hexagon:2611.01287v1

Subexponential-Time Classical Sampling of Quantum Circuits from Not ETH

Alexandru Gheorghiu[1] and Aparna Gupte[2]

[1]IBM Research
[2]Massachusetts Institute of Technology

Note. Writeup by ChatGPT Astra, based on ideas and feedback from the authors.

Abstract

The recent subexponential-time algorithm for counting satisfying assignments of a 3-CNF formula, together with the Boolean tensor-contraction theorem underlying its sharper bound, gives classical sampling algorithms for two families of quantum circuits. The first consists of n-qubit circuits with $O(n)$ gates from any fixed finite set of bounded-arity algebraic gates. The second consists of circuits over $\{H, Z, \mathrm{CZ}, \mathrm{CCZ}\}$ with $O(n)$ Hadamard gates and $d \le n\, 2^{o(\sqrt{\log n})}$ diagonal gates. More precisely, for $d \le Cn^{11/10}$ with fixed C, we obtain exact worst-case sampling in time $n^{O(1)}2^{O(n\log(2+d/n)/\sqrt{\log n})}$ and polynomial space. We explain the reductions using Feynman sums over paths. Montanaro's construction [Mon17] gives a cubic polynomial with $O(n)$ variables and $O(n+d)$ monomials; an explicit density bound for polynomial-system counting yields the stated time. In both cases, each prefix probability is computed as a single circuit amplitude, and the chain rule gives a sample. The second sampler uses exactly as many fair bits as the circuit has Hadamards.

1 Statements and conventions

Let U be a circuit on n qubits, initialized in $|0^n\rangle$ and measured in the computational basis. Its output distribution is

$$p_U(z) = \left|\langle z|U|0^n\rangle\right|^2, \qquad z \in \{0,1\}^n.$$

We take the recent counting results of [OAI26] as input and derive the following two consequences. Throughout, logarithms are base two, polylog n means a fixed power of $\log n$, and subexponential time means $2^{o(n)}$ in the number of qubits. All running times count bit operations. Finitely many small input sizes can be handled by exhaustive computation.

Theorem 1 (Linear-size circuits)**.** *Fix a finite set $\mathcal{G}$ of quantum gates, each acting on at most a constant number of qubits and having algebraic matrix entries. For every fixed $A > 0$, the output of any n-qubit circuit over $\mathcal{G}$ with at most An gates can be sampled exactly by a classical randomized algorithm in expected time*

$$n^{O(1)}2^{O(n/\sqrt{\log n})} = 2^{o(n)}. \tag{1}$$

For any $0 < \varepsilon < 1$, a sample within total variation distance ε can instead be produced in worst-case time

$$2^{O(n/\sqrt{\log n})}\,\mathrm{poly}(n, \log(1/\varepsilon)).$$

The constants depend only on $\mathcal{G}$ and A.

The gate-set assumption gives a precise model for exact arithmetic. It includes, for example, the usual gate set $\{H, T, \mathrm{CNOT}\}$. The distinction between expected and worst-case time concerns the generation of exact random choices when probabilities are irrational; the probability computations themselves have deterministic time bounds.

Theorem 2 (A linear number of Hadamards and more diagonal gates)**.** *Assume Theorem D.1 and Lemmas 5.1–5.2 of [OAI26]. Fix constants $A, C > 0$. Let U be an n-qubit circuit over $\{H, Z, \mathrm{CZ}, \mathrm{CCZ}\}$ with $h \le An$ Hadamard gates and $d \le Cn^{11/10}$ diagonal gates. Its output can be sampled exactly by a classical randomized algorithm in worst-case time*

$$n^{O_{A,C}(1)}2^{O_{A,C}(n\log(2+d/n)/\sqrt{\log n})} \tag{2}$$

and polynomial space. Its random choices use exactly h fair bits. The gate description is an explicit list of gate types and wire indices.

The bound is subexponential whenever $\log(2 + d/n) = o(\sqrt{\log n})$, equivalently $d \le n\,2^{o(\sqrt{\log n})}$. For $d = O(n\,\mathrm{polylog}\,n)$ it recovers the exponent $O(n \log\log n/\sqrt{\log n})$. It also permits, for example, $d = n\,2^{(\log n)^{1/3}}$, giving exponent $O(n/(\log n)^{1/6})$. At the technical endpoint $d = Cn^{11/10}$, (2) gives exponent $O(n\sqrt{\log n})$ and does not certify subexponential time.

The proof has three steps: compute a prefix probability as one amplitude, express that amplitude by exact counts, and choose the output bits using conditional probabilities. For the second theorem we retain the monomial density explicitly in the polynomial-system construction of [OAI26], following the strengthened argument in [Den26]. This requires the full contraction theorem. The fixed polylogarithmic-density counting statement alone does not cover an input-dependent exponent of $\log n$.

2 The counting results

We use two counting statements. The first is the fixed-density case of [OAI26, Theorem 1.1]. The second extends the incidence construction of [OAI26, Lemma D.2] to an explicit monomial-density parameter. We give its proof in Appendix A, using [OAI26, Theorem D.1 and Lemmas 5.1–5.2]. All constants in the hypotheses are fixed independently of the input size; the parameter B in part (ii) may depend on the input.

Theorem 3 (Counting results)**.**

(i) For each fixed $D > 0$, the number of satisfying assignments of a 3-CNF formula with N variables and at most DN clauses can be computed deterministically in

$$N^{O(1)}2^{O(N/\sqrt{\log N})}$$

time and polynomial space.

(ii) Fix an integer $k \ge 1$ and constants $c, C > 0$. Suppose that $P_1, \dots, P_a$ are polynomials over $\mathbb{F}_2$ in $N \ge 20$ declared Boolean variables, where $a \le cN$. Each P_i has degree at most k and is

specified by at most B monomials, with $1 \le B \le CN^{1/10}$. Assuming [OAI26, Theorem D.1 and Lemmas 5.1–5.2], their common zeros can be counted deterministically in

$$N^{O_{k,c,C}(1)} 2^{O_{k,c,C}(N \log(B+2)/\sqrt{\log N})}$$

time and polynomial space.

Bounded values of N can be handled by enumeration. When $B = O((\log N)^c)$ for a fixed c, part (ii) has exponent $O(N \log\log N/\sqrt{\log N})$, as in the original sparse-polynomial statement. Its explicit dependence on B is what permits more diagonal gates in Theorem 2.

3 From prefix probabilities to samples

For a string $a \in \{0,1\}^\ell$, define its prefix probability by

$$p_U(a) = \sum_{b \in \{0,1\}^{n-\ell}} p_U(ab). \tag{3}$$

In particular, $p_U(\varnothing) = 1$ and $p_U(a) = p_U(a0) + p_U(a1)$. Starting from the empty prefix, choose the next bit to be 0 with probability

$$q(a) = \frac{p_U(a0)}{p_U(a)}, \tag{4}$$

and choose 1 otherwise. Every prefix reached with positive probability has a positive denominator. For any output z of positive probability, the chance that this procedure produces z is

$$\prod_{j=1}^{n} \frac{p_U(z_1 \cdots z_j)}{p_U(z_1 \cdots z_{j-1})} = p_U(z).$$

Zero-probability outputs are never chosen. Since the current value $p_U(a)$ can be retained and $p_U(a1)$ obtained by subtraction, only one new prefix-probability computation is needed per output bit.

3.1 A prefix probability is one amplitude

The sum in (3) can be represented without enumerating its $2^{n-\ell}$ terms. Add ℓ ancilla qubits initialized to zero, and let C_ℓ copy the first ℓ computational-basis bits into these ancillas:

$$C_\ell |z\rangle |0^\ell\rangle = |z\rangle |z_1 \cdots z_\ell\rangle.$$

This unitary is a product of ℓ CNOT gates. For $a \in \{0,1\}^\ell$, let $X^a = \bigotimes_{j=1}^{\ell} X^{a_j}$ and set

$$V_a = (I \otimes X^a)(U^\dagger \otimes I) C_\ell (U \otimes I). \tag{5}$$

Lemma 4. *The circuit V_a acts on $n+\ell$ qubits and satisfies*

$$\langle 0^{n+\ell} | V_a | 0^{n+\ell} \rangle = p_U(a). \tag{6}$$

If U has s gates, then V_a has at most $2s + 2\ell$ gates, using the original gates, their adjoints, CNOT, and X.

Proof. Write $\alpha_z = \langle z|U|0^n\rangle$. Immediately after C_ℓ, the state is

$$\sum_z \alpha_z |z\rangle |z_1 \cdots z_\ell\rangle.$$

The final ancilla bra selects exactly the strings with prefix a, while $\langle 0^n|U^\dagger|z\rangle = \overline{\alpha_z}$. The amplitude in (6) is therefore $\sum_{z:z_{1:\ell}=a} |\alpha_z|^2 = p_U(a)$. The gate count follows directly from (5). □

Thus sampling reduces to n amplitude computations for circuits whose size increases only by a constant factor and an additive $O(n)$ term. Equation (6) is an equality with the probability itself; its right-hand side is already the squared-amplitude sum required by the Born rule.

4 Linear-size circuits: paths and 3-CNF formulas

Let $W = G_s \cdots G_1$ be a circuit on r qubits. Inserting computational-basis states between successive gates gives its Feynman sum over paths. A path specifies the basis bits on the wires as the circuit is traversed; its weight is the product of the corresponding gate entries. For fixed input x and output y,

$$\langle y|W|x\rangle = \sum_{\pi: x \longrightarrow y} \prod_{j=1}^{s} g_j(\pi), \tag{7}$$

where $g_j(\pi)$ is the local matrix entry of gate G_j along the path.

The number of Boolean variables needed to describe a path is $O(r+s)$. To see this, keep the current bit label on every wire and introduce new labels only for the output wires of the next gate. A gate of bounded arity introduces only constantly many labels. Bits on the other wires retain their labels. Input and output conditions fix the labels at the boundary.

4.1 Grouping paths with the same weight

For a fixed finite gate set, there are only constantly many distinct nonzero gate entries. List them as $\beta_1, \ldots, \beta_R$, where R is a constant. Discard paths containing a zero entry. If a path uses β_j exactly m_j times, its weight is $\prod_j \beta_j^{m_j}$. Hence

$$\langle y|W|x\rangle = \sum_{\substack{\mathbf{m}\in\mathbb{Z}^R_{\geq 0} \\ m_1+\cdots+m_R=s}} N_{\mathbf{m}} \prod_{j=1}^{R} \beta_j^{m_j}, \tag{8}$$

where $N_{\mathbf{m}}$ is the number of paths with the indicated multiplicities and boundary conditions. There are at most $(s+1)^R$ vectors to consider.

For each $\mathbf{m}$, these conditions can be checked by a Boolean circuit of size $O(r+s)$. Each gate's local transition and entry type are determined by constantly many path bits. To count how often a given type occurs, add its s indicator bits using a balanced binary tree of adders. At level j there are $O(s/2^j)$ additions of $O(j)$-bit numbers, so the total number of Boolean gates is

$$O\left(\sum_{j\geq 1} \frac{sj}{2^j}\right) = O(s).$$

Comparison with the required multiplicities and checking the boundary conditions also fit within $O(r+s)$ gates.

4.2 Preserving the count in the conversion to 3-CNF

Introduce one auxiliary variable for each gate of this Boolean checking circuit, and impose clauses expressing its exact value. For example, $w = u \wedge v$ is expressed by

$$(\neg w \vee u) \ \wedge \ (\neg w \vee v) \ \wedge \ (w \vee \neg u \vee \neg v).$$

The constraints for OR and NOT are equally small. Finally, require the checking circuit's output to equal 1.

For every assignment to the path bits, the auxiliary gate values are uniquely determined. Consequently, the number of satisfying assignments of the resulting 3-CNF formula is exactly $N_{\mathbf{m}}$. The formula has $O(r+s)$ variables and clauses, with a constant clause-to-variable ratio. Theorem 3(i) therefore computes each count in

$$\mathrm{poly}(r+s)\, 2^{O((r+s)/\sqrt{\log(r+s)})}$$

time. The polynomial number of multiplicity vectors does not change the exponential factor.

All the β_j lie in a fixed number field. Products of at most s such entries, and their sums with the integer coefficients $N_{\mathbf{m}} \le 2^{O(r+s)}$, have exact representations using polynomially many bits. Thus the arithmetic in (8) adds only polynomial time. We have proved the following amplitude bound.

Proposition 5. *For a fixed finite set of bounded-arity algebraic gates, an amplitude of an r-qubit, s-gate circuit can be computed exactly in*

$$\mathrm{poly}(r+s)\, 2^{O((r+s)/\sqrt{\log(r+s)})}$$

deterministic time and polynomial space.

For a circuit in Theorem 1, every V_a from Lemma 4 has $r + s = O(n)$. Enlarging the fixed gate set by adjoints, CNOT, and X preserves the hypotheses of Proposition 5. Every prefix probability is therefore available within (1). The chain rule and the random-choice implementation in Section 6 prove Theorem 1.

5 Few Hadamards: cubic polynomials over $\mathbb{F}_2$

For the gate set $\{H, Z, \mathrm{CZ}, \mathrm{CCZ}\}$, the path sum has a simpler form. Diagonal gates preserve the basis labels and contribute signs. Only Hadamards introduce new path variables. This is the reason that the number of Hadamards controls the number of variables even when there are many diagonal gates.

5.1 Montanaro's construction

Use the following form of Montanaro's circuit-polynomial correspondence [Mon17, Proposition 1]. Suppose an r-qubit circuit C starts and ends with $H^{\otimes r}$, and its interior has t Hadamards and d diagonal gates. Give a variable to each wire segment between interior Hadamards. There are $N = r + t$ variables. Add a term to f_C for each interior gate:

$$H: \ uv, \qquad Z: \ u, \qquad \mathrm{CZ}: \ uv, \qquad \mathrm{CCZ}: \ uvw,$$

using the variables on its incident segments. Addition is over $\mathbb{F}_2$. The polynomial has degree at most three and at most $t + d$ monomials. Since

$$\langle v|H|u\rangle = 2^{-1/2}(-1)^{uv},$$

the path expansion gives

$$\langle 0^r|C|0^r\rangle = 2^{-r-t/2}\,\mathrm{gap}(f_C), \qquad \mathrm{gap}(f) = \sum_{x\in\{0,1\}^N} (-1)^{f(x)}. \tag{9}$$

Writing $Z(f) = |\{x : f(x) = 0\}|$, we have

$$\mathrm{gap}(f) = 2Z(f) - 2^N. \tag{10}$$

An arbitrary r-qubit circuit W can be put in the required form as $H^{\otimes r}(H^{\otimes r}WH^{\otimes r})H^{\otimes r}$, adding $4r$ Hadamards without changing W.

For example, $H^{\otimes 3}\,\mathrm{CCZ}\,H^{\otimes 3}$ gives $f(x) = x_1x_2x_3$. It has seven zeros and one one, so its all-zero amplitude is $(7-1)/8 = 3/4$ and its all-zero output probability is $9/16$.

5.2 Counting a cubic polynomial with many monomials

The polynomial in (9) may have many more monomials than variables. We distribute its monomials among linearly many equations, using only $O(N)$ additional variables. This makes the density in Theorem 3(ii) proportional to the number of monomials per original variable.

Lemma 6. *Fix $C > 0$. Under the assumptions of Theorem 3(ii), let f be a degree-at-most-three polynomial over $\mathbb{F}_2$ on N Boolean variables, specified by at most $M \le CN^{11/10}$ monomials. Its number of zeros and its gap can be computed exactly in*

$$N^{O_C(1)}2^{O_C(N\log(2+M/N)/\sqrt{\log N})}$$

time and polynomial space.

Proof. Constant polynomials can be handled directly. Otherwise, set $B = \max\{1, \lceil M/N\rceil\}$, partition the monomials into $q \le N$ groups of at most B terms, and write

$$f(x) = f_1(x) + \cdots + f_q(x) \qquad \text{over } \mathbb{F}_2.$$

Introduce variables $y_1, \ldots, y_q$ for the successive partial sums and impose

$$\begin{aligned} y_1 + f_1(x) &= 0, \\ y_j + y_{j-1} + f_j(x) &= 0 \qquad (2 \le j \le q), \\ y_q &= 0. \end{aligned} \tag{11}$$

Once x is fixed, the first q equations uniquely determine $y_j = \sum_{i\le j} f_i(x)$. The last equation holds exactly when $f(x) = 0$. The number of common zeros of (11) is therefore $Z(f)$, with no multiplicative correction.

The system has $N' = N + q \in [N, 2N]$ variables and at most $N + 1$ equations. Every equation has degree at most three and at most $B + 2$ monomials. Moreover,

$$B + 2 \le CN^{1/10} + 3 \le (C + 3)(N')^{1/10}.$$

Theorem 3(ii) applies with fixed constants depending only on C. Since $N' = \Theta(N)$ and $\log(B+4) = O(\log(2 + M/N))$, it gives $Z(f)$ in the stated time. Equation (10) then gives $\mathrm{gap}(f)$. □

The reduction uses one new variable per group. There are at most N groups even as their size grows with the monomial density.

5.3 Applying the construction to prefix probabilities

Let U have h Hadamards and d diagonal gates. All four gates in this gate set are self-adjoint. In the prefix circuit (5), replace each CNOT and each X by

$$\mathrm{CNOT}_{u\to v} = H_v\,\mathrm{CZ}_{u,v}\,H_v, \qquad X = HZH.$$

For a prefix of length ℓ, write $\mathrm{wt}(a)$ for its number of ones. The resulting V_a acts on $r = n + \ell \le 2n$ qubits and has

$$h_a = 2h + 2\ell + 2\,\mathrm{wt}(a) \le 2h + 4n, \tag{12}$$

$$d_a = 2d + \ell + \mathrm{wt}(a) \le 2d + 2n. \tag{13}$$

Pad it with boundary Hadamards as above. Its number of interior Hadamards is $t_a = h_a + 2r$, so Montanaro's polynomial has

$$N_a = r + t_a = h_a + 3r = 2h + 3n + 5\ell + 2\,\mathrm{wt}(a) \tag{14}$$

variables. In particular, $3n \le N_a \le (2A + 10)n$. Its number M_a of listed monomials satisfies

$$M_a \le t_a + d_a = N_a + 2d - n + \mathrm{wt}(a) \le N_a + 2d.$$

Thus $d \le Cn^{11/10}$ implies $M_a \le (1 + 2C)N_a^{11/10}$, within the fixed-constant range of Lemma 6. Also

$$N_a = \Theta_A(n), \qquad \log(2 + M_a/N_a) = O(\log(2 + d/n)).$$

Combining Lemma 4 with (9) gives the exact prefix probability

$$p_U(a) = 2^{-r-t_a/2}(2Z(f_a) - 2^{N_a}). \tag{15}$$

The integer h_a is even, so t_a is even and the denominator here is an integer power of two. Lemma 6 computes this probability within (2). Repeating for at most n prefixes preserves that bound. The exact sampling procedure below completes the proof of Theorem 2.

6 Implementing the random choices

6.1 Algebraic probabilities

For Theorem 1, every prefix probability is a real element of a fixed number field whose coordinates use polynomially many bits. The same is true of a conditional probability $q \in [0, 1]$ from (4). Arithmetic and order comparisons in this fixed field take polynomial time in the bit length of their inputs.

To generate a Bernoulli random variable with success probability q, expose independent fair bits of a uniform number $R \in [0, 1]$. After b bits, R lies in a known half-open dyadic interval $I_b = [u_b, v_b)$ of length 2^{-b}. If $v_b \le q$, declare success; if $u_b \ge q$, declare failure. Otherwise expose another bit. The event $R = q$ has probability zero.

The probability of being undecided after b bits is at most 2^{-b}. Each comparison costs $\mathrm{poly}(n + b)$ time, so the expected arithmetic overhead is polynomial in n. The procedure halts with probability one and makes the exact conditional choice. Together with the n prefix computations, this proves the exact expected-time claim.

For a worst-case bound, stop after $L = \lceil \log_2(n/\varepsilon) \rceil$ bits for each conditional choice, resolving any remaining ambiguity arbitrarily. Couple this procedure to the exact sampler using the same random bits. At each step the conditional chance of an unresolved comparison is at most 2^{-L}; the chance of any discrepancy is at most $n2^{-L} \le \varepsilon$. Consequently the output distribution $\widetilde{p}$ satisfies

$$d_{\mathrm{TV}}(p_U, \widetilde{p}) := \frac{1}{2} \sum_z |p_U(z) - \widetilde{p}(z)| \le \varepsilon.$$

The additional worst-case work is $\mathrm{poly}(n, \log(1/\varepsilon))$, as stated in Theorem 1.

6.2 A finite exact sampler for the second gate set

For a circuit over $\{H, Z, \mathrm{CZ}, \mathrm{CCZ}\}$ with h Hadamards, each path has weight $\pm 2^{-h/2}$. Every output amplitude is therefore $A_z 2^{-h/2}$ for an integer A_z, and

$$p_U(z) = \frac{A_z^2}{2^h}, \qquad W(a) := 2^h p_U(a) \in \mathbb{Z}_{\ge 0}. \tag{16}$$

Although the ratios in (4) need not be dyadic, all prefix weights in (16) are integers with a common denominator. This permits exact sampling with a fixed number of fair bits:

1. Draw a uniform integer $R \in \{0, \dots, 2^h - 1\}$ using h fair bits, and start with $a = \varnothing$.
2. Compute $W(a0)$. If $R < W(a0)$, append 0 to a. Otherwise replace R by $R - W(a0)$ and append 1.
3. Repeat until a has length n, then output a.

Conditioned on the current prefix a, the integer R is uniform in $\{0, \dots, W(a) - 1\}$. Splitting this interval into lengths $W(a0)$ and $W(a1)$ makes the next choice with exactly the probabilities in (4), and preserves the same invariant at the next step. Intervals of length zero are never selected.

The prefix weights and the random integer have $O(h + 1) = O(n)$ bits. The sampler makes n deterministic counting calls and has polynomial additional work and space. Hence its exact worst-case running time is (2), using exactly h fair bits. This includes $h = 0$, when the initial interval has one element and no random bits are needed.

Concurrent work. The results in this note were publicly announced on 3 November 2026 on the ETH community discussion blog, together with a link to the manuscript [Ghe26]. Concurrent work by Vaibhav Tulsyan, posted to Hexagon on 4 November 2026, also derives subexponential-time exact simulation and sampling algorithms for sparse quantum circuits from the recent counting results [Tul26].

A Counting with an explicit monomial density

We prove Theorem 3(ii), following the construction of [OAI26, Lemma D.2] with the dependence on B retained, as in [Den26]. The main point is that the count can be expressed using incidence monomials of total degree $O(N)$, even though there may be $O(NB)$ incidences. Specifying such a monomial takes $O(N \log(B + 2))$ bits. These bits will index the edges of a tensor network.

A.1 The contraction input

Here is the form of [OAI26, Theorem D.1] that we use. Fix constants $C_0, A_0, b_0 > 0$. Consider a bipartite tensor network whose edge indices are Boolean, with at most $C_0 s$ edges, maximum degree at most $\lceil s^{9/10} \rceil$, and polynomially many vertices. Its entries belong to

$$(\mathbb{Z}/2^{P_0}\mathbb{Z})[V]/(V^{\ell}), \qquad 1 \le P_0, \ell \le C_0 s.$$

Suppose the network and its entry procedures have total description length L. Each entry must be computable uniformly in time $(s+L)^{b_0} 2^{A_0 s^{9/10}}$ and space $(s+L)^{b_0}$. The theorem computes the contraction deterministically in

$$(s+L)^{O(1)} 2^{O(s/\sqrt{\log s})}$$

time and polynomial space. The constants depend on the fixed constants in these hypotheses. In particular, tensors are specified by procedures for evaluating their entries; an explicit table of all entries is not required. A contraction sums, over all assignments to the edge indices, the product of the entries at the vertices.

A.2 Controlling variable occurrences

First multilinearize each input polynomial and cancel repeated monomials over $\mathbb{F}_2$. A constant-one equation makes the answer zero. Discard constant-zero equations, returning 2^N if none remain. Let I_i be the set of variables occurring in P_i. Then

$$|I_i| \le kB, \qquad E_0 := \sum_i |I_i| \le kcNB.$$

An incidence is a pair (i, v) with $v \in I_i$.

We first fix variables that occur in many equations. Put

$$\lambda = \lceil \log N \rceil, \qquad D_* = \max\{1, \lceil E_0/N \rceil\} \lceil \sqrt{\lambda} \rceil.$$

At most $E_0/D_* \le N/\sqrt{\lambda}$ variables occur in more than D_* sets I_i. Enumerate their assignments, creating at most $2^{N/\sqrt{\lambda}}$ disjoint branches. On each branch normalize the remaining equations again. A constant-one equation gives branch count zero. Discard constant-zero equations; if none remain, the branch count is two to the power of the number of free variables. Every free variable now occurs in at most

$$D_* = O_{k,c}(B\sqrt{\log N})$$

equations, and each remaining equation still involves at most kB variables. Free variables absent from every equation remain declared. We retain the original N in all precision and size bounds, and use $a \le cN$ for the number of remaining equations on a branch.

A.3 A bounded-degree expression for the count

For one branch, give each incidence (i, v) its own formal variable z_{iv}. Let $L_i(z_i)$ be the integer polynomial obtained from P_i by interpreting its monomials over the integers and replacing x_v by z_{iv}. The parity lift in [OAI26, Lemmas 5.1–5.2] provides integer polynomials $e_0, \dots, e_N$ of degree at most one such that

$$\sum_{j=0}^{N} e_j(Y)(Y(Y-1))^j \equiv \mathbf{1}_{\{Y \text{ even}\}} \pmod{2^{N+1}} \qquad (Y \in \mathbb{Z}). \tag{17}$$

Their coefficients have polynomial bit length and can be constructed in polynomial time.

Work over the ring

$$\mathcal{R} = (\mathbb{Z}/2^{N+1}\mathbb{Z})[V]/(V^{N+1})$$

and form

$$B_i(z_i, V) = \sum_{j=0}^{N} e_j\big(L_i(z_i)\big)\big(L_i(z_i)(L_i(z_i)-1)\big)^j V^j. \tag{18}$$

Multilinearize each coefficient in the incidence variables. This leaves its values on Boolean inputs unchanged. Variables belonging to different equations are distinct, so

$$F(z, V) = \prod_{i=1}^{a} B_i(z_i, V)$$

is multilinear in all incidence variables. The coefficient of V^j has total incidence degree at most $k(a+2j)$: a product using exponents $j_1, \dots, j_a$ has degree at most $\sum_i k(1+2j_i)$, and $\sum_i j_i = j$. Thus every incidence monomial surviving the truncation has degree at most

$$K = k(a+2N) = \Theta_{k,c}(N).$$

Let S_{N+1} sum the coefficients of $1, V, \dots, V^N$ in $\mathcal{R}$. The branch count, modulo 2^{N+1}, is

$$S_{N+1}\left(\sum_{x \text{ free Boolean}} F\big((z_{iv} = x_v)_{iv}, V\big)\right). \tag{19}$$

To see this, first multiply the expressions in (17) with $Y = L_i(x)$. Their product is the indicator that all equations hold. Every factor $L_i(x)(L_i(x)-1)$ is even. Hence a term of total V-degree at least $N+1$ is divisible by 2^{N+1} and can be omitted. Summing the remaining coefficients gives (19). The coefficient-sum map is applied after multiplication.

A.4 Encoding the incidence monomials

Partition the equations and the free variables separately into $J = \lceil N^{1/4} \rceil$ groups of sizes differing by at most one. Write T_g for an equation group and X_h for a variable group. Let E_{gh} count the incidences between these two groups, and set $E = \sum_{g,h} E_{gh} \le E_0$.

Enumerate the integer arrays (d_{gh}) with $0 \le d_{gh} \le E_{gh}$ and $\sum_{g,h} d_{gh} \le K$. There are at most

$$(K+1)^{J^2} = 2^{O_{k,c}(\sqrt{N}\log N)}$$

such arrays. For a fixed array, create one vertex for each group and join T_g to X_h by

$$\left\lceil \log \binom{E_{gh}}{d_{gh}} \right\rceil$$

Boolean edges. The bits on these edges index a subset of d_{gh} incidences, in a fixed order agreed by both endpoints. An index beyond the number of subsets gives entry zero. If the binomial coefficient is one, no edge is needed. Ranking and unranking these subsets use polynomial-time arithmetic on binomial coefficients of polynomial bit length.

The number of edges is at most

$$\begin{aligned} J^2 + \log \prod_{g,h} \binom{E_{gh}}{d_{gh}} &\le J^2 + \log \binom{E+K}{K} \\ &\le J^2 + K \log \left(\frac{e(E+K)}{K} \right) \\ &= O_{k,c}(N \log(B+2)). \end{aligned}$$

For the first inequality, the indexed choices are subsets of at most K of the E incidences. There are at most $\binom{E+K}{K}$ such subsets: encode one by a multiset of size K from the E incidences and one extra padding symbol. The final bound uses $E = O_{k,c}(NB)$ and $K = \Theta_{k,c}(N)$. This is the source of the factor $\log(B+2)$.

We next specify the tensor entries. At an equation-group vertex, the incident edge values determine a subset $S_i \subseteq I_i$ for each $i \in T_g$. Write $z_{i,S_i} = \prod_{v \in S_i} z_{iv}$. The entry is the following element of $\mathcal{R}$:

$$\prod_{i \in T_g} [z_{i,S_i}] B_i(z_i, V).$$

Each coefficient can be found from Boolean evaluations by

$$[z_{i,S_i}] B_i = \sum_{T \subseteq S_i} (-1)^{|S_i| - |T|} B_i(\mathbf{1}_T, V), \tag{20}$$

where $\mathbf{1}_T$ sets precisely the variables in T to one and all other incidence variables of that equation to zero. Expanding a multilinear polynomial in this sum cancels every monomial except z_{i,S_i}. Since $|S_i| \le kB$, there are at most 2^{kB} evaluations, each taking polynomial time using (18). Compute and multiply the coefficients for the equations one at a time. An entry therefore takes $N^{O(1)} 2^{O_k(B)}$ time and polynomial space.

At a variable-group vertex, let b_v be the number of selected incidences involving $v \in X_h$. Its entry is

$$\sum_{x \in \{0,1\}^{X_h}} \prod_{v \in X_h} x_v^{b_v} = 2^{|\{v \in X_h : b_v = 0\}|} \pmod{2^{N+1}}.$$

An exponent zero contributes the constant one; a positive exponent requires $x_v = 1$. This entry is computable in polynomial time and accounts for free variables absent from every equation.

Every incidence monomial of F belongs to exactly one degree array and has exactly one valid subset index for each pair of groups. The equation-group entries supply its coefficient, and the variable-group entries sum its value over assignments after the incidences of each variable are identified. Summing the contractions over all arrays therefore gives the expression inside S_{N+1} in (19).

A.5 Checking the time and space bounds

An equation group contains $O_c(N^{3/4})$ equations, each with at most kB incidences. A variable group contains $O(N^{3/4})$ variables, each in at most D_* equations. Since $\lceil \log \binom{E_{gh}}{d_{gh}} \rceil \le E_{gh}$, every vertex has degree at most

$$O_{k,c}(N^{3/4} B \sqrt{\log N}) \le O_{k,c,C}(N^{17/20} \sqrt{\log N}).$$

Set $s = \lceil N \log(B+2) \rceil$. For all sufficiently large N, this degree is at most $\lceil s^{9/10} \rceil$. There are $2J$ vertices and $O_{k,c}(s)$ edges. The graph and entry procedures have polynomial-size descriptions, and

the precision and truncation order $N+1$ are $O(s)$. Finally, the entry cost $N^{O(1)}2^{O_k(B)}$ is within the allowed $N^{O(1)}2^{O(s^{9/10})}$, with polynomial space.

The contraction theorem thus applies. Since $\log s = \Theta(\log N)$ throughout $1 \le B \le CN^{1/10}$, each contraction takes

$$N^{O_{k,c,C}(1)}2^{O_{k,c,C}(s/\sqrt{\log s})} = N^{O_{k,c,C}(1)}2^{O_{k,c,C}(N\log(B+2)/\sqrt{\log N})}$$

time and polynomial space. Multiplying by the number of branches and degree arrays preserves this bound: their exponents are $O(N/\sqrt{\log N})$ and $O(\sqrt{N}\log N)$.

Process the branches and arrays sequentially, applying S_{N+1} and adding branch counts modulo 2^{N+1}. All ring elements and accumulators have polynomial size. The total number of common zeros lies in $[0, 2^N]$, so its least nonnegative residue modulo 2^{N+1} is the exact answer. This proves Theorem 3(ii).

References

[OAI26] OpenAI. *The Exponential Time Hypothesis is False.* Manuscript, 17 October 2026; revised 25 October 2026. Manuscript and source. The inputs used here are Theorem 1.1, Theorem D.1, and Lemmas 5.1–5.2. The proof of Lemma D.2 supplies the incidence construction extended in Appendix A.

[Den26] *Exact Quantum Sampling with More Diagonal Gates: A density extension with explicit hypotheses and parameter bounds.* Unpublished note, 30 October 2026. Theorem 1.1, Theorem 2.1, and Corollary 2.2.

[Mon17] Ashley Montanaro. Quantum circuits and low-degree polynomials over $\mathbb{F}_2$. *Journal of Physics A: Mathematical and Theoretical*, 50(8):084002, 2017. doi:10.1088/1751-8121/aa565f. arXiv:1607.08473.

[Ghe26] Alexandru Gheorghiu. Announcement of joint work with Aparna Gupte. Comment on *New Results Related to ETH*, ETH: Community Examination and Discussion, 3 November 2026. Online announcement.

[Tul26] Vaibhav Tulsyan. *Exact amplitudes of sparse quantum circuits in subexponential time, assuming the new #SAT algorithms.* Hexagon preprint 2611.00090v1, 4 November 2026. Manuscript.