Optimal Ancilla Complexity for Approximate Toffoli Gates in the Mixed Clifford+T Model

Arul Rhik Mazumder
Carnegie Mellon University
arulm@andrew.cmu.edu

1 Previous Results

Theorem 1 (Upper bound: mixed T-count)**.** *For any $n \geq 1$ and $\epsilon > 0$,*

$$T_\epsilon^{\text{mixed}}(\text{Toff}_n) \;\leq\; T_0^{\text{unitary}}\big(\text{Toff}_{\lceil \log(1/\epsilon)\rceil+3}\big) \;=\; O(\log(1/\epsilon))\,. \tag{1}$$

Theorem 2 (Lower bound: unitary T-count)**.** *For any $\epsilon \in [0, 1/2)$ and sufficiently large n, $T_\epsilon^{\text{unitary}}(\text{Toff}_n) \geq n - 2$.*

Theorem 3 (Lower bound: mixed/adaptive T-count)**.** *For sufficiently large n and $1/\epsilon$,*

$$T_\epsilon^{\text{mixed}}(\text{Toff}_n) \;\geq\; T_\epsilon^{\text{adaptive}}(\text{Toff}_n) \;=\; \Omega(\min\{n, \log(1/\epsilon)\})\,. \tag{2}$$

Theorem 4 (General Boolean functions)**.** *For $f : \{0,1\}^n \to \{0,1\}$ and $\epsilon > 0$, $T_\epsilon^{\text{mixed}}(U_f) = O\Big(\|\widehat{f}\|_1^2 \,\log(1/\epsilon)\Big)$.*

Theorem 5 (Parity decision tree connections)**.** *For any $f : \{0,1\}^n \to \{0,1\}$ and $\epsilon \geq 0$,*

$$\text{PDT}^{\text{na}}(f) - 1 \;\leq\; T_{1/3}^{\text{unitary}}(U_f) \;=\; O(\text{gatePDT}^{\text{na}}(f))\,, \tag{3}$$

$$\text{RPDT}_\epsilon^{\text{na}}(f) - 1 \;\leq\; T_\epsilon^{\text{mixed}}(U_f) \;=\; O(\text{gateRPDT}_\epsilon^{\text{na}}(f))\,. \tag{4}$$

Theorem 6 (Exact adaptive lower bound)**.** *For any Boolean function f, $T_0^{\text{adaptive}}(U_f) \geq \text{PDT}^{\text{na}}(f) - 1$.*

Extending this to $\varepsilon > 0$ remains open.

2 Definitions

2.1 Distance Measures

Definition 1 (Trace distance)**.** $D(\rho, \sigma) = \frac{1}{2}\|\rho - \sigma\|_1$*, where* $\|A\|_1 = \text{Tr}\Big(\sqrt{A^\dagger A}\Big)$.

Definition 2 (Fidelity). $F(\rho,\sigma) = \big\|\sqrt{\rho}\sqrt{\sigma}\big\|_1^2$. *For pure* $\sigma = |\psi\rangle\langle\psi|$*:* $F(\rho,|\psi\rangle\langle\psi|) = \langle\psi|\rho|\psi\rangle$.

Lemma 1 (Fuchs–van de Graaf). $1 - \sqrt{F(\rho,\sigma)} \le D(\rho,\sigma) \le \sqrt{1-F(\rho,\sigma)}$.

Theorem 7 (Holevo–Helstrom). *The optimal single-copy distinguishing probability for* ρ *vs.* σ *is* $\frac{1}{2} + \frac{1}{2}D(\rho,\sigma)$.

Definition 3 (Diamond distance). *For quantum channels* E_1, E_2 *on* n *qubits,* $D_\diamond(E_1,E_2) = \sup_{\ell,\,\rho}\ D\big((E_1\otimes I_\ell)(\rho),\ (E_2\otimes I_\ell)(\rho)\big)$.

2.2 T-count Models

Definition 4 (Unitary T-count). $T_\epsilon^{unitary}(U)$ *is the minimum* T*-gate count in any Clifford+*T *circuit* V *(with ancillae in* $|0\rangle$*) satisfying* $D_\diamond\big(\mathrm{Tr}_{anc}[\Phi_V],\,U\big) \le \epsilon$.

Definition 5 (Mixed T-count). *Given a distribution* $\{p_i\}$ *over Clifford+*T *circuits* $\{V_i\}$*, define* $\mathcal{E}(\rho) = \mathrm{Tr}_{anc}\big[\sum_i p_i V_i(\rho\otimes|0_a\rangle\langle 0_a|)V_i^\dagger\big]$*. Then* $T_\epsilon^{mixed}(U)$ *is the minimum (over channels with* $D_\diamond(\mathcal{E},U)\le\epsilon$*) of* $\max_i T\text{-}count(V_i)$.

Definition 6 (Adaptive T-count). *An adaptive Clifford+*T *circuit interleaves Clifford gates,* T *gates, and computational-basis measurements, with each gate depending on prior outcomes.* $T_\epsilon^{adaptive}(U)$ *is the minimum worst-case expected* T*-count over circuits with* $D_\diamond(\mathcal{E},U)\le\epsilon$.

3 Algorithm and Upper Bound

Algorithm 1 Approximate Toff_n via random XOR sampling

Require: Controls $|x_1,\dots,x_{n-1}\rangle$, target $|b\rangle$, error $\varepsilon > 0$
Ensure: Mixed channel $\mathcal{E}$ with $D_\diamond(\mathcal{E},\mathrm{Toff}_n)\le\varepsilon$

1: $k \leftarrow \lceil \log_2(1/\varepsilon)\rceil + 2$
2: **for** $j = 1$ **to** k **do**
3: Sample $S_j \subseteq [n-1]$ uniformly at random
4: **end for**
5: Define $g(x) = \mathrm{OR}_k\big(\bigoplus_{i\in S_1} x_i,\ \dots,\ \bigoplus_{i\in S_k} x_i\big)$
6: Implement W_g: (a) $X^{\otimes n}$ conjugation, (b) compute k parities via CNOTs, (c) compute OR_k via Toff_{k+1}, (d) CNOT to target and uncompute, (e) undo conjugation.

Theorem 8. $\mathcal{E}(\rho) = \mathbb{E}_{S_1,\dots,S_k}[W_g\,\rho\,W_g^\dagger]$ *satisfies* $D_\diamond(\mathcal{E},\mathrm{Toff}_n)\le 4\cdot 2^{-k}$.

4 Lower Bound via Stabilizer Nullity

We prove the following matching lower bound.

Theorem 9 (Lower bound: mixed/adaptive T-count)**.** *For sufficiently large n and $1/\epsilon$,*

$$T_\epsilon^{\text{mixed}}(\text{Toff}_n) \;\geq\; T_\epsilon^{\text{adaptive}}(\text{Toff}_n) \;=\; \Omega(\min\{n, \log(1/\epsilon)\})\,. \tag{5}$$

Combined with Theorem 10, this gives $T_\epsilon^{\text{mixed}}(\text{Toff}_n) = \Theta(\log(1/\epsilon))$ for any fixed $\epsilon > 0$. The proof uses stabilizer nullity as a "magic counter."

Definition 7 (Stabilizer nullity)**.** $\nu(\rho) = n - \log|\{P \in \{\pm 1\} \cdot \{I, X, Y, Z\}^{\otimes n} : P\rho = \rho\}|$.

Lemma 2 (Properties)**.** *Properties of Stabilizer nullity*

(i) $\nu(\rho) = 0$ *iff* ρ *is stabilizer.*

(ii) Cliffords preserve ν*.*

(iii) Each T *gate increases* ν *by at most* 1*.*

(iv) Pauli postselection does not increase ν*.*

(v) $\nu(\rho \otimes \sigma) = \nu(\rho) + \nu(\sigma)$*.*

Lemma 3 (Maximal nullity of target)**.** *Let* $|\Phi\rangle = C^{n-1}Z|+\rangle^{\otimes n}$ *with* $n \geq 3$*. If* $D(\omega, |\Phi\rangle\langle\Phi|) < 2/2^n$*, then* $\nu(\omega) = n$*.*

Proposition 1 (Adaptive $\to$ non-adaptive reduction)**.** *An adaptive circuit using t expected T gates can be converted to a Clifford-plus-postselection circuit consuming $2t$ copies of $|T\rangle$, with output trace distance $\leq \sqrt{6\epsilon}$ from the target.*

5 Improved Ancilla Counts (Novel Stuff)

6 Preliminaries

We work in the Clifford+T gate set. The n-qubit Toffoli gate is defined as

$$\text{Toff}_n|x_1, \ldots, x_{n-1}\rangle|b\rangle = |x_1, \ldots, x_{n-1}\rangle|b \oplus (x_1 \wedge \cdots \wedge x_{n-1})\rangle.$$

We use the mixed T-count model of [2], where the implementation is a channel $\mathcal{E}(\rho) = \sum_g p_g\, U_g \rho\, U_g^\dagger$ and each U_g is a Clifford+T circuit. The approximation quality is measured in diamond distance: $D_\diamond(\mathcal{E}, \text{Toff}_n) \leq \varepsilon$.

Theorem 10 (Gosset–Kothari–Zhang [2])**.** *For any positive integer n and $\varepsilon > 0$, $\mathcal{T}_\varepsilon^{\text{mixed}}(\text{Toff}_n) = O(\log(1/\varepsilon))$, achieved by sampling $k = \lceil \log_2(1/\varepsilon) \rceil + 2$ random subsets $S_1, \ldots, S_k \subseteq [n-1]$, computing random parities $p_j = \bigoplus_{\ell \in S_j} x_\ell$, and flipping the target if and only if $\text{OR}_k(p_1, \ldots, p_k) = 0$ (i.e., all parities vanish).*

The OR subroutine, used throughout, is:

Algorithm 2 $\textsc{Or}(q_1, \ldots, q_r,\ t)$: Compute OR_r into target via De Morgan

Require: Input qubits $|q_1, \ldots, q_r\rangle$, target qubit $|t\rangle$
Ensure: $|t\rangle \leftarrow |t \oplus \mathrm{OR}(q_1, \ldots, q_r)\rangle$
1: Apply $X^{\otimes r}$ on $q_1, \ldots, q_r$
2: Apply Toff_{r+1}: $|t\rangle \leftarrow |t \oplus (q_1 \wedge \cdots \wedge q_r)\rangle$
3: Apply $X^{\otimes r}$ on $q_1, \ldots, q_r$
4: Apply X on t

7 The 2-Level Construction ($\sqrt{\log(1/\varepsilon)}$ Ancillae)

We first present the 2-level construction, which serves as the base case for the general d-level strategy.

Algorithm 3 Approximate Toff_n via 2-Level OR Tree (Workspace + Storage)

Require: Controls $|x_1, \ldots, x_{n-1}\rangle$, target $|b\rangle$, $\varepsilon > 0$, m workspace qubits $|w_i\rangle$ $(1 \le i \le m)$, $r = \lceil k/m \rceil$ storage qubits $|s_j\rangle$ $(1 \le j \le r)$
Ensure: $D_\diamond(\mathcal{E}, \mathrm{Toff}_n) \le \varepsilon$
1: $k \leftarrow \lceil \log_2(1/\varepsilon) \rceil + 2$; sample $S_1, \ldots, S_k \subseteq [n-1]$ uniformly at random
2: Apply $X^{\otimes(n-1)}$ to controls
3: **for** $i = 1$ **to** r **do**
4: $\quad m_i \leftarrow \min(m,\ k - (i-1)m)$
5: $\quad$ Compute parities: $|w_j\rangle \oplus= \bigoplus_{\ell \in S_{(i-1)m+j}} x_\ell$ for $j = 1, \ldots, m_i$
6: $\quad \textsc{Or}(w_1, \ldots, w_{m_i},\ s_i)$
7: $\quad$ Uncompute parities (reverse of line 5)
8: **end for**
9: $\textsc{Or}(s_1, \ldots, s_r,\ b)$; apply X to b
10: **for** $i = r$ **downto** 1 **do**
11: $\quad$ Recompute parities; $\textsc{Or}^\dagger(w_1, \ldots, w_{m_i},\ s_i)$;
12: **end for**
13: Apply $X^{\otimes(n-1)}$ to controls

Lemma 4 (Functional equivalence)**.** *The Boolean function computed by Algorithm 3 is identical to that of [2, Algorithm 1]:*

$$\mathrm{OR}_r\big(\mathrm{OR}_{m_1}(p_1, \ldots, p_{m_1}),\ \ldots,\ \mathrm{OR}_{m_r}(p_{(r-1)m+1}, \ldots, p_k)\big) = \mathrm{OR}_k(p_1, \ldots, p_k),$$

where $r = \lceil k/m \rceil$ *and* $m_i = \min(m,\ k - (i-1)m)$.

Proof. Immediate from the associativity of OR. □

Theorem 11 (2-level resource counts)**.** *Algorithm 3 with* $m = \lceil \sqrt{k} \rceil$ *implements a mixed channel* $\mathcal{E}$ *with* $D_\diamond(\mathcal{E}, \mathrm{Toff}_n) \le \varepsilon$ *using* $O(\sqrt{\log(1/\varepsilon)})$ *ancilla qubits and* $O(\log(1/\varepsilon))$ T *gates.*

Proof. **Ancillae.** Total is $f(m) = m + \lceil k/m \rceil$, minimized at $m = \lceil \sqrt{k} \rceil$, giving $O(\sqrt{k}) = O(\sqrt{\log(1/\varepsilon)})$.
T-count. Each Toff_{m_i+1} decomposes into $2(m_i - 2)$ standard Toffoli-3 gates (compute + uncompute in the linear chain), each costing 4 T gates. Summing:

$$T\text{-count} = 2\sum_{i=1}^{r} 8(m_i - 2) + 8(r - 2) = 16k - 24r - 16 = O(k) = O(\log(1/\varepsilon)).$$

□

8 The d-Level Tree Generalization

8.1 Construction

The 2-level tree batches k parities into groups of m, computes OR of each batch into a storage qubit, then combines all storage qubits with a final OR. The d-level tree applies this idea recursively: each level groups the outputs of the previous level into batches and computes their OR into the next level's storage qubits, with all intermediate results uncomputed after they are consumed.

Definition 8 (d-level OR tree)**.** *Fix $k \geq 1$ and fan-ins $m_1, \ldots, m_{d-1} \geq 2$. A* d-level OR tree *of k inputs is defined recursively:*

- ***Level 0** (leaves): the k parity bits $p_1, \ldots, p_k$.*
- ***Level** ℓ ($1 \leq \ell \leq d-1$): partition the outputs of level $\ell-1$ into consecutive groups of size m_ℓ, and compute* OR *of each group into a* level-ℓ storage qubit*. This produces $n_\ell = \lceil n_{\ell-1}/m_\ell \rceil$ outputs, where $n_0 = k$.*
- ***Root** (level d): compute* OR *of the n_{d-1} top-level storage qubits into the target b, then apply X to b.*

The key to reducing ancillae is the depth-first execution order: we traverse the tree depth-first, computing each subtree's OR, passing the result up, and then uncomputing the subtree's intermediate qubits so they can be reused by the next subtree.

Algorithm 4 d-Level Approximate Toff_n

Require: Controls $|x_1, \ldots, x_{n-1}\rangle$, target $|b\rangle$, $\varepsilon > 0$, tree depth $d \geq 2$, fan-ins $m_1, \ldots, m_{d-1}$
Ensure: $D_\diamond(\mathcal{E}, \text{Toff}_n) \leq \varepsilon$
1: $k \leftarrow \lceil \log_2(1/\varepsilon) \rceil + 2$; sample $S_1, \ldots, S_k \subseteq [n-1]$ uniformly at random
2: Apply $X^{\otimes(n-1)}$ to controls
3: FillStorage$(d-1,\ 1,\ k)$
4: Or$\big(s_1^{(d-1)}, \ldots, s_r^{(d-1)},\ b\big)$;
5: apply X to b
6: FillStorage$(d-1,\ 1,\ k)^\dagger$
7: Apply $X^{\otimes(n-1)}$ to controls

Algorithm 5 FillStorage(ℓ, start, count): Depth-first fill of level-ℓ storage qubits

Require: Level index ℓ, starting parity index *start*, number of parities *count*

Ensure: Level-ℓ qubits $s_1^{(\ell)}, \ldots, s_r^{(\ell)}$ satisfy $s_i^{(\ell)} = \mathrm{OR}(\text{parities in group } i)$; all workspace and intermediate storage qubits are returned to $|0\rangle$

1: **if** $\ell = 1$ **then**
2: $r \leftarrow \lceil \text{count}/m_1 \rceil$
3: **for** $i = 1$ **to** r **do**
4: $m_i' \leftarrow \min(m_1,\ \text{count} - (i-1)m_1)$
5: $|w_j\rangle \oplus= \bigoplus_{k \in S_{\text{start}+(i-1)m_1+j-1}} x_k$ for $j = 1, \ldots, m_i'$
6: Or$(w_1, \ldots, w_{m_i'},\ s_i^{(1)})$
7: Uncompute parities (reverse line 5)
8: **end for**
9: Return
10: **end if**
11: $r \leftarrow \lceil \text{count}/\prod_{j=1}^{\ell} m_j \rceil$
12: **for** $i = 1$ **to** r **do**
13: $\text{start}_i \leftarrow \text{start} + (i-1)\prod_{j=1}^{\ell} m_j$
14: $\text{cnt}_i \leftarrow \min\big(\prod_{j=1}^{\ell} m_j,\ \text{count} - (i-1)\prod_{j=1}^{\ell} m_j\big)$
15: FillStorage$(\ell - 1, \text{start}_i, \text{cnt}_i)$
16: $r_i' \leftarrow \lceil \text{cnt}_i/\prod_{j=1}^{\ell-1} m_j \rceil$
17: Or$\big(s_1^{(\ell-1)}, \ldots, s_{r_i'}^{(\ell-1)},\ s_i^{(\ell)}\big)$
18: FillStorage$(\ell - 1, \text{start}_i, \text{cnt}_i)^{\dagger}$
19: **end for**

8.2 Analysis

Lemma 5 (Simultaneous qubit occupancy)**.** *During depth-first execution of Algorithm 4 with fan-ins $m_1, \ldots, m_{d-1}$, the number of ancilla qubits simultaneously occupied at any instant is at most*

$$A(d) \;=\; \sum_{\ell=1}^{d-1} m_\ell \;+\; n_{d-1},$$

where $n_{d-1} = \lceil k/\prod_{\ell=1}^{d-1} m_\ell \rceil$ is the number of top-level storage qubits.

Proof. We identify which qubits are occupied at the moment of deepest nesting. Consider an instant during the innermost recursive call. The depth-first traversal has exactly one "open" batch at each level:

- **Level 0** (workspace): m_1 qubits hold the current batch of parities.
- **Level** ℓ $(1 \le \ell \le d-2)$: Within the current level-$(\ell+1)$ batch, at most $m_{\ell+1}$ level-ℓ storage qubits are occupied. (Previous level-$(\ell+1)$ batches have already uncomputed their level-ℓ storage.)

- **Level** $d-1$ (top storage): n_{d-1} qubits accumulate until the final OR. These are not freed until the global uncompute phase.

Workspace contributes m_1; each intermediate level ℓ ($1 \le \ell \le d-2$) contributes at most $m_{\ell+1}$ (the fan-in of the level above it, which is the number of level-ℓ results needed before the level-$(\ell+1)$ OR fires); the top level contributes n_{d-1}. Re-indexing, the intermediate levels contribute $m_2 + m_3 + \cdots + m_{d-1}$. Total:

$$A(d) = m_1 + (m_2 + \cdots + m_{d-1}) + n_{d-1} = \sum_{\ell=1}^{d-1} m_\ell + n_{d-1}. \qquad \square$$

Theorem 12 (Optimal symmetric fan-in)**.** *Setting all fan-ins equal, $m_1 = \cdots = m_{d-1} = m$, the ancilla count is*

$$A(d) = (d-1)\, m + \lceil k/m^{d-1} \rceil.$$

This is minimized at $m = \lceil k^{1/d} \rceil$, giving

$$A(d) = O\Big(d \cdot k^{1/d}\Big) = O\Big(d \cdot \log(1/\varepsilon)^{1/d}\Big).$$

Proof. With symmetric fan-in, $n_{d-1} = \lceil k/m^{d-1} \rceil$. Treating $A(m) = (d-1)m + k\, m^{-(d-1)}$ as continuous and differentiating:

$$\frac{dA}{dm} = (d-1) - (d-1)\, k\, m^{-d} = 0 \quad \Longrightarrow \quad m^d = k \quad \Longrightarrow \quad m = k^{1/d}.$$

Substituting back:

$$A = (d-1)\, k^{1/d} + k^{1/d} = d\, k^{1/d}. \qquad \square$$

Corollary 1 (Constant fan-in limit)**.** *Setting $m_\ell = c$ for a constant $c \ge 2$ at all levels requires $d = \lceil \log_c k \rceil$ levels, giving*

$$A = O(d) = O(\log k) = O(\log\log(1/\varepsilon))$$

ancillae with $O(\log(1/\varepsilon))$ T gates.

Proof. With $m = c$ and $d = \lceil \log_c k \rceil$, we have $n_{d-1} = \lceil k/c^{d-1} \rceil = O(1)$. The total ancillae are $(d-1)\, c + O(1) = O(d) = O(\log_c k) = O(\log\log(1/\varepsilon))$. $\square$

Theorem 13 (d-level resource counts)**.** *For any positive integer n, $\varepsilon > 0$, and tree depth $d \ge 2$, Algorithm 4 with $k = \lceil \log_2(1/\varepsilon) \rceil + 2$ and symmetric fan-in $m = \lceil k^{1/d} \rceil$ implements a mixed channel $\mathcal{E}$ with $D_\diamond(\mathcal{E}, \mathrm{Toff}_n) \le \varepsilon$ using:*

(i) $O\big(d \cdot \log(1/\varepsilon)^{1/d}\big)$ *ancilla qubits,*

(ii) $O(\log(1/\varepsilon))$ T *gates.*

In particular, with $d = \lceil \log_c k \rceil$ for any constant $c \ge 2$:

(i′) $O(\log\log(1/\varepsilon))$ *ancilla qubits,*

(ii′) $O(\log(1/\varepsilon))$ T *gates.*

Proof. **Correctness.** By induction on d, the d-level OR tree computes the same function $\mathrm{OR}_k(p_1, \ldots, p_k)$ as a flat OR (by associativity of OR, generalizing Lemma 4). Thus each sampled unitary W_g is identical to that of [2], and the diamond distance bound $D_\diamond(\mathcal{E}, \mathrm{Toff}_n) \le \varepsilon$ follows from Theorem 10.
Ancilla count. Follows from Theorem 12 and Corollary 1.
T-count. In the tree, there are $O(k)$ total OR nodes (each leaf produces one parity, and the tree has fewer internal nodes than leaves). Each OR node with fan-in m_ℓ contributes $O(m_\ell)$ Toffoli-3 gates. For the compute phase and uncompute phase combined, the total number of Toffoli-3 invocations is at most

$$2 \sum_{\text{nodes}} O(m_\ell) = O\Big(\sum_{\ell=1}^{d-1} n_{\ell-1}\Big) = O(k),$$

since $\sum_\ell n_{\ell-1} \le k \cdot (1 + 1/m + 1/m^2 + \cdots) = O(k)$ for $m \ge 2$. Each Toffoli-3 costs 4 T gates, so the total is $O(k) = O(\log(1/\varepsilon))$. □

9 Summary of Results

Construction	**Ancillae**	**T-count**	**Model**
GKZ [2] (flat)	$O(\log(1/\varepsilon))$	$O(\log(1/\varepsilon))$	Mixed
2-level tree (Thm. 11)	$O(\sqrt{\log(1/\varepsilon)})$	$O(\log(1/\varepsilon))$	Mixed
d-level tree (Thm. 13)	$O(d \cdot \log(1/\varepsilon)^{1/d})$	$O(\log(1/\varepsilon))$	Mixed
Constant fan-in (Cor. 1)	$O(\log\log(1/\varepsilon))$	$O(\log(1/\varepsilon))$	Mixed
Exact (Beverland et al. [1])	$n-2$	$\Theta(n)$	Unitary
Approx. unitary [2]	any	$\Omega(n)$	Unitary

Table 1: Comparison of resource costs for implementing Toff_n to error ε in diamond distance. All mixed-model constructions have T-count independent of n.

10 Proving a Lower-Bound on Ancillae

Throughout this section, we fix $n \ge 3$ and assume $\varepsilon < 2^{-(n-1)}$, so that $k = \lceil \log_2(1/\varepsilon) \rceil \ge n-1 = m$. This is the regime where the lower bound is nontrivial; when $\varepsilon \ge 2^{-(n-1)}$, a constant number of ancillae suffices.

10.1 Formal redefinition

Definition 9 (Mixed Circuit)**.** *A mixed circuit on n system qubits with a ancillae is a finite probability distribution $\{(p_g, U_g)\}_{g \in \mathcal{G}}$ over classical-reversible*

unitaries on $n + a$ qubits, i.e., unitaries that permute the computational basis. Equivalently, each U_g is a reversible Boolean circuit composed of NOT, CNOT, and Toffoli gates (which may be decomposed into Clifford+T gates for cost accounting). The induced channel is

$$\mathcal{E}(\rho) \;=\; \sum_g p_g \; Tr_{\mathrm{anc}}\big[U_g\big(\rho \otimes |0^a\rangle\langle 0^a|0^a\big)U_g^\dagger\big].$$

The T-count of the circuit is $\max_g T(U_g)$.

Remark 1. *In this model, we require each U_g to be a classical-reversible permutation of the computational basis. This is a genuine restriction, since a general Clifford+T unitary can create superpositions from basis states (e.g., via Hadamards) and would not yield a well-defined Boolean function per circuit branch. This captures the T-count optimal regime discussed in [2].*

Lemma 6. *If $D_\diamond(\mathcal{E}, \mathcal{T}_n) \le \varepsilon$ then:*

$$\textstyle\sum_g p_g \, f_g(1^m) \ge 1 - \varepsilon, \tag{6}$$

$$\forall\, x \neq 1^m : \quad \textstyle\sum_g p_g \, f_g(x) \le \varepsilon. \tag{7}$$

Proof. By definition, the diamond norm $D_\diamond(\mathcal{E}, \mathcal{T}_n) \le \varepsilon$ implies that for every input state ρ,

$$\tfrac{1}{2}\big\|(\mathcal{E} - \mathcal{T}_n)(\rho)\big\|_1 \;\le\; \varepsilon. \tag{8}$$

Fix an arbitrary $x \in \{0,1\}^m$ and take $\rho = |x,0\rangle\langle x,0|$. For the ideal channel:

$$\mathcal{T}_n(|x,0\rangle\langle x,0|) \;=\; |x, \mathrm{AND}_m(x)\rangle\langle x, \mathrm{AND}(x)|,$$

For the mixed channel, we expand using the definition $\mathcal{E}(\rho) = \sum_g p_g \, \mathrm{Tr}_{\mathrm{anc}}[U_g(\rho \otimes |0^a\rangle\langle 0^a|0^a)U_g^\dagger]$. Since each U_g is a classical-reversible unitary (i.e., a permutation of basis states), it maps $|x,0,0^a\rangle$ to a single basis state:

$$U_g|x,0,0^a\rangle \;=\; |x'_g(x,0),\; b'_g(x,0),\; s_g(x,0)\rangle,$$

where $x'_g(x,0) \in \{0,1\}^m$ is the output control register, $b'_g(x,0) \in \{0,1\}$ is the output target bit, and $s_g(x,0) \in \{0,1\}^a$ is the output ancilla state. For brevity, we will denote this as (x'_g, b'_g, s_g). Taking the outer product and tracing over the ancillae, we get:

$$\begin{aligned}
\mathrm{Tr}_{\mathrm{anc}}\big[U_g\big(|x,0\rangle\langle x,0| \otimes |0^a\rangle\langle 0^a|\big)U_g^\dagger\big] &= \mathrm{Tr}_{\mathrm{anc}}\big[|x'_g, b'_g, s_g\rangle\langle x'_g, b'_g, s_g|\big] \\
&= |x'_g, b'_g\rangle\langle x'_g, b'_g| \cdot \mathrm{Tr}[|s_g\rangle\langle s_g|] \qquad (9)\\
&= |x'_g, b'_g\rangle\langle x'_g, b'_g|. \qquad (10)
\end{aligned}$$

Summing over the mixture:

$$\mathcal{E}(|x,0\rangle\langle x,0|) \;=\; \sum_g p_g \, |x'_g, b'_g\rangle\langle x'_g, b'_g|. \tag{11}$$

Substituting (11) into (8):

$$\tfrac{1}{2}\Big\|\sum_g p_g\,|x'_g, b'_g\rangle\langle x'_g, b'_g| - |x, \mathrm{AND}_m(x)\rangle\langle x, \mathrm{AND}_m(x)|\Big\|_1 \;\le\; \varepsilon. \qquad (12)$$

Both terms are diagonal in the computational basis: $\sum_g p_g\,|x'_g, b'_g\rangle\langle x'_g, b'_g|$ assigns probability mass p_g to each basis state $|x'_g, b'_g\rangle$, while $|x, \mathrm{AND}_m(x)\rangle\langle x, \mathrm{AND}_m(x)|$ assigns mass 1 to $|x, \mathrm{AND}_m(x)\rangle$ and 0 to all other states.
For two probability distributions P and Q over a common finite set, $\frac{1}{2}\|P-Q\|_1 = 1 - \sum_j \min(P_j, Q_j)$. Applying this with $P =$ the distribution $\{p_g$ on $(x'_g, b'_g)\}_g$ and $Q =$ the point mass on $(x, \mathrm{AND}_m(x))$:

$$\sum_j \min(P_j, Q_j) \;=\; \min\Big(\sum_{g:\,(x'_g,b'_g)=(x,\mathrm{AND}_m(x))} p_g,\; 1\Big) \;=\; \sum_{g:\,(x'_g,b'_g)=(x,\mathrm{AND}_m(x))} p_g,$$

since $\sum_g p_g = 1$ and the inner sum is ≤ 1. Therefore:

$$\tfrac{1}{2}\|\cdots\|_1 \;=\; 1 - \sum_{g:\,(x'_g,b'_g)=(x,\mathrm{AND}_m(x))} p_g.$$

Combining with (12):

$$\sum_{g:\,(x'_g,b'_g)\,=\,(x,\mathrm{AND}_m(x))} p_g \;\ge\; 1-\varepsilon. \qquad (13)$$

Completeness. Set $x = 1^m$, so $\mathrm{AND}_m(1^m) = \prod_{i=1}^m 1 = 1$. Then (13) becomes:

$$\sum_{g:\,(x'_g,b'_g)\,=\,(1^m,1)} p_g \;\ge\; 1-\varepsilon. \qquad (14)$$

Recall $f_g(1^m) = b'_g(1^m, 0) \in \{0, 1\}$. We compare the sum $\sum_g p_g f_g(1^m)$ with (14). For each g:

- If $(x'_g, b'_g) = (1^m, 1)$, then $b'_g = 1$, so $f_g(1^m) = 1 \ge \mathbf{1}[(x'_g, b'_g) = (1^m, 1)]$.
- If $(x'_g, b'_g) \ne (1^m, 1)$ but $b'_g = 1$ (i.e., $x'_g \ne 1^m$), then $f_g(1^m) = 1$ while $\mathbf{1}[(x'_g, b'_g) = (1^m, 1)] = 0$, so $f_g(1^m) \ge \mathbf{1}[(x'_g, b'_g) = (1^m, 1)]$.
- If $b'_g = 0$, then $f_g(1^m) = 0 = \mathbf{1}[(x'_g, b'_g) = (1^m, 1)]$.

In all cases, $f_g(1^m) \ge \mathbf{1}[(x'_g, b'_g) = (1^m, 1)]$. Therefore:

$$\sum_g p_g\, f_g(1^m) \;\ge\; \sum_g p_g\, \mathbf{1}[(x'_g, b'_g) = (1^m, 1)] \;=\; \sum_{g:\,(x'_g,b'_g)=(1^m,1)} p_g \;\ge\; 1-\varepsilon.$$

This establishes (6).
Soundness. For $x \ne 1^m$ (so $\mathrm{AND}_m(x) = 0$), (13) gives

$$\sum_{g:\,(x'_g,b'_g)=(x,0)} p_g \ge 1-\varepsilon \Rightarrow \sum_{g:\,(x'_g,b'_g)\ne(x,0)} p_g \le \varepsilon$$

If $f_g(x) = 1$ (i.e., $b'_g = 1$), then $(x'_g, b'_g) \neq (x, 0)$, so:

$$\textstyle\sum_g p_g\, f_g(x) \;\le\; \sum_{g:\,(x'_g,b'_g)\neq(x,0)} p_g \;\le\; \varepsilon.$$

This gives (7). □

Define the *accept set* $A_g = f_g^{-1}(1) \subseteq \{0,1\}^m$ and the *false-accept set* $B_g = A_g \setminus \{1^m\}$.

Lemma 7. *There exists a subset $\mathcal{G}' \subseteq \mathcal{G}$ with $\sum_{g\in\mathcal{G}'} p_g \ge \frac{1}{2}$ such that for every $g \in \mathcal{G}'$, $|B_g| \le 2^{m-k+1}$, where $k = \lceil \log_2(1/\varepsilon) \rceil$.*

Proof. By (7), summing over all non-all-ones inputs:

$$\sum_g p_g\, |B_g| \;=\; \sum_{x\neq 1^m} \sum_g p_g\, f_g(x) \;\le\; (2^m - 1)\,\varepsilon \;\le\; 2^m \varepsilon \;\le\; 2^{m-k},$$

where the last inequality uses $\varepsilon \le 2^{-k}$ (by definition of $k = \lceil \log_2(1/\varepsilon) \rceil$). By Markov's applied to $|B_g|$ under the distribution (p_g):

$$\Pr_g[|B_g| > 2 \cdot 2^{m-k}] \;\le\; \tfrac{1}{2}.$$

Hence at least half the probability mass satisfies $|B_g| \le 2^{m-k+1}$. □

Definition 10 (Reed–Muller code)**.** *The r-th order Reed–Muller code $RM(r, m)$ is the set of all evaluation vectors $(f(x))_{x\in\mathbb{F}_2^m} \in \mathbb{F}_2^{2^m}$ of degree $\le r$.*

Lemma 8 (Reed–Muller minimum distance; see [3, Ch. 13])**.** *The minimum Hamming weight of a nonzero codeword of $RM(r, m)$ is 2^{m-r}. That is, for any $f : \mathbb{F}_2^m \to \mathbb{F}_2$ with $\deg(f) \le r$ and $f \not\equiv 0$:*

$$|f^{-1}(1)| \;\ge\; 2^{m-r}.$$

Lemma 9 (Degree lower bound)**.** *Let $f : \mathbb{F}_2^m \to \mathbb{F}_2$ satisfy $f(1^m) = 1$ and $|f^{-1}(1)| \le 2^{m-k+1}$ for an integer $k \ge 2$. Then $\deg(f) \ge k - 1$.*

Proof. Since $f(1^m) = 1$, f is not identically zero. Suppose for contradiction that $\deg(f) \le k - 2$. By Lemma 8, $|f^{-1}(1)| \ge 2^{m-(k-2)} = 2^{m-k+2}$. But $2^{m-k+2} > 2^{m-k+1}$, contradicting $|f^{-1}(1)| \le 2^{m-k+1}$. Therefore $\deg(f) \ge k - 1$. □

Lemma 10 (Support Bound)**.** *There exists a subset $\mathcal{G}'' \subseteq \mathcal{G}$ with $\sum_{g\in\mathcal{G}''} p_g \ge \frac{1}{3}$ such that for every $g \in \mathcal{G}''$, $|A_g| \le 2^{m-k+1}$.*

Proof. From Lemma 7, at least half the weight has $|B_g| \le 2^{m-k+1}$. Among these circuits, consider the conditional distribution. For a circuit $g \in \mathcal{G}'$ to have $|A_g| = 2^{m-k+1} + 1$, it must have $|B_g| = 2^{m-k+1}$ and $1^m \in A_g$ simultaneously. Using a tighter Markov bound we get:

$$\Pr_g\big[|B_g| > \tfrac{3}{2} \cdot 2^{m-k}\big] \;\le\; \frac{2^{m-k}}{\frac{3}{2} \cdot 2^{m-k}} \;=\; \tfrac{2}{3}.$$

Hence at least $\frac{1}{3}$ of the weight has $|B_g| \le \frac{3}{2} \cdot 2^{m-k}$. Since $|B_g|$ is an integer, this forces $|B_g| \le \lfloor \frac{3}{2} \cdot 2^{m-k} \rfloor$. Since $k \ge m$ by our standing assumption, we have $2^{m-k} \le 1$, and:

$$\lfloor \tfrac{3}{2} \cdot 2^{m-k} \rfloor \;\le\; 2^{m-k+1} - 1$$

(when $m = k$, we have $\lfloor \frac{3}{2} \rfloor = 1 = 2^1 - 1 = 2^{m-k+1} - 1$; when $m < k$, $\frac{3}{2} \cdot 2^{m-k} < 1$ so the floor is $0 \le 2^{m-k+1} - 1$). Therefore, at least $\frac{1}{3}$ of the weight has $|B_g| \le 2^{m-k+1} - 1$. For these circuits, $|A_g| = |B_g| + \mathbf{1}[1^m \in A_g] \le 2^{m-k+1}$. Taking $\mathcal{G}''$ to be this set of circuits (which has weight $\ge \frac{1}{3}$) completes the proof. □

Corollary 2. *For a positive fraction of the mixture weight (over circuits $g \in \mathcal{G}''$), the function f_g satisfies $f_g(1^m) = 1$ and $|f_g^{-1}(1)| \le 2^{m-k+1}$, and therefore has $\mathbb{F}_2$-degree $\ge k - 1$.*

Proof. Lemma 10 gives a set $\mathcal{G}''$ of weight $\ge \frac{1}{3}$ with $|A_g| \le 2^{m-k+1}$ for all $g \in \mathcal{G}''$. We claim that most of $\mathcal{G}''$ also satisfies $f_g(1^m) = 1$. By (6), the total weight of circuits with $f_g(1^m) = 0$ is at most ε(since $\sum_g p_g f_g(1^m) \ge 1 - \varepsilon$ and $f_g(1^m) \in \{0, 1\}$, the weight of circuits with $f_g(1^m) = 0$ is $1 - \sum_g p_g f_g(1^m) \le \varepsilon$). Therefore, the weight of circuits in $\mathcal{G}''$ with $f_g(1^m) = 0$ is at most ε.
By inclusion-exclusion, the weight of circuits satisfying both $g \in \mathcal{G}''$ and $f_g(1^m) = 1$ is at least

$$\tfrac{1}{3} - \varepsilon \;>\; 0,$$

where the strict inequality holds because $\varepsilon < 2^{-(n-1)} \le 2^{-2} = \frac{1}{4} < \frac{1}{3}$ for $n \ge 3$. In particular, the set of such circuits is nonempty. For each such g, we have $f_g(1^m) = 1$ and $|f_g^{-1}(1)| = |A_g| \le 2^{m-k+1} = 2^{m-(k-1)}$. Applying Lemma 9 gives $\deg(f_g) \ge k - 1$. □

Lemma 11. *Any $\mathbb{F}_2$-polynomial of degree d computed by a circuit of fan-in-2 AND gates and unlimited XOR/NOT gates has AND-gate depth $\ge \lceil \log_2 d \rceil$.*

Proof. Let f be the polynomial over $\mathbb{F}_2$ computed by the circuit, with $\deg(f) = d$. We prove the following claim by induction on the AND-depth D of the circuit:
Claim. Every gate v in a circuit of AND-depth D computes a polynomial of degree $\le 2^D$.
Base case ($D = 0$): no AND gates are present, so every gate computes an $\mathbb{F}_2$-linear combination of the input variables (via XOR and NOT), which has degree $\le 1 = 2^0$.
Inductive step: Consider any gate v in a circuit of AND-depth $D \ge 1$. If v is a XOR or NOT gate, its output is an $\mathbb{F}_2$-linear combination of its inputs, so $\deg(\text{output of } v) \le \max_i \deg(\text{input}_i)$. If v is an AND gate with inputs u_1, u_2, then $\deg(\text{output of } v) = \deg(p_{u_1} \cdot p_{u_2}) \le \deg(p_{u_1}) + \deg(p_{u_2})$, where p_{u_i} is the polynomial computed at gate u_i. Each input u_i of the AND gate v lies in a sub-circuit of AND-depth $\le D - 1$, so by induction $\deg(p_{u_i}) \le 2^{D-1}$. Therefore $\deg(\text{output of } v) \le 2^{D-1} + 2^{D-1} = 2^D$.
Since f is computed at the output gate of a circuit of AND-depth D, we have $d = \deg(f) \le 2^D$. Therefore $D \ge \log_2 d$, and since D is a non-negative integer, $D \ge \lceil \log_2 d \rceil$. □

Lemma 12 (Qubit–pebble correspondence)**.** *Let U_g be a classical-reversible circuit realizing f_g as the XOR into the target qubit, with ancillae returned to $|0^a\rangle$. The AND sub-circuit of U_g (i.e., the part computing the leading monomial of f_g into an intermediate qubit) can be simulated by a reversible pebbling strategy for a binary tree of height $h = \mathrm{depth}(\textit{AND sub-circuit})$, using the following qubits as pebble slots:*

- *The* 1 *target qubit;*
- *The a ancilla qubits;*
- *The $m = n - 1$ control qubits, which may be* temporarily *repurposed as intermediate computation registers via fan-out Toffoli gates.*

This gives $n + a$ total available pebble slots, and therefore any reversible realization of a depth-h AND tree requires $n + a \geq h + 1$ by Lemma 13.

Proof. Any reversible circuit computing a Boolean function can be viewed as implementing a pebbling strategy for the DAG of its AND-gate dependencies. Specifically:

- *Place a pebble* on node v: compute the value at v into a fresh qubit (using a Toffoli/CCX gate if v is an AND node, or a CNOT if v is an XOR node).
- *Remove a pebble* from node v: uncompute the value at v by applying the same gate in reverse (reversible circuits are their own inverses up to gate ordering).

At any point in time, the "live" qubits holding intermediate AND results correspond exactly to pebbles on non-leaf nodes of the AND tree. Source nodes (leaves, i.e., linear forms of the input variables) do not require persistent storage: their values are computed on-the-fly by CNOT from the input control qubits, occupy at most 2 qubits simultaneously (the two children of the current AND gate), and are immediately uncomputed.
A control qubit x_i can serve as a non-leaf pebble slot by first copying its value via a Toffoli gate: $|x_i, 0\rangle \mapsto |x_i, x_i\rangle$, using the copy as the intermediate register, and then uncopying at the end. This uses 2 Toffoli gates per control qubit so repurposed, at $O(1)$ T-gates each, and at most h control qubits are simultaneously repurposed during any phase of the computation (since only h AND levels are active at once). Therefore, the total number of qubits available as non-leaf pebble slots is at most $1 + a + m = n + a$, and any reversible computation of a depth-h AND tree requires at least as many pebbles as specified by Lemma 13. □

Lemma 13 (Tree pebbling; cf. [4, Ch. 10])**.** *In the standard black pebbling game where leaves are always available (i.e., leaf nodes can be pebbled for free at any time), the pebbling number of a complete binary tree of height $h \geq 2$ is $h + 1$ non-leaf pebbles.*

Theorem 14. *Fix $n \geq 3$. For all sufficiently small $\varepsilon > 0$ (specifically, $\varepsilon < 2^{-(n-1)}$), every mixed circuit (in the sense of Definition 9) with a ancillae and T-count $O(\log(1/\varepsilon))$ satisfying $D_\diamond(\mathcal{E}, \mathcal{T}_n) \leq \varepsilon$ requires*

$$a \;\geq\; \left\lceil \log_2 \left\lceil \log_2(1/\varepsilon) \right\rceil \right\rceil - n.$$

In particular, $a = \Omega(\log\log(1/\varepsilon))$ for fixed n.

Proof of Theorem 14. Set $k = \lceil \log_2(1/\varepsilon) \rceil$. Our assumption $\varepsilon < 2^{-(n-1)}$ ensures $k \geq n - 1 = m$. First, note from Lemma 6, Diamond-norm error $\leq \varepsilon$ forces conditions (6) and (7). Then from Lemma 10, at least $\frac{1}{3}$ of the mixture weight (call this set $\mathcal{G}''$) satisfies $|A_g| \leq 2^{m-k+1}$. By Corollary 2, a subset of weight $\geq \frac{1}{3} - \varepsilon > 0$ additionally satisfies $f_g(1^m) = 1$. Further from Corollary 2, Lemma 9, Lemma 8 and the Reed-Muller minimum distance, for every such g, the function f_g has $\mathbb{F}_2$-degree $\deg(f_g) \geq k - 1$. From Lemma 11, the AND subcircuit of each such U_g has degree $\geq k-1$, so AND-gate depth $h \geq \lceil \log_2(k-1) \rceil$. We claim $\lceil \log_2(k-1) \rceil + 1 \geq \lceil \log_2 k \rceil$ for all $k \geq 2$, so that $h + 1 \geq \lceil \log_2 k \rceil$. To see this: let $j = \lceil \log_2 k \rceil$, so $2^{j-1} < k \leq 2^j$. Then $k - 1 \geq 2^{j-1}$, hence $\lceil \log_2(k-1) \rceil \geq j - 1 = \lceil \log_2 k \rceil - 1$, which gives $\lceil \log_2(k-1) \rceil + 1 \geq \lceil \log_2 k \rceil$ as claimed. Finally by Lemmas 12 and 13, any reversible realization of a depth-h AND tree requires $h + 1$ non-leaf pebbles, and the total available pebble slots are $n + a$. Therefore:

$$n + a \;\geq\; h + 1 \;\geq\; \lceil \log_2 k \rceil.$$

Rearranging and using $k = \lceil \log_2(1/\varepsilon) \rceil$, we obtain directly:

$$a \;\geq\; \left\lceil \log_2 \left\lceil \log_2(1/\varepsilon) \right\rceil \right\rceil - n.$$

For fixed n and $\varepsilon \to 0$: $a = \Omega(\log\log(1/\varepsilon)) = \omega(1)$. □

References

[1] Michael Beverland et al. "Lower bounds on the non-Clifford resources for quantum computations". In: *Quantum Science and Technology* 5.3 (May 2020), p. 035009. ISSN: 2058-9565. DOI: 10.1088/2058-9565/ab8963. URL: http://dx.doi.org/10.1088/2058-9565/ab8963.

[2] David Gosset, Robin Kothari, and Chenyi Zhang. *Multi-qubit Toffoli with exponentially fewer T gates*. 2025. arXiv: 2510.07223 [quant-ph]. URL: https://arxiv.org/abs/2510.07223.

[3] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*. 2nd. North-holland Publishing Company, 1978.

[4] John E. Savage. *Models of Computation: Exploring the Power of Computing*. 1st. USA: Addison-Wesley Longman Publishing Co., Inc., 1997. ISBN: 0201895390.