is a sum of products of at most d inputs. Truncation modulo z^J means discarding coefficients of degrees at least J ; it cannot increase the norm or the denominator bound.

Fix a node t , a pair (E, U) , and a measurement tuple, and abbreviate $S_g = S_{t,g}$. Let $d_g = \sum_{e: \sigma_t(e)=g} |I_e|$. The group- g sum has at most $2^{|S_g|}$ terms, each of norm at most $C_s^{d_g}$ and with coefficient denominators dividing $B_s^{d_g}$. Its norm is therefore at most $2^{|S_g|} C_s^{d_g}$. Multiplying the group sums gives norm at most $2^N C_s^M$ and common denominator B_s^M , because $\sum_g d_g = M$. Each factor occurs only once in any expanded product.

There are at most $(2D + 1)^{2\tau N}$ measurement tuples and at most $\sum_{E \subseteq [N]} 2^{|E|} = 3^N$ pairs (E, U) . Consequently every partial signed sum used to accumulate Taylor data at a node has norm at most

$$3^N (2D + 1)^{2\tau N} 2^N C_s^M \leq 6^N (2D + 1)^{2\tau N} C_s^{3N} = 2^{O_s(N)}.$$

This uses the triangle inequality and holds before any cancellation. For an explicit implementation, accumulate each group sum over the denominator $B_s^{d_g}$. Multiplying the group sums gives denominator B_s^M ; retain that denominator for all measurement and signed support sums. Finally, multiply the numerators by B_s^{3N-M} to express all coefficients over B_s^{3N} . Additions therefore do not multiply denominators. The displayed norm bound implies that the integer numerators and this denominator have $O_s(N)$ bits. The same bounds cover partial group sums and products.

Each truncated polynomial has $J = O_s(N)$ coefficients. Standard exact integer operations and polynomial multiplication therefore take polynomially many bit operations per enumerated tuple or assignment. Lemma 3.4 supplies the schedules in polynomial bit time. Group feasibility tests evaluate integer linear measurements on Boolean assignments and compare them with the enumerated values; these tests also have polynomial bit cost.

Interpolation. Finally, recover H as the unique polynomial of degree less than QJ with the computed Taylor data. Put $\ell = QJ$ and write $H(T) = \sum_{d=0}^{\ell-1} h_d T^d$. The coefficient matrix $\mathcal{H}$ is the square Hermite matrix with entries

$$\binom{d}{j} t^{d-j} \quad (1 \leq t \leq Q, \ 0 \leq j < J, \ 0 \leq d < QJ),$$

interpreted as zero when $d < j$. Uniqueness was proved by the root multiplicity argument following (2.7). Here $\ell = Q(q+1) \leq 3NR + Q = O_s(N)$. For $d \geq j$ the entry is at most $2^d Q^d = (2Q)^d$, so each integer entry has $O_s(N)$ bits. Multiplying the right side by B_s^{3N} gives an integer vector b with $O_s(N)$ bits per entry. The resulting system is $\mathcal{H}v = b$, where $v_d = B_s^{3N} h_d$. The original definition of H gives the same denominator and norm bounds as above: $\|H\|_1 \leq 2^N C_s^M$. Thus every v_d is itself an integer with $O_s(N)$ bits.

Use fraction-free Gaussian elimination on the augmented matrix $[\mathcal{H} \mid b]$. Order its rows lexicographically by (t, j) and its coefficient columns by increasing d . Initialize $p_{\text{prev}} = 1$. At stage $k = 1, \dots, \ell - 1$, if the pivot a_{kk} is zero, swap with the first remaining row having a nonzero entry in column k . Nonsingularity guarantees that such a row exists. Save $p = a_{kk}$ and replace the trailing entries, including the right-side column, by

$$a_{ij} \leftarrow \frac{p a_{ij} - a_{ik} a_{kj}}{p_{\text{prev}}}, \quad i, j > k,$$

keeping the pivot row and column unchanged during these updates. Then set the entries below the pivot to zero and set $p_{\text{prev}} = p$ for the next stage.

Here is the determinant invariant that makes every division exact. In the current row order, let C be the leading $(k-1) \times (k-1)$ block of the original matrix. Inductively, $\det C = p_{\text{prev}}$, and each active entry is the minor obtained by bordering C with its row and column. At the first stage C is empty and $\det C = 1$. Eliminating C over $\mathbb{Q}$ shows that each such minor is $\det C$ times the corresponding residual entry. The update numerator is therefore $(\det C)^2$ times a two-by-two residual determinant. The minor bordered by both new rows and columns is $\det C$ times that same determinant. Division by p_{prev} thus gives the larger integer minor. This is Sylvester's determinant identity. Swapping remaining rows leaves C and p_{prev} unchanged, so the argument also covers pivoting.

Apart from the inserted zeros, all stored entries are consequently signed minors of the original augmented matrix, with the row permutations taken into account. For a minor of order $k \leq \ell$, Hadamard's bound gives

$$|\det| \leq (\sqrt{k} 2^{O_s(N)})^k,$$

so its bit length is $O_s(N^2 + N \log(N+1))$. The temporary products and differences in an update obey the same asymptotic bit bound, since they combine only two such entries at a time. There are polynomially many updates.

Back substitution also uses exact integer divisions, since the solution entries v_d are integers. Each numerator is a right-side entry minus a sum of at most $\ell - 1$ products of a triangular coefficient and an already computed v_d . The minor bound and the $O_s(N)$ -bit bound on v_d therefore keep all these intermediate integers within $O_s(N^2 + N \log(N+1))$ bits. Standard integer arithmetic gives polynomial bit time for elimination and back substitution. Finally divide by B_s^{3N} to recover H ; its constant coefficient is the exact integer $\#F$.

Use of AI: The author prompted Claude and ChatGPT interactively to simplify the proofs, focusing only on disproving basic statement of ETH, without optimizing log factors. The author expanded the proofs manually and made them more human-readable and added explanations and a worked example.

References

- [1] OpenAI. *The Exponential Time Hypothesis Is False*. Manuscript, 17 October 2026; revised 25 October 2026. [Public manuscript](#). The construction reconstructed here is in Sections 2–4; Appendix B.4 discusses the corresponding exact arithmetic.
- [2] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k -SAT. *Journal of Computer and System Sciences* 62(2):367–375, 2001. [doi:10.1006/jcss.2000.1727](#).
- [3] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences* 63(4):512–530, 2001. [doi:10.1006/jcss.2001.1774](#). Corollary 1, p. 523, is the sparsification lemma used here.