hexagon:2611.00090v1

EXACT AMPLITUDES OF SPARSE QUANTUM CIRCUITS IN SUBEXPONENTIAL TIME, ASSUMING THE NEW #SAT ALGORITHMS

VAIBHAV TULSYAN

Abstract. Classical simulation lower bounds for quantum circuits have been derived from the Exponential Time Hypothesis (ETH). However, in October 2026 OpenAI and Anthropic claimed deterministic subexponential algorithms for #k-SAT, refuting ETH, and neither manuscript mentions quantum circuits. We record the corollary. Assuming three internal results of the Anthropic manuscript, one deterministic Turing machine computes the amplitude $\langle y|C|x\rangle$ of a Clifford+T circuit with G two-qubit gates in arbitrary connectivity exactly, as an element of $\mathbb{Z}[\omega]/\sqrt{2}^{\,k}$ with $\omega = e^{i\pi/4}$, in time $2^{O(G/\sqrt{\log G})}$ whenever the input length L satisfies $\log L \le G^{0.9}$. The proof reads the sum over paths, one bit per wire segment, as an Anthropic-type network, with a Kronecker map for $\mathbb{Z}[\omega]$ and a Galois bound. We machine-check this theorem in Lean 4 (`amplitude_TM_rate'`, `amplitude_TM_rate`, `amplitude_TM`; axioms `propext`, `Classical.choice`, `Quot.sound`) on top of Anthropic's Lean development (sources at https://github.com/xennygrimmato/eth-hamiltonicity). On paper only, the method gives exact marginals and exact sampling, depth-d circuits in $2^{O(dn/\sqrt{\log n})}$, and, through OpenAI's tensor-contraction theorem, polynomial space when the number of Hadamard-type gates is $O(G)$. Conversely, constant-depth circuits with $O(n)$ gates encode #3-SAT with $O(n)$ clauses, so the rate matches that of the refutations. We check the fine-grained supremacy conjectures we found against their sources; a counting form of the Calabro–Impagliazzo–Paturi width reduction refutes the gap conjecture of Morimae and Tamaki's depth-scaling paper. Dense statements (cubic phases, permanents, random circuits) survive. Nothing here concerns approximate sampling, the constants are galactic, and everything is conditional on the claimed refutations, which are under review.

1. Introduction

Part of the case for "quantum computational supremacy" rests on fine-grained hypotheses. Morimae and Tamaki [20, 19, 21], Huang, Newman and Szegedy [14, 15] and Dalzell, Harrow, Koh and La Placa [12] derive lower bounds on the classical simulation of quantum circuits from ETH, SETH and relatives. They usually do so by writing #SAT, or the gap $\mathrm{gap}(f) = \sum_x (-1)^{f(x)}$ of a Boolean function, as a circuit amplitude. The Exponential Time Hypothesis (ETH) of Impagliazzo and Paturi [16] asserts that there is $\delta > 0$ such that 3-SAT on n variables cannot be solved in time $O(2^{\delta n})$; by the sparsification lemma [17] it is equivalent to the same statement for the number m of clauses.

However, in October 2026 two manuscripts appeared, by OpenAI [23] and by Anthropic [3], each claiming a deterministic algorithm that counts the satisfying assignments of a k-CNF in subexponential time, and hence a refutation of ETH. Both come with Lean 4 formalisations of their top-level statements. These are *claimed* results, still under review by the community. We write [A] and [O] for the two manuscripts. The Main Theorem of [A] counts the models of a k-CNF with m clauses on n variables deterministically in time $2^{O(\min\{km/\sqrt{\log m},\ k^2 n \log\log n/\sqrt{\log n}\})}$; the main theorem of [O] does so in $(n+m)^{O_k(1)} 2^{O_k(n \log\log n/\sqrt{\log n})}$ bit operations and polynomial space, and in $2^{O_{k,D}(n/\sqrt{\log n})}$ when $m \le Dn$. Neither manuscript mentions quantum circuits (Section 8). So we ask the converse question: what do the new algorithms give for the classical simulation of quantum circuits, and which of the lower bounds above survive? For depth $d = O(1)$ in arbitrary connectivity we know of no previous $2^{o(n)}$ exact method: the interaction graph of such a circuit can be a bounded-degree expander, whose treewidth is $\Theta(n)$.

Date: November 2026.

2020 *Mathematics Subject Classification.* 68Q12, 68Q25, 81P68.

Key words and phrases. Quantum circuit simulation, strong simulation, Exponential Time Hypothesis, fine-grained complexity, quantum supremacy, tensor networks, Clifford+T.

1.1. **The problem and a running example.** We consider the *exact amplitude problem*: given a circuit C and basis states x, y, output $\langle y|C|x\rangle$ exactly, as an element of a fixed number ring. We fix the setting as follows. A circuit C acts on n qubits and is a sequence of gates from the set

$$\mathcal{G} = \{H, T, T^\dagger, S, S^\dagger, X, Z\} \cup \{\mathrm{CNOT}, \mathrm{CZ}, \mathrm{CS}, \mathrm{SWAP}, \mathrm{CH}\}.$$

It has G two-qubit gates, in arbitrary connectivity, and G_1 single-qubit gates. Put $\omega = e^{i\pi/4}$, so that $\mathbb{Z}[\omega] = \mathbb{Z}[x]/(x^4+1)$, $i = \omega^2$ and $\sqrt{2} = \omega - \omega^3 \in \mathbb{Z}[\omega]$. Every gate $U \in \mathcal{G}$ is stored as a pair (N_U, κ_U) with N_U a matrix over $\mathbb{Z}[\omega]$ and $U = N_U/\sqrt{2}^{\kappa_U}$, where $\kappa_U = 1$ for H and CH and $\kappa_U = 0$ otherwise (for CH, $N = \mathrm{diag}(\sqrt{2}, \sqrt{2}) \oplus \left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$). Put $k(C) := \sum_U \kappa_U$, the number of H and CH gates. Running times are counted as in [A]: operations on words long enough to hold an address, plus table queries; all logarithms are to base 2. The input has length $\mathrm{poly}(n + G + G_1)$.

Example 1 (Running example)**.** Let C act on $n = 2$ qubits: first H on qubit 1, then CNOT with control 1 and target 2, then CZ. So $G = 2$, $G_1 = 1$ and $k(C) = 1$. The state before CZ is $(|00\rangle + |11\rangle)/\sqrt{2}$, so $\langle 11|C|00\rangle = -1/\sqrt{2}$, that is, $(a_0, a_1, a_2, a_3) = (-1, 0, 0, 0)$ over $\sqrt{2}^{-1}$. Each qubit line runs from the first two-qubit gate to the second, so it carries one *wire segment* between them. The amplitude is therefore a sum over the 2^2 values of the two segment bits; we return to this in Example 2.

1.2. **The naive routes and the principle.** At first one might use a known exact method (Table 1). The Feynman path sum takes time $2^{O(G)}$. Tensor contraction [18] takes time $2^{O(\mathrm{tw})}$, and $\mathrm{tw} = \Theta(n)$ is possible at $d = O(1)$. So neither is $2^{o(n)}$ for shallow circuits on expanders. Instead, we rely on a different principle: *a qubit circuit is a sum over binary variables, one per wire segment, coupled by bounded-arity local tables, so it has $O(G)$ shared bits; that is the kind of input the network engine of* [A] *consumes, whatever the connectivity.* What remains is arithmetic. Amplitudes live in $\mathbb{Z}[\omega]$, which is not a subring of $\mathbb{Z}$, nor of the truncated 2-adic ring of [O]. A Kronecker substitution maps $\mathbb{Z}[\omega]$ to $\mathbb{Z}/(2^{4b}+1)\mathbb{Z}$ injectively on a box (Theorem 3.3), and a Galois-conjugate bound shows that the answer lies in the box (Theorem 3.2).

1.3. **What we prove.** We use the following hypotheses, which name exactly what we import from the two manuscripts (Section 2).

Hypothesis ($\mathrm{H_A}$)**.** *The following statements of* [A] *hold as stated there: Lemma 8, case (i) (the reduction from general networks to cliques, Theorem 2.3); Theorem A(a),(b) (one algorithm for the (K_ℓ, λ)-cliques of all levels ℓ, Theorem 2.4); and Theorem B (pieces with parameters $(1/\sqrt{2}, c)$, Theorem 2.5).*

Hypothesis ($\mathrm{H_O}$)**.** *The Boolean tensor-contraction theorem in the appendix of* [O] *holds as stated there (Theorem 2.6).*

Theorem 1.1 (Exact amplitudes; machine-checked)**.** *Assume* ($\mathrm{H_A}$)*. There are an absolute constant c and one deterministic Turing machine that, given a circuit C over $\mathcal{G}$ on n qubits with G two-qubit gates, and $x, y \in \{0,1\}^n$, encoded as a word of length L with $\log_2(L+2) \le (G+2)^{0.9}$, outputs integers a_0, a_1, a_2, a_3 with*

$$\langle y|C|x\rangle = \frac{a_0 + a_1\omega + a_2\omega^2 + a_3\omega^3}{\sqrt{2}^{\,k(C)}}$$

within $\lfloor c \cdot 2^{c(G+2)/\sqrt{\log_2(G+2)}} \rfloor$ steps. Without the condition on L, the same holds within $\lfloor c \cdot 2^{cG/\sqrt{\log_2(G+2)}} (L+2)^{c(G+2)^{1/15}} \rfloor$ steps, and for every $\delta > 0$ there is a machine running within $\lfloor c_\delta \, 2^{\delta G} (L+2)^{c_\delta} \rfloor$ steps.

Since the input word has length $L = 3n + 1 + (G + G_1)(2n + 4)$, we have $L + 2 \le 6n(G + G_1 + 1)$ for $n \ge 1$, so the condition on L holds, for example, whenever $n \ge 1$ and $6n(G + G_1 + 1) \le 2^{G^{0.9}}$; on paper, the proof below also gives $2^{cG/\sqrt{\log G}}\,\mathrm{poly}(n + G_1)$ under the weaker condition $\log(G_1 + 2) \le G^{0.9}$, but that variant is not machine-checked and not claimed.

Theorem 1.2 (Polynomial space; proved on paper only)**.** *Assume* ($\mathrm{H_O}$)*. Put $s := \max\{4G, k(C), 4\}$. The output of Theorem 1.1 can be computed in $\mathrm{poly}(n + G + G_1)\, 2^{O(s/\sqrt{\log s})}$ bit operations and $\mathrm{poly}(n + G + G_1)$ bits of space. In particular, if $k(C) = O(G)$ (for example, if the number of H and* CH *gates is $O(G)$), the time is $2^{O(G/\sqrt{\log G})}\,\mathrm{poly}(n + G_1)$ and the space is polynomial.*

Corollary 1.3 (Marginals and exact sampling; proved on paper only)**.** *Assume* ($\mathrm{H_A}$)*, and let* C, x *be as in Theorem 1.1, including its condition on* L*. For every* $S \subseteq [n]$ *and* $y_S \in \{0,1\}^S$*, the marginal probability* $p(y_S) = \sum_{y_{[n]\setminus S}} |\langle y|C|x\rangle|^2$ *can be computed exactly, as* $(r_0 + r_1\sqrt{2})/2^{k(C)}$ *with* $r_0, r_1 \in \mathbb{Z}$*, within* $2^{c'G/\sqrt{\log G}}\,\mathrm{poly}(n+G_1)$ *operations, for an absolute constant* c'*. Consequently there is a Las Vegas algorithm that outputs* $y \in \{0,1\}^n$ *with probability exactly* $|\langle y|C|x\rangle|^2$ *in expected time* $n \cdot 2^{c'G/\sqrt{\log G}}\,\mathrm{poly}(n+G_1)$*. Under* ($\mathrm{H_O}$) *the same holds in polynomial space when* $k(C) = O(G)$*.*

Corollary 1.4 (Constant and slowly growing depth; proved on paper only)**.** *Assume* ($\mathrm{H_A}$)*. Amplitudes, marginals and exact samples of depth-*d *circuits over* $\mathcal{G}$ *on* $n \geq 4$ *qubits can be computed within* $2^{O(dn/\sqrt{\log n})}$ *operations; the exponent is* $o(n)$ *whenever* $d = o(\sqrt{\log n})$*.*

Proposition 1.5 (The rate is that of #3-SAT; proved on paper only)**.** *There is a polynomial-time map from 3-CNFs* φ *with* n *variables and* m *clauses to circuits* C_φ *over* $\mathcal{G}$ *of depth* $O(1)$ *with* $O(n+m)$ *qubits and* $O(n+m)$ *gates, and to an integer* n'*, such that* $\langle 0|C_\varphi|0\rangle = \#\varphi/2^{n'}$*. Hence an algorithm for exact amplitudes of* G*-gate circuits in time* $T(G)$ *counts the models of 3-CNFs with* $m \geq n$ *clauses in time* $T(O(m))\,\mathrm{poly}(m)$*.*

So, assuming the refutations, the exact amplitude problem for circuits with $O(n)$ gates is exactly as hard as #3-SAT with $O(n)$ clauses, up to constants in the exponent: a bound $2^{o(G/\sqrt{\log G})}$ in Theorem 1.1 would improve the rate of [A] for $m = O(n)$. Before 2026 the same reduction showed, under the counting version #ETH [13], that $2^{\Omega(n)}$ time was needed. Table 1 compares Theorem 1.1 with the classical methods for exact strong simulation.

TABLE 1. Exact strong simulation of a circuit with n qubits, G two-qubit gates and depth d, arbitrary connectivity. The last two rows are conditional.

Method	Time	Space	Source
state vector	2^n poly	2^n	folklore
Feynman path sum	$2^{O(G)}$	poly	folklore
recursive (Savitch-type)	$O(n(2d)^{n+1})$	$O(n \log d)$	[2, Thm. 4.1]
tensor contraction	$2^{O(\mathrm{tw})}$, $\mathrm{tw} = \Theta(n)$ possible at $d = O(1)$	$2^{O(\mathrm{tw})}$	[18]
geometrically local, dim. ≥ 2	subexponential in G	–	[31]
Theorem 1.1	$2^{O(G/\sqrt{\log G})} \leq 2^{O(dn/\sqrt{\log n})}$	as time	under ($\mathrm{H_A}$)
Theorem 1.2	same, if $k(C) = O(G)$	poly	under ($\mathrm{H_O}$)

Against the polynomial-space bound $(2d)^{O(n)} = 2^{O(n \log d)}$ of Aaronson and Chen [2], Theorem 1.2 gives $2^{O(dn/\sqrt{\log n})}$ in the worst case $G = \Theta(dn)$, which is asymptotically smaller only when $d/\log d = O(\sqrt{\log n})$.

1.4. **Contributions.** Our contributions are:

- We prove that one deterministic Turing machine outputs the exact amplitudes of Clifford+T circuits with G two-qubit gates in time $2^{O(G/\sqrt{\log G})}$, assuming ($\mathrm{H_A}$) (Theorem 1.1, Section 3), and we machine-check this theorem in Lean on top of [A]'s development (Section 10.2).
- We give the route through [O] in polynomial space, and show that its restriction $k(C) = O(G)$ comes from the precision cap of [O]'s ring (Theorem 1.2, Section 4).
- We derive exact marginals, exact sampling and the depth-d bound, and prove a converse reduction from #3-SAT, so a better rate here would improve the rate of [A] for $m = O(n)$ (Theorems 1.3 to 1.5, Section 5).
- We check the fine-grained supremacy statements we found, 24 rows in all, against their primary sources, and record which are refuted, which lose their premise and which survive; for this we prove a counting form of the width reduction of Calabro, Impagliazzo and Paturi (Theorem 6.1, Section 6).
- We explain why cubic phases resist the clause-count-free bound of [A], and prove the weaker 2-adic truncation that does transfer (Theorem 7.1, Section 7).
- We check the reductions in exact arithmetic on random small circuits (Section 10).

Mathematically, Theorem 1.2 is an immediate corollary of the appendix theorem of [O], which contracts Boolean tensor networks with one-bit wires: anyone who reads that theorem next to Markov and Shi will see it. What we add is the care needed to make it a theorem: the coefficient ring, the precision, which comes from Theorem 3.2, the resulting restriction $k(C) = O(G)$ in the route through [O] and its absence in the route through [A], marginals and exact sampling, the converse reduction, the check of the literature, and the analysis of cubic phases. We claim the careful statement and proof, not the idea.

1.5. **Status of each claim.** We separate four kinds of claim. *Machine-checked in Lean*: Theorem 1.1 only, as `EthFollowup.Quantum.amplitude_TM_rate'` (the bound under the condition on L), `EthFollowup.Quantum.amplitude_TM_rate` (all circuits) and `EthFollowup.Quantum.amplitude_TM` (every $\delta > 0$), with the axioms `propext`, `Classical.choice` and `Quot.sound` only. The sources are in our public repository [30], https://github.com/xennygrimmato/eth-hamiltonicity, at commit `ca17a37` (Section 10.2). The bound under the condition on L and the bound for all circuits are separate Lean theorems; the Lean proof of the former reuses the machine of the latter, so one machine gives both. In Lean, ($\mathrm{H_A}$) is not a hypothesis: [A]'s Lean development proves the forms of the three imported results that our Lean proof uses (Section 2), and the kernel checks them together with ours. *Proved on paper only*: everything else, in particular Theorem 1.2, Theorems 1.3 and 1.4, Theorem 1.5, Theorem 6.1, Theorem 6.2, Theorem 7.1 and the entries of the list in Section 6. We do not claim these as machine-checked. *Assumed*: on paper, ($\mathrm{H_A}$), ($\mathrm{H_O}$) or the main theorem of [A] or [O], as each result states; in Lean, nothing beyond the three axioms. Either way the results stand or fall with the claimed refutations (Section 9). The bound after Theorem 7.1 is a sketch and is not claimed. *Checked numerically*: the reductions of Sections 3 and 4 and the marginals of Theorem 1.3, on circuits with $n \le 6$ and $G \le 20$, and Theorem 7.1, on cubic phases with $n \le 7$ and $m \le 14$, all in exact arithmetic (Section 10). The circuit of Theorem 1.5 is not scripted; only its analogue for IQP circuits is. The running example is checked by hand.

Outline. Section 2 recalls the imported results. Section 3 proves Theorem 1.1, Section 4 gives the route through [O], and Section 5 the corollaries and the converse. Section 6 checks the fine-grained supremacy literature, and Section 7 treats cubic phases. Sections 8 to 10 give related work, limitations, and the numerical checks and the formalisation.

2. The imported results

In this section we recall the definitions and results of [A] and [O] that we use, in the notation of [A] except where a letter clashes with ours, as in [29]. We state them exactly as we use them, so that ($\mathrm{H_A}$) and ($\mathrm{H_O}$) can be checked against the sources. Conventions of [A]: $\log = \log_2$ and $[x = u] \in \{0, 1\}$.

Definition 2.1 ([A, Definition 1])**.** A function $F\colon X \times X' \to \mathbb{Z}$ *factors through c bits* if there are $f\colon X \times \{0,1\}^c \to \mathbb{Z}$ and $f'\colon X' \times \{0,1\}^c \to \mathbb{Z}$ with $F(z, z') = \sum_{\theta \in \{0,1\}^c} f(z, \theta) f'(z', \theta)$. In particular, for any $h\colon X \to \{0,1\}^c$ and $h'\colon X' \to \{0,1\}^c$ the function $[h(z) = h'(z')] = \sum_\theta [h(z) = \theta][h'(z') = \theta]$ factors through c bits.

Definition 2.2 ([A, Definition 2])**.** Let $P, B, \tau \ge 1$ be integers. A *network on J with parameters P, B, τ* consists of a finite set J of nodes; for each node a a finite set S_a of settings and a function $W_a\colon S_a \to \mathbb{Z}$; and for each unordered pair $\{a, b\}$ of distinct nodes a function $\Psi_{a,b}\colon S_a \times S_b \to \mathbb{Z}$ (with $\Psi_{b,a}(z_b, z_a) = \Psi_{a,b}(z_a, z_b)$), an integer $c_{a,b} = c_{b,a} \ge 0$ and a factorisation $(f_{a,b}, f_{b,a})$ of $\Psi_{a,b}$ through $c_{a,b}$ bits. The settings of a are encoded as bit strings of one fixed length $P_a \le P$, and

(i) $\sum_{b \ne a} c_{a,b} \le P$ for every $a \in J$;

(ii) all values of all W_a, $\Psi_{a,b}$, $f_{a,b}$ are below 2^B in absolute value;

(iii) an algorithm receives P, B and the numbers P_a, $c_{a,b}$ as input, and can decide membership in S_a and compute any value of W_a, $\Psi_{a,b}$, $f_{a,b}$ with at most τ operations.

The *value* of the network is $Z = \sum_{z \in \prod_a S_a} \prod_{a \in J} W_a(z_a) \prod_{\{a,b\}} \Psi_{a,b}(z_a, z_b)$, and its *total* is $U = \sum_{(a,b)\colon a \ne b} c_{a,b}$.

In the (K, λ)-*clique* of [A, Definition 3] each of K nodes holds a λ-bit string for each other node, $c_{w,w'} = \lambda$, $\Psi_{w,w'} = [z_w(w') = z_{w'}(w)]$, and the tables $|W_w| < 2^B$ are queried at unit cost.

Imported Lemma 2.3 ([A, Lemma 8, case (i)]). *Let $K \geq 2$, $g = 3K^8$ and $\zeta = 1/(2K)$. Take a network on $[g]$ with parameters P, B, τ and total U, and put $L = \lceil (U + gP/K)/K^2 \rceil$ and $B' = \lceil 2gP/K \rceil + g^2 B$. If an algorithm outputs the value of every (K, L)-clique with $|W_w| < 2^{B'}$ within T operations, then the value of the network is computed within*

$$\big(K^g + T \cdot 2^{2gP/K} \cdot \tau\big) \cdot \big(g(P+B)\big)^{C_1} \tag{2.1}$$

operations, by one procedure, for an absolute constant C_1.

Imported Theorem 2.4 ([A, Theorem A]). *Suppose there are pieces with parameters (ρ, c) in the sense of* [A, Definition 7]*, and let $\alpha' = \log(1/\rho)$. Fix an integer $x_0 \geq 18$ with $2^{x_0} \geq c + 5$ and define $K_0 = 2^{x_0}$, $\zeta_\ell = 2^{-\ell-5}$, $g_\ell = 2^{3\ell+x_0} K_\ell$, $K_{\ell+1} = g_\ell^2$, $e_0 = \frac{1}{2}$ and $e_{\ell+1} = (\rho + 3\zeta_\ell) e_\ell + 3/K_\ell$.*

(a) *$\log K_\ell = (3x_0 + 6)2^\ell - 6\ell - 2x_0 - 6$, so $K_\ell < K_{\ell+1} \leq K_\ell^{10}$; and $\frac{1}{2}\rho^\ell \leq e_\ell \leq 0.85\,\rho^\ell$, where $\rho^\ell \leq ((3x_0+6)/\log K_\ell)^{\alpha'}$.*

(b) *One deterministic algorithm, given $\ell \geq 0$, $\lambda, B \geq 1$ and query access to tables with $|W_w| < 2^B$, outputs the value of the (K_ℓ, λ)-clique within*

$$2^{e_\ell K_\ell^2 \lambda} \cdot \big(K_\ell(\lambda + B)\big)^{C_0 K_\ell^2} \tag{2.2}$$

operations, with C_0 depending only on c.

Imported Theorem 2.5 ([A, Theorem B]). *There are pieces with parameters $(1/\sqrt{2}, c)$ for an absolute constant c.*

Imported Theorem 2.6 ([O, Appendix, Boolean tensor contraction], up to notation). *A Boolean tensor network is a bipartite multigraph in which each edge carries a single bit; each vertex v has an ordered list $\delta(v)$ of incident edges and a tensor T_v assigning a ring element to each Boolean assignment to those edges; its contraction is $Z = \sum_{x \in \{0,1\}^E} \prod_v T_v(x|_{\delta(v)})$, an isolated vertex contributing its scalar entry. Fix positive constants C, A, b. Let $s \geq 2$ be an integer, and let $\mathcal{L}$ be the total length of the graph and tensor descriptions of a bipartite tensor network with at most Cs edges and maximum degree at most $\lceil s^{9/10} \rceil$. Suppose that the number of vertices is bounded by a fixed polynomial in s, and that the entries belong to*

$$\mathcal{B}_{P_0,\ell} = (\mathbb{Z}/2^{P_0}\mathbb{Z})[V]/(V^\ell), \qquad 1 \leq P_0, \ell \leq Cs.$$

Suppose, uniformly for every vertex and Boolean choice of its ports, that its entry can be computed from the description in time $(s+\mathcal{L})^b 2^{As^{9/10}}$ and space $(s+\mathcal{L})^b$. Then the contraction can be computed deterministically in $(s+\mathcal{L})^{O(1)} 2^{O(s/\sqrt{\log s})}$ bit operations and $(s+\mathcal{L})^{O(1)}$ bits of space, with constants depending only on C, A, b.

[A]'s Lean development proves Theorem 2.3 (in a form weaker by a polynomial factor), Theorem 2.4(b) and Theorem 2.5 for arbitrary networks in its own format, and our Lean proof of Theorem 1.1 applies them directly to the circuit network (Section 10.2). The appendix of [O] is not formalised, so Theorem 1.2 is proved on paper only.

3. Proof of Theorem 1.1: one bit per wire segment

In this section we write the amplitude as a sum over the bits of the wire segments (Section 3.1), map its coefficients from $\mathbb{Z}[\omega]$ into the integers, read the sum as a network in the sense of [A] (Section 3.3), and bound the running time (Section 3.4).

3.1. The sum over paths on wire segments.

For a qubit q let $\gamma_{q,1}, \ldots, \gamma_{q,r_q}$ be the two-qubit gates acting on q, in circuit order; q is *touched* if $r_q \geq 1$. Every single-qubit gate on a touched line is multiplied into the next two-qubit gate on that line, and the single-qubit gates after γ_{q,r_q} are multiplied into γ_{q,r_q} from the left. This is legitimate because a single-qubit gate on line q commutes with every gate that does not act on q. The result is a list of G *merged* gates $\hat{\Gamma}_i = \hat{N}_i/\sqrt{2}^{\hat{\kappa}_i}$ with $\hat{N}_i \in M_4(\mathbb{Z}[\omega])$, where $\hat{\kappa}_i$ is the sum of the κ of the gates merged into it. An untouched line contributes the scalar $\langle y_q|R_q|x_q\rangle = u_q/\sqrt{2}^{\kappa_q}$, where R_q is the product of its gates. The merging costs $\mathrm{poly}(n + G + G_1)$ operations, since all coefficients have $O(k(C) + G + G_1)$ bits.

A *segment* is a piece of line q between $\gamma_{q,j}$ and $\gamma_{q,j+1}$; it carries one bit σ_e. The input leg of $\gamma_{q,1}$ is fixed to x_q and the output leg of γ_{q,r_q} to y_q. With n_t touched lines there are $s_0 = \sum_q (r_q - 1) = 2G - n_t \leq 2G$

segments. For a merged gate i write $\sigma_{\text{in}}(i), \sigma_{\text{out}}(i) \in \{0,1\}^2$ for the values on its input and output legs (a segment bit or a fixed x_q, y_q).

Lemma 3.1 (Sum over paths)**.** *Put $k_t := \sum_i \hat{\kappa}_i$ and $A := \sum_{\sigma \in \{0,1\}^{s_0}} \prod_{i=1}^{G} \hat{N}_i[\sigma_{\text{out}}(i), \sigma_{\text{in}}(i)] \in \mathbb{Z}[\omega]$. Then*

$$\langle y|C|x\rangle = \Big(\prod_{q \text{ untouched}} u_q \Big) \cdot \frac{A}{\sqrt{2}^{k(C)}}, \qquad k(C) = k_t + \sum_{q \text{ untouched}} \kappa_q,$$

and $A = \sqrt{2}^{k_t} \langle y_t|C_t|x_t\rangle$, where C_t is the circuit C restricted to the touched lines.

Proof. Insert $\sum_{b \in \{0,1\}} |b\rangle\langle b|$ on every segment and expand the matrix product. □

3.2. **Coefficients: a Galois bound and a Kronecker map.** For $\alpha = \sum_j a_j \omega^j \in \mathbb{Z}[\omega]$ write $\|\alpha\|_\infty = \max_j |a_j|$ and $\|\alpha\|_1 = \sum_j |a_j|$. Multiplication by ω^j permutes the coefficients up to sign, so $\|\alpha\beta\|_1 \le \|\alpha\|_1 \|\beta\|_1$.

Lemma 3.2 (Galois bound)**.** *Let M be a product of gates from $\mathcal{G}$ (on any number of qubits), let κ be the sum of their κ_U, and let N be the product of their numerators, so $M = N/\sqrt{2}^{\kappa}$. Then every entry α of N satisfies $\|\alpha\|_\infty \le 2^{\kappa/2}$.*

Proof. The Galois group of $\mathbb{Q}(\omega)/\mathbb{Q}$ consists of $\sigma_t \colon \omega \mapsto \omega^t$, $t \in \{1,3,5,7\}$; it is abelian and contains complex conjugation σ_7. Hence, for a unitary U with entries in $\mathbb{Q}(\omega)$, $\sigma_t(U)\sigma_t(U)^* = \sigma_t(UU^*) = I$, so $\sigma_t(U)$ is unitary. Applying σ_t entrywise to all factors, and using $\sigma_t(\sqrt{2}) = \pm\sqrt{2}$, we get $\sigma_t(N) = (\pm\sqrt{2})^{\kappa}\sigma_t(M)$ with $\sigma_t(M)$ unitary, so $|\sigma_t(\alpha)| \le 2^{\kappa/2}$. Finally $\sum_t \omega^{rt} = 4[r \equiv 0]$ for $r \in \{-3, \dots, 3\}$, which gives $a_j = \frac{1}{4}\sum_t \sigma_t(\alpha)\,\omega^{-jt}$ and $|a_j| \le 2^{\kappa/2}$. □

By Theorem 3.1 and Theorem 3.2 (applied to C_t), $\|A\|_\infty \le 2^{k_t/2}$, and every entry of every $\hat{N}_i$ has $\|\cdot\|_\infty \le 2^{\hat{\kappa}_i/2}$, hence $\|\cdot\|_1 \le 4 \cdot 2^{\hat{\kappa}_i/2}$.

Lemma 3.3 (Kronecker map)**.** *Let $b \ge 2$ and $M_b := 2^{4b} + 1$. The substitution $x \mapsto 2^b$ defines a ring homomorphism $\kappa_b \colon \mathbb{Z}[\omega] = \mathbb{Z}[x]/(x^4+1) \to \mathbb{Z}/M_b\mathbb{Z}$, because $2^{4b} \equiv -1 \pmod{M_b}$. It is injective on $\{\alpha : \|\alpha\|_\infty \le 2^{b-1} - 1\}$, and α is recovered from $\kappa_b(\alpha)$ by taking the representative in $(-M_b/2, M_b/2]$ and reading its balanced base-2^b digits.*

Proof. For α in the box, $|\sum_j a_j 2^{bj}| \le (2^{b-1}-1)(1+2^b+2^{2b}+2^{3b}) < 2^{4b-1} < M_b/2$, and balanced digits are unique. □

Put $\text{Kron}_b(\alpha) := \sum_j a_j 2^{bj} \in \mathbb{Z}$, an integer representative of $\kappa_b(\alpha)$, and fix $b := \lceil k_t/2 \rceil + 2$, so that $\|A\|_\infty \le 2^{b-1} - 1$.

No ring embedding is available ($x^4 = -1$ has no solution modulo 4, nor modulo V); a homomorphism injective on a box containing the answer suffices.

3.3. **The network.** Fix $g \ge 1$ and a partition of the merged gates into groups $V_1, \dots, V_g$ with $|V_a| \le \lceil G/g \rceil$ (some groups may be empty). A segment is *internal* to a if both its end gates lie in V_a, and *crossing* otherwise. Let int_a be the number of segments internal to a and P_a the number of crossing segments with an end in V_a.

Definition 3.4. The network $\mathcal{N}(C, x, y, (V_a), b)$ on $[g]$ has settings $S_a = \{0,1\}^{P_a}$, the bits of the crossing segments at a;

$$W_a(z_a) := \text{Kron}_b\Big(\sum_{\text{internal bits of } a} \prod_{i \in V_a} \hat{N}_i[\sigma_{\text{out}}(i), \sigma_{\text{in}}(i)] \Big),$$

the sum taken in $\mathbb{Z}[\omega]$ over the 2^{int_a} values of the internal segment bits, with the crossing bits read from z_a; and $\Psi_{a,b}(z_a, z_b) = \prod_e [z_a(e) = z_b(e)]$ over the segments e crossing between a and b, which factors through $c_{a,b} := \#\{e \text{ crossing between } a \text{ and } b\}$ bits by Theorem 2.1, with $f_{a,b} \in \{0,1\}$.

Example 2 (Running example, continued)**.** In Example 1 both lines are touched, so there are no untouched scalars. The merged gates are $\hat{\Gamma}_1 = \text{CNOT}\,(H \otimes I)$, with $\hat{\kappa}_1 = 1$, and $\hat{\Gamma}_2 = \text{CZ}$, and there are $s_0 = 2G - n_t = 2$ segments. Since $\hat{N}_1[\sigma, 00] = 1$ for $\sigma \in \{00, 11\}$ and 0 otherwise, and $\text{CZ}[11,11] = -1$, Theorem 3.1 gives $A = -1$ and $\langle 11|C|00\rangle = -1/\sqrt{2}$. Take $g = 2$, $V_1 = \{1\}$ and $V_2 = \{2\}$. Both segments cross, so $P_1 = P_2 = c_{1,2} = 2$ and $U = 4 \le 4G$. Here $k_t = 1$, so $b = 3$ and $M_b = 2^{12} + 1$. The value of the network is $Z = \sum_\sigma W_1(\sigma)W_2(\sigma) = -1$, and Theorem 3.3 decodes $A = -1$.

Proposition 3.5 (Parameters and value)**.** *Let* $P := 4\lceil G/g\rceil + 1$, $B := 3b + 2 + 4\lceil G/g\rceil + \lceil k_t/2\rceil$ *and* $\tau := 2^{2\lceil G/g\rceil}\,\mathrm{poly}(G, k(C))$. *Then* $\mathcal{N} = \mathcal{N}(C, x, y, (V_a), b)$ *is a network on* $[g]$ *with parameters* P, B, τ *in the sense of Theorem 2.2, its total is* $U \le 4G$, *and its value* Z *satisfies* $Z \equiv \kappa_b(A) \pmod{M_b}$. *Hence* A *is obtained from* Z *by Theorem 3.3.*

Proof. Condition (i). $\sum_{b\neq a} c_{a,b} = P_a$, and every merged gate has four legs, so $P_a \le 4|V_a| \le P$. This is where bounded gate arity enters; nothing about connectivity is used. *The total.* $U = \sum_a P_a = 2 \cdot \#\{\text{crossing segments}\} \le 2s_0 \le 4G$. *Condition (ii).* The values of Ψ and f are 0 or 1. For W_a, the inner sum α_a has $\|\alpha_a\|_\infty \le \|\alpha_a\|_1 \le 2^{\mathrm{int}_a} \prod_{i\in V_a} 4 \cdot 2^{\hat{\kappa}_i/2}$, and $\mathrm{int}_a \le 2|V_a|$, so $\log \|\alpha_a\|_\infty \le 4\lceil G/g\rceil + k_t/2$; and $|\operatorname{Kron}_b(\alpha)| < \|\alpha\|_\infty 2^{3b+1}$. So $|W_a| < 2^B$. *Condition (iii).* Membership in S_a is trivial, Ψ and f compare bits, and $W_a(z_a)$ is a sum of $2^{\mathrm{int}_a} \le 2^{2\lceil G/g\rceil}$ products of $|V_a|$ elements of $\mathbb{Z}[\omega]$ with $O(k(C)+G)$-bit coefficients, read from the precomputed merged gates. *Value.* $\operatorname{Kron}_b(\alpha) \equiv \kappa_b(\alpha) \pmod{M_b}$ and κ_b is a ring homomorphism, so $Z \equiv \kappa_b\big(\sum_z \prod_a \alpha_a(z_a) \prod_{\{a,b\}} \Psi_{a,b}\big)$. The factors Ψ restrict the sum to settings that agree on every crossing segment, and the inner sums run over the internal segments, so the argument of κ_b is A. □

3.4. **Running time.** The algorithm merges the gates, computes k_t, b and the untouched scalars, chooses a level ℓ (below), puts $K := K_\ell$ and $g := 3K^8$, takes any partition into groups of size at most $\lceil G/g\rceil$, computes $P, B, P_a, c_{a,b}$, and runs the procedure of Theorem 2.3 on $\mathcal{N}$ with the algorithm of Theorem 2.4(b) at level ℓ as the clique algorithm. It reduces the value Z modulo M_b, decodes A by Theorem 3.3, and multiplies by the untouched scalars.

Hypotheses. Theorem 2.3 needs a network on $[g]$ with integer parameters $P, B, \tau \ge 1$ and $g = 3K^8$; Theorem 3.5 supplies it. It then needs an algorithm for every (K, L)-clique with $|W| < 2^{B'}$: Theorem 2.4(b) at level ℓ, with $\lambda = L \ge 1$ and $B = B' \ge 1$, gives $T = 2^{e_\ell K^2 L}(K(L + B'))^{C_0 K^2}$. Theorem 2.4 needs pieces, which Theorem 2.5 supplies with $\rho = 1/\sqrt{2}$, $\alpha' = \frac{1}{2}$, and c, x_0 absolute. The circuit reaches the clique algorithm only through the tables that Theorem 2.3 builds from the W_a and $f_{a,b}$, and Theorem 2.3 itself searches the K^g groupings of the g nodes for one with small pairwise totals, so no balancing beyond the partition is needed.

The exponent. By (2.1) and (2.2), with $e := e_\ell \le 1$ and $K^2 L \le U + gP/K + K^2$, the cost is 2^X with

$$X \le 1 + \max\Big\{g \log K,\ e\big(U + \tfrac{gP}{K} + K^2\big) + C_0 K^2 \log\big(K(L + B')\big) + \tfrac{2gP}{K} + \log\tau\Big\} + C_1 \log\big(g(P + B)\big). \tag{3.1}$$

For $G < G_0 := K_0^{30}$, an absolute constant, the algorithm instead sums over the $\le 2G$ segment bits directly, in $O(1) \cdot \mathrm{poly}(n + G_1)$ operations. For $G \ge G_0$ let ℓ be the largest index with $K_\ell^{30} \le G$. Then $K_{\ell+1} > G^{1/30}$, and $K_{\ell+1} \le K_\ell^{10}$ gives $G^{1/300} < K \le G^{1/30}$, so $g = 3K^8 \le 3G^{8/30}$ and, by Theorem 2.4(a),

$$e \le 0.85\Big(\frac{3x_0 + 6}{\log K}\Big)^{1/2} \le \frac{0.85\,(300(3x_0 + 6))^{1/2}}{\sqrt{\log G}} =: \frac{c_e}{\sqrt{\log G}}.$$

Also $gP \le g(4G/g + 5) = 4G + 5g$, $L \le U + gP + 1 = O(G)$ and $B' \le 2gP + 1 + g^2 B = O(G^{1.6}(1 + k(C)))$. Since $k(C) \le G + G_1$ and $\log(G_1 + 2) \le G^{0.9}$, we have $\log(k(C) + 2) = O(G^{0.9})$. The terms of (3.1) are bounded as follows:

$$\begin{aligned}
e\,U &\le 4c_e\, G/\sqrt{\log G},\\
(e + 2)\, gP/K &\le 3(4G + 5g)/G^{1/300} = O(G^{1-1/300}),\\
eK^2 + C_0 K^2 \log(K(L + B')) &= O\big(G^{1/15}(\log G + \log(k(C) + 2))\big) = O(G^{0.97}),\\
\log\tau &\le 2G/g + O(\log G + \log(k(C) + 2)) = O(G^{1-8/300} + G^{0.9}),\\
g \log K &= O(G^{8/30} \log G),\\
C_1 \log(g(P + B)) &= O(\log G + \log(k(C) + 2)) = O(G^{0.9}).
\end{aligned}$$

Every term except the first is $o(G/\sqrt{\log G})$, uniformly in the circuit. The value Z has $O(U + gB)$ bits, so reducing and decoding it, and the preparation, cost $\mathrm{poly}(n+G+G_1)$. Hence the total is $2^{cG/\sqrt{\log G}}\,\mathrm{poly}(n+G_1)$ for an absolute c (any $c > 4c_e$ works for large G). This proves Theorem 1.1. □

Remark 3.6 (The single-qubit gate count). G_1 enters polynomially (input, merging) and through $k(C) \le G+G_1$, which sets the table bit length $B = O(G/g + k(C))$; B reaches the exponent only via $C_0 K^2 \log(K(L + B'))$,

hence the hypothesis $\log(G_1+2) \le G^{0.9}$. If it fails, the direct sum over $2G$ segment bits is quasi-polynomial in the input length. The route through [A] is exact whatever $k(C)$ is.

4. The route through [O]: polynomial space when $k(C) = O(G)$

In this section we prove Theorem 1.2. We check the hypotheses of Theorem 2.6 one by one, with $s = \max\{4G, k(C), 4\}$ and $C = 4$.

- *Bipartite, one-bit edges, at most Cs edges.* The merged gates form one side. Every segment is subdivided by a binary equality tensor, as [O] does in its own CNF encoding; the equality vertices form the other side. There are $2s_0 \le 4G \le s$ edges. Fixed legs $(x_q,\, y_q)$ are absorbed into the gate tensor, and a gate all of whose legs are fixed is an isolated vertex.
- *Maximum degree at most $\lceil s^{9/10} \rceil$.* The degree is at most 4, and $\lceil s^{9/10} \rceil \ge 4$ for $s \ge 4$.
- *Number of vertices polynomial in s.* It is $G + s_0 \le 3G$.
- *Entries in $\mathcal{B}_{P_0,\ell}$ with $1 \le P_0, \ell \le Cs$.* Map $x \mapsto V$. The entries of $\hat{N}_i$ become integer polynomials of degree at most 3, reduced modulo 2^{P_0}, and the equality tensors have entries $0, 1$. Over $\mathbb{Z}$ the contraction is $Q(V) := \sum_\sigma \prod_i \hat{N}_i(V)$, of degree at most $3G$, so with $\ell := 3G + 1 \le s$ nothing is truncated and the contraction in $\mathcal{B}_{P_0,\ell}$ is $Q(V) \bmod 2^{P_0}$. Reducing Q modulo $V^4 + 1$ over $\mathbb{Z}/2^{P_0}$ gives A modulo 2^{P_0} coefficientwise, because $Q \bmod (x^4+1) = A$ in $\mathbb{Z}[\omega]$. By Theorem 3.2, $|a_j| \le 2^{k_t/2} < 2^{P_0 - 1}$ for $P_0 := \lceil k(C)/2 \rceil + 3$, so the balanced residues are the a_j. Here $P_0 \le s/2 + 4 \le 2s$. *This is the only place where $k(C)$ enters, and it is why $s \ge k(C)$*: [O] works modulo a power of 2 with precision at most Cs.
- *Entries computable in time $(s+\mathcal{L})^b 2^{As^{9/10}}$ and space $(s+\mathcal{L})^b$.* Each entry is a lookup in a merged gate, precomputed in polynomial time and space; $\mathcal{L} = \mathrm{poly}(G + k(C))$.

The conclusion of Theorem 2.6 gives $(s+\mathcal{L})^{O(1)} 2^{O(s/\sqrt{\log s})}$ bit operations and $(s+\mathcal{L})^{O(1)}$ space; merging and the untouched scalars cost polynomial time and space. □

When $k(C) \gg G$ we see no way around the precision cap inside [O]'s ring: the numerator of a two-qubit Clifford+T unitary can need arbitrarily many bits. A different route avoids the cap at a weaker rate: Bhattacharjee's factor-system theorem [4, Thm. 5.3] sums products of bounded-scope factors with values in any commutative $\mathbb{Q}$-algebra in $\mathrm{poly}(n)\, 2^{O(\rho(n))}$ operations of the algebra and polynomial storage, where $\rho(n) = n(\log\log n)^2/\sqrt{\log n}$. Applied to our path sum with the algebra $\mathbb{Q}(\omega)$ it suggests polynomial space with no restriction on $k(C)$; we have not checked the bit complexity in $\mathbb{Q}(\omega)$, which that theorem does not state, and we do not claim this variant.

5. Corollaries, and why the rate is that of #3-SAT

In this section we prove Theorems 1.3 and 1.4 and the converse reduction, Theorem 1.5.

5.1. Marginals: the doubled network.

Proof of Theorem 1.3, marginals. Untouched lines contribute $|u_q|^2/2^{\kappa_q}$ for $q \in S$ and 1 for $q \notin S$. On the touched lines take two copies of the merged gates, the second with every entry conjugated ($\overline{a_0 + a_1\omega + a_2\omega^2 + a_3\omega^3} = a_0 - a_3\omega - a_2\omega^2 - a_1\omega^3$). Both copies get input x. For $q \in S$ the output legs of both copies are fixed to y_q; for a touched $q \notin S$ the two output legs are joined into one new *trace* segment. The resulting sum over paths has $2G$ merged gates and at most $2s_0 + n_t \le 4G$ segments, and its value is $R = \sum_{y_{\bar{S}}} A(y)\overline{A(y)} = 2^{k_t} p_t(y_S)$, where p_t is the marginal of C_t. R is real, so $R = r_0 + r_1\sqrt{2}$ with $r_0, r_1 \in \mathbb{Z}$. For every t, $\sigma_t(R) = 2^{k_t} \sum_{y_{\bar{S}}} |\langle y|\sigma_t(C_t)|x\rangle|^2 \in [0, 2^{k_t}]$ (the proof of Theorem 3.2), so $\|R\|_\infty \le 2^{k_t}$ and $b := k_t + 2$ suffices in Theorem 3.3. Sections 3.3 and 3.4 now apply verbatim with $2G$ merged gates, $U \le 8G$, $P = 4\lceil 2G/g \rceil + 1$ and k_t replaced by $2k_t$; the exponent changes by a constant factor. Under ($\mathrm{H_O}$) take $s = \max\{8G, 2k(C), 4\}$, $\ell = 6G + 1$ and $P_0 = k(C) + 3$. □

Proof of Theorem 1.3, sampling. Draw $u \in [0, 1)$ uniformly, revealing its binary digits only when needed, and output the $y \in \{0,1\}^n$ whose interval in the lexicographic partition of $[0, 1)$ into intervals of lengths $|\langle y|C|x\rangle|^2$ contains u; then $\Pr[y] = |\langle y|C|x\rangle|^2$ exactly. The digits $y_1, y_2, \ldots$ are found one at a time: if the interval of the prefix $y_{<j}$ is $[L, L + p(y_{<j}))$, then $y_j = 0$ iff $u < L + p(y_{<j}0)$. This takes n marginal computations, with endpoints in $\mathbb{Z}[\sqrt{2}]/2^{k(C)}$, compared exactly with dyadic rationals (compare squares of integers). A

comparison is undecided after t digits with probability at most 2^{-t}, so $\log n + O(1)$ digits are needed in expectation. □

Proof of Theorem 1.4. $G \le dn/2$ and $G_1 \le dn$. If $G \ge 2$ and $\log(G_1+2) \le G^{0.9}$, use Theorems 1.1 and 1.3 and $G/\sqrt{\log G} \le dn/\sqrt{\log n}$ ($n \ge 4$). Otherwise $G \le (\log(dn+2))^{1.12}$ and the direct sum over at most $4G$ segment bits suffices. □

5.2. **The converse reduction.**

Proof of Theorem 1.5. Make every variable occur at most three times, parsimoniously (Tovey's copying [28]): replace the $d \ge 2$ occurrences of a variable v by fresh copies $v_1, \dots, v_d$ and add the 2-clauses $(\bar v_1 \vee v_2), \dots, (\bar v_{d-1} \vee v_d), (\bar v_d \vee v_1)$, which force all copies to be equal. The result φ' has $n' = O(n+m)$ variables, $m' = O(n+m)$ clauses of width at most 3, each variable in at most three clauses, and $\#\varphi' = \#\varphi$.

Use one qubit per variable, one qubit y_j per clause and one ancilla per clause, and put $D := \prod_j D_j$ with $D_j|x,y\rangle = (-1)^{y_j \cdot [\text{clause } j \text{ is violated by } x]}|x,y\rangle$. Since [clause j violated] is the AND of the negated literals, D_j is a multi-controlled Z on y_j and at most three literal qubits, conjugated by X on negative literals. A controlled-controlled-controlled-Z is exact over $\mathcal{G}$ with one ancilla returned to $|0\rangle$: compute the AND of two controls into the ancilla with a Toffoli gate, apply CCZ, and uncompute; Toffoli and CCZ have exact Clifford+T circuits with seven T gates and $O(1)$ CNOT and H gates [22]. Every gadget touches $O(1)$ qubits and every variable qubit lies in at most three gadgets, so the gadgets can be coloured greedily into $O(1)$ layers of disjoint gadgets. Let $C_\varphi = H^{\otimes(n'+m')} D H^{\otimes(n'+m')}$, the Hadamards acting on variable and clause qubits only. Then

$$\langle 0|C_\varphi|0\rangle = 2^{-(n'+m')} \sum_{x,y} \prod_j (-1)^{y_j [\text{clause } j \text{ violated}]} = 2^{-(n'+m')} \sum_x \prod_j 2[\text{clause } j \text{ satisfied}] = \frac{\#\varphi}{2^{n'}}.$$

C_φ has depth $O(1)$ and $O(n+m)$ gates. Reading off $\#\varphi$ needs only the amplitude. □

This construction is not scripted; the analogue for IQP circuits [9] $\langle 0|H^{\otimes n} D_f H^{\otimes n}|0\rangle = \mathrm{gap}(f)/2^n$ for degree-3 f, with CCZ compiled into Clifford+T, is checked by `qc_check.py` (Section 10).

6. Fine-grained supremacy statements after the refutations

Throughout this section "refuted" means *refuted assuming the claimed refutations of ETH*. Entries 1, 2, 5, 12 and 13 of the list in Section 6 need only the main theorem of either [A] or [O], which is what the labs have formalised in Lean.

6.1. **Two facts about SETH-type statements.**

ETH fails if and only if $s_k = 0$ for every $k \ge 3$, where s_k is the infimum of the exponents δ with k-SAT in time $2^{\delta n}$ [17]; then $s_\infty = \lim_k s_k = 0$, so the Strong Exponential Time Hypothesis $s_\infty = 1$ [16] fails too. Several statements below are phrased for CNFs of unbounded width with at most cn clauses. For the decision version, Calabro, Impagliazzo and Paturi [11] relate them to k-SAT through Schuler's width reduction [26]. We need the counting version, which follows from the same reduction because its two branches are disjoint.

Lemma 6.1 (Counting width reduction)**.** *Let $k \ge 1$ and $s > 0$, and suppose a deterministic algorithm counts the models of k-CNFs with n' variables and m' clauses in time $2^{sn'}\,\mathrm{poly}(n',m')$. Then the models of an arbitrary CNF with n variables and m clauses can be counted deterministically in time $2^{sn}\exp\big(2^{-sk}(m + n/k)\big)\,\mathrm{poly}(n,m)$.*

Proof. Write $\#_X F$ for the number of models of F over the variable set X. If F has a clause $C = \{l_1, \dots, l_{k'}\}$ with $k' > k$, put $C' = \{l_1, \dots, l_k\}$, let $F_1 = (F \setminus \{C\}) \cup \{C'\}$, and let F_0 be F with $l_1, \dots, l_k$ set to false (satisfied clauses deleted, false literals removed). Since $C' \subseteq C$, the models of F_1 are exactly the models of F that satisfy C'; the models of F that violate C' are the extensions of the models of F_0 by $l_1 = \cdots = l_k = \text{false}$. These two sets are disjoint, so $\#_X F = \#_X F_1 + \#_{X \setminus \mathrm{var}(C')} F_0$. Recurse until no clause is longer than k (a leaf whose formula contains the empty clause counts 0), and count each leaf with the given algorithm. Along a root-to-leaf path every left branch removes a clause longer than k and right branches create none, so there are at most m left branches, and a leaf reached by r right branches has $n - kr$ variables, so $r \le n/k$. The number of leaves reached by exactly r right branches is at most $\binom{m+r}{r}$ [11, proof of Lemma 5]. Hence the time is at most $\mathrm{poly}(n,m) \sum_{r \le n/k} \binom{m+r}{r} 2^{s(n-kr)} \le \mathrm{poly}(n,m)\, 2^{sn} \sum_r \binom{m+\lfloor n/k \rfloor}{r} 2^{-skr} \le \mathrm{poly}(n,m)\, 2^{sn} (1+2^{-sk})^{m+n/k}$. □

Corollary 6.2. *Assume the main theorem of* [A] *or of* [O]. *For all $c, \varepsilon > 0$ there is a deterministic algorithm that counts the models of CNFs (of any width) with n variables and at most cn clauses in time $O(2^{\varepsilon n})$. Hence* $\operatorname{gap}(f) = 2^n - 2\#f$ *is computed in the same time.*

Proof. Put $s = \varepsilon/3$ and choose k with $2^{-sk}(c+1)\log e \le \varepsilon/3$. For this fixed k, the main theorems count the models of k-CNFs in time $2^{o(n')}\operatorname{poly}(n', m') \le 2^{sn'}\operatorname{poly}(n', m')$ for n' large; Theorem 6.1 gives $2^{2\varepsilon n/3}\operatorname{poly}(n)$. □

Theorem 6.2 does not give CNF-SAT with $m = n^{O(1)}$ clauses of unbounded width in time $2^{(1-\delta)n}\operatorname{poly}(m)$: the term $2^{-sk}m$ then forces $k = \Omega(\log n)$, where neither main theorem is subexponential. That form of SETH is not known to follow, and some statements below rest on it.

6.2. **The table.** The list below gives the statements we found, with the wording checked against the primary sources (arXiv versions cited). "Witness family" means the circuits built in the proof. *Refuted* means the statement itself is false under the refutations; *premise refuted* means a theorem whose hypothesis is now false, while its conclusion is not contradicted by anything here; *survives* means we know no consequence of the refutations that touches it.

Fine-grained quantum-supremacy statements, assuming the refutations. Each entry gives the statement and its source, its status, and the reason.

1. [20] Conj. 1: ETH as $2^{\Omega(m)}$ for 3-CNF decision, m clauses. *Status:* refuted. [A]: #3-SAT in $2^{O(m/\sqrt{\log m})}$.
2. [20] Conj. 2 ("SETH"): deterministic CNF-SAT, $\le cn$ clauses, any width, needs $2^{(1-a)n}$. *Status:* refuted. Theorem 6.2.
3. [20] Conj. 3 (NSETH, polynomial-size circuits). *Status:* survives. General circuits.
4. [20] Conj. 4–6: nondeterministic gap $\neq 0$ for poly-size, log-depth, reversible circuits needs $2^{(1-a)n}$. *Status:* survive. $\operatorname{poly}(n)$-size instances; Theorem 1.1 gives nothing below 2^n.
5. [20] Conj. 7: nondeterministic $\operatorname{gap}(f) \neq 0$, 3-CNF with m clauses, needs $2^{\Omega(m)}$. *Status:* refuted. $\operatorname{gap}(f) = 2^n - 2\#f$ in deterministic $2^{O(m/\sqrt{\log m})}$.
6. [20] Thm. 6, [15] Thm. 1: ETH ⇒ no $2^{o(t)}$ strong simulation of Clifford+T with t T gates (additive error $2^{-(3t+14)/7}$, resp. deciding $\langle 0|C|0\rangle \neq 0$). *Status:* premise refuted. Witness circuits have $O(t)$ gates and are solved exactly in $2^{O(t/\sqrt{\log t})}$; with $\operatorname{poly}(n)$ Cliffords the conclusion is open.
7. [20] Thms. 7, 8: Conj. 7 ⇒ no $2^{o(t)}$ multiplicative-error sampling of Clifford+T, resp. t-qubit PBC. *Status:* premise refuted. Witness circuits ($O(t)$ gates) sampled exactly in $2^{O(t/\sqrt{\log t})}$; conclusions open.
8. [20] Thm. 10: Conj. 2 ⇒ exact probabilities with h Hadamards not in $2^{(1-a)h}$. *Status:* premise refuted. Witness circuits have $\Theta(ch^2)$ gates; conclusion open.
9. [20] Thm. 12: ETH ⇒ efficient stabilizer rank of t-Toffoli resources is $2^{\Omega(t)}$. *Status:* premise refuted. Conclusion open.
10. [14] Thm. 3: CNF-SAT needs $2^{n-o(n)}\operatorname{poly}(m)$ ⇒ strong simulation of $\operatorname{poly}(n)$-size circuits to precision $2^{-n}/2$ needs $2^{n-o(n)}$. *Status:* survives. Premise is the CNF-SAT form with $\operatorname{poly}(m)$ clauses, not known to follow; unconditional monotone bound unaffected.
11. [15] Thm. 2: an $O(2^{2.2451\cdot 10^{-8}N})$ simulator (N T gates) gives 3-SAT in $O(1.3^n)$. *Status:* moot. The refutations give the consequent outright.
12. [19] Conj. 1 ("SETH", $\le cn$ clauses, deterministic decision). *Status:* refuted. Theorem 6.2; Thm. 1 loses its premise.
13. [19] Conj. 2: nondeterministic $\operatorname{gap}(f) \neq 0$, CNF with $\le cn$ clauses, needs $2^{(1-a)n}$. *Status:* refuted. Theorem 6.2 (counting form). Thm. 2 loses its premise; its witness circuits have $\ge nm$ qubits.
14. [19] Conj. 3–4 (OV), 5–6 (3-SUM). *Status:* survive. OV is implied by SETH, not equivalent; 3-SUM unrelated.
15. [12] poly3-NSETH(a). *Status:* survives. Worst case has $\Theta(n^3)$ monomials (Section 7).
16. [12] per-int-NSETH(b). *Status:* survives. Dense permanents.
17. [12] poly3-ave-SBSETH(a'). *Status:* survives. Average case, dense, additive error.
18. [12] Thm. 2: ETH ⇒ deterministic poly3-NONBALANCED needs 2^{an}. *Status:* premise refuted. Reduction instances have $O(n)$ monomials; worst case open.
19. [21] Conj. 1: random cubic, multiplicative error, NTIME[n^2] oracle, 2^{an}. *Status:* survives. Dense, average case, approximate.
20. [21] Conj. 2: random quadratic plus worst-case log-depth circuit. *Status:* survives. General log-depth circuits.

21. [21] Conj. 3: random quadratic plus worst-case 3-CNF with m clauses, 2^{am}. *Status:* survives (not shown safe). $\Theta(n^2+m)$ gates; nothing for $m=O(n)$.
22. [2] QUATH (poly time, random circuits with $\geq n^2$ gates) and HOG. *Status:* unaffected. Average case, approximate.
23. [10] basic QSETH: no quantum $O(2^{n(1-\delta)/2}m^{O(1)})$ CNF-SAT. *Status:* survives. CNF-SAT form; the k-CNF variant of their footnote 8 is refuted.
24. $2^{\Omega(t)}$ T-count hardness with unboundedly many Cliffords; $\chi(|T\rangle^{\otimes t})=2^{\Omega(t)}$ [8]. *Status:* open. ETH support gone; best known exact stabilizer-rank simulation $2^{0.3963t}$ poly [24], after [8, 6]; compressed circuits have $\Theta(t^2)$ gates.

Entry 13 was left open in our working notes: there we checked only the decision form of the width–density duality. Theorem 6.1 settles it.

6.3. **Sampling versus exact amplitudes.** By Theorem 1.3, any hardness claim for sampling, with multiplicative or additive error, stated as $2^{\Omega(n)}$ for a family with $G=o(n\sqrt{\log n})$ gates is refuted, since exact samples are approximate ones. Nothing is said about families with $G\geq n\sqrt{\log n}$, which include every random-circuit sampling proposal at depth $d\gg\sqrt{\log n}$, nor about polynomial-time hardness or unconditional separations for shallow circuits [27, 7].

The estimates of [12] (208 IQP qubits and 500 gates, assuming poly3-NSETH(1/2) and a cost linear in the number of gates) extrapolate from the dense worst case to a sparse family, which is asymptotically $2^{O(n/\sqrt{\log n})}$ under the refutations; at $n=208$ this has no force.

7. Cubic phases: only a 2-adic truncation transfers

Let $f=\sum_{t\in T}m_t$ over $\mathbb{F}_2$, with m_t monomials of degree at most 3 in n variables and $|T|=m$, and $\mathrm{gap}(f)=\sum_x(-1)^{f(x)}$. This is the object of poly3-NSETH in [12], and $\langle 0|H^{\otimes n}D_fH^{\otimes n}|0\rangle=\mathrm{gap}(f)/2^n$ for the diagonal circuit D_f built from Z, CZ and CCZ. Compiling CCZ into Clifford+T gives $G=\Theta(m)$, so Theorem 1.1 computes $\mathrm{gap}(f)$ in $2^{O(m/\sqrt{\log m})}$, which is $2^{o(n)}$ only for $m=o(n\sqrt{\log n})$. For CNFs, [A] has a second bound that does not depend on m. Does one hold for cubic phases? As far as we can see, no.

7.1. **Why the clause-count-free mechanism does not transfer.** As we read [A, Appendix D], the m-free bound writes [no clause is violated at ξ] by inclusion–exclusion over the set $V(\xi)$ of violated clauses, truncates it at a number of marks of order $n/\log\Delta$, and handles the assignments with more violated clauses separately, using that they contribute *zero* to #SAT. For phases there is no such set: $(-1)^{f(x)}=\prod_t(1-2m_t(x))=\sum_{\Pi\subseteq V(x)}(-2)^{|\Pi|}=(-1)^{|V(x)|}$, where $V(x)$ is the set of monomials equal to 1 at x, so every assignment contributes ±1, and the assignments with many "violated" monomials carry a non-zero total in general. This is checked on random instances (check C4 in Section 10). What does transfer is a weaker, 2-adic truncation.

Proposition 7.1 (2-adic truncation; proved on paper)**.** *For $S\subseteq T$ let $V(S)$ be the set of variables occurring in the monomials of S.*

(a) $\mathrm{gap}(f)=\sum_{S\subseteq T}(-2)^{|S|}\,2^{n-|V(S)|}$.

(b) *Every term with $|S|\geq|V(S)|+2$, in particular every term with $|S|\geq n+2$, is divisible by 2^{n+2}. Since $|\mathrm{gap}(f)|\leq 2^n$, $\mathrm{gap}(f)$ is the representative in $(-2^{n+1},2^{n+1}]$ of $\sum_{|S|\leq n+1}(-2)^{|S|}2^{n-|V(S)|}$ modulo 2^{n+2}.*

Proof. (a) Expand $\prod_t(1-2m_t(x))$ and use $\sum_x\prod_{t\in S}m_t(x)=2^{n-|V(S)|}$. (b) The 2-adic valuation of the term of S is $|S|+n-|V(S)|$. □

The truncation at $n+1$ marks is sharp up to a constant factor: truncating at $\lfloor n/2\rfloor$ marks gives wrong answers on some instances (check C3). So the budget is $\Theta(n)$ marks, where [A] pays about $n/\log\Delta$.

Sketch (not proved here). Feeding (b) into the incidence encoding that [O] uses for sparse $\mathbb{F}_2$-polynomial systems (each monomial a test vertex of weight $1-2\prod_{v\in t}z_{t,v}$, precision 2^{n+2}, $\ell=1$, incidence degree bound $3(n+1)$, and branching on the variables in more than $(3m/n)\sqrt{\log n}$ monomials) should give $O(n\log(em/n)+\sqrt{n}\log n)$ wires, hence, under ($\mathrm{H_O}$), $\mathrm{gap}(f)$ in time $2^{O(n\log(em/n)/\sqrt{\log n})}\,\mathrm{poly}(m)$: modulo 2^{n+2} only sets of at most $n+1$ marked monomials matter, at about $\log(em/n)$ bits each. We have not checked this application line by line against the degree and entry hypotheses of [O] (the maximum degree should be $n^{3/4}2^{o(\sqrt{\log n})}\leq s^{9/10}$ and the entries are products of 1 and -2, so we expect no obstacle). If

correct, it is $2^{o(n)}$ if and only if $m \le n2^{o(\sqrt{\log n})}$, more than Theorem 1.1 gives, and it is worse than brute force for dense f ($m = \Theta(n^3)$ gives $2^{O(n\sqrt{\log n})}$). Making cubic phases m-independent would decide dense poly3-NONBALANCED in $2^{o(n)}$ and refute poly3-NSETH(a) for every a; we record it as a target, not a result.

8. Related work

We organise related work by idea: fine-grained supremacy, exact strong simulation, the counting engines, and permanents.

Fine-grained supremacy from ETH and SETH. Morimae and Tamaki [20, 19, 21], Huang, Newman and Szegedy [14, 15] and Dalzell, Harrow, Koh and La Placa [12] derive lower bounds on classical simulation from ETH, SETH, NSETH and their relatives. Aaronson and Chen [2] base random-circuit sampling on QUATH and HOG, and Buhrman, Patro and Speelman [10] formulate quantum analogues of SETH. In contrast, we go in the converse direction: we assume the refutations and ask which of these statements survive (Section 6).

Exact strong simulation. The state vector and the Feynman path sum are folklore; Aaronson and Chen [2, Thm. 4.1] give a recursive polynomial-space method. Markov and Shi [18] contract tensor networks in time exponential in the treewidth, and Wahl and Strelchuk [31] obtain time subexponential in the number of gates for geometrically local circuits in dimension at least 2. For circuits with t T gates, stabilizer-rank methods run in time exponential in t [8, 6, 24]. These methods are unconditional and, unlike ours, some are practical. In contrast, our bound is subexponential in G in arbitrary connectivity, but it is conditional and galactic (Table 1).

The counting engines. OpenAI [23] and Anthropic [3] claim deterministic subexponential #SAT algorithms, with Lean developments, under review. Our earlier work [29] counts Hamiltonian cycles on top of [A]'s development, and we reuse its conventions. Neither manuscript mentions quantum circuits: in the source of [O] the words "quantum", "qubit" and "amplitude" do not occur, and [A] mentions tensor networks, including Markov and Shi's paper on simulating quantum computation by tensor contraction [18], only in its related work. Nor did we find the corollary in the press or discussion threads we searched. Closest to our work is the appendix theorem of [O] (Theorem 2.6), of which Theorem 1.2 is an immediate corollary. Bhattacharjee's factor-system theorem over commutative $\mathbb{Q}$-algebras [4, Thm. 5.3] is also close: it does not mention quantum circuits, but it suggests a polynomial-space route without the restriction $k(C) = O(G)$, at the weaker rate $2^{O(G(\log\log G)^2/\sqrt{\log G})}$ (Section 4). In contrast, we work out the coefficient ring and the precision, take the route through [A] at the rate $2^{O(G/\sqrt{\log G})}$ without the restriction $k(C) = O(G)$, and machine-check that route.

Permanents and BosonSampling. BosonSampling amplitudes are permanents of $n \times n$ submatrices of Haar-random unitaries [1], and Ryser's 2^n poly(n) formula [25] is still essentially the best exact method (for matrices of poly(n)-bit integers it has been improved only to $2^{n-\Omega(\sqrt{n/\log n})}$ [5]). Nothing above applies to them. A qubit circuit is a sum over *binary* variables, one per wire segment, coupled by bounded-arity local tables, so it has $O(G)$ shared bits. A dense permanent is a sum over *permutations*; written as an interferometer it has $\Theta(m^2)$ beam splitters on modes carrying up to n photons, and any local encoding seems to need about $\log n$ bits per row. Informal notes in the project repository [30], not claimed here, argue that "row-local" networks for the permanent in the model of [A] (rows grouped into clusters of at most s) have total at least $n\log(n/s)$, too much for the engine to beat 2^n; only sparse matrices go through. In contrast to the qubit case, the conjecture per-int-NSETH of [12] is therefore untouched (Section 6).

9. Limitations

The results stand or fall with the claimed refutations. We list this and the other limitations plainly.

(a) *Everything algorithmic is conditional* on ($\mathrm{H_A}$), ($\mathrm{H_O}$), or (for Theorem 6.2) the main theorem of either manuscript. Expert review of the manuscripts is pending; if the hypotheses fail, the results lose their proofs, and we offer no independent route. Theorem 1.1 is machine-checked on top of [A]'s Lean development, so it needs that development, built from source, to be sound. Our other results use the imported statements in sharp forms that are not machine-checked in the generality needed here; their proofs are on paper, written with AI assistance, and not yet checked by a human.

(b) *Approximate sampling; many gates.* The results concern exact simulation of circuits with $G = o(n\sqrt{\log n})$ gates. Random-circuit sampling rests on the non-collapse of the polynomial hierarchy and on average-case conjectures about approximate amplitudes (QUATH, HOG [2]) for circuits with $G \geq n^2$ gates; none of this is touched, nor are cross-entropy benchmarking and polynomial-depth families. For $G \geq n\sqrt{\log n}$ the bound is no better than 2^n.

(c) *The constants are galactic.* Using [A, Theorem A(a)] with the smallest admissible $x_0 = 18$, the leading term of the exponent is $e_\ell U$ with $U \approx 4G$ when most wire segments cross the partition, and $15/\sqrt{\log G} \leq e_\ell \leq 114/\sqrt{\log G}$ at the level chosen in Section 3.4. For depth d, so $G \approx dn/2$, the leading term alone is below n only if $\log n$ exceeds roughly $900\, d^2$ (with the lower estimate) to $5.2 \cdot 10^4\, d^2$ (with the upper one), and lower-order terms only make this worse. [O] gives no explicit constants. Experiments with 50 to 1000 qubits are not affected.

(d) *Denominators in the route through* [O]. Its 2-adic precision cap forces $k(C) = O(s)$.

(e) *Sketches.* The bound after Theorem 7.1 is a sketch; the circuit of Theorem 1.5 is not scripted; the reading of [A, Appendix D] is ours.

(f) *Literature.* Every entry of the list in Section 6 was checked against the arXiv text of its source, and all citations were checked against the primary sources on 30 Oct 2026, with a further spot-check on 4 Nov 2026; theorem and conjecture numbers refer to the arXiv versions named in the bibliography. We have not searched systematically for prior statements of Theorem 1.1; given how direct it is, it may well have been noticed independently.

(g) *Scale of the numerical checks.* They cover $n \leq 6$ and $G \leq 20$ and test the reductions (encodings, decodings, parameters, ring emulation), not the engines, which are replaced by brute force and variable elimination.

10. Numerical checks and formalisation

In this section we describe the numerical checks of the reductions and the Lean development of Theorem 1.1.

10.1. **Numerical checks.** The scripts are `research/scripts/qc_*.py` in the project repository [30]; they are not part of this submission. They use exact integer and $\mathbb{Z}[\omega]$ arithmetic only and print PASS or FAIL; `qc_lib.py` implements the objects of Sections 3 to 5.

`qc_check.py [seed] [count]` draws random circuits ($n \in [2,6]$, $G \in [1,20]$, $G_1 \in [0,25]$, random gates, x, y, and balanced or contiguous partitions into $g \leq 4$ groups) and checks: (Q1) Theorem 3.1 against the state vector; (Q2) the integer network by brute force over the crossing bits, by the literal sum of Theorem 2.2 on small cases and by variable elimination, decoded by Theorem 3.3; (Q3) decoding of the unreduced Kronecker polynomial; (Q4) condition (i), $U \leq 4G$ and $|W_a| < 2^B$; (Q5) the contraction in $\mathcal{B}_{P_0,\ell}$ with $P_0 = \lceil k/2 \rceil + 3$, $\ell = 3G + 1$, reduced modulo $V^4 + 1$ and lifted; (Q6) Theorem 3.2, with $|\sigma_t(A)|^2 \leq 2^k$ checked exactly in $\mathbb{Z}[\sqrt{2}]$; (Q7) marginals via the doubled network, in $\mathbb{Z}[\sqrt{2}]$ and equal to 1 for $S = \emptyset$. It also checks 20 IQP instances ($\mathrm{gap}(f)/2^n$ with CCZ compiled into Clifford+T) and four ablations that must be caught: b one bit too small, $\ell = 4$, P_0 one bit short, and no conjugation in the doubled network. Seed 1 (150 circuits): 529 networks by brute force, 418 also by the literal sum, 65 truncated-ring brute forces, 91 marginals, at most 23 crossing segments, 48 partitions with a node of more than 14 boundary bits skipped. Seed 2: 544, 432, 62, 86, 24, 39. Both PASS, with all ablations caught; the agreement is equality of integers.

`qc_cubic_gap.py` checks, for random f with $n \leq 7$, $m \leq 14$: (C1)–(C2) Theorem 7.1; (C3) the valuation bound, and failure of truncation at $\lfloor n/2 \rfloor$ marks on some instance; (C4) a non-zero total of the assignments with more than $\lfloor n/3 \rfloor$ "violated" monomials on some instance. Seeds 1 (120 instances) and 3 (400) PASS.

10.2. **Formalisation.** We machine-check Theorem 1.1 in Lean 4 on top of [A]'s development, as three theorems in the namespace `EthFollowup.Quantum`: `amplitude_TM_rate'` (the stated bound under the condition on L), `amplitude_TM_rate` (all circuits) and `amplitude_TM` (every $\delta > 0$). The sources are in our public repository [30], `https://github.com/xennygrimmato/eth-hamiltonicity`, as of commit `ca17a37`, in the directories `lean/EthFollowup/Quantum` and `lean-bridge-anthropic/bridge/EthFollowup/Quantum`; they are not part of this submission. The semantics a reader must check are short: the gate matrices over $\mathbb{C}$, the circuit matrix as the ordered product, and the amplitude $\langle y|C|x\rangle$ (`Circuit.lean`, §1), and the input and output encodings (§7). The theorems depend only on the axioms `propext`, `Classical.choice` and `Quot.sound`, and fresh `leanchecker` replays of their closures (`EthFollowup.Quantum.QRate`, which contains

`QTheorem`, and `EthFollowup.Quantum.Paths`) passed. Theorem 1.2, Theorem 1.3 and Theorem 6.2 are proved on paper only and are not claimed as machine-checked; nor are the other results listed as proved on paper in Section 1.5.

Acknowledgements

This work was done with AI assistance (Claude Code, Anthropic): the observation, the proofs, the Lean development, the literature checks and the numerical verification scripts were produced in sessions with Claude Code. Theorem 1.1 is checked by the Lean kernel; everything else has been checked numerically or against the primary sources as described, but awaits human expert review. Responsibility for any errors rests with the author.

References

[1] S. Aaronson and A. Arkhipov, *The computational complexity of linear optics*, Theory Comput. **9** (2013), 143–252.
[2] S. Aaronson and L. Chen, *Complexity-theoretic foundations of quantum supremacy experiments*, Proc. 32nd Computational Complexity Conference (CCC 2017), LIPIcs, vol. 79, 2017, pp. 22:1–22:67; theorem numbers as in arXiv:1612.05903v2.
[3] Claude (Anthropic), *A $2^{o(n)}$ time algorithm for 3-SAT*, preprint dated 26 October 2026. Paper and Lean formalisation in the repository `https://github.com/anthropics/formal-math`, folder `eth`; we use commit `16a5ce41304fcdaea4e0be7c132e1c361dc26da5`.
[4] D. Bhattacharjee, *Subexponential weighted counting and exact optimisation*, Hexagon preprint hexagon:2610.00594v1, 29 October 2026, `https://hexagonmath.org/abs/2610.00594`.
[5] A. Björklund, *Below all subsets for some permutational counting problems*, Proc. 15th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2016), LIPIcs, vol. 53, 2016, pp. 17:1–17:11.
[6] S. Bravyi and D. Gosset, *Improved classical simulation of quantum circuits dominated by Clifford gates*, Phys. Rev. Lett. **116** (2016), no. 25, 250501.
[7] S. Bravyi, D. Gosset and R. König, *Quantum advantage with shallow circuits*, Science **362** (2018), no. 6412, 308–311.
[8] S. Bravyi, G. Smith and J. A. Smolin, *Trading classical and quantum computational resources*, Phys. Rev. X **6** (2016), no. 2, 021043.
[9] M. J. Bremner, R. Jozsa and D. J. Shepherd, *Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy*, Proc. R. Soc. A **467** (2011), no. 2126, 459–472.
[10] H. Buhrman, S. Patro and F. Speelman, *The quantum strong exponential-time hypothesis*, arXiv:1911.05686 (v2), 2019. Published as *A framework of quantum strong exponential-time hypotheses*, Proc. 38th STACS, LIPIcs, vol. 187, 2021, pp. 19:1–19:19.
[11] C. Calabro, R. Impagliazzo and R. Paturi, *A duality between clause width and clause density for SAT*, Proc. 21st Annual IEEE Conference on Computational Complexity (CCC 2006), 2006, pp. 252–260.
[12] A. M. Dalzell, A. W. Harrow, D. E. Koh and R. L. La Placa, *How many qubits are needed for quantum computational supremacy?*, Quantum **4** (2020), 264; arXiv:1805.05224.
[13] H. Dell, T. Husfeldt, D. Marx, N. Taslaman and M. Wahlén, *Exponential time complexity of the permanent and the Tutte polynomial*, ACM Trans. Algorithms **10** (2014), no. 4, Art. 21.
[14] C. Huang, M. Newman and M. Szegedy, *Explicit lower bounds on strong quantum simulation*, arXiv:1804.10368 (v2), 2018; published in IEEE Trans. Inform. Theory **66** (2020), no. 9, 5585–5600.
[15] C. Huang, M. Newman and M. Szegedy, *Explicit lower bounds on strong simulation of quantum circuits in terms of T-gate count*, arXiv:1902.04764 (v2), 2019.
[16] R. Impagliazzo and R. Paturi, *On the complexity of k-SAT*, J. Comput. System Sci. **62** (2001), no. 2, 367–375.
[17] R. Impagliazzo, R. Paturi and F. Zane, *Which problems have strongly exponential complexity?*, J. Comput. System Sci. **63** (2001), no. 4, 512–530.
[18] I. L. Markov and Y. Shi, *Simulating quantum computation by contracting tensor networks*, SIAM J. Comput. **38** (2008), no. 3, 963–981.
[19] T. Morimae and S. Tamaki, *Depth-scaling fine-grained quantum supremacy based on SETH and qubit-scaling fine-grained quantum supremacy based on Orthogonal Vectors and 3-SUM*, arXiv:1902.08382 (v1), 2019. Version 2 (November 2019), with R. Hayakawa as additional author, is retitled *Fine-grained quantum supremacy based on Orthogonal Vectors, 3-SUM and All-Pairs Shortest Paths*.
[20] T. Morimae and S. Tamaki, *Fine-grained quantum computational supremacy*, arXiv:1901.01637 (v4), 2019; published in Quantum Inf. Comput. **19** (2019), no. 13&14, 1089–1115.
[21] T. Morimae and S. Tamaki, *Additive-error fine-grained quantum supremacy*, Quantum **4** (2020), 329; arXiv:1912.06336.
[22] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.
[23] OpenAI, *The Exponential Time Hypothesis is false*, preprint dated 17 October 2026, revised 25 October 2026. Paper and Lean formalisation in the repository `https://github.com/openai/math`.
[24] H. Qassim, H. Pashayan and D. Gosset, *Improved upper bounds on the stabilizer rank of magic states*, Quantum **5** (2021), 606; arXiv:2106.07740.
[25] H. J. Ryser, *Combinatorial Mathematics*, Carus Mathematical Monographs, no. 14, Mathematical Association of America, 1963.

[26] R. Schuler, *An algorithm for the satisfiability problem of formulas in conjunctive normal form*, J. Algorithms **54** (2005), no. 1, 40–44.
[27] B. M. Terhal and D. P. DiVincenzo, *Adaptive quantum computation, constant depth quantum circuits and Arthur–Merlin games*, Quantum Inf. Comput. **4** (2004), no. 2, 134–145.
[28] C. A. Tovey, *A simplified NP-complete satisfiability problem*, Discrete Appl. Math. **8** (1984), no. 1, 85–89.
[29] V. Tulsyan, *Counting Hamiltonian cycles in sparse graphs in subexponential time, assuming the new #SAT algorithms*, Hexagon preprint hexagon:2610.00672v1, October 2026; source and Lean development at https://github.com/xennygrimmato/eth-hamiltonicity.
[30] V. Tulsyan, *eth-hamiltonicity: Lean 4 formalisation and research notes*, public repository https://github.com/xennygrimmato/eth-hamiltonicity, 2026; Lean sources as of commit `ca17a37`.
[31] T. B. Wahl and S. Strelchuk, *Simulating quantum circuits using efficient tensor network contraction algorithms with subexponential upper bound*, Phys. Rev. Lett. **131** (2023), no. 18, 180601; arXiv:2208.01498.