Improved Conditional Lower Bounds for the Strong Simulation of Clifford+T Circuits

Franz J. Schreiber[1, *]

[1] *Dahlem Center for Complex Quantum Systems, Freie Universität Berlin, 14195 Berlin, Germany*

Strong simulation of Clifford+T circuits is often analyzed in terms of the number N of T-gates, the standard measure of non-Clifford cost. It is therefore natural to ask how small an exponent η in a runtime of the form $O^*(2^{\eta N})$ can be without implying unexpected consequences for classical algorithms. Following the reduction strategy of Huang, Newman, and Szegedy [1], we relate strong simulation of Clifford+T circuits to 3-SAT. Our contribution is quantitative rather than conceptual: we replace their Toffoli-based SAT encoding by a direct $4T$ implementation of Boolean clause evaluations, avoiding the final conjunction tree, and tighten the 3-SAT-specific sparsification analysis. As a consequence, if polynomial-size Clifford+T circuits with N T-gates could be strongly simulated in time $O^*(2^{\eta N})$ for $\eta < 5.07 \times 10^{-7}$, then one would obtain an $O^*(1.3^n)$ algorithm for 3-SAT, improving over currently quoted worst-case bases. This improves the previous explicit constant by roughly one order of magnitude.

I. INTRODUCTION

Strong simulation of quantum circuits is a natural benchmark for the power and limits of classical algorithms. For Clifford+T circuits, the relevant parameter is often the number N of T-gates, since T-gates capture the non-Clifford resource and many simulation methods scale primarily with N. This raises a basic fine-grained question: how small can the exponent η in a runtime of the form $O^*(2^{\eta N})$ be without implying an unexpected consequences in classical algorithms?

A route to such lower bounds was given by Huang, Newman, and Szegedy [1], who reduced 3-SAT to strong simulation of Clifford+T circuits and thereby obtained an explicit conditional lower bound in terms of T-count. In this work we follow their strategy closely. Our contribution is quantitative rather than conceptual: we optimize the argument at two points. First, we replace the Toffoli-based SAT encoding by a direct Clifford+T implementation of Boolean clause evaluations using clean-target $4T$ AND/OR gadgets. We also project onto all clause-output bits being 1, rather than computing a final conjunction. For a mixed 3-CNF with m_i clauses of size i, this uses $4(m_2 + 2m_3)$ T-gates. Second, we tighten the 3-SAT-specific sparsification analysis used to pass from formula size to the number of variables.

Combining these two improvements yields a stronger explicit conditional lower bound. In particular, if polynomial-size Clifford+T circuits with N T-gates could be strongly simulated in time $O^*(2^{\eta N})$ for

$$\eta < 5.07 \times 10^{-7}, \tag{1}$$

then one would obtain an $O^*(1.3^n)$ algorithm for 3-SAT. Thus, unless the 1.3^n benchmark for 3-SAT is broken, the exponent η in T-count-based strong simulation cannot be this small. An $O^*(1.3^n)$-time algorithm would improve over currently known worst-case exponential bases for 3-SAT, which are slightly better than $O^*(1.30704^n)$ [1] [2–4]. It would not by itself imply a standard collapse in complexity theory. However, despite concentrated efforts, the improvements in worst-case complexity of 3-SAT solvers have been very incremental within the last two decades. We take this as weak evidence that the optimal runtime of 3-SAT is close to $O^*(1.3^n)$. This translates to (at least) weak evidence that our conditional lower bound is true.

II. ENCODING k-SAT IN CLIFFORD+T

We present an encoding of k-SAT into Clifford+T circuits. For an m clause k-CNF, our construction uses at most $4(k-1)m$ T-gates and $O(km)$ Clifford gates. Let

$$\phi(x_1, \ldots, x_n) = \bigwedge_{i=1}^{m} C_i, \qquad C_i = \bigvee_{j=1}^{k_i} \ell_{i,j}, \tag{2}$$

with literals $\ell_{i,j} \in \{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}$.

* f.schreiber@fu-berlin.de.

[1] Slight improvements over $O^*(1.30704^n)$ are proven, but not numerically characterized.

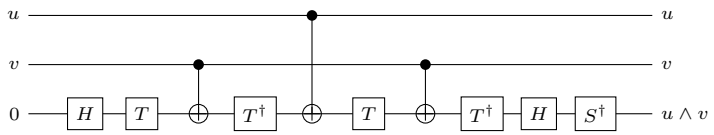

Figure 1. A $4T$ AND gadget. Except for the T-gates, all gates are Clifford.

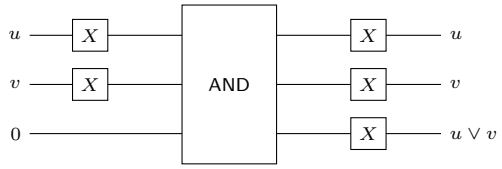

Figure 2. An OR gadget obtained from the AND gadget by De Morgan's law.

Definition 1 (Formula and clause-evaluation length)**.** *The* formula length $L(\phi)$ *is the number of binary nonlinear gates ($\wedge$ and $\vee$) in the standard binary decomposition of ϕ. Thus*

$$L(\phi) := \sum_{i=1}^{m}(k_i - 1) + (m - 1) = \sum_{i=1}^{m} k_i - 1. \tag{3}$$

Define $(r)_+ := \max\{0, r\}$. We will also use the clause-evaluation length

$$R(\phi) := \sum_{i=1}^{m}(k_i - 1)_+, \tag{4}$$

which counts only the OR-gates needed to evaluate the clauses, not the final AND tree. For a k-CNF with m clauses, $R(\phi) \le (k-1)m$. For a mixed 3-CNF with m_j clauses of size j, one has

$$R(\phi) = m_2 + 2m_3. \tag{5}$$

Lemma 2 (AND gadget)**.** *There exists a three-qubit Clifford+T circuit* AND *with T-count exactly 4 such that*

$$\mathsf{AND}\,|u, v, 0\rangle = |u, v, u \wedge v\rangle\,, \qquad u, v \in \{0, 1\}, \tag{6}$$

with no input-dependent relative phase on the subspace with target initialized to $|0\rangle$.

Proof. Use the circuit in Fig. 1. A direct computation in the computational basis, with wire order $|u, v, t\rangle$, gives

$$|0, 0, 0\rangle \mapsto |0, 0, 0\rangle\,, \qquad |0, 1, 0\rangle \mapsto |0, 1, 0\rangle\,, \tag{7}$$
$$|1, 0, 0\rangle \mapsto |1, 0, 0\rangle\,, \qquad |1, 1, 0\rangle \mapsto |1, 1, 1\rangle\,. \tag{8}$$

Thus the target-zero subspace has exactly the desired classical AND action and carries no input-dependent phase. Phases may occur on basis states with target initially 1, but those states are never used as inputs to the gadget in the construction below. For optimality of the Margolus-type construction, see [5, 6]. □

Lemma 3 (OR gadget)**.** *There exists a three-qubit Clifford+T circuit* OR *with T-count exactly 4 such that*

$$\mathsf{OR}\,|u, v, 0\rangle = |u, v, u \vee v\rangle\,, \qquad u, v \in \{0, 1\}. \tag{9}$$

Proof. By De Morgan:

$$u \vee v = \neg(\neg u \wedge \neg v). \tag{10}$$

Hence, as shown in Fig. 2,

$$\mathsf{OR} = (X \otimes X \otimes X)\,\mathsf{AND}\,(X \otimes X \otimes I), \tag{11}$$

so the T-count is unchanged and the target-zero action is again phase-free. □

Lemma 4 (Reversible evaluator circuit (clause outputs))**.** *For every CNF formula ϕ on n variables with m clauses and clause-evaluation length $R := R(\phi)$, there exists a reversible Clifford+T circuit U_ϕ on $n + a + m$ qubits, with $a \le R$, such that*

$$U_\phi \left|x\right\rangle \left|0^a\right\rangle \left|0^m\right\rangle = \left|x\right\rangle \left|g(x)\right\rangle \left|C_1(x), \ldots, C_m(x)\right\rangle \qquad \forall x \in \{0,1\}^n, \tag{12}$$

for some garbage function $g : \{0,1\}^n \to \{0,1\}^a$. The circuit U_ϕ consists of $4R$ T-gates and $O(R + m)$ additional Clifford gates.

Proof. Compute each unit clause into its own output bit using only Clifford gates. Compute each clause C_i of size $k_i \ge 2$ by a binary tree/chain of OR gadgets, requiring exactly $k_i - 1$ such gadgets, and use the last target as the output bit of that clause. All intermediate targets are fresh ancillas and are left as garbage. Hence the number of nonlinear gadgets is

$$\sum_{i=1}^{m} (k_i - 1)_+ = R(\phi) = R, \tag{13}$$

so the T-count is $4R$. A left-associated implementation uses at most one fresh target per nonlinear gadget, so $a \le R$. The rest is Clifford overhead only. □

Lemma 5 (Garbage neutrality under Hadamards)**.** *For every $g \in \{0,1\}^a$,*

$$\left\langle 0^a\right| H^{\otimes a} \left|g\right\rangle = \frac{1}{\sqrt{2^a}}. \tag{14}$$

Proof.

$$H^{\otimes a} \left|g\right\rangle = 2^{-a/2} \sum_{z \in \{0,1\}^a} (-1)^{g \cdot z} \left|z\right\rangle, \tag{15}$$

so the $\left|0^a\right\rangle$ coefficient is $2^{-a/2}$. □

Theorem 6 (Multi-output amplitude encoding)**.** *Define*

$$C_\phi := \left(H^{\otimes n} \otimes H^{\otimes a} \otimes I^{\otimes m}\right) U_\phi \left(H^{\otimes n} \otimes I^{\otimes (a+m)}\right), \tag{16}$$

where U_ϕ is from Lemma 4. Then

$$\alpha_\phi := \left\langle 0^n, 0^a, 1^m\right| C_\phi \left|0^n, 0^a, 0^m\right\rangle = \frac{\#\mathrm{SAT}(\phi)}{2^{n+a/2}}. \tag{17}$$

Hence $\alpha_\phi \neq 0$ iff ϕ is satisfiable.

The circuit uses $N = 4R(\phi)$ T-gates and $O(R + m + n)$ additional Clifford gates.

Proof. Starting from $\left|0^n, 0^a, 0^m\right\rangle$, the first Hadamards give

$$2^{-n/2} \sum_x \left|x, 0^a, 0^m\right\rangle. \tag{18}$$

Applying U_ϕ:

$$2^{-n/2} \sum_x \left|x, g(x), C_1(x), \ldots, C_m(x)\right\rangle. \tag{19}$$

Projecting after the final $H^{\otimes n} \otimes H^{\otimes a}$ onto $\left|0^n, 0^a, 1^m\right\rangle$ gives

$$\alpha_\phi = 2^{-n/2} \sum_x \left\langle 0^n\right| H^{\otimes n} \left|x\right\rangle \left\langle 0^a\right| H^{\otimes a} \left|g(x)\right\rangle \mathbf{1}[\phi(x) = 1]. \tag{20}$$

Now $\left\langle 0^n\right| H^{\otimes n} \left|x\right\rangle = 2^{-n/2}$ and by Lemma 5, $\left\langle 0^a\right| H^{\otimes a} \left|g(x)\right\rangle = 2^{-a/2}$, so

$$\alpha_\phi = \frac{1}{2^n \sqrt{2^a}} \sum_{x : \phi(x) = 1} 1 = \frac{\#\mathrm{SAT}(\phi)}{2^{n+a/2}}. \tag{21}$$

The T-count is $4R(\phi)$ by Lemma 4. □

Corollary 7 (k-SAT specialization)**.** *Let ϕ be a k-SAT instance with $m = \Delta n$ clauses. Then*

$$R(\phi) \le (k-1)m, \qquad N = 4R(\phi) \le 4(k-1)m \le 4(k-1)\Delta n. \tag{22}$$

In particular, for a mixed 3-CNF with m_j clauses of size j, one has $N = 4(m_2 + 2m_3)$.

III. REDUCTION TO k-SAT

We showed that any k-CNF-SAT formula can be reduced to determining a computational-basis amplitude of a quantum circuit with $N \leq 4(k-1)m$ T-gates and $O(km) = \mathrm{poly}(n)$ Clifford gates. As a consequence, worst-case runtimes for strong Clifford+T simulation have implications for deciding k-CNF-SAT formulas. In the following, $O^*(\cdot)$ suppresses factors polynomial in the number of variables, circuit size, and requested precision.

Theorem 8 (k-SAT to Clifford+T reduction)**.** *Assume that strongly simulating a Clifford+T circuit, that is determining the output amplitudes to inverse exponential precision, can be done in time $O^*(2^{\eta N})$, where $\eta > 0$ and N denotes the number of T-gates. Then, k-SAT with m clauses can be decided in time $O^*(2^{4\eta(k-1)m})$.*

Proof. By Theorem 6, there is a Clifford+T circuit consisting of $N \leq 4(k-1)m$ T-gates and $O(km)$ Clifford gates that encodes the number of satisfying assignments of a given k-CNF-SAT formula in the specified computational-basis amplitude α_ϕ. Simulating this circuit to sufficient precision such that $\alpha_\phi = 0$ and $\alpha_\phi = 1/2^{n+a/2}$, that is to inverse exponential precision, is sufficient to decide that formula. □

Unfortunately, we cannot directly relate Theorem 8 to many results in the SAT literature, because the number of clauses need not scale linearly in n; in fact $m = O(n^k)$ in the worst case. In particular, we would like to relate the simulation time of Clifford+T circuits to the benchmark of deciding 3-SAT in time better than $O^*(1.3^n)$.

A. Sparsification Lemma for k-CNFs

Lemma 9 (Sparsification Lemma for k-CNFs, Corollary 1 in [7])**.** *For every integer k and $\epsilon > 0$, there is a constant $c(k, \epsilon)$ such that any k-CNF φ can be written as a disjunction*

$$\varphi = \bigvee_{i=1}^{t} \phi_i \tag{23}$$

with $t \leq 2^{\epsilon n}$, where each ϕ_i is a k-CNF with $m = c(k, \epsilon)\, n$ clauses. The algorithm computing the disjunction runs in time $O^*(2^{\epsilon n})$.

Looking at Theorem 8, the *sparsification constant* $c(k, \epsilon)$ directly enters the exponential factor of the runtime. Therefore, we require explicit upper bounds on $c(k, \epsilon)$.

B. An optimized $k = 3$ sparsification constant

For 3-CNF formulas one can sharpen the generic sparsification analysis by specializing to the three possible sunflower types, namely $(2,1)$-, $(3,2)$-, and $(3,1)$-sunflowers, following [1]. We slightly tighten their bookkeeping by pruning contradictory unit branches and by tracking immigrant 2-clauses exactly.

Proposition 10 (Refined 3-SAT sparsification)**.** *Fix integers $\theta_2 \geq \theta_1 \geq 2$, and let*

$$H(p) := -p \log_2 p - (1-p) \log_2 (1-p) \tag{24}$$

denote the binary entropy function. There is an algorithm that, given a 3-CNF formula ϕ on n variables, outputs 3-CNF formulas $\phi_1, \ldots, \phi_t$ such that

$$\phi \equiv \bigvee_{i=1}^{t} \phi_i, \tag{25}$$

each ϕ_i is produced by the branching and subsumption reductions described below, and

$$t \leq 2^{\Gamma(\theta_1, \theta_2) n}, \qquad R(\phi_i) \leq B(\theta_1, \theta_2)\, n, \tag{26}$$

where

$$\Gamma(\theta_1, \theta_2) := (3\theta_1 - 2)\, H\left(\frac{1/\theta_1 + (3\theta_1 - 3)/\theta_2}{3\theta_1 - 2} \right) \tag{27}$$

and

$$B(\theta_1, \theta_2) := \theta_1 - 1 + \frac{4}{3}(\theta_2 - 1). \tag{28}$$

The algorithm runs in time $O^*(2^{\Gamma(\theta_1,\theta_2)n})$.

Proof. We use the same recursive branching scheme as in the 3-SAT-specific sparsification of [1], with thresholds θ_1 for $(2,1)$- and $(3,2)$-sunflowers and threshold θ_2 for $(3,1)$-sunflowers, and priority order

$$(2,1) \succ (3,2) \succ (3,1). \tag{29}$$

Here a (s,h)-sunflower means a collection of s-clauses with a common h-literal heart H, written

$$C_j = H \vee P_j, \tag{30}$$

where the P_j are the petal clauses of size $s-h$. No pairwise-disjointness of the petals is assumed or needed. The branching rule uses

$$\bigwedge_j (H \vee P_j) \equiv H \vee \bigwedge_j P_j, \tag{31}$$

so each branch replaces a good sunflower either by its heart or by its petals, followed by subsumption reduction. Additionally, whenever a branch contains both $\{x_j\}$ and $\{\neg x_j\}$ for some variable x_j, or contains the empty clause, we discard that branch immediately, since it is unsatisfiable. All counting below is done before optional deterministic unit propagation at a leaf; performing such propagation can only decrease the clause-evaluation length R.

We first bound the output clause-evaluation length. At a leaf there is no good sunflower. Hence each literal occurs in at most $\theta_1 - 1$ clauses of size 2, so

$$2m_2 \le 2n(\theta_1 - 1), \tag{32}$$

and each literal occurs in at most $\theta_2 - 1$ clauses of size 3, so

$$3m_3 \le 2n(\theta_2 - 1). \tag{33}$$

Therefore, for every leaf formula ϕ_i,

$$R(\phi_i) = m_2 + 2m_3 \le n(\theta_1 - 1) + \frac{4}{3}n(\theta_2 - 1) = B(\theta_1, \theta_2)n. \tag{34}$$

We next bound the number of leaves. Fix a root-to-leaf path and call a clause *immigrant* if it is not present at the root. Let r_2 denote the maximum number of immigrant 2-clauses sharing a literal. We claim that throughout the recursion,

$$r_2 \le 2\theta_1 - 2. \tag{35}$$

If a branch takes the heart of a $(3,2)$-sunflower, then it adds one new 2-clause. Since the parent node contains no good $(2,1)$-sunflower, each literal lies in at most $\theta_1 - 1$ pre-existing 2-clauses, so afterwards $r_2 \le \theta_1$. If a branch takes the petals of a $(3,1)$-sunflower, then it adds new immigrant 2-clauses. Because $(3,2)$-sunflowers have higher priority, no literal can occur in θ_1 or more of these new petals; otherwise the parent would already contain a good $(3,2)$-sunflower of size at least θ_1. Hence each literal occurs in at most $\theta_1 - 1$ newly added immigrant 2-clauses, while it occurs in at most $\theta_1 - 1$ old 2-clauses, so $r_2 \le 2\theta_1 - 2$.

Consequently, when a new immigrant unit clause is created, it can eliminate at most $2\theta_1 - 2$ immigrant 2-clauses. Since there are at most n immigrant unit clauses on the path, the number of immigrant 2-clauses that ever get eliminated is at most $(2\theta_1 - 2)n$. At the leaf, at most $n(\theta_1 - 1)$ immigrant 2-clauses remain. Therefore the total number of immigrant 2-clauses ever introduced on the path is at most

$$(3\theta_1 - 3)n. \tag{36}$$

Every branching step introduces at least one immigrant clause. Hence the depth of the recursion tree is at most

$$n + (3\theta_1 - 3)n = (3\theta_1 - 2)n. \tag{37}$$

Among these branching steps, taking petals of a $(2,1)$- or $(3,2)$-sunflower introduces at least θ_1 immigrant unit clauses, so there are at most n/θ_1 such petal branches on any path. Taking petals of a $(3,1)$-sunflower introduces at least θ_2 immigrant 2-clauses, so there are at most $(3\theta_1 - 3)n/\theta_2$ such petal branches on any path. Thus the total number of petal branches on any path is at most

$$\left(\frac{1}{\theta_1} + \frac{3\theta_1 - 3}{\theta_2}\right) n. \tag{38}$$

Since the algorithm is deterministic once the sequence of heart/petal choices is fixed, the number of leaves is at most

$$\sum_{i=0}^{q} \binom{N}{i}, \tag{39}$$

where

$$N := (3\theta_1 - 2)n, \qquad q := \left(\frac{1}{\theta_1} + \frac{3\theta_1 - 3}{\theta_2}\right) n. \tag{40}$$

By the standard entropy bound on binomial sums, this is at most

$$2^{NH(q/N)} = 2^{\Gamma(\theta_1,\theta_2)n}. \tag{41}$$

The running time is $O^*(2^{\Gamma(\theta_1,\theta_2)n})$ since the work per node is polynomial. □

Corollary 11 (Optimized asymptotic constant for 3-CNF clause-evaluation length)**.** *Let $\epsilon > 0$ and choose θ_1 minimally subject to*

$$\Gamma(\theta_1, 3\theta_1^2) \le \epsilon. \tag{42}$$

Then every 3-CNF formula on n variables can be written as a disjunction of at most $2^{\epsilon n}$ many 3-CNF formulas, each of clause-evaluation length at most

$$B(\theta_1, 3\theta_1^2)\, n = (64 + o(1)) \left(\frac{\log_2(1/\epsilon)}{\epsilon}\right)^2 n. \tag{43}$$

Proof. Set $\theta_2 = c\theta_1^2$ with $c > 0$ fixed. Then

$$\Gamma(\theta_1, c\theta_1^2) = \left(2 + \frac{6}{c} + o(1)\right) \frac{\log_2 \theta_1}{\theta_1}, \tag{44}$$

so the condition $\Gamma(\theta_1, c\theta_1^2) \le \epsilon$ is achieved with

$$\theta_1 = \left(2 + \frac{6}{c} + o(1)\right) \frac{\log_2(1/\epsilon)}{\epsilon}. \tag{45}$$

Since

$$B(\theta_1, c\theta_1^2) = \frac{4}{3} c\, \theta_1^2 + o(\theta_1^2), \tag{46}$$

the leading coefficient is

$$\frac{4}{3} c \left(2 + \frac{6}{c}\right)^2. \tag{47}$$

This is minimized at $c = 3$, where it equals 64. □

Remark (No better asymptotic coefficient inside the HNS threshold family)**.** *More generally, suppose one assigns thresholds $\alpha\lambda$ to $(2,1)$-sunflowers, $\beta\lambda$ to $(3,2)$-sunflowers, and $\rho\lambda^2$ to $(3,1)$-sunflowers, with $\alpha, \beta, \rho > 0$ fixed and $\lambda \to \infty$. The same counting argument gives an asymptotic clause-evaluation coefficient*

$$\frac{16}{3} \rho \left(\frac{1}{\min\{\alpha, \beta\}} + \frac{2\alpha + \beta}{\rho}\right)^2. \tag{48}$$

Optimizing over ρ yields

$$\rho = (2\alpha + \beta)\min\{\alpha, \beta\}, \tag{49}$$

and hence the coefficient becomes

$$\frac{64}{3}\frac{2\alpha + \beta}{\min\{\alpha, \beta\}} \geq 64, \tag{50}$$

with equality if and only if $\alpha = \beta$. Thus the coefficient 64 from Corollary 11 is already asymptotically optimal within this entire HNS-style family of threshold choices.

Corollary 12 (Implication for strong Clifford+T simulation)**.** *Let*

$$\delta := \log_2(1.3). \tag{51}$$

Assume that a polynomial-size Clifford+T circuit with N T-gates can be strongly simulated in time $O^*(2^{\eta N})$. Using the reduction $N = 4R(\phi)$ from a 3-CNF formula ϕ to a Clifford+T circuit, together with Proposition 10, one obtains a 3-SAT solver running in time

$$O^*\left(2^{(\Gamma(\theta_1,\theta_2)+4\eta\, B(\theta_1,\theta_2))n}\right) \tag{52}$$

for every choice of integers $\theta_2 \geq \theta_1 \geq 2$.

Therefore any

$$\eta < \sup_{\theta_2 \geq \theta_1 \geq 2} \frac{\delta - \Gamma(\theta_1, \theta_2)}{4B(\theta_1, \theta_2)} \tag{53}$$

would imply an $O(1.3^n)$-time algorithm for 3-SAT on instances with $m = \mathrm{poly}(n)$.

A direct numerical optimization of the right-hand side gives

$$\sup_{\theta_2 \geq \theta_1 \geq 2} \frac{\delta - \Gamma(\theta_1, \theta_2)}{4B(\theta_1, \theta_2)} \approx 5.07 \times 10^{-7}. \tag{54}$$

For example, the explicit choice

$$(\theta_1, \theta_2) = (104, 40147) \tag{55}$$

gives

$$\Gamma(104, 40147) = 0.2695641807\ldots, \qquad B(104, 40147) = 53631, \tag{56}$$

and hence

$$\eta < \frac{\log_2(1.3) - 0.2695641807\ldots}{4 \cdot 53631} = 5.0785666\ldots \times 10^{-7}. \tag{57}$$

IV. ON LOWER BOUNDS FOR WEAK SIMULATION

For weak simulation, the question of obtaining exponential lower bounds is essentially still open. In [8], such a lower bound is derived, but the result is conditioned on a custom complexity conjecture. The simple repeated-sampling argument below applies to exact weak simulation, or to sufficiently accurate multiplicative-error weak simulation. It does not directly apply to constant-additive-error, or constant-total-variation-distance, weak simulation when the relevant event has probability as small as $2^{-\gamma n}$, since such additive error can swamp the event. For basing such a lower bound on SETH, the following problem seems central:

Open problem. *Is there a quantum algorithm using $O(N)$ T-gates that, for any CNF-SAT formula φ, outputs 0 if φ is unsat and, if φ is sat, outputs 1 with probability at least $2^{-\gamma n}$ for some positive $\gamma < 1$?*

If one could devise such an algorithm, then a sufficiently fast exact or multiplicative-error weak simulator for Clifford+T circuits would give a SAT decision procedure whose runtime is dominated by repeated sampling. If the probability to sample a 1 is at least $2^{-\gamma n}$, then the failure probability after K trials is

$$(1 - 2^{-\gamma n})^K \leq \exp\left(-2^{-\gamma n} K\right). \tag{58}$$

To have this at most δ, it suffices to set

$$K \geq 2^{\gamma n} \ln\left(\frac{1}{\delta}\right). \tag{59}$$

Thus any positive $\gamma < 1$ would be incompatible with SETH when combined with a subexponential-time exact or multiplicative-error weak simulator. For additive-error weak simulation, one would instead need the accepting event to have at least inverse-polynomial probability, or the simulator's additive error on that event to be $o(2^{-\gamma n})$.

---

[1] C. Huang, M. Newman, and M. Szegedy, Explicit lower bounds on strong simulation of quantum circuits in terms of t-gate count, arXiv preprint arXiv:1902.04764 (2019).

[2] T. Hertli, 3-sat faster and simpler—unique-sat bounds for ppsz hold in general, SIAM Journal on Computing **43**, 718 (2014).

[3] D. Scheder and J. Steinberger, Ppsz for general k-sat and csp—making hertli's analysis simpler and 3-sat faster, computational complexity **33**, 13 (2024).

[4] D. Scheder, Ppsz is better than you think, in *62nd IEEE Annual Symposium on Foundations of Computer Science (FOCS 2021)* (IEEE, 2021) pp. 205–216.

[5] G. Song and A. Klappenecker, The simplified toffoli gate implementation by margolus is optimal (2003), arXiv preprint quant-ph/0312225.

[6] D. Maslov, Advantages of using relative phase toffoli gates with an application to multiple control toffoli optimization, Physical Review A **93**, 022311 (2016).

[7] R. Impagliazzo, R. Paturi, and F. Zane, Which problems have strongly exponential complexity?, Journal of computer and system sciences **63**, 512 (2001).

[8] T. Morimae and S. Tamaki, Additive-error fine-grained quantum supremacy, Quantum **4**, 329 (2020).