How Much Workspace Does Optimal-T Synthesis Need? Ancilla Complexity in the Mixed Clifford+T Model

Junseo Lee* Arul Rhik Mazumder†

Abstract

Contents

*Seoul National University. Email: junseolee@fas.harvard.edu
†Carnegie Mellon University. Email: arulm@andrew.cmu.edu

1 Introduction

The synthesis of quantum circuits over the Clifford+T gate set is a central problem in fault-tolerant quantum computing. Because T gates dominate implementation cost in error-corrected architectures, a large body of work has focused on minimizing T-count [AM19; Bar+95; Sel13]. Ancilla qubits are a second scarce resource, and the tradeoff between non-Clifford cost and workspace is a basic question in circuit synthesis.

Recent work of Gosset, Kothari, and Zhang (GKZ) [GKZ25] introduced a separation between exact and approximate synthesis for multi-controlled Toffoli gates. They showed that in a mixed Clifford+T model, where one may sample from a distribution over Clifford+T circuits, the n-qubit Toffoli gate admits an ε-approximation using only

$$T = O(\log(1/\varepsilon))$$

T gates. Moreover, they proved a matching lower bound, establishing

$$T_\varepsilon^{\text{mixed}}(\text{Toff}_n) = \Theta(\log(1/\varepsilon))$$

whenever $\log(1/\varepsilon) \leq n$.

Consequently, the asymptotic T count complexity of approximate Toffoli synthesis is already understood. A natural remaining question is the ancilla complexity required to achieve this optimal scaling. The GKZ construction evaluates

$$k = \Theta(\log(1/\varepsilon))$$

random parity tests and computes their logical OR. In its standard form, all intermediate values are stored simultaneously, resulting in an ancilla requirement linear in k. This raises the following question.

> Can one preserve the optimal $T = \Theta(\log(1/\varepsilon))$ scaling while substantially reducing the ancilla count?

The question was first posed for the Toffoli gate, and there the answer is deflating: a single ancilla suffices at the optimal T-count, for every ε. In the genuinely-approximate regime $\varepsilon \geq 2^{-(n-3)}$ (equivalently $k \leq n-1$), a linear-algebra construction (Section 5) samples $k = \Theta(\log(1/\varepsilon))$ parity vectors, conditions on their linear independence, realizes them as coordinates of the input register through an $\mathbb{F}_2$ basis change (a CNOT circuit), and computes the GKZ decision with one call to Barenco et al.'s one-ancilla k-fold Toffoli [Bar+95]. For smaller ε this construction no longer applies, but the error is then so small that Barenco's *exact* one-ancilla decomposition suffices, and its $O(n)$ T-gates are $O(\log(1/\varepsilon))$ in that regime. Either way one ancilla attains the optimal T-count (Theorem 7), so the ancilla/T tradeoff for Toff_n is *degenerate* and the OR-tree and counter constructions one might reach for are never needed. We treat this case as a warm-up.

The contribution of this paper is what happens for general Boolean oracle unitaries U_f, where the workspace question does *not* collapse. Under the GKZ Fourier-sampling framework the target is a signed threshold of $k = \Theta(\|\widehat{f}\|_1^2 \log(1/\varepsilon))$ parities rather than an OR, and the linear-algebra shortcut has no analogue. Here the depth-first, workspace-reusing evaluation behind the OR-tree becomes a running tally, and we obtain a genuine two-dimensional ancilla/T tradeoff with two corners: $(\lambda, T) = (O(\log k), O(k))$ by the tally, and $(\lambda, T) = (O(1), \text{poly}(n, k))$ by a Barrington branching program. We prove the tally optimal among single-pass evaluators, and reduce the one remaining question, whether re-reading the parities beats $O(\log k)$ ancillae at a linear T-count, to a bounded-width branching-program problem.

1.1 Results

(1) **(Warm-up.)** For every $n \geq 2$ and $\varepsilon \in (0,1)$, Toff_n has a mixed Clifford+T ε-approximation using one ancilla and optimal $T = O(\log(1/\varepsilon))$ (Theorem 7): a linear-algebra construction (Theorem 16) for $\varepsilon \geq 2^{-(n-3)}$, and the exact Barenco decomposition below that. The ancilla/T tradeoff for Toff_n is degenerate.

(2) **(Main: a general-oracle tradeoff.)** For a general $f : \{0,1\}^n \to \{0,1\}$ the GKZ top function is a signed threshold rather than an OR, and we give two corners of an ancilla/T tradeoff for U_f, with $k = \Theta(\|\widehat{f}\|_1^2 \log(1/\varepsilon))$:

$$(\lambda, T) = \big(O(\log k),\ O(k)\big) \text{ via a running tally (Theorem 22), and } (\lambda, T) = \big(O(1),\ \mathrm{poly}(n,k)\big) \text{ via Barrington (Proposition 24).}$$

The running tally is the general-oracle form of the depth-first workspace-reuse principle developed for Toff_n in Section 4.

(3) **(Lower bound and barrier.)** The tally is optimal among single-pass evaluators (Proposition 25). Whether re-reading the parities can beat $O(\log k)$ ancillae at the T-optimal count is open; we reduce it to a linear-length bounded-width branching-program question, pinpointing why it is hard.

(4) **(Application.)** Inside Tan's unitary synthesis framework [Tan25], for $\max_x \|\widehat{f}_x\|_1 = O(1)$ the running-tally gadget gives $O(\log(n + \log(1/\varepsilon)))$ ancillae, exponentially smaller than both Tan's $O(2^{N/2})$ and the flat-GKZ count.

1.2 Related Work

Exact Toffoli synthesis. Barenco et al. [Bar+95] showed Toff_n can be implemented exactly with $O(n)$ T-gates and $n-2$ ancillae, and GKZ showed this is essentially tight in the unitary model [GKZ25].

General unitary synthesis. The LKS–GKW product bound $(n+\lambda)\cdot R = \Omega(2^{2n})/\mathrm{poly}(\log(1/\varepsilon))$ is established in [LKS24; GKW24]. Tan [Tan25] achieves $R = O(2^{3N/2})$ T-gates with $\lambda = O(2^{N/2})$ ancillae for block size κ and $N = n - \kappa$. In Section 7 we show that for unitaries with bounded Fourier ℓ_1 norm, the mixed model reduces ancillae exponentially relative to Tan's exact construction.

Ancilla–T tradeoffs. Tradeoffs between ancilla count and gate count have been studied for Toffoli decompositions [Bar+95; Sel13], phase-polynomial circuits [Amy+13], and the Reed–Muller/T-count minimization framework of Amy and Mosca [AM19]. None of these work in the mixed model, which changes the problem qualitatively.

2 Preliminaries

We work in the mixed Clifford+T model throughout. The n-qubit Toffoli gate is

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

Sections 6–7 also treat Boolean oracle unitaries U_f for arbitrary $f : \{0,1\}^n \to \{0,1\}$.

2.1 Distance Measures and the Mixed Model

Definition 1 (Diamond distance)**.** For quantum channels E_1, E_2 on n qubits,

$$D_\diamond(E_1, E_2) = \sup_{\ell,\,\rho} D\big((E_1 \otimes I_\ell)(\rho),\ (E_2 \otimes I_\ell)(\rho)\big),$$

where $D(\rho, \sigma) = \frac{1}{2}\|\rho - \sigma\|_1$ is trace distance.

Definition 2 (Mixed T-count)**.** A *mixed Clifford+T implementation* of a unitary U at error ε is a distribution $\{(p_i, V_i)\}$ over Clifford+T circuits such that the channel

$$\mathcal{E}(\rho) = \mathrm{Tr}_{\mathrm{anc}}\left[\sum_i p_i\, V_i(\rho \otimes |0_a\rangle\langle 0_a|)V_i^\dagger\right]$$

satisfies $D_\diamond(\mathcal{E}, U) \le \varepsilon$. The mixed T-count $T_\varepsilon^{\mathrm{mixed}}(U)$ is the minimum over all such implementations of $\max_i T\text{-count}(V_i)$.

2.2 GKZ Results

We recall the results of [GKZ25] that our constructions build on.

Theorem 3 ([GKZ25])**.** *For any n and $\varepsilon > 0$, $T_\varepsilon^{\mathrm{mixed}}(\mathrm{Toff}_n) = O(\log(1/\varepsilon))$. The bound is achieved by sampling $k = \lceil \log_2(1/\varepsilon) \rceil + 2$ random subsets $S_1, \ldots, S_k \subseteq [n-1]$, computing parities $p_j = \bigoplus_{\ell \in S_j} x_\ell$, and flipping the target iff $\mathrm{OR}_k(p_1, \ldots, p_k) = 0$.*

Theorem 4 ([GKZ25])**.** *The GKZ mixed channel satisfies $D_\diamond(\mathcal{E}, \mathrm{Toff}_n) \le 4 \cdot 2^{-k}$.*

Theorem 5 ([GKZ25])**.** *For sufficiently large n and $1/\varepsilon$,*

$$T_\varepsilon^{\mathrm{mixed}}(\mathrm{Toff}_n) \ \ge\ T_\varepsilon^{\mathrm{adaptive}}(\mathrm{Toff}_n) \ =\ \Omega\big(\min\{n, \log(1/\varepsilon)\}\big).$$

Together, Theorems 3 and 5 give $T_\varepsilon^{\mathrm{mixed}}(\mathrm{Toff}_n) = \Theta(\log(1/\varepsilon))$ whenever $\log(1/\varepsilon) \le n$. Our constructions do not improve this asymptotic T-count; they reduce ancilla count while preserving it. We will also use the following extension to general Boolean functions.

Theorem 6 ([GKZ25])**.** *For $f : \{0,1\}^n \to \{0,1\}$ and $\varepsilon > 0$,*

$$T_\varepsilon^{\mathrm{mixed}}(U_f) = O\Big(\|\widehat{f}\|_1^2 \log(1/\varepsilon)\Big).$$

We use the following standard OR subroutine throughout.

Algorithm 1 $\textsc{Or}(q_1, \ldots, q_r,\ t)$

Require: 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 $X^{\otimes r}$ to $q_1, \ldots, q_r$
2: Apply Toff_{r+1} with controls $q_1, \ldots, q_r$ and target t
3: Apply X to t; apply $X^{\otimes r}$ to $q_1, \ldots, q_r$ =0

3 Warm-up: One Ancilla Suffices for Toffoli

We begin with the special case that motivated this work, $f = \mathrm{Toff}_n$, where the workspace question collapses. The result below combines the one-ancilla construction of Section 5 with the exact decomposition of Barenco et al. It is a warm-up: the contribution of this paper is the general-oracle picture of Section 6, where no such collapse occurs and the ancilla count becomes a genuine resource.

Theorem 7 (One ancilla suffices for Toff_n)**.** *For every $n \geq 2$ and every $\varepsilon \in (0, 1)$, there is a mixed Clifford+T implementation of Toff_n with $D_\diamond(\mathcal{E}, \mathrm{Toff}_n) \leq \varepsilon$ using a single ancilla qubit and $T = O(\log(1/\varepsilon))$ T-gates. This T-count is optimal up to constants.*

Proof. Split on the error rate at $\varepsilon = 2^{-(n-3)}$, i.e. at $k = \lceil \log_2(1/\varepsilon) \rceil + 2 = n - 1$.

Genuine approximation, $\varepsilon \geq 2^{-(n-3)}$ (so $k \leq n - 1$). The construction of Theorem 16 gives a single ancilla, $T = O(\log(1/\varepsilon))$, and $D_\diamond \leq \varepsilon$.

Small error, $\varepsilon < 2^{-(n-3)}$ (so $k > n - 1$). The linear-algebra construction no longer applies, but Barenco et al.'s exact one-ancilla decomposition [Bar+95, Cor. 7.4] does: a deterministic (hence mixed) Clifford+T circuit with $D_\diamond = 0 \leq \varepsilon$, one ancilla, and $T = O(n) = O(\log(1/\varepsilon))$ since $\log(1/\varepsilon) > n - 3$.

Either way one ancilla and $O(\log(1/\varepsilon))$ T-gates suffice. Optimality is the GKZ lower bound (Theorem 5): $T_\varepsilon^{\mathrm{mixed}}(\mathrm{Toff}_n) = \Omega(\min\{n, \log(1/\varepsilon)\})$, which is $\Omega(\log(1/\varepsilon))$ in the first regime and $\Omega(n) = \Omega(\log(1/\varepsilon))$ in the second. □

Remark 8. The two halves meet at $\varepsilon \approx 2^{-n}$, the point below which approximating Toff_n is the same as computing it exactly. So for Toff_n the ancilla/T tradeoff is degenerate. The recursive OR-tree and Hamming-weight-counter constructions one might reach for (Section 4), which spend $\omega(1)$ ancillae, are never needed here; their value is as the workspace-reuse template for the general-oracle case, where the collapse does not happen.

4 The Bridge: Depth-First Workspace Reuse

This section develops the workspace-reuse principle behind the paper. It is dominated for Toff_n by the one-ancilla warm-up of Theorem 7, but it is the template that carries to general Boolean oracles in Section 6, where the one-ancilla shortcut is unavailable and the principle gives the best bounds we know. For Toff_n the GKZ top function is an OR, which is associative and so admits the tree form below; for a general oracle it becomes a threshold, and the same depth-first idea reappears as a running tally (Section 6).

The GKZ construction stores all k parity outputs simultaneously, requiring $\Theta(\log(1/\varepsilon))$ ancillae. By associativity of OR, the computation $\mathrm{OR}_k(p_1, \ldots, p_k)$ can instead be organized hierarchically: intermediate values are computed, consumed, and then uncomputed depth-first. This trades storage for recomputation while leaving the Boolean function and hence the GKZ error guarantee unchanged.

4.1 Two-Level OR Tree

Algorithm 2 Approximate Toff_n via 2-Level OR Tree

Require: Controls $|x_1, \ldots, x_{n-1}\rangle$, target $|b\rangle$, $\varepsilon > 0$; m workspace qubits $|w_i\rangle$; $r = \lceil k/m \rceil$ storage qubits $|s_j\rangle$

Ensure: $D_\diamond(\mathcal{E}, \mathrm{Toff}_n) \le \varepsilon$

1: $k \leftarrow \lceil \log_2(1/\varepsilon) \rceil + 2$; sample $S_1, \ldots, S_k$ uniformly
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: $\quad \textsc{Or}(w_1, \ldots, w_{m_i},\ s_i)$; uncompute parities
7: **end for**
8: $\textsc{Or}(s_1, \ldots, s_r,\ b)$; apply X to b
9: **for** $i = r$ **downto** 1 **do**
10: $\quad$ Recompute parities; $\textsc{Or}^\dagger(w_1, \ldots, w_{m_i},\ s_i)$; uncompute parities
11: **end for**
12: Apply $X^{\otimes(n-1)}$ to controls $=0$

Lemma 9. *Algorithm 2 computes the same Boolean function as the flat GKZ construction, i.e.,*

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

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

Proof. Associativity of OR. □

Theorem 10. *Algorithm 2 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. By Lemma 9, each sampled circuit W_g is identical to the GKZ unitary, so $D_\diamond(\mathcal{E}, \mathrm{Toff}_n) \le \varepsilon$ by Theorem 4.

For the ancilla count, we track peak simultaneous occupancy, which occurs just before the outer Or fires on line 7. At that point the m workspace qubits hold the current parity batch and the $r = \lceil k/m \rceil$ storage qubits hold the inner-OR outputs; prior batches have been uncomputed. The Barenco decomposition of each $\textsc{Or}_{m_i}$ chains Toffoli-3 gates sharing a workspace qubit as the internal ancilla [Bar+95], so no extra qubits are needed beyond those already counted. Total ancilla count is $f(m) = m + \lceil k/m \rceil$, minimized at $m = \lceil \sqrt{k} \rceil$, giving $O(\sqrt{k}) = O(\sqrt{\log(1/\varepsilon)})$.

For the T-count, each inner $\textsc{Or}_{m_i}$ is computed once and uncomputed once, contributing $2 \times 8(m_i - 2)$ T-gates via the relative-phase Toffoli decomposition of [Bar+95]. The outer $\textsc{Or}_r$ likewise contributes $2 \times 8(r-2)$ T-gates. Summing:

$$\begin{aligned} T\text{-count} &= 2\sum_{i=1}^{r} 8(m_i - 2) + 2 \cdot 8(r-2) \\ &= 16(k - 2r) + 16(r-2) = 16k - 16r - 32 = O(k) = O(\log(1/\varepsilon)), \end{aligned}$$

using $\sum_{i=1}^{r} m_i = k$ and $r = O(\sqrt{k})$. □

4.2 d-Level Tree

The two-level construction groups k parities into r batches of size m, computes OR of each batch, then ORs the results. The d-level tree applies this recursively.

Definition 11 (d-level OR tree)**.** Fix $k \geq 1$ and fan-ins $m_1, \ldots, m_{d-1} \geq 2$. Level 0 consists of the k parity bits $p_1, \ldots, p_k$. For $1 \leq \ell \leq d-1$, level ℓ partitions the outputs of level $\ell - 1$ into consecutive groups of size m_ℓ and computes OR of each group into a *level-ℓ storage qubit*, producing $n_\ell = \lceil n_{\ell-1}/m_\ell \rceil$ outputs with $n_0 = k$. The root (level d) ORs the n_{d-1} top-level storage qubits into the target and applies X.

The ancilla savings come from *depth-first execution*: after computing and storing a subtree's OR output, we immediately uncompute the subtree's internal qubits before moving to the next subtree, so registers are reused across groups.

Algorithm 3 d-Level Approximate Toff_n

Require: Controls $|x_1, \ldots, x_{n-1}\rangle$, target $|b\rangle$, $\varepsilon > 0$, depth $d \geq 2$, fan-ins $m_1, \ldots, m_{d-1}$
Ensure: $D_\diamond(\mathcal{E}, \mathrm{Toff}_n) \leq \varepsilon$

$k \leftarrow \lceil \log_2(1/\varepsilon) \rceil + 2$; sample $S_1, \ldots, S_k$
Apply $X^{\otimes(n-1)}$ to controls
FillStorage$(d-1,\ 1,\ k)$
Or$\big(s_1^{(d-1)}, \ldots, s_{n_{d-1}}^{(d-1)},\ b\big)$; apply X to b
FillStorage$(d-1,\ 1,\ k)^\dagger$
Apply $X^{\otimes(n-1)}$ to controls =0

Algorithm 4 FillStorage$(\ell,\ start,\ count)$

Require: Level ℓ, starting parity index *start*, count *count*
Ensure: $s_i^{(\ell)} = \mathrm{OR}$(group-$i$ parities) for each i; all workspace and intermediate qubits in $|0\rangle$

if $\ell = 1$ **then**
  for $i = 1$ **to** $\lceil count/m_1 \rceil$ **do**
    $m_i' \leftarrow \min(m_1,\ count - (i-1)m_1)$
    Compute parities $|w_j\rangle \oplus= \bigoplus_{\ell' \in S_{start+(i-1)m_1+j-1}} x_{\ell'}$ for $j = 1, \ldots, m_i'$
    Or$(w_1, \ldots, w_{m_i'},\ s_i^{(1)})$; uncompute parities
  end for
  return
end if
$r \leftarrow \lceil count / \prod_{j=1}^{\ell} m_j \rceil$
for $i = 1$ **to** r **do**
  $start_i \leftarrow start + (i-1)\prod_{j=1}^{\ell} m_j$; $cnt_i \leftarrow \min\big(\prod_{j=1}^{\ell} m_j,\ count - (i-1)\prod_{j=1}^{\ell} m_j\big)$
  FillStorage$(\ell - 1,\ start_i,\ cnt_i)$
  $r_i' \leftarrow \lceil cnt_i / \prod_{j=1}^{\ell-1} m_j \rceil$
  Or$\big(s_1^{(\ell-1)}, \ldots, s_{r_i'}^{(\ell-1)},\ s_i^{(\ell)}\big)$
  FillStorage$(\ell - 1,\ start_i,\ cnt_i)^\dagger$
end for=0

Lemma 12 (Peak ancilla occupancy)**.** *During depth-first execution of Algorithm 3 with fan-ins $m_1, \dots, m_{d-1}$, the number of ancilla qubits simultaneously occupied is at most*

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

where $n_{d-1} = \lceil k / \prod_{\ell=1}^{d-1} m_\ell \rceil$.

Proof. Peak occupancy occurs just before the OR call at the deepest level of the first group. At that moment: the m_1 workspace qubits hold the current parity batch (prior batches uncomputed); for each $1 \le \ell \le d-2$, at most $m_{\ell+1}$ level-ℓ storage qubits are occupied, one per filled child of the current level-$(\ell+1)$ batch; and all n_{d-1} top-level storage qubits, allocated at line 3 of Algorithm 3, remain occupied until the global uncompute at line 5. Summing gives $m_1 + (m_2 + \cdots + m_{d-1}) + n_{d-1}$. □

Theorem 13. *With $m_1 = \cdots = m_{d-1} = m$, the ancilla count is $A(d) = (d-1)m + \lceil k/m^{d-1} \rceil$, minimized at $m = \lceil k^{1/d} \rceil$, giving $A(d) = O(d \cdot k^{1/d}) = O(d \cdot \log(1/\varepsilon)^{1/d})$.*

Proof. Treating $A(m) = (d-1)m + k\, m^{-(d-1)}$ as a continuous function of m and setting $dA/dm = 0$ gives $m = k^{1/d}$, at which $A = d\, k^{1/d}$. □

Corollary 14. *Taking $m_\ell = c$ for a fixed constant $c \ge 2$ requires $d = \lceil \log_c k \rceil$ levels and yields $A = O(\log\log(1/\varepsilon))$ ancillae with $O(\log(1/\varepsilon))$ T-gates.*

Proof. With $m = c$ and $d = \lceil \log_c k \rceil$, $n_{d-1} = \lceil k/c^{d-1} \rceil = O(1)$, so $A = (d-1)c + O(1) = O(\log_c k) = O(\log\log(1/\varepsilon))$. □

Theorem 15. *For any n, $\varepsilon > 0$, and $d \ge 2$, Algorithm 3 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 $O(d \cdot \log(1/\varepsilon)^{1/d})$ ancilla qubits and $O(\log(1/\varepsilon))$ T-gates. At constant fan-in ($d = \lceil \log_c k \rceil$ for fixed $c \ge 2$), the ancilla count is $O(\log\log(1/\varepsilon))$.*

Proof. Correctness follows by induction on d: the d-level OR tree computes $\mathrm{OR}_k(p_1, \dots, p_k)$ by associativity of OR (Lemma 9), so each sampled circuit is identical to the GKZ unitary and $D_\diamond(\mathcal{E}, \mathrm{Toff}_n) \le \varepsilon$ by Theorem 4. The ancilla bound is Theorem 13 and Corollary 14.

For the T-count, the tree has $O(k)$ OR nodes, each at some level ℓ with fan-in m_ℓ. Computing and uncomputing all nodes contributes

$$2 \sum_{\text{nodes}} O(m_\ell) = O\Big(\sum_{\ell=0}^{d-1} n_\ell\Big) = O(k),$$

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

5 A One-Ancilla Construction for Toffoli

For Toff_n the workspace question collapses: a single ancilla suffices whenever the approximation is genuine (that is, $\varepsilon \ge 2^{-(n-3)}$, equivalently $k \le n-1$). The construction is due to Ryan O'Donnell.[1] It chooses the k parity vectors $a_1, \dots, a_k$ to be linearly independent over $\mathbb{F}_2$, applies

[1] Personal communication. We include it, with his permission, as the warm-up that motivates the general-oracle question of Section 6.

the corresponding basis change as a CNOT circuit, and calls Barenco et al.'s 1-ancilla k-fold Toffoli directly. Throughout this section, $k = \lceil \log_2(1/\varepsilon) \rceil + 2$ and $m = n - 1$.

Algorithm 5 One-Ancilla Approximate Toff_n

Require: Controls $|x_1, \ldots, x_m\rangle$, target $|b\rangle$, $\varepsilon \geq 2^{-(n-3)}$
Ensure: $D_\diamond(\mathcal{E}, \mathrm{Toff}_n) \leq \varepsilon$
1: $k \leftarrow \lceil \log_2(1/\varepsilon) \rceil + 2$
2: Draw $a_1, \ldots, a_k \in \mathbb{F}_2^m$ i.i.d. uniformly; resample the entire k-tuple if linearly dependent (expected $O(1)$ draws, since $\Pr[\text{lin. indep.}] \geq \prod_{i \geq 1}(1 - 2^{-i}) > 0.28$)
3: Extend $\{a_1, \ldots, a_k\}$ to a basis of $\mathbb{F}_2^m$ by appending $m - k$ standard basis vectors; let M be the $m \times m$ invertible matrix with rows $a_1, \ldots, a_m$
4: Apply $X^{\otimes m}$ to controls
5: Apply the CNOT circuit $|x\rangle \mapsto |Mx\rangle$
6: Apply $X^{\otimes k}$ to the first k control qubits
7: Apply Barenco's 1-ancilla k-fold Toffoli [Bar+95, Cor. 7.4 + Lem. 7.2] with controls $(Mx)_1, \ldots, (Mx)_k$ and target b
8: Apply $X^{\otimes k}$ to the first k control qubits
9: Apply the inverse CNOT circuit $|Mx\rangle \mapsto |x\rangle$
10: Apply $X^{\otimes m}$ to controls $=0$

Theorem 16. *For any $n \geq 2$ and $\varepsilon \geq 2^{-(n-3)}$ (equivalently $k = \lceil \log_2(1/\varepsilon) \rceil + 2 \leq n - 1$), Algorithm 5 implements a mixed channel $\mathcal{E}$ with $D_\diamond(\mathcal{E}, \mathrm{Toff}_n) \leq \varepsilon$ using 1 ancilla qubit and $O(\log(1/\varepsilon))$ T-gates per circuit.*

Proof. Function computed. The map M places the k parities $a_j \cdot x$ in the first k positions of Mx. The $X^{\otimes k}$ followed by the k-fold Toffoli flips b iff all k leading bits of Mx are 0, i.e. $a_j \cdot x = 0$ for $j = 1, \ldots, k$. Uncomputing M and removing the GKZ $X^{\otimes m}$ conjugation gives the GKZ random unitary W_g for parity vectors $a_1, \ldots, a_k$.

Diamond-distance bound. We show that conditioning on linear independence does not spoil the GKZ error bound. Fix any nonzero $y \in \mathbb{F}_2^m$ (a value of the negated controls); the sampled circuit flips the target erroneously on y exactly when all k parities vanish, i.e. $a_j \cdot y = 0$ for every j. Under μ^*, the uniform distribution over linearly independent k-tuples,

$$\Pr_{\mu^*}\big[\, a_j \cdot y = 0 \text{ for all } j \,\big] = \frac{\#\{\text{independent } k\text{-tuples in } y^\perp\}}{\#\{\text{independent } k\text{-tuples in } \mathbb{F}_2^m\}} = \prod_{i=0}^{k-1} \frac{2^{m-1} - 2^i}{2^m - 2^i} \;\leq\; 2^{-k},$$

since each factor is at most $\frac{1}{2}$ (and vanishes once $i = m - 1$, so the bound holds throughout $k \leq m$). The conditioned per-input error is thus at most 2^{-k}, no larger than under uniform sampling, so the diamond-distance analysis of [GKZ25] gives $D_\diamond(\mathcal{E}_{\mu^*}, \mathrm{Toff}_n) \leq 4 \cdot 2^{-k} \leq \varepsilon$ for $k = \lceil \log_2(1/\varepsilon) \rceil + 2$. Conditioning on independence carries no penalty, so the construction is valid for the full range $k \leq n - 1$.

Ancilla count. CNOT and X layers are ancilla-free. Barenco's k-fold Toffoli uses exactly 1 ancilla [Bar+95, Cor. 7.4 + Lem. 7.2].

T-count. Barenco's decomposition uses $\Theta(k)$ Toffoli-3 gates at 4 T-gates each, for total $\Theta(k) = \Theta(\log(1/\varepsilon))$. □

Remark 17. The one-ancilla construction of Theorem 16 covers the entire regime $\varepsilon \geq 2^{-(n-3)}$ (i.e. $k \leq n - 1$) with $\lambda = 1$ and $T = O(\log(1/\varepsilon))$. For $\varepsilon < 2^{-(n-3)}$ it no longer applies (k would exceed

the $n-1$ available independent parities), but the error is now so small that one simply computes Toff_n exactly: Barenco et al.'s exact one-ancilla decomposition [Bar+95, Cor. 7.4] gives $\lambda = 1$ and $T = O(n) = O(\log(1/\varepsilon))$ (as $\log(1/\varepsilon) > n-3$ here), which is optimal by Theorem 5. Either way a single ancilla attains the optimal T-count; this is Theorem 7.

The d-tree of Section 4, which uses $\Theta(\log\log(1/\varepsilon))$ ancillae, is therefore never needed for Toff_n. Its role is not as a Toffoli construction but as the workspace-reuse template: it is the case of the depth-first evaluation principle that survives to general Boolean oracles (Section 6), where the top function is a signed threshold, Barenco's shortcut has no analogue, and the workspace question does not collapse.

Remark 18. Algorithm 5 relies on OR_k reducing to AND_k via De Morgan, so Barenco's 1-ancilla k-fold Toffoli computes g directly. For a general Boolean function f, the GKZ top function g (Lemma 19) is a signed threshold, not an OR, so Barenco's shortcut does not apply. One can still evaluate g with $O(1)$ ancillae via a Barrington branching program (Proposition 24), but only at a $\mathrm{poly}(n,k)$ T-count; the running-tally construction of Section 6 instead keeps the T-count optimal at the price of $O(\log k)$ ancillae. Section 6.3 discusses the resulting tradeoff.

6 General Oracles: A Workspace–T Tradeoff

Theorem 7 settles Toff_n: its optimal T-count needs only one ancilla. This section is the main event. For a general Boolean oracle U_f the workspace question does *not* collapse, and we map out what does happen. Using the GKZ Fourier-sampling framework of Theorem 6, the target becomes a signed threshold rather than an OR, and the ancilla cost becomes a genuine parameter. We give matching upper and lower bounds in the natural single-pass model and pin the general question to a branching-program barrier.

6.1 Structure of the Top Function

Let $f : \{0,1\}^n \to \{0,1\}$ with Fourier expansion $f(x) = \sum_{S\subseteq[n]} \widehat{f}(S)\,(-1)^{\langle S,x\rangle}$. GKZ samples $k = \Theta(\|\widehat{f}\|_1^2 \log(1/\varepsilon))$ sets $S_1,\dots,S_k$ with probability proportional to $|\widehat{f}(S_j)|$, computes parities $z_j = \bigoplus_{\ell\in S_j} x_\ell$, and applies a *top function* $g : \{0,1\}^k \to \{0,1\}$ to decide whether to flip the target.

Lemma 19. *Let $S_1,\dots,S_k$ be i.i.d. with $\Pr[S_j = S] = |\widehat{f}(S)|/\|\widehat{f}\|_1$ and let $\sigma_j = \mathrm{sign}(\widehat{f}(S_j))$. The GKZ rounding rule produces*

$$g(z_1,\dots,z_k) = \mathbf{1}\left[\sum_{j=1}^{k} \sigma_j\,(-1)^{z_j} \;\geq\; \frac{k}{2\|\widehat{f}\|_1}\right].$$

Proof. Each term $\sigma_j(-1)^{z_j}$ is an unbiased estimator of $\|\widehat{f}\|_1 \cdot \mathrm{sign}(\widehat{f}(S_j)) \cdot (-1)^{\langle S_j,x\rangle}$ weighted by $|\widehat{f}(S_j)|/\|\widehat{f}\|_1$. Taking expectation:

$$\mathbb{E}\left[\frac{\|\widehat{f}\|_1}{k}\sum_{j=1}^{k}\sigma_j(-1)^{z_j}\right] = \sum_S \widehat{f}(S)(-1)^{\langle S,x\rangle} = f(x).$$

The threshold form follows by rearranging $\frac{\|\widehat{f}\|_1}{k}\sum_j \sigma_j(-1)^{z_j} \geq \frac{1}{2}$. □

Remark 20. For $f = \mathrm{OR}_n$, one verifies $\|\widehat{\mathrm{OR}_n}\|_1 \leq 2$, so the threshold is $\sum_j \sigma_j(-1)^{z_j} \geq k/4$. After the GKZ $X^{\otimes(n-1)}$ conjugation, $\sigma_j = -1$ for all j and the threshold reduces to $\mathrm{OR}_k(p_1,\dots,p_k) = 1$, recovering Theorem 3.

6.2 Ancilla reduction via a running tally

For most f, the top function g is a signed threshold over $\{+1,-1\}^k$ rather than an OR. To evaluate it reversibly with few ancillae, we maintain a running tally.

Definition 21 (Tally register)**.** A *tally register* stores a running partial sum $T_j = \sum_{i=1}^{j} \sigma_i(-1)^{z_i} \in [-j, j]$ in $\lceil \log_2(k+1) \rceil + 1$ bits, with a final comparison to the threshold $\lceil k/(2\|\widehat{f}\|_1) \rceil$.

Unlike OR, a signed threshold does not decompose into a tree of partial thresholds: thresholding a subtree discards the magnitude of its partial sum, which the root still needs. We therefore evaluate g with a single running tally instead of a tree, iterating over the k signed parities and accumulating them into one shared register. This gives the following bound.

Theorem 22. *Let $f : \{0,1\}^n \to \{0,1\}$, $\varepsilon > 0$, and $k = \Theta(\|\widehat{f}\|_1^2 \log(1/\varepsilon))$. There is a mixed Clifford+T implementation of U_f with $D_\diamond(\mathcal{E}, U_f) \le \varepsilon$ using*

$$T = O\big(\|\widehat{f}\|_1^2 \log(1/\varepsilon)\big) \qquad \text{and} \qquad \lambda = O(\log k) = O\big(\log(\|\widehat{f}\|_1^2 \log(1/\varepsilon))\big).$$

Proof. Run GKZ importance sampling to produce signed parities $(z_1, \sigma_1), \ldots, (z_k, \sigma_k)$, where each sign $\sigma_j \in \{+1, -1\}$ is fixed by the sample. Writing $c_j = \mathbf{1}[\sigma_j = -1]$ and $b_j = z_j \oplus c_j$, we have $\sigma_j(-1)^{z_j} = 1 - 2b_j$, so $\sum_j \sigma_j(-1)^{z_j} = k - 2P$ with $P = \sum_j b_j$. The threshold of Lemma 19 is therefore a single comparison on the Hamming weight P.

Maintain one counter of $\lceil \log_2(k+1) \rceil$ bits, initialized to 0. For $j = 1, \ldots, k$: compute $z_j = \bigoplus_{\ell \in S_j} x_\ell$ into a single workspace qubit ($O(1)$ ancilla, $O(n)$ Clifford gates), increment the counter iff $b_j = 1$, then uncompute z_j. After the loop the counter holds P; compare it to the threshold, flip the target on the comparison bit, and uncompute the counter by reversing the loop.

Ancilla. The counter uses $\lceil \log_2(k+1) \rceil$ qubits and the workspace $O(1)$, giving $\lambda = O(\log k)$.

T-count. The counter increases monotonically from 0 to $P \le k$, so across the k increments bit i flips $O(k/2^i)$ times and the total is $O(k)$ Toffoli-3 gates; the single final comparison costs $O(\log k)$ T-gates. Hence $T = O(k) = O(\|\widehat{f}\|_1^2 \log(1/\varepsilon))$. Correctness follows from Lemma 19 and the diamond-distance analysis of [GKZ25]. □

Example 23 (MAJ_n)**.** The majority function satisfies $\|\widehat{\mathrm{MAJ}_n}\|_1 = \Theta(\sqrt{n})$ [ODo21], so $k = \Theta(n \log(1/\varepsilon))$. By Lemma 19, the top function is

$$g(z_1, \ldots, z_k) = \mathbf{1}\left[\sum_{j=1}^{k} \sigma_j(-1)^{z_j} \ge \Theta(k/\sqrt{n}) \right],$$

and Theorem 22 gives $T = O(n \log(1/\varepsilon))$ and $\lambda = O(\log(n \log(1/\varepsilon)))$.

The one-ancilla construction of Section 5 does not generalize here, since g is a genuine threshold rather than an OR. The running tally evaluates g with $O(\log k) = O(\log(n \log(1/\varepsilon)))$ ancillae at the T-optimal count; alternatively, since $g \in \mathsf{NC}^1$, Proposition 24 evaluates it with $O(1)$ ancillae at a $\mathrm{poly}(n, k)$ T-count. Which point on this tradeoff is forced remains open (Section 6.3).

6.3 The ancilla/T tradeoff and a lower-bound barrier

For Toff_n the top function is an OR and Theorem 7 collapses the tradeoff to a single ancilla. For a general threshold top function the two constructions above sit at opposite corners,

$$\text{(tally)} \quad (\lambda, T) = \big(O(\log k),\, O(k)\big), \qquad \text{(Barrington)} \quad (\lambda, T) = \big(O(1),\, \mathrm{poly}(n, k)\big),$$

where the second corner is the following.

Proposition 24 (Constant-ancilla evaluation)**.** *For any $f : \{0,1\}^n \to \{0,1\}$ and $\varepsilon > 0$, there is a mixed Clifford+T implementation of U_f with $D_\diamond(\mathcal{E}, U_f) \le \varepsilon$ using $O(1)$ ancilla qubits and* $\mathrm{poly}(n,k)$ *T-gates, where $k = \Theta(\|\widehat{f}\|_1^2 \log(1/\varepsilon))$.*

Proof. Each sampled GKZ circuit applies the top function g of Lemma 19, a threshold of the k parities $z_j = a_j \cdot x$. As a function of x, g is a threshold of parities and hence lies in NC^1. By Barrington's theorem [Bar89], g is computed by a width-5 permutation branching program of length $\mathrm{poly}(n,k)$. Such a program is reversible: encode its S_5 state in 3 qubits, apply the length-many controlled permutations (each a constant-size Clifford+T operation controlled on one input bit), copy the accept indicator onto the target, and run the program backward to uncompute. This uses $O(1)$ ancillae and $\mathrm{poly}(n,k)$ T-gates and computes the same g as the tally, so the diamond-distance guarantee of [GKZ25] is unchanged. □

Between these corners the tally is already optimal in the natural class of *single-pass* evaluators: those that read the k sampled values once each, in a fixed order, and decide from a workspace register updated after each read. (The tally is the canonical example; so is any streaming variant.)

Proposition 25 (Optimality of the tally among single-pass evaluators)**.** *Let $\theta \in \{1,\dots,k\}$ and let $\mathrm{THR}_\theta : \{0,1\}^k \to \{0,1\}$, $\mathrm{THR}_\theta(v) = \mathbf{1}[\sum_i v_i \ge \theta]$. Any single-pass evaluator of THR_θ uses a workspace of at least $\lceil \log_2(\min\{\theta,\, k-\theta\}+1)\rceil$ qubits. For the GKZ threshold, which sits at $\theta = \Theta(k)$ for U_{MAJ_n}, this is $\Omega(\log k)$, matching the tally.*

Proof. Fix the read order and assume $\theta \le k/2$, so $\min\{\theta, k-\theta\} = \theta$ (the case $\theta > k/2$ is symmetric under $v \mapsto \bar{v}$, which swaps θ and $k-\theta$). For $s \in \{0,1,\dots,\theta\}$ let P_s be a prefix that fills the first θ read positions with exactly s ones, and let $W(s)$ be the workspace state after reading P_s. Fix $0 \le s < s' \le \theta$ and append the common suffix that places exactly $\theta - s'$ ones among the remaining $k-\theta$ positions (valid since $0 \le \theta - s' \le \theta \le k-\theta$). The two completed inputs have sums $s + (\theta - s') = \theta - (s' - s) < \theta$ and $s' + (\theta - s') = \theta$, so THR_θ outputs 0 and 1 respectively. Since the two runs share this suffix and the output is a function of the final workspace alone, $W(s) \ne W(s')$. Hence $W(0),\dots,W(\theta)$ are pairwise distinct, requiring $\lceil \log_2(\theta+1)\rceil$ qubits. □

Any single-pass implementation of the GKZ top function evaluates THR_θ on the realized values, so its workspace obeys the same bound whenever the sampled parities realize $\Omega(\min\{\theta, k-\theta\})$ distinct partial sums, which holds with high probability for the random GKZ sample. The open question is therefore precisely whether *re-reading* the parities can beat this.

> **Open problem.** Is $\lambda = \Omega(\log k)$ necessary for a mixed Clifford+T implementation of U_{MAJ_n}, or of a general threshold oracle, at the T-optimal count $T = O(k)$, when re-reading is allowed?

This is not a gap in our analysis but a genuine barrier. Proposition 25 already forces $\Omega(\log k)$ workspace for *single-pass* evaluation, so the entire question is the value of re-reading. In our model the parities are Clifford-computed and may be recomputed from x arbitrarily often at no T-cost, and Proposition 24 exploits exactly this to decide the threshold with $O(1)$ reversible state (at a polynomial T-count). The barrier is thus whether re-reading also helps at a *linear* T-count.

Consequently any resolution touches a well-known open problem. Keeping the input register read-only, an $O(1)$-ancilla implementation at the T-optimal count $T = O(k)$ is exactly a constant-width, *linear*-length branching program (with parity queries) for a threshold function: the $O(1)$ ancilla are the bounded-width state, and the $O(k)$ T-gates are the linear length. Barrington's theorem yields constant width at *polynomial* length; whether a threshold admits constant width at

linear length is open, and ruling it out would be a new superlinear length lower bound for bounded-width branching programs. A matching mixed-model lower bound is no easier: by the extremality of the unitary channel U_{MAJ_n}, a diamond-close mixed implementation forces each sampled branch to be near-reversible with its ancilla disentangled, so randomization confers no additional power and the question reduces to the same branching-program frontier. We therefore state it as an open problem rather than claim a resolution.

7 Application to Tan's General Unitary Synthesis

We apply the running-tally gadget to the general unitary synthesis framework of Tan [Tan25], showing that the ancilla savings extend beyond the single-gate setting to structured families of unitaries.

7.1 Tan's Framework

For block size κ and $N = n - \kappa$, Tan writes $U = \sum_{x\in\{0,1\}^N} |x\rangle\langle x| \otimes V_x$ and synthesizes $M = 2^N$ *delegation gadgets*, one per x. Each gadget computes a Boolean control function $f_x : \{0,1\}^N \to \{0,1\}$, conditionally applies the κ-qubit rotation V_x, and then uncomputes f_x. Tan's exact implementation uses the four-Russians method at cost $O(2^{N/2})$ T-gates and $O(2^{N/2})$ ancillae per gadget, for total T-count $O(2^{3N/2})$ and ancilla count $\Lambda_{\mathrm{Tan}} = O(2^{N/2})$.

7.2 Improved gadget via a running tally

By Theorem 6, each U_{f_x} has mixed T-count $O(\|\widehat{f_x}\|_1^2 \log(M/\varepsilon))$. Splitting the error budget evenly (ε/M per gadget, union bound) and writing $K = O(\Phi^2(N + \log(1/\varepsilon)))$ for $\Phi = \max_x \|\widehat{f_x}\|_1$, the flat GKZ implementation requires $O(K)$ ancillae per gadget. Applying Theorem 22 instead reduces this to $O(\log K)$. (Alternatively, Proposition 24 makes each gadget use $O(1)$ ancillae at a $\mathrm{poly}(N, K)$ T-count, trading T-count for workspace as in Section 6.3.)

Theorem 26. *Let $\Phi = \max_x \|\widehat{f_x}\|_1$ and $K = \Phi^2(N + \log(1/\varepsilon))$. For any $\varepsilon > 0$, the tally-augmented Tan algorithm implements a mixed channel $\mathcal{E}$ with $D_\diamond(\mathcal{E}, U) \le \varepsilon$ using*

$$R = O\big(2^N \cdot \Phi^2(N + \log(1/\varepsilon))\big) \quad T\text{-gates},$$
$$\Lambda = O(\log K) \quad \text{ancillae}.$$

Proof. Diamond distance is sub-additive under sequential composition, so the ε/M-per-gadget budget gives total error at most ε. Each gadget costs $O(K)$ T-gates (Theorem 6), giving $R = O(2^N K)$. Gadgets run sequentially in Tan's pipeline, so the per-gadget ancilla count equals the total: $\Lambda = O(\log K)$ by Theorem 22. □

Corollary 27. *If $\|\widehat{f_x}\|_1 = O(1)$ for all x, then*

$$R = O\big(2^N(N + \log(1/\varepsilon))\big), \qquad \Lambda = O\big(\log(N + \log(1/\varepsilon))\big).$$

Remark 28. Corollary 27 applies to products of $O(1)$-qubit gates, diagonal unitaries with low-degree phase polynomials, and Toff_n itself ($\Phi \le 2$ by the GKZ Fourier analysis, recovering our earlier results). In each case the ancilla count $O(\log(N + \log(1/\varepsilon)))$ is exponentially smaller than both Tan's exact count $O(2^{N/2})$ and the flat-GKZ count $O(N + \log(1/\varepsilon))$.

For a Haar-random unitary, $\|\widehat{f_x}\|_1 = \Theta(2^{N/2})$ with high probability, so $R = O(2^{2N}N)$, which is worse than Tan's $O(2^{3N/2})$. This is expected: the $\Omega(2^n)$ lower bound of [GKW24] applies to

adaptive (hence also mixed) circuits, so no randomized synthesis can reduce the worst-case T-count for generic unitaries below $\Omega(2^n)$. The tally improvement is therefore confined to the structured regime.

8 Summary

Table 1 collects the resource costs of all constructions in this paper together with the relevant prior results.

Construction	**Target**	**Ancillae λ**	**T-count**	**Model**
One ancilla, warm-up (Thm. 7)	Toff_n	1	$O(\log(1/\varepsilon))$	Mixed
GKZ [GKZ25]	Toff_n	$O(\log(1/\varepsilon))$	$O(\log(1/\varepsilon))$	Mixed
d-level tree (Thm. 15)‡	Toff_n	$O(d \cdot \log(1/\varepsilon)^{1/d})$	$O(\log(1/\varepsilon))$	Mixed
Const. fan-in (Cor. 14)‡	Toff_n	$O(\log\log(1/\varepsilon))$	$O(\log(1/\varepsilon))$	Mixed
Fourier tally (Thm. 22)	U_f	$\mathbf{O(\log k)}$	$O(\|\hat{f}\|_1^2 \log(1/\varepsilon))$	Mixed
Barrington (Prop. 24)	U_f	$\mathbf{O(1)}$	$\mathrm{poly}(n,k)$	Mixed
Tan [Tan25]	general U	$O(2^{N/2})$	$O(2^{3N/2})$	Unitary
Tan+tally, structured (Cor. 27)	$\Phi{=}O(1)$	$O(\log(N{+}\log(1/\varepsilon)))$	$O(2^N(N{+}\log(1/\varepsilon)))$	Mixed
Tan+tally, general (Thm. 26)	general U	$O(\log(\Phi^2 K))$	$O(2^N\Phi^2 K)$	Mixed
Barenco et al. [Bar+95]	Toff_n	$n-2$	$\Theta(n)$	Unitary
GKZ lower bound [GKZ25]	Toff_n	any	$\Omega(n)$	Unitary

Table 1: Resource costs. Here $N = n - \kappa$ is the number of control qubits in Tan's decomposition, $k = \Theta(\|\widehat{f}\|_1^2 \log(1/\varepsilon))$, $K = \Phi^2(N + \log(1/\varepsilon))$, and $\Phi = \max_x \|\widehat{f}_x\|_1$. ‡Subsumed by Theorem 7, which attains a single ancilla at optimal T-count for every ε; listed only to show the interpolation curve. The Fourier tally and Barrington rows are the two extreme corners of the general-oracle tradeoff (Section 6.3).

Acknowledgments

We thank Ryan O'Donnell for suggesting the workspace question and for the one-ancilla Toffoli construction of Section 5, which he communicated to us and generously allowed us to include as the warm-up here.

References

[AM19] Matthew Amy and Michele Mosca. "T-Count Optimization and Reed–Muller Codes". In: *IEEE Transactions on Information Theory* 65.8 (Aug. 2019), pp. 4771–4784. ISSN: 1557-9654. DOI: 10.1109/tit.2019.2906374. URL: http://dx.doi.org/10.1109/TIT.2019.2906374.

[Amy+13] M. Amy et al. "A Meet-in-the-Middle Algorithm for Fast Synthesis of Depth-Optimal Quantum Circuits". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 32.6 (2013), pp. 818–830. ISSN: 1937-4151. DOI: 10.1109/tcad.2013.2244643. URL: http://dx.doi.org/10.1109/TCAD.2013.2244643.

[Bar+95] Adriano Barenco et al. "Elementary gates for quantum computation". In: *Physical Review A* 52.5 (1995). arXiv:quant-ph/9503016, pp. 3457–3467. DOI: 10.1103/PhysRevA.52.3457.

[Bar89] David A. Barrington. "Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1". In: *Journal of Computer and System Sciences* 38.1 (1989), pp. 150–164.

[GKW24] David Gosset, Robin Kothari, and Kewen Wu. *Quantum state preparation with optimal T-count*. 2024. arXiv: 2411.04790 [quant-ph]. URL: https://arxiv.org/abs/2411.04790.

[GKZ25] 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.

[LKS24] Guang Hao Low, Vadym Kliuchnikov, and Luke Schaeffer. "Trading T gates for dirty qubits in state preparation and unitary synthesis". In: *Quantum* 8 (2024). arXiv:1812.00954, p. 1375. DOI: 10.22331/q-2024-06-17-1375.

[ODo21] Ryan O'Donnell. *Analysis of Boolean Functions*. 2021. arXiv: 2105.10386 [cs.DM]. URL: https://arxiv.org/abs/2105.10386.

[Sel13] Peter Selinger. "Quantum circuits of T-depth one". In: *Physical Review A* 87.4 (2013). arXiv:1210.0974, p. 042302. DOI: 10.1103/PhysRevA.87.042302.

[Tan25] Xinyu Tan. *Unitary synthesis with fewer T gates*. 2025. arXiv: 2509.25702 [quant-ph]. URL: https://arxiv.org/abs/2509.25702.