Fast Cliffords When Your Quantum Memory Is Full

Marten Folkertsma[*1], Ian Mertz[†2], Sergii Strelchuk[‡3], and Sathyawageeswar Subramanian[§3]

[1] *University of Amsterdam, QuSoft*
[2] *Charles University, Prague*
[3] *Department of Computer Science, University of Oxford, Parks Rd, Oxford OX1 3QG, United Kingdom*

Abstract

Additional qubits can reduce the depth of a quantum circuit by providing workspace for parallel computation, but standard constructions assume that this workspace is initialized in a known state. In this work we study catalytic implementations, asking whether dirty qubits can instead be used, provided that their joint state including any entanglement with other registers is restored exactly at the end of the computation.

We show that every n-qubit Clifford circuit has a catalytic implementation of depth $O(\log n)$ using $O(n^2/\log^2 n)$ dirty qubits (and no clean qubits), matching the asymptotic depth achievable when clean workspace is available. We extend this approach to diagonal elements of any fixed level C_k of the Clifford hierarchy, which admit catalytic implementations of depth $O(\log(n+1))$ with $O(n^k/\log^2(n))$ dirty qubits, as well as to semi-Clifford Gates.

Contents

1 Introduction **2**

1.1 Our results 3

1.2 Related work 4

1.3 A conjecture 5

2 Preliminaries **5**

2.1 Quantum circuits 5

2.2 Quantum catalytic circuits 6

3 Technical overview **7**

[*] m.j.folkertsma@uva.nl
[†] iwmertz@iuuk.mff.cuni.cz
[‡] Sergii.Strelchuk@cs.ox.ac.uk
[§] Sathya.Subramanian@cs.ox.ac.uk

4 Clifford circuits **7**

4.1 Background 7

4.2 Catalytic implementation of Theorem 6 9

4.3 Removing the clean output register 11

4.4 From CNOT circuits to Clifford circuits 13

4.5 Application to constructing low-T count approximate Toffoli 13

5 Generalizations **16**

5.1 Diagonal elements of the Clifford hierarchy 16

5.2 Semi-Clifford circuits 26

1 Introduction

Auxiliary qubits are a basic resource in quantum computing. By storing intermediate information, they allow different parts of a computation to proceed in parallel, and many circuit-synthesis results achieve depth reduction by trading in additional workspace for fewer sequential layers. This workspace, however, is typically assumed to be initialized in a known state. This assumption introduces an important distinction between available qubits from available temporary memory. In particular, a device may contain many idle qubits that cannot be treated as *clean* ancillas for use as workspace because they already store quantum information that must be preserved.

This motivates a different notion of workspace, namely qubits that may already contain arbitrary quantum information, but can be used during the computation provided that their state is restored exactly at the end. We call such workspace catalytic.

Computing with non-initialized memory has been investigated in several forms. In space-bounded computation, catalytic memory has developed into a broader model in which a computation may use a full auxiliary memory provided that it is restored at the end [BCK+14], with subsequent work showing that such memory can substitute for or augment other computational resources and can yield nontrivial space and time savings [AM25; CLM+25; KMP+25; CP26; CDK+26a; CDK+26b]. A quantum analogue of catalytic space was introduced in [BFM+25], and related in-place techniques have also been used in quantum circuit constructions [RV25].

In a related line of work, quantum algorithms that do not require initialization of their auxiliary registers have been studied for problems including Deutsch–Jozsa, Simon's problem, and period finding [CKL01; PSL05], and Takahashi and Tani have studied the power of uninitialized qubits in shallow quantum circuits [TT21]. These models allow auxiliary qubits to start in unknown states, but do not require them to be restored for every initial state, in contrast to the quantum catalytic space model of [BFM+25]. Thus it remains less understood whether such information-bearing auxiliary memory can replace the clean ancillas used to reduce the depth of a prescribed quantum circuit, while preserving every state of the auxiliary register.

For CNOT and Clifford circuits, Jiang et al. [JST+20] established an optimal trade-off between depth and the number of initialized auxiliary qubits, showing in particular that sufficiently many clean ancillas can be used to reduce arbitrary n-qubit Clifford circuits to logarithmic depth. Their construction therefore gives a precise benchmark for the power of clean workspace. We ask whether the same depth reduction remains possible when the initialized ancillas are replaced entirely by catalytic qubits whose initial state is unknown, arbitrary, and must be preserved.

Recent work already gives partial affirmative answers to this question. Li, Tian, He, and Sun show that any n-qubit CNOT circuit can be implemented in depth $O(\log n)$ using $O(n^2)$ dirty ancillas, via a decomposition into bipartite CNOT circuits [LTH+25]. Related recent work of Du, Cheng, and Ma develops a general catalytic compiler for bounded-fan-in, bounded-fan-out XOR computations, preserving their asymptotic depth while replacing clean intermediate storage by dirty workspace [DCM26]. Our results complement these recent developments by recovering the full depth–workspace tradeoff available with initialized ancillas: every n-qubit Clifford circuit admits a catalytic implementation of depth $O\left(\frac{n}{s \log n}\right)$ using $O(sn)$ catalytic qubits, for $1 \leq s \leq n/\log^2 n$. In particular, achieving depth $O(\log n)$ requires only $O\left(n^2/\log^2 n\right)$ catalytic qubits and no clean qubits.

Our approach is structurally different from the preceding constructions. Starting from the parallel CNOT compiler of Jiang et al, we cycle the catalytic register through a fixed family of linear transformations whose contributions cancel the dependence on its unknown initial state, and then use a factorization due to Urschel [Urs23] along with the Clifford normal form of Aaronson and Gottesman [AG04] to eliminate the clean output register altogether. This approach allows us to extend to diagonal gates in the Clifford hierarchy using the classification of Cui, Gottesman, and Krishna [CGK17] and exact phase synthesis based on Barenco et al. [BBC+95].

These results show that clean workspace can be replaced by catalytic memory for broad classes of quantum circuits, while retaining the depth advantages provided by initialized workspace. They also suggest a broader question: how far can this replacement principle be pushed for general quantum computation?

1.1 Our results

Our main theorem shows that the depth–workspace tradeoff for Clifford circuits can be achieved using catalytic workspace in place of clean auxiliary qubits.

Theorem 1. *Let $s := s(n) \leq n/\log^2 n$. For every n-qubit Clifford circuit C, there exists a catalytic Clifford circuit $\mathcal{C}_C$ of depth $O\left(\frac{n}{s \log n}\right)$ using $O(sn)$ catalytic qubits and no clean qubits which implements C.*

As an application of these catalytic circuits, we show in Section 4.5 that the Toffoli construction of [GKZ25], which uses exponentially fewer T gates at the cost of being approximate, can be made optimal in depth in the presence of catalytic qubits.

Theorem 2 (See Theorem 15)**.** *For every $\epsilon > 0$, there exists a catalytic Clifford circuit $\mathcal{C}_\epsilon$ of depth $O(\log n)$ using $O(\log(1/\epsilon))$ T gates, $O(n \log(1/\epsilon))$ catalytic qubits, and $O(\log(1/\epsilon))$ clean qubits which implements the n-qubit ϵ-approximate Toffoli gate.*

The same catalytic depth reduction extends beyond Clifford circuits to diagonal gates in the Clifford hierarchy. For a fixed level k, the phase of such a gate is described by a polynomial of degree at most k, and we exploit this bounded structure to cancel the dependence on the initial catalyst state. This gives logarithmic depth with polynomial catalytic workspace.

Theorem 3 (See Theorem 23)**.** *Let $k \geq 2$ and $s := s(n) \leq n^{k-1}/\log^2 n$. For every n-qubit diagonal unitary D in the kth level of the Clifford hierarchy, there exists a catalytic Clifford circuit $\mathcal{C}_D$ of depth $O\left(\frac{n^{k-1}}{s \log n} + \log n\right)$ using $O(sn)$ catalytic qubits and no clean qubits which implements D. In particular, there is a depth $O(\log n)$ implementation using $O(n^k/\log^2 n)$ catalytic qubits.*

We supplement this in Lemma 24 with a lower bound showing that our construction achieves the optimal depth. The construction also yields the same asymptotic bounds for semi-Clifford unitaries given a Clifford–diagonal decomposition (see Corollary 25).

When catalytic qubits are abundant, the remaining depth overhead of $O(\log n)$ comes from implementing the fan-out gate. With native access to the fan-out gate, the depth of the high-order results reduces to $O(1)$, at the cost of an additional factor $O(\log n)$ in the number of catalytic qubits. This holds only in the regime of $O(n^2/\log n)$ catalytic qubits for Clifford unitaries, and $O(n^k/\log n)$ catalytic qubits for diagonal gates in the k-th level of the Clifford hierarchy (see Remark 4.1).

1.2 Related work

Classical catalysis. Buhrman et al. introduced catalytic space as a model in which a computation may use an auxiliary memory containing arbitrary information, provided that the memory is restored at the end [BCK+14]. Subsequent work has shown that catalytic memory can increase computational power beyond ordinary low-space computation [BCK+14; AM25; CDK+26a], can replace other resources such as randomness and non-determinism [CLM+25; KMP+25], admits robust characterizations [FMS+25; KMS26], and can be used to speed up low-space computation [CP26; CDK+26b]. These techniques have been used in novel derandomization approaches to space complexity [LPT24; DPT+25] as well as breakthrough relationships between time and space [CM25; Wil25; Sha26] (for a survey on problems and techniques see [Kou16; Mer23]).

Quantum catalysis. Quantum computation has several related notions of non-clean workspace, but they impose different requirements. initialization-free quantum algorithms have been studied [CKL01; PSL05], and Takahashi and Tani investigated uninitialized qubits in shallow circuits [TT21].These works allow auxiliary qubits to begin in an unknown state, but do not formulate the requirement we need here that the entire auxiliary register be restored for every initial state. The quantum catalytic space model introduced by Buhrman et al. [BFM+25] adopts this stronger restoration requirement and develops its complexity-theoretic consequences; in particular, it gives an equivalent circuit formulation of quantum catalytic space. The usual information-theoretic notion of quantum catalysis is different again, as there the catalyst is typically a fixed state specifically chosen for the transformation, whereas our catalyst may be arbitrary and must be preserved exactly.

Dirty ancillas have long been used as temporary workspace in quantum circuit synthesis. Numerous constructions for multi-controlled gates, arithmetic circuits, and related primitives exploit auxiliary qubits whose initial state is unknown but restored before the computation finishes. Representative techniques include toggle detection and, more recently, conditionally clean ancillas, which reduce or eliminate the overhead associated with borrowing dirty qubits [Gid15; Gid17; KG25]. These constructions demonstrate that dirty workspace can successfully replace clean ancillas in specific circuit primitives, but do not study the global depth–workspace tradeoffs that we consider.

Catalysis in the quantum circuit model notably allows for in-place computations. This in-place catalysis is utilized in [RV25] to construct efficient algorithms for multi-controlled Tofolli gates and addition circuits. By contrast, classical in-place computation is a provably impossible challenge for even relatively simple functions; for example, assuming any reasonable cryptography, neither integer multiplication nor even constant-depth constant-locality circuits can be computed by an in-place machine [CGM+25].

Li et al. obtain $O(\log n)$-depth implementations of arbitrary CNOT circuits using $O(n^2)$ dirty ancillas, while Du, Cheng, and Ma give a catalytic compiler for bounded-fan-in, bounded-fan-out XOR circuits [LTH+25; DCM26]. Their constructions demonstrate that dirty workspace can support substantial depth reductions, but do not give the depth–workspace tradeoff of Theorem 1.

Our construction uses a different technique: the dependence on the initial work state is cancelled algebraically by cycling the work register through a fixed family of binary linear transformations, and a separate binary matrix factorization step to remove the clean output register. We then extend the same cancellation principle to diagonal gates in the Clifford hierarchy.

1.3 A conjecture

The results above suggest a broader question: can catalytic workspace replace clean workspace for general quantum circuits, while preserving efficient circuit size and incurring only a small increase in depth? We formulate this as the following conjecture.

Conjecture 1 (Space collapse)**.** Every n-qubit unitary U with a size s, depth d, space a implementation over a universal gate set G has a corresponding catalytic implementation of size $\mathrm{poly}(s)$, depth $d \cdot \mathrm{polylog}(s)$, and catalytic space $\mathrm{poly}(n, s, a)$.

There are several natural weakenings of this conjecture, for example by restricting the gate set or dropping the depth bound. The results of this paper establish such a statement for Clifford and semi-Clifford circuits. Determining whether a comparable collapse holds for general quantum circuits is a novel avenue for further work that appears to require new ways of using and restoring arbitrary quantum memory.

2 Preliminaries

2.1 Quantum circuits

Our base model is the usual notion of quantum circuits.

Definition 1 (Quantum circuit)**.** A *quantum circuit* is a sequence of gates from a fixed finite gate set G, with each gate acting on a bounded number of qubits. Its *size* is the number of gates, its *depth* is the number of layers, and its *width* is the total number of qubits. Gates in the same layer act on disjoint sets of qubits.

We impose no restriction on which pairs of qubits may interact. All logarithms are to base two. We write $\mathbb{F}_2$ for the field with two elements and use addition over $\mathbb{F}_2$ for bit strings.

Our results will pertain to the Clifford circuit class, which we will next define.

Definition 2 (Pauli group)**.** The single-qubit Pauli group P_1 consists of the operators I, X, Y, Z multiplied by phases from $\{1, -1, i, -i\}$. The n-qubit Pauli group P_n consists of n-fold tensor products of I, X, Y, Z with the same four possible overall phases, and has 4^{n+1} elements.

Definition 3 (Clifford circuits)**.** The Clifford group C_2 consists of the n-qubit unitaries U satisfying $UPU^\dagger \in P_n$ for every $P \in P_n$. Up to a global phase, it is generated by the gate set

$$G_c := \{H, S, \mathrm{CNOT}\},$$

where H is the Hadamard gate, $S = \mathrm{diag}(1, i)$, and CNOT is the controlled-NOT gate. A circuit over G_c is called a Clifford circuit.

The Clifford hierarchy extends this condition recursively by allowing a Pauli operator to be mapped into the preceding level under conjugation.

Definition 4 (Clifford hierarchy)**.** For a fixed number n of qubits, the Clifford hierarchy is the sequence $\{C_k\}_{k\geq 1}$ defined by

$$C_1 = P_n, \qquad C_k = \{U \in \mathrm{U}(2^n) : UPU^\dagger \in C_{k-1} \text{ for every } P \in P_n\}, \quad k \geq 2.$$

The first two levels are groups, while the higher levels need not be. We use the Clifford normal form in Section 4.1 and the diagonal normal form in Section 5.1.

2.2 Quantum catalytic circuits

We study circuits whose work qubits may start in any quantum state and must be restored after the computation. These are also called dirty qubits. A clean work qubit starts and ends in $|0\rangle$.

Definition 5 (Quantum catalytic circuits)**.** A *quantum catalytic circuit* C for a unitary U acts on an input register I and a work register W and satisfies

$$C\big(|\psi\rangle_I |\tau\rangle_W\big) = (U \otimes I_W)\big(|\psi\rangle_I |\tau\rangle_W\big)$$

for every input state $|\psi\rangle_I$ and every work state $|\tau\rangle_W$, where I_W is the identity on W. We refer to the (dirty) qubits in the work register as *catalytic qubits*.

Note that the extension of the quantum Turing Machine definition to the catalytic regime is equivalent to Definition 5 [BFM+25].

We sometimes identify circuits up to one global phase. In that case, the catalytic condition is

$$C = e^{i\theta}(U \otimes I_W),$$

where θ is independent of both the input and the work state. The following observation allows us to verify this condition on computational basis states.

Lemma 4 (Basis state criterion)**.** *Suppose that a circuit C satisfies*

$$C\,|x\rangle_I\,|z\rangle_W = e^{i\theta}(U\,|x\rangle_I)\,|z\rangle_W$$

for every computational basis state $|x\rangle_I\,|z\rangle_W$, with the same phase $e^{i\theta}$ for all x, z. Then

$$C = e^{i\theta}(U \otimes I_W).$$

In particular, the circuit preserves every state of the work register, including its entanglement with other registers.

Proof. The computational basis states span the joint input and workspace, so the operator identity follows by linearity. Tensoring this identity with the identity on any external register gives the final assertion. □

Returning each work basis state up to a phase is not sufficient. For example, Z preserves each computational basis state up to a phase but changes $|+\rangle$ to $|-\rangle$. The phase must be the same for every work basis state.

When a circuit is applied conditionally, its overall phase becomes a relative phase. Throughout the constructions that follow, we specify whether the circuit equals $U \otimes I_W$ exactly or only up to one global phase.

3 Technical overview

A CNOT circuit acts as an invertible binary linear map on computational basis states. For an $n \times n$ binary matrix M and n-bit strings x, y, the associated addition

$$|x\rangle\,|y\rangle \longmapsto |x\rangle\,|y + Mx\rangle$$

is reversible even when M is singular. This addition is the main step in our construction. The parallel circuit of Jiang et al. restores its work register, but the output can depend on the initial work string. We cancel this dependence by combining three copies of their circuit with changes of basis on the work register, as shown in Lemma 10.

To implement an invertible map on the input itself, we decompose its matrix into six additions of this form. Each addition uses one half of the input as its control register and the other half as its target register, so no separate clean output register is needed. The resulting CNOT construction extends to Clifford circuits through the normal form of Aaronson and Gottesman [AG04]. In particular, Lemma 13 gives depth $O(\log n)$ with $O(n^2/\log^2 n)$ catalytic qubits and no clean qubits.

For diagonal elements of the Clifford hierarchy, we must further cancel the dependence of a phase on the work state. The normal form writes the required phase as a sum of monomials of bounded degree and precision. For each monomial, we compare the phase before and after adding the input to a work block, then repeat this comparison over a cycle of binary linear maps. The accumulated phase depends only on the input. Separate work blocks allow the monomials to be implemented in parallel, with an additional logarithmic depth to distribute shared input controls. Section 5.1 gives the construction and its extension to semi-Clifford gates.

4 Clifford circuits

We now give the Clifford construction outlined above, allowing every catalytic qubit to start in an arbitrary state. We first recall the Clifford normal form and the role of clean auxiliary qubits in the Jiang construction, then show how to remove the initialisation requirements from the work and output registers. Clifford unitaries are identified up to one global phase throughout this section.

4.1 Background

The normal form of Aaronson and Gottesman reduces Clifford synthesis to a constant number of CNOT circuits separated by single-qubit gates.

Theorem 5 (Clifford normal form [AG04, Theorem 8])**.** *Every Clifford circuit over G_c can be converted into an equivalent circuit with eleven rounds:*

$$H - C - S - C - S - C - H - S - C - S - C,$$

where each H is a layer of Hadamard gates, each S is a layer of S gates, and each C is a CNOT *circuit.*

Since the single-qubit layers have constant depth, it suffices to parallelise the CNOT circuits. With clean auxiliary qubits, Jiang et al. [JST+20] obtained the following optimal relation between depth and workspace.

Theorem 6 ([JST+20, Theorem 4.1])**.** *Let $1 \le s \le O(n/\log^2 n)$. Every n-qubit CNOT circuit can be transformed into an equivalent CNOT circuit of depth*

$$O\left(\frac{n}{s \log n}\right)$$

that uses $(3s+1)n$ ancillas.

They also provide a matching lower bound to proof that their result is optimal:

Theorem 7 ([JST+20, Theorem 1.3], with $m = sn$)**.** *Let $0 \le \epsilon < \sqrt{2}/2$ be a constant and let $0 < s < n/\log^2 n$. For a $1 - o(1)$ fraction of n-qubit CNOT circuits, every ϵ-approximate n-qubit quantum circuit with sn ancillas has depth*

$$\Omega\left(\frac{n}{(1+s)\log n}\right).$$

Applying this result to the constant number of CNOT circuits in Theorem 5 gives the same bound for Clifford circuits.

Corollary 8 (of [JST+20, Theorem 4.1 and Corollary 1.2])**.** *Let $1 \le s \le O(n/\log^2 n)$. Every n-qubit Clifford circuit can be implemented, without measurements, by a circuit of depth*

$$O\left(\frac{n}{s \log n}\right)$$

that uses $(3s+1)n$ ancillas initialized to $|0\rangle$ and returns them to $|0\rangle$. This is optimal in the worst case, since CNOT circuits are Clifford circuits and Corollary 7 applies to them.

Our construction achieves logarithmic depth using $O(n^2/\log^2 n)$ catalytic qubits, showing that this degree of parallelization in fact does not require clean workspace. To explain the adaptation, we first describe CNOT circuits as binary linear maps.

For bits x, y, a CNOT gate acts as

$$\mathrm{CNOT}\,|x\rangle\,|y\rangle = |x\rangle\,|y+x\rangle\,,$$

which is multiplication of the column vector $(x, y)^{\mathsf{T}}$ by

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

We write $\mathrm{GL}(n, 2)$ for the group of invertible $n \times n$ binary matrices. Every n-qubit CNOT circuit defines a matrix in this group, and every such matrix can be reduced to the identity by row elimination. Since a row addition is a CNOT gate and a row swap is a product of three row additions, each $M \in \mathrm{GL}(n, 2)$ has a CNOT circuit $\mathcal{C}$ satisfying

$$\mathcal{C}\,|x\rangle = |Mx\rangle$$

for every $x \in \mathbb{F}_2^n$ [PMH08]. Matrix and circuit products act from right to left.

The usual implementation with a clean output register uses reversible addition. Let I and O be n-qubit input and output registers, and let $\mathcal{C}_1$ implement the block matrix

$$\begin{bmatrix} I & 0 \\ M & I \end{bmatrix},$$

so that

$$\mathcal{C}_1 \left|x\right\rangle_I \left|y\right\rangle_O = \left|x\right\rangle_I \left|y + Mx\right\rangle_O ,$$

where the identity blocks act on the corresponding registers. A second circuit $\mathcal{C}_2$ implements

$$\begin{bmatrix} I & 0 \\ M^{-1} & I \end{bmatrix}.$$

Starting with O in $|0\rangle$, the sequence of $\mathcal{C}_1$, a swap of the two registers, and $\mathcal{C}_2$ computes Mx in I and restores O to $|0\rangle$:

$$\begin{aligned} \mathcal{C}_1 \left|x\right\rangle_I \left|0\right\rangle_O &= \left|x\right\rangle_I \left|Mx\right\rangle_O , \\ \mathrm{SWAP} \left|x\right\rangle_I \left|Mx\right\rangle_O &= \left|Mx\right\rangle_I \left|x\right\rangle_O , \\ \mathcal{C}_2 \left|Mx\right\rangle_I \left|x\right\rangle_O &= \left|Mx\right\rangle_I \left|x + M^{-1}Mx\right\rangle_O \\ &= \left|Mx\right\rangle_I \left|0\right\rangle_O . \end{aligned}$$

Additional work qubits allow parts of these additions to run in parallel. Our next lemma describes the resulting action when the work register is not assumed to start in $|0\rangle$.

4.2 Catalytic implementation of Theorem 6

Lemma 4.2 of Jiang et al. [JST+20] gives the required parallel addition circuit. Although stated for an invertible off-diagonal block, its proof also applies to arbitrary binary matrices. We use the numbering in the full version of their paper.

Lemma 9 (Adapted from Lemma 4.2 of [JST+20])**.** *Let $n \geq 2$ and let s be an integer with $1 \leq s \leq n/\log^2 n$. For every $M \in \mathbb{F}_2^{n\times n}$, there exists a* CNOT *circuit $\mathcal{C}_M$ of depth*

$$O\left(\frac{n}{s \log n}\right)$$

using a work register W of $m = O(sn)$ qubits. Its input and output registers I, O each contain n qubits, and for some $B \in \mathbb{F}_2^{n\times m}$ it implements the matrix

$$C_M = \begin{bmatrix} I_I & 0 & 0 \\ M & I_O & B \\ 0 & 0 & I_W \end{bmatrix}.$$

Equivalently,

$$\mathcal{C}_M \left|x\right\rangle_I \left|y\right\rangle_O \left|z\right\rangle_W = \left|x\right\rangle_I \left|y + Mx + Bz\right\rangle_O \left|z\right\rangle_W$$

for $x, y \in \mathbb{F}_2^n$ and $z \in \mathbb{F}_2^m$. Subscripts on identity blocks indicate their registers.

Proof. Although Jiang et al. state the lemma for invertible M, their proof constructs M one block at a time by partitioning it into groups of $s \log^2 n$ columns and applying their Corollary 4.5 to each group. That corollary allows an arbitrary binary block, so the construction and its resource bounds apply unchanged to any $M \in \mathbb{F}_2^{n\times n}$. Jiang et al. do not specify the blocks acting on W, since their workspace starts in $|0\rangle$. Their circuit uses I only as a control, so the (I, W) block is 0. It restores W by applying the inverse of the circuit that computed into it, so the (W, W) block is I_W. □

Remark 4.1. The construction of Jiang et al. [JST+20] can be improved in depth if one has access to $O(n^2/\log(n))$ auxiliary qubits and the fan-out gate (one control, many NOT targets) at unit cost. Then the remaining $O(\log(n))$ depth reduces to depth $O(1)$. The implementation then consists of copying the data x to many registers, or of computing the parity gate, and each of these happens only a constant number of times. Utilizing fan-out and parity natively comes at the cost of an additional multiplicative factor $O(\log(n))$ in the number of auxiliary qubits. The improvement in depth therefore only holds when one has access to $O(n^2/\log(n))$ auxiliary qubits. All our constructions incur depth scaling in n either through the Jiang et al. construction or through fan-out used to copy information. Therefore, this improvement in depth also holds for our constructions. This might be especially interesting for hardware implementations with native multi-target interactions. For example, the proposed implementations of fan-out using collective interactions in trapped ions [MN18], cavity systems [YLN10], and Rydberg atoms [MLW+09]. These implementations use a number of collective operations independent of the number of targets.

With both the output and work registers initially zero, this circuit computes Mx in the output register:

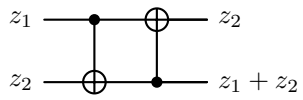
$$\mathcal{C}_M \left|x\right\rangle_I \left|0\right\rangle_O \left|0\right\rangle_W = \left|x\right\rangle_I \left|Mx\right\rangle_O \left|0\right\rangle_W .$$

For an arbitrary work string z, the output instead contains the additional term Bz. We eliminate this term by cycling the work register through three binary linear maps whose sum is zero.

The construction uses

$$P := \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}.$$

This matrix maps $(z_1, z_2)^\mathsf{T}$ to $(z_2, z_1 + z_2)^\mathsf{T}$ and has a two-CNOT implementation:



Its powers satisfy

$$P^3 = I \qquad \text{and} \qquad I + P + P^2 = 0$$

over $\mathbb{F}_2$. Applying these maps to pairs of work qubits lets us cancel Bz while restoring the entire work register.

Lemma 10 (Catalytic binary addition)**.** *Let $n \geq 2$ and let s be an integer with $1 \leq s \leq n/\log^2 n$. For every $M \in \mathbb{F}_2^{n\times n}$, there exists a* CNOT *circuit $\widetilde{\mathcal{C}}_M$ of depth*

$$O\left(\frac{n}{s \log n}\right)$$

using $O(sn)$ catalytic work qubits that implements

$$\widetilde{C}_M = \begin{bmatrix} I_I & 0 & 0 \\ M & I_O & 0 \\ 0 & 0 & I_W \end{bmatrix}.$$

Equivalently, for every $x, y \in \mathbb{F}_2^n$ and every work state $\left|\tau\right\rangle_W$,

$$\widetilde{\mathcal{C}}_M \left|x\right\rangle_I \left|y\right\rangle_O \left|\tau\right\rangle_W = \left|x\right\rangle_I \left|y + Mx\right\rangle_O \left|\tau\right\rangle_W .$$

Proof. Let m be the number of work qubits in the circuit of Lemma 9. We first treat even m and combine three conjugated copies of its matrix C_M. Define

$$\mathbf{P} := \underbrace{\begin{pmatrix} P & 0 & \cdots & 0 \\ 0 & P & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & P \end{pmatrix}}_{m/2 \text{ copies}} \qquad \text{and} \qquad D := \begin{pmatrix} I_I & 0 & 0 \\ 0 & I_O & 0 \\ 0 & 0 & \mathbf{P} \end{pmatrix}.$$

Thus, $\mathbf{P}$ applies P to each consecutive pair of work qubits. The matrix D can be implemented in depth two using two CNOT gates per pair.

For $j \in \{0, 1, 2\}$, define

$$C_{M,j} := D^{-j} C_M D^j = \begin{bmatrix} I & 0 & 0 \\ M & I & B\mathbf{P}^j \\ 0 & 0 & I \end{bmatrix}.$$

Because $P^3 = I$, we have $P^{-1} = P^2$. Each $C_{M,j}$ therefore consists of one application of $\mathcal{C}_M$ and constant-depth CNOT circuits on W before and after it. Finally,

$$\begin{aligned} C_{M,0} C_{M,1} C_{M,2} &= \begin{bmatrix} I & 0 & 0 \\ M + M + M & I & B + B\mathbf{P} + B\mathbf{P}^2 \\ 0 & 0 & I \end{bmatrix} \\ &= \begin{bmatrix} I_I & 0 & 0 \\ M & I_O & 0 \\ 0 & 0 & I_W \end{bmatrix} = \widetilde{C}_M. \end{aligned}$$

Here we used $M + M + M = M$ and $I + \mathbf{P} + \mathbf{P}^2 = 0$ over $\mathbb{F}_2$.

If m is odd, append one additional catalytic qubit and extend B by one zero column. The enlarged work register has even size, and the same argument applies. The additional qubit does not change the asymptotic space bound. □

The matrix identity holds for every computational basis state of the work register. By linearity, it holds for every quantum state, including states entangled with an external register. In particular, the circuit acts exactly as the identity on the catalytic register.

4.3 Removing the clean output register

The preceding lemma implements addition into an arbitrary output register. To use it for a transformation of the input itself, we apply the following special case of Urschel's factorisation [Urs23].

Theorem 11 (Urschel [Urs23, Theorem 1])**.** *For every $M \in \mathrm{GL}(2n, 2)$, there exist binary matrices $A_1, \ldots, A_6 \in \mathbb{F}_2^{n \times n}$ such that*

$$M = \begin{pmatrix} I & 0 \\ A_1 & I \end{pmatrix} \begin{pmatrix} I & A_2 \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ A_3 & I \end{pmatrix} \begin{pmatrix} I & A_4 \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ A_5 & I \end{pmatrix} \begin{pmatrix} I & A_6 \\ 0 & I \end{pmatrix}.$$

Urschel proves the factorisation for matrices of determinant one over an arbitrary field. Since every invertible binary matrix has determinant one, it applies in the form above. Each factor is an addition from one half of the input to the other, which we implement using Lemma 10.

Theorem 12. *Let $n \geq 2$ and let s be an integer with $1 \leq s \leq n/\log^2 n$. For every $M \in \mathrm{GL}(n,2)$, there exists a catalytic* CNOT *circuit $\mathcal{C}_c$ of depth*

$$O\left(\frac{n}{s \log n}\right)$$

using $O(sn)$ catalytic qubits. For every $x \in \mathbb{F}_2^n$ and every work state $|\tau\rangle_W$,

$$\mathcal{C}_c\big(|x\rangle_I |\tau\rangle_W\big) = |Mx\rangle_I |\tau\rangle_W .$$

Proof. We construct the circuit on computational basis states and then use linearity. It suffices to treat sufficiently large n, since bounded values can be handled by Gaussian elimination without additional qubits. We may also take $n = 2r$ to be even: for odd n, append one catalytic qubit and apply the even-dimensional construction to $\begin{pmatrix} M & 0 \\ 0 & 1 \end{pmatrix}$. The extended map leaves this qubit unchanged and has the same asymptotic resource bounds. By Theorem 11, there exist $A_1, \ldots, A_6 \in \mathbb{F}_2^{r \times r}$ such that

$$M = M_1 M_2 M_3 M_4 M_5 M_6,$$

where

$$M_i = \begin{cases} \begin{pmatrix} I & 0 \\ A_i & I \end{pmatrix}, & i \text{ odd}, \\ \begin{pmatrix} I & A_i \\ 0 & I \end{pmatrix}, & i \text{ even}. \end{cases}$$

Split the input into two r-qubit halves and use the first half as the control register for odd i, reversing the roles for even i. To apply Lemma 10 in dimension r, choose $s' := \min\{s, \lfloor r/\log^2 r \rfloor\}$. For sufficiently large n, $s' \geq 1$ and $s' = \Theta(s)$, so this choice preserves both asymptotic bounds. The lemma gives a circuit $\mathcal{C}_i$ for each M_i. For odd i, call the two halves I and O and write their basis labels as x_{in} and x_{out}. The circuit acts as

$$\mathcal{C}_i |x_{\text{in}}\rangle_I |x_{\text{out}}\rangle_O |z\rangle_W = |x_{\text{in}}\rangle_I |x_{\text{out}} + A_i x_{\text{in}}\rangle_O |z\rangle_W .$$

The analogous identity holds for even i with the two data registers interchanged.

The circuit for M is their product

$$\mathcal{C}_c := \mathcal{C}_1 \mathcal{C}_2 \mathcal{C}_3 \mathcal{C}_4 \mathcal{C}_5 \mathcal{C}_6,$$

with the rightmost factor acting first. For every computational basis state $|x\rangle$ and work basis state $|z\rangle$, it satisfies

$$\mathcal{C}_c |x\rangle_I |z\rangle_W = |Mx\rangle_I |z\rangle_W .$$

No phase depends on x or z, since $\mathcal{C}_c$ consists only of CNOT gates. Lemma 4 extends the identity to arbitrary states of the input and work registers.

Each $\mathcal{C}_i$ uses $O(s'r) = O(sn)$ catalytic qubits and has depth $O(r/(s' \log r)) = O(n/(s \log n))$. Since every factor restores the work register, all six circuits can reuse it, giving total workspace $O(sn)$ and depth $O(n/(s \log n))$. □

This result is in fact optimal. The lower bound given in Theorem 7 by Jiang et. al. [JST+20] holds for clean qubits, therefore in particular, it also holds for catalytic qubits.

4.4 From CNOT circuits to Clifford circuits

The Clifford normal form contains only a constant number of CNOT circuits. Applying Theorem 12 to each of them therefore gives the same asymptotic depth and workspace for Clifford circuits, which will be used to prove Theorem 1.

Lemma 13. *Let $n \geq 2$ and let s be an integer with $1 \leq s \leq n/\log^2 n$. For every n-qubit Clifford circuit $\mathcal{C}$, there exists a Clifford circuit $\mathcal{C}_c$ of depth*

$$O\left(\frac{n}{s \log n}\right)$$

using $O(sn)$ catalytic qubits, such that, for every input state $|\psi\rangle$ and every work state $|\tau\rangle_W$,

$$\mathcal{C}_c\big(|\psi\rangle_I |\tau\rangle_W\big) = (\mathcal{C} \otimes I_W)\big(|\psi\rangle_I |\tau\rangle_W\big),$$

up to the global phase convention for Clifford circuits.

Proof. Use the eleven rounds of Theorem 5 and replace each CNOT circuit by its catalytic implementation from Theorem 12. These circuits restore the work register after every round, so they may all use the same catalytic qubits. The intervening single-qubit gates add only constant depth, which gives the claimed bounds. □

4.5 Application to constructing low-T count approximate Toffoli

An important subroutine in many quantum algorithms is the multi-controlled Toffoli gate, which computes AND_n of n control qubits into a target qubit. For instance, up to Clifford gates, this gate forms the diffusion operator used in Grover's algorithm. It has been shown that an exact implementation of this gate requires at least n T gates, or magic states, even when measurements and classical feed-forward are allowed [BCH+20].

Gosset et al. [GKZ25, Theorem 1] showed that this T-count can be exponentially improved when allowing some error on the implementation of the Toffoli gate: $O(\log(1/\epsilon))$ T gates suffice for every n and $\epsilon \in (0, 1/2]$. This result holds specifically in the mixed model of unitary implementation. Mixed here means that they generate an ensemble of unitary circuits, of which one is picked at random and then implemented. For every computational basis input, the chosen circuit acts correctly with probability at least $1 - \epsilon$, and the averaged channel is ϵ-close to the Toffoli gate in diamond norm [GKZ25, Definition 10]. The T-count bound holds for all circuits in the ensemble.

In this section we show that their construction can be improved in depth when given access to catalytic qubits, and be made optimal when enough are available. This could be of use in general compilers and reduce the overall overhead in quantum circuits. The depth reduction scales with the number of catalytic qubits available. The direct implementation of the construction of [GKZ25] has depth $O(\log(n)\log(1/\epsilon))$. An algorithm that uses this gate $\mathrm{poly}(n)$ times needs per-gate error $\epsilon = 1/\mathrm{poly}(n)$ for a constant total error, by the union bound. The depth per gate then becomes $O(\log^2(n))$. We reduce this to $O\big(\log(n)\frac{\log(1/\epsilon)}{s} + \log\log(1/\epsilon)\big)$ when having access to sn catalytic qubits. This reduces to $O(\log(n))$ depth when $s = \Theta(\log(n))$, which is optimal for the n-qubit Toffoli gate among circuits of two-qubit gates.

Lemma 14 (Random-parity OR, based on [GKZ25, Algorithm 1 and Theorem 12])**.** *For any positive integer n and $\epsilon < 1/2$, there exists an ensemble of circuits $\{C_j\}_j$ of depth $O(\log(n)\log(1/\epsilon))$ using*

$O(\log(1/\epsilon))$ T gates and $O(\log(1/\epsilon))$ clean work space, such that every C_j returns W to $|0\rangle$ and, for every $x \in \{0,1\}^n$ and $b \in \{0,1\}$, with probability at least $1-\epsilon$ over a uniformly random j,

$$C_j \left|x\right\rangle_I \left|b\right\rangle_O \left|0\right\rangle_W = \left|x\right\rangle_I \left|b \oplus \mathsf{OR}_n(x)\right\rangle_O \left|0\right\rangle_W .$$

Moreover, the channel

$$\mathcal{E}(\rho) = \mathbb{E}_j \left[C_j \left(\rho \otimes |0\rangle\langle 0|_W\right) C_j^\dagger \right]$$

equals $\mathcal{E}'(\rho) \otimes |0\rangle\langle 0|_W$ with $D_\diamond\big(\mathcal{E}', U_{\mathsf{OR}_n}\big) \le \epsilon$.

Proof. For simplicity, start by setting $k = \lceil \log(1/\epsilon) \rceil + 2$, so that $2^{-k} \le \epsilon/4$; the extra 2 will be needed only for the diamond-distance bound at the end. Instead of implementing an exact OR_n gate, the circuit implements k Parity_S gates on uniformly random subsets S. Start by picking a collection of subsets $\{S_i\}_{i=1}^k$, each $S_i \subseteq [n]$ chosen uniformly at random and independently. If $\mathsf{OR}_n(x) = 0$, then $\mathsf{Parity}_{S_i}(x) = 0$ for every $i \in [k]$, but if $\mathsf{OR}_n(x) = 1$, then for every i, $\mathsf{Parity}_{S_i}(x) = 1$ with probability $1/2$, independently. Let C'_j be the circuit that sequentially applies Parity_{S_i} to x, with a separate output qubit in W for every i:

$$C'_j \left|x\right\rangle_I \left|b\right\rangle_O \left|0\right\rangle_W = \left|x\right\rangle_I \left|b\right\rangle_O \bigotimes_{i=1}^{k} \left|\mathsf{Parity}_{S_i}(x)\right\rangle_W .$$

Let C_j now be the circuit that, after C'_j, applies the exact OR_k gate with the work register as input and the output register as output. Then apply $C_j'^\dagger$ to clean the work register. Every C_j therefore acts as $C_j \left|x\right\rangle_I \left|b\right\rangle_O \left|0\right\rangle_W = \left|x\right\rangle_I \left|b \oplus g_j(x)\right\rangle_O \left|0\right\rangle_W$ with $g_j = \mathsf{OR}_k(\mathsf{Parity}_{S_1}, \ldots, \mathsf{Parity}_{S_k})$. If $x = 0$, thus $\mathsf{OR}_n(x) = 0$, then $g_j(x) = 0$ for every j, as required. If $\mathsf{OR}_n(x) = 1$, then the probability that all $\mathsf{Parity}_{S_i}(x) = 0$ is $\left(\frac{1}{2}\right)^k \le \epsilon/4$. Therefore, with probability at least $1-\epsilon$, one of the $\mathsf{Parity}_{S_i}(x) = 1$ and $g_j(x) = 1 = \mathsf{OR}_n(x)$.

C_j is the circuit for one choice of the collection $\{S_i\}$. Every j corresponds to another collection $\{S_i\}$, and the same analysis holds for all of them. We can count the complexity of C_j. There are k Parity_{S_i} gates, each a CNOT tree of depth $O(\log n)$, and one exact OR_k gate, a tree of Toffoli gates of depth $O(\log k)$ with $O(k)$ T gates. Remeber that $k = \lceil \log(1/\epsilon) \rceil + 2$, therefore, the depth is $O(\log(n)\log(1/\epsilon) + \log\log(1/\epsilon)) = O(\log(n)\log(1/\epsilon))$, it uses $O(\log(1/\epsilon))$ T gates, and it uses $O(\log(1/\epsilon))$ additional clean qubits.

For the diamond distance, let $U_j \left|x\right\rangle \left|b\right\rangle = \left|x\right\rangle \left|b \oplus g_j(x)\right\rangle$. By linearity, $C_j(|\psi\rangle \otimes |0\rangle_W) = (U_j |\psi\rangle) \otimes |0\rangle_W$ for every state $|\psi\rangle$ on I, O and a reference system, so $\mathcal{E}$ factors with $\mathcal{E}'(\rho) = \mathbb{E}_j[U_j \rho U_j^\dagger]$. This is the channel of [GKZ25, Algorithm 1] up to fixed X gates, since $U_{\mathsf{OR}_n} = (X^{\otimes n} \otimes I)\,\mathrm{Toff}_{n+1}\, X^{\otimes n+1}$ and the same gates relate U_j to their W_g. Fixed unitaries before and after preserve the diamond distance, so [GKZ25, Theorem 12] gives $D_\diamond(\mathcal{E}', U_{\mathsf{OR}_n}) \le 4 \cdot 2^{-k} \le \epsilon$. □

The depth of this ensemble of circuits is dominated by the k Parity_{S_i} gates. The OR_k gate adds only depth $O(\log k) = O(\log\log(1/\epsilon))$, so

$$\mathrm{depth}(C_j) = 2\,\mathrm{depth}(C'_j) + O(\log\log(1/\epsilon)).$$

The circuit C'_j is a CNOT circuit: it computes the linear map $\left|x\right\rangle_I \left|y\right\rangle_W \mapsto \left|x\right\rangle_I \left|y \oplus Mx\right\rangle_W$, where the rows of $M \in \mathbb{F}_2^{k \times n}$ are the indicator vectors of $S_1, \ldots, S_k$. Linear maps of this form are exactly what the catalytic compiler parallelizes. We use it to give a depth-improved implementation of the approximate OR_n gate.

Theorem 15. *For any integer $n \geq 2$, any ϵ with $2^{-n} \leq \epsilon < 1/2$ and any integer s with $1 \leq s \leq \lceil \log(1/\epsilon) \rceil$, there exists an ensemble of catalytic circuits $\{C_j\}_j$ of depth $O\big(\log(n)\frac{\log(1/\epsilon)}{s} + \log\log(1/\epsilon)\big)$ using sn catalytic qubits, $O(\log(1/\epsilon))$ T gates and $O(\log(1/\epsilon))$ clean qubits, such that the channel*

$$\mathcal{E}(\rho \otimes |0\rangle\langle 0|_{W_c} \otimes |\tau\rangle\langle\tau|_{W_{cat}}) = \mathbb{E}_j\big[C_j\,(\rho \otimes |0\rangle\langle 0|_{W_c} \otimes |\tau\rangle\langle\tau|_{W_{cat}})\,C_j^\dagger\big]$$

returns the registers W_c and W_{cat} exactly for every catalyst state $|\tau\rangle$,

$$\mathcal{E}(\rho \otimes |0\rangle\langle 0|_{W_c} \otimes \tau_{W_{cat}}) = \mathcal{E}'(\rho) \otimes |0\rangle\langle 0|_{W_c} \otimes \tau_{W_{cat}},$$

and is close to the OR_n gate in diamond norm, $D_\diamond\big(\mathcal{E}', U_{\mathsf{OR}_n}\big) \leq \epsilon$.

Proof. In principle, we use the same construction as in Lemma 14, with the adjustment that the circuit C_j' consisting of the Parity_{S_i} gates is implemented using the catalytic compiler. We prove this first for the catalyst in a computational basis state $|z\rangle$ and then invoke Lemma 4 to prove it for general τ. Let $k = \lceil \log(1/\epsilon) \rceil + 2$, and split the catalyst into s rows $W_{cat} = Z_1 \cdots Z_s$ of n qubits each, holding $z_1, \ldots, z_s$. Batch the k parity gates into $\lceil k/s \rceil$ batches of at most s gates. For a batch with subsets $S_1, \ldots, S_s$ (after relabelling) and target qubits $a_1, \ldots, a_s$ in W_c, perform the following circuit. Use fanout from Lemma 19 to copy x into the catalyst:

$$F\,|x\rangle \bigotimes_{l=1}^{s} |0\rangle_{a_l}\,|z_l\rangle_{Z_l} \longmapsto |x\rangle \bigotimes_{l=1}^{s} |0\rangle_{a_l}\,|z_l \oplus x\rangle_{Z_l}\,.$$

Then, apply the batch of Parity_{S_l} gates in parallel, the l-th one reading row Z_l and writing into a_l:

$$\bigotimes_{l=1}^{s} U_{\mathsf{Parity}_{S_l}}\ |x\rangle \bigotimes_{l=1}^{s} |0\rangle_{a_l}\,|z_l \oplus x\rangle_{Z_l} \longmapsto |x\rangle \bigotimes_{l=1}^{s} \big|\mathsf{Parity}_{S_l}(x \oplus z_l)\big\rangle_{a_l}\,|z_l \oplus x\rangle_{Z_l}\,.$$

Each $U_{\mathsf{Parity}_{S_l}}$ is an in-place CNOT tree on the qubits of Z_l in S_l, one CNOT into a_l, and the inverse tree. It has depth $O(\log n)$ and leaves Z_l unchanged, and the rows are disjoint, so the batch runs in parallel. Remove x from the catalytic register using fanout again:

$$F\,|x\rangle \bigotimes_{l=1}^{s} \big|\mathsf{Parity}_{S_l}(x \oplus z_l)\big\rangle_{a_l}\,|z_l \oplus x\rangle_{Z_l} \longmapsto |x\rangle \bigotimes_{l=1}^{s} \big|\mathsf{Parity}_{S_l}(x \oplus z_l)\big\rangle_{a_l}\,|z_l\rangle_{Z_l}\,.$$

Finally, apply the batch of Parity_{S_l} gates one more time:

$$\bigotimes_{l=1}^{s} U_{\mathsf{Parity}_{S_l}}\ |x\rangle \bigotimes_{l=1}^{s} \big|\mathsf{Parity}_{S_l}(x \oplus z_l)\big\rangle_{a_l}\,|z_l\rangle_{Z_l} \longmapsto |x\rangle \bigotimes_{l=1}^{s} \big|\mathsf{Parity}_{S_l}(x)\big\rangle_{a_l}\,|z_l\rangle_{Z_l}\,,$$

where we use the fact that $\mathsf{Parity}_S(x \oplus z) \oplus \mathsf{Parity}_S(z) = \mathsf{Parity}_S(x)$. Repeat this for every batch, reusing the same rows Z_l; this implements C_j' and returns the catalyst to $|z\rangle$. The rest of the circuit is the same as in Lemma 14.

Because the catalytic implementation of C_j' has exactly the same action as the sequential one on every basis state, and fixes z, each C_j acts as the circuit of Lemma 14 on I, O and W_c, and as the identity on W_{cat}. By Lemma 4 this holds for every catalyst state τ, also when W_{cat} is entangled with a reference system. Hence $\mathcal{E}$ factors as stated, and the diamond-norm bound of Lemma 14 carries over, with W_{cat} taken as part of the reference system.

Accounting for complexity: one batch has depth $O(\log n + \log s)$, from two fanouts of depth $O(\log s)$ and two parity layers of depth $O(\log n)$. Since $\epsilon \geq 2^{-n}$, we have $s \leq \log(1/\epsilon) \leq n$, so $\log s \leq \log n$. There are $\lceil k/s \rceil \leq 2k/s = O\big(\frac{\log(1/\epsilon)}{s}\big)$ batches, so the catalytic C_j' has depth $O\big(\log(n)\frac{\log(1/\epsilon)}{s}\big)$ and uses sn catalytic qubits. The OR_k gate has depth $O(\log k) = O(1 + \log\log(1/\epsilon))$ and uses $O(\log(1/\epsilon))$ T gates. The final depth is therefore $O\big(\log(n)\frac{\log(1/\epsilon)}{s} + \log\log(1/\epsilon)\big)$, where the constant is absorbed by the first term. □

5 Generalizations

5.1 Diagonal elements of the Clifford hierarchy

The cancellation used for binary addition has an analogue for diagonal phases. We develop it for n-qubit diagonal unitaries in a fixed level C_k, with $k \geq 2$, where the level bounds both the degree and the precision of the phase polynomial. As before, W denotes the catalytic work register and I_W its identity operator.

For $x \in \{0,1\}^n$ and a nonempty set $S \subseteq \{1,\ldots,n\}$, write $x_S := \prod_{i \in S} x_i$.

Theorem 16 (Diagonal normal form [CGK17, Theorem 3])**.** *Let S range over the nonempty subsets of $\{1,\ldots,n\}$ with $|S| \leq k$, and choose integers $0 \leq a_S < 2^{k-|S|+1}$. Up to a global phase, the n-qubit diagonal elements of C_k are precisely the unitaries defined by*

$$p(x) = \sum_S 2^{|S|-1} a_S x_S, \qquad D\,|x\rangle = e^{2\pi i p(x)/2^k}\,|x\rangle\,. \tag{1}$$

A constant term in p contributes only an overall phase and is omitted. To see that the remaining coefficients are unique, evaluate $p(x)$ modulo 2^k on strings whose entries are 1 exactly at the positions in S, considering these sets in increasing order of size. Once the coefficients for proper subsets of S are known, subtracting their contributions leaves $2^{|S|-1}a_S$ modulo 2^k. This determines a_S uniquely in the specified range.

For $k = 2$, the normal form expresses D as a product of powers of S and CZ gates. At $k = 3$, the corresponding gates are T, controlled-S, and CCZ, where $T = \mathrm{diag}(1, e^{i\pi/4})$. More generally, for a monomial of degree $d = |S|$ and integers a and $r \geq 1$, the phase $\exp(2\pi i a x_S/2^r)$ defines a diagonal gate in C_k whenever $r + d - 1 \leq k$. The hierarchy level thus bounds both the degree of each monomial and the denominator needed to express its phase.

The gate set consists of $G_c = \{H, S, \mathrm{CNOT}\}$ together with powers of

$$R_k := \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix}.$$

For this fixed k, write

$$G := G_c \cup \{R_k^a : a = 0,\ldots,2^k - 1\}.$$

All circuits in this subsection use the finite gate set G, with each power of R_k counted as a single gate. As in the Clifford construction, two-qubit gates may act on any pair of qubits, and gates in the same layer act on disjoint qubits. We add bit strings over $\mathbb{F}_2$ and compare integer phase numerators modulo 2^r when the denominator is 2^r.

5.1.1 Implementing the normal form I: one monomial

For a single monomial, the controlled-unitary construction of Barenco et al. [BBC$^+$95, Lemma 7.5 and Corollary 7.6] gives a circuit whose size is quadratic in the monomial degree. We give the construction explicitly to verify that every required rotation belongs to G.

Lemma 17 (Local phase synthesis)**.** *Let d, r be positive integers satisfying $r + d - 1 \leq k$, and let a be an integer. For bits $u_1,\ldots,u_d$, define*

$$Q\,|u_1,\ldots,u_d\rangle = \exp\left(\frac{2\pi i a}{2^r} \prod_{\ell=1}^{d} u_\ell\right) |u_1,\ldots,u_d\rangle\,.$$

Then Q has an exact circuit over G of size and depth $O(d^2)$, with no additional qubits.

Proof. Writing the phase angle as $\theta = 2\pi a/2^r$, we implement the case $d = 1$ with a single rotation. For $d = 2$, set $u = u_1$ and $v = u_2$ and use the identity

$$2uv = u + v - (u \oplus v).$$

The first two terms give phase rotations of angle $\theta/2$ on u and v. To implement the last term, compute $u \oplus v$ into v with a CNOT gate, apply a phase rotation of angle $-\theta/2$, and undo the CNOT. This gives three rotations and two CNOT gates.

The construction for $d \geq 3$ also uses Toffoli gates. In this case, $r + d - 1 \leq k$ implies $k \geq 3$, so these gates can be implemented exactly over G. To see this, for bits u, v, w, the identity

$$4uvw = u + v + w - (u \oplus v) - (u \oplus w) - (v \oplus w) + (u \oplus v \oplus w)$$

expresses the phase of CCZ as a sum of parity terms with angles $\pm\pi/4$. Each parity is computed into one of the three qubits using CNOT gates and uncomputed after the phase is applied. This requires no additional qubits, and all the rotations are powers of R_k. Applying H to the third qubit before and after CCZ gives an exact Toffoli gate.

To reduce the monomial degree from d to $d - 1$, set

$$f = u_1 \cdots u_{d-2}, \qquad u = u_{d-1}, \qquad v = u_d,$$

and use

$$2fuv = fv + uv - (u \oplus f)v. \tag{2}$$

The transformation $u \mapsto u \oplus f$ is a NOT gate controlled by the first $d-2$ qubits. It can be implemented using $O(d)$ Toffoli and CNOT gates, with v as a dirty auxiliary qubit [BBC+95, Corollary 7.4]. The circuit leaves the controls unchanged and restores v exactly. Replacing each Toffoli gate by the construction above gives a circuit over G.

Apply this operation, then the phase $\exp(-i\theta uv/2)$, and then apply the operation again. Next apply the phases $\exp(i\theta uv/2)$ and $\exp(i\theta fv/2)$. Equation (2) shows that the total phase is $\exp(i\theta fuv)$. The use of v as a dirty qubit does not affect this calculation, since each controlled operation restores v before a phase is applied and the second operation also restores u.

The two-qubit phases use the $d = 2$ construction, and the phase involving fv is implemented recursively on $d - 1$ qubits. At each step, the degree falls by one and the angle is halved. The one-qubit rotation angles are integer multiples of $2\pi/2^{r+d-1}$, and hence of $2\pi/2^k$. Thus all rotations are powers of R_k. Each step adds $O(d)$ gates and uses only the original d qubits, giving size and depth $O(d^2)$ after summing over the degrees. □

We next apply the local circuit to a work register and compare its phase before and after adding the input. Repeating this comparison over a cycle of binary linear maps cancels the dependence on the initial work state.

Lemma 18. *Let d, r be positive integers satisfying $r + d - 1 \leq k$, and let a be an integer. For $x \in \{0,1\}^d$, define*

$$D\,|x\rangle = \exp\left(\frac{2\pi i a}{2^r} \prod_{\ell=1}^{d} x_\ell\right) |x\rangle\,.$$

For every integer $t \geq d$ satisfying $2^r \mid a2^t$, there is a circuit $\mathcal{C}$ over G using t catalytic qubits and no clean qubits such that

$$\mathcal{C} = D \otimes I_W.$$

Its size and depth are $O(t^2 2^t)$. In particular, $t \geq \max\{d, r\}$ always suffices.

Proof. Let I be the d-qubit input register and identify the basis labels of W with $\mathbb{F}_2^t$. Choose an invertible $t \times t$ binary matrix A whose powers cycle through all nonzero vectors. Such a matrix represents multiplication by a generator of the multiplicative group of a field with 2^t elements, written in a binary basis. It satisfies $A^{2^t-1} = I$ and, for every $z \neq 0$,

$$\{A^j z : 0 \leq j \leq 2^t - 2\} = \mathbb{F}_2^t \setminus \{0\},$$

with each nonzero vector occurring once.

For $t = 2$, we may take $A = P$, where P is the matrix used in Lemma 10. Its powers cycle through the three nonzero vectors of $\mathbb{F}_2^2$. The choice of A extends this property to work blocks of arbitrary size.

For $z \in \mathbb{F}_2^t$, set $f(z) := \prod_{\ell=1}^{d} z_\ell$, and write $\bar{x} := (x_1, \ldots, x_d, 0, \ldots, 0) \in \mathbb{F}_2^t$. Let F consist of d disjoint CNOT gates with action

$$F \left|x\right\rangle_I \left|z\right\rangle_W = \left|x\right\rangle_I \left|z + \bar{x}\right\rangle_W .$$

The zero coordinates of $\bar{x}$ specify targets that are unchanged by F, so they impose no initialisation condition on the work qubits. The two remaining operations act on W:

$$Q \left|z\right\rangle = \exp\!\left(\frac{2\pi i a}{2^r} f(z)\right) \left|z\right\rangle , \qquad \mathcal{A} \left|z\right\rangle = \left|Az\right\rangle .$$

Extending these operators by the identity on I, define

$$\mathcal{C} = (\mathcal{A} Q F Q^{-1} F)^{2^t-1},$$

where the rightmost factor acts first. One round has action

$$\mathcal{A} Q F Q^{-1} F \left|x\right\rangle_I \left|z\right\rangle_W = \exp\!\left(\frac{2\pi i a}{2^r}\big(f(z) - f(z + \bar{x})\big)\right) \left|x\right\rangle_I \left|Az\right\rangle_W .$$

After j rounds the work label is $A^j z$, and after all $2^t - 1$ rounds it returns to z.

We evaluate the sum of differences in the exponent separately for zero and nonzero work strings. If $z = 0$, each difference equals $-f(\bar{x})$ because $f(0) = 0$. For $z \neq 0$, the powers of A visit every nonzero work string, and translation by $\bar{x}$ permutes $\mathbb{F}_2^t$, giving

$$\sum_{y \in \mathbb{F}_2^t} \big(f(y) - f(y + \bar{x})\big) = 0.$$

Removing the term $y = 0$ leaves $f(\bar{x})$, so the two cases give

$$\sum_{j=0}^{2^t-2} \big(f(A^j z) - f(A^j z + \bar{x})\big) = \begin{cases} (1 - 2^t) f(\bar{x}), & z = 0, \\ f(\bar{x}), & z \neq 0. \end{cases} \tag{3}$$

After multiplication by a, both expressions are congruent to $a f(\bar{x})$ modulo 2^r, because $2^r \mid a 2^t$. Thus the circuit has exactly the phase of D on every joint basis state. By linearity, $\mathcal{C} = D \otimes I_W$.

Binary row elimination implements $\mathcal{A}$ with $O(t^2)$ CNOT gates, while each of Q and Q^{-1} uses $O(d^2)$ gates by Lemma 17, without further workspace. Including the two applications of F, one round has size and depth $O(t^2)$, and the $2^t - 1$ rounds give the claimed bounds. □

The operator identity also preserves entanglement with an external register. In choosing t, one may first cancel common powers of two from $a/2^r$ and omit coefficients that vanish modulo 2^r.

5.1.2 Implementing the normal form II: full expression

The full diagonal unitary is the product of its monomial phase gates. We implement these gates in parallel using separate work blocks. For one monomial, the input addition F in Lemma 18 consists of disjoint CNOT gates. Across several monomials, the same input qubit may control targets in different work blocks. The following lemma implements all gates sharing one control in logarithmic depth, without requiring the targets to be initialised. We use it below to implement the combined input addition, also denoted by F.

Lemma 19 (Parallel CNOT gates with a common control)**.** *Let $b \geq 1$ be an integer and let $x, z_1, \ldots, z_b$ be bits. The transformation*

$$|x\rangle\,|z_1, \ldots, z_b\rangle \longmapsto |x\rangle\,|z_1 + x, \ldots, z_b + x\rangle$$

has a CNOT *implementation of depth $2\lceil \log_2 b \rceil + 1$ and size $2b - 1$, with no additional qubits. The targets may be in an arbitrary state.*

Proof. Let $e_1 = (1, 0, \ldots, 0)^{\mathsf{T}}$ and $\mathbf{1} = (1, \ldots, 1)^{\mathsf{T}}$. On the b target qubits, construct a binary tree of CNOT gates whose matrix L satisfies $Le_1 = \mathbf{1}$. Starting from the first target, each layer uses the qubits already reached as controls for distinct new targets. This requires $b - 1$ gates in $\lceil \log_2 b \rceil$ layers and defines an invertible linear map on all target strings.

Write $z = (z_1, \ldots, z_b)^{\mathsf{T}}$. Apply the inverse target circuit, then one CNOT from x to the first target, and then the target circuit itself. The resulting string is

$$L(L^{-1}z + xe_1) = z + x\mathbf{1}.$$

The gate count and depth follow. For $b = 1$, the target circuit is empty. □

Lemma 20 (Parallel implementation of monomial phases)**.** *Let D be given by* (1)*, with its nonzero coefficients specified. Let $s \geq 1$ be an integer, and let N be the number of nonzero monomials.*

There is a circuit $\mathcal{C}_D$ over G using at most $k \min\{N, sn\}$ catalytic qubits and no clean qubits such that

$$\mathcal{C}_D = D \otimes I_W.$$

Its depth and size are, respectively,

$$O\left(2^k \left\lceil \frac{N}{sn} \right\rceil \left(k^2 + \log(\min\{N, sn\} + 1)\right)\right) \qquad \text{and} \qquad O(k^2 2^k N).$$

In particular, for fixed k its depth is

$$O\left(\left(1 + \frac{N}{sn}\right) \log(n + 1)\right),$$

its size is $O(N)$, and it uses $O(sn)$ catalytic qubits. If $sn \geq N$, its depth is $O(\log(n + 1))$, and its size and workspace are $O(n^k)$.

Proof. The case $N = 0$ is the identity and requires no gates. Otherwise, partition the nonzero monomials into $\lceil N/(sn) \rceil$ batches, each containing at most sn monomials. Apply the construction to each batch in turn, reusing the same work register.

Consider a batch containing b monomials, and assign a separate block of k work qubits to each monomial. For a support $S = \{i_1, \ldots, i_d\}$, the coefficient in $p(x)/2^k$ is $a_S/2^{k-d+1}$. Use Lemma 18

with $t = k$, $r = k - d + 1$, and $a = a_S$. Order each support once and use this order throughout the circuit. The input CNOT gates add $x_{i_1}, \ldots, x_{i_d}$ to the first d targets in its work block. All blocks use the same matrix A and the same $2^k - 1$ rounds.

Let F denote the product of these input CNOT gates over all blocks in the batch. Each input bit controls at most b targets. Implement these gates using Lemma 19. Since different input bits have disjoint target sets and distinct controls, their circuits run in parallel. A complete application of F has depth $O(\log(b+1))$ and size $O(kb)$.

Each round applies F, all Q^{-1} circuits, F again, all Q circuits, and finally all $\mathcal{A}$ circuits, in this order. The phase operations and binary linear transformations act on disjoint work blocks and begin only after the preceding application of F is complete. Although the fanout circuit uses gates between different blocks, it implements F exactly.

After all rounds, every work label is restored. By (3), each block contributes its monomial phase independently of the initial work string. The circuit for the batch thus applies the required phases and acts as the identity on the work register. This register can be reused for the next batch. The product of the phases over all batches is $\exp\bigl(2\pi i p(x)/2^k\bigr)$, giving $\mathcal{C}_D = D \otimes I_W$.

In each round, the phase operations and binary linear transformations have depth $O(k^2)$ and size $O(k^2 b)$. Including the two applications of F, the round has depth $O(k^2 + \log(b+1))$ and size $O(k^2 b)$. Multiplying by $2^k - 1$ rounds and $\lceil N/(sn) \rceil$ batches gives the depth bound. The batch sizes sum to N, so the total size is $O(k^2 2^k N)$. Reusing the work register requires only k qubits for each monomial in the largest batch, giving at most $k \min\{N, sn\}$ catalytic qubits.

Finally,

$$N \leq \sum_{d=1}^{\min(k,n)} \binom{n}{d} = O(n^k)$$

for fixed k. Together with $\lceil N/(sn) \rceil \leq 1 + N/(sn)$, this gives the remaining bounds. □

The work register can be smaller for a given list of coefficients. Reduce each nonzero fraction $a_S/2^{k-|S|+1}$ to an odd numerator and denominator 2^{r_S}, and choose $t_S = \max\{|S|, r_S\}$. Within each batch, round $j = 0, 1, \ldots$ acts only on blocks with $j < 2^{t_S} - 1$, excluding completed blocks from the fanout target sets. The workspace for a batch is the sum of its block sizes. Since the register is reused, the largest of these sums suffices for the full circuit. Each t_S is at most k, so the same worst-case bounds hold.

Terms with $|S| = 1$ can instead be applied directly to the input in one layer, without any work qubits. The matrix A for each block size can be chosen once for fixed k, and the circuit is then explicit from the coefficients and supports.

If the operation in Lemma 19 is available as a single native gate, including on targets in an arbitrary state, each application of F has constant depth. For fixed k, the construction above then has depth $O(\lceil N/(sn) \rceil)$. In particular, implementing all N monomials in parallel gives constant depth using at most kN catalytic qubits. The construction below uses less workspace but also requires catalytic circuits for binary linear maps, whose depth must be accounted for separately.

5.1.3 Reducing the workspace

The workspace count is dominated by the monomials with the highest degree, i.e. those that apply a phase -1 depending on k input qubits. We can construct an optimal compiler for these phases using the CNOT compiler developed above. We first specify the specific diagonal terms:

Definition 6. Let $k \geq 2$, let S range over subsets of $\{1, \ldots, n\}$ with $|S| = k$, and choose bits $a_S \in \{0, 1\}$. Define the diagonal unitary D_k by

$$p(x) = \sum_S a_S x_S, \qquad D_k \left|x\right\rangle = (-1)^{p(x)} \left|x\right\rangle,$$

where $p(x)$ is evaluated over $\mathbb{F}_2$.

To optimize these terms we take inspiration from [HC18], who group degree-three terms by a shared control to obtain controlled quadratic phases. We extend this grouping to degree k and reduce each quadratic with Dickson's theorem [Dic01].

The main idea is to batch different terms together, splitting the full polynomial into a sum of degree-$(k-2)$ monomials, each multiplied by a quadratic form. We can then use Dickson's theorem to optimize each quadratic form separately. This reduces the number of terms in each quadratic form from $O(n^2)$ to $O(n)$ monomials, so the total number of monomials after the basis changes is $O(n^{k-1})$, at the cost of applying these basis changes. Each basis change consists of a CNOT circuit followed by local X gates. The number of basis changes equals the number of degree-$(k-2)$ monomials, which is at most $\binom{n}{k-2} = O(n^{k-2})$.

All basis changes can be run in parallel, and one basis change costs $O(n^2/\log^2 n)$ catalytic qubits at depth $O(\log n)$. The total cost is therefore $O(n^k/\log^2 n)$ catalytic qubits, which is the dominant cost in the new circuit. Compared with applying Lemma 20 directly to all $O(n^k)$ monomials, the efficient catalytic CNOT circuits therefore reduce the workspace for the high-degree phases by a factor $\log^2 n$.

We first introduce, and prove, Dickson's theorem:

Lemma 21 (Dickson's Theorem [Dic01])**.** *Let $q : \mathbb{F}_2^n \to \mathbb{F}_2$ be a polynomial of degree at most 2 with $q(0) = 0$, and write it as:*

$$q(x) = \sum_{0 \leq i < j \leq n-1} B_{ij} x_i x_j + \sum_{i=0}^{n-1} A_i x_i.$$

Then there exists an affine transformation, given by a matrix $M \in GL(n, 2)$ and a vector $c \in \mathbb{F}_2^n$ such that $y = Mx + c$, and an integer d with $2d \leq n$, such that

$$q(x) = \sum_{i=0}^{d-1} y_{2i} y_{2i+1} + \delta y_{2d} + \epsilon,$$

where $\delta, \epsilon \in \{0, 1\}$, at most one of them is 1, and the linear term is omitted if $2d = n$.

Proof. The goal is to construct a basis transformation that constructs the right y values. If $q(x)$ contains a quadratic term, pick one such term $x_u x_v$ and relabel the variables such that it becomes $x_0 x_1$, so $B_{01} = 1$. A relabelling is a permutation of the coordinates, which is also an element of $GL(n, 2)$. Now look at all terms involving x_0 and x_1. These can be written as:

$$q_{x_0, x_1}(x) = x_0 x_1 + x_0 l_{\geq 2}(x) + x_1 k_{\geq 2}(x) + a x_0 + b x_1,$$

where $l_{\geq 2}(x)$ is the sum of the x_i such that $B_{0i} = 1$ for $i \geq 2$, $k_{\geq 2}(x)$ the same such that $B_{1i} = 1$ for $i \geq 2$, and $a = A_0$, $b = A_1$. Now let

$$y_0 = x_0 + k_{\geq 2}(x) + b, \qquad y_1 = x_1 + l_{\geq 2}(x) + a,$$

then it follows that,

$$y_0y_1 = x_0x_1 + x_0l_{\geq 2}(x) + x_1k_{\geq 2}(x) + ax_0 + bx_1 + l_{\geq 2}(x)k_{\geq 2}(x) + ak_{\geq 2}(x) + bl_{\geq 2}(x) + ab.$$

Importantly the additional terms $l_{\geq 2}(x)k_{\geq 2}(x) + ak_{\geq 2}(x) + bl_{\geq 2}(x)$ contain at most quadratic terms in x_i such that $i \geq 2$. Additionally notice that it might happen that in the product $l_{\geq 2}(x)k_{\geq 2}(x)$ some term x_i^2 appears, but because this is over $\mathbb{F}_2$ it holds that $x_i^2 = x_i$, which means that it adds at most a linear term. The term ab is an additive constant, which in the end will possibly produce the term ϵ. The new variables y_0 and y_1 only add terms in x_i with $i \geq 2$ and constants to x_0 and x_1, so this transformation is invertible. After this step we can now rewrite $q(x)$:

$$q(x) = y_0y_1 + \sum_{2\leq i<j\leq n-1} B'_{ij}x_ix_j + \sum_{i=2}^{n-1} A'_ix_i + \epsilon',$$

where B' contains the extra quadratic terms from $l_{\geq 2}(x)k_{\geq 2}(x)$, A' the extra linear terms produced by the additional terms, and $\epsilon' = ab$.

This step can now be repeated on the remaining variables using the updated B' and A'. Sequentially repeat this, each time picking a remaining quadratic term, until after d steps no quadratic term is left. Relabelling the paired variables as $y_0, \ldots, y_{2d-1}$ gives

$$q(x) = \sum_{i=0}^{d-1} y_{2i}y_{2i+1} + \sum_{i=2d}^{n-1} A''_ix_i + \epsilon',$$

where ϵ' is the sum of all terms ab modulo 2. Notice that the linear terms on the paired variables are already absorbed by the constants a and b in the definition of the y's. The remaining linear terms $\sum_{i=2d}^{n-1} A''_ix_i$ only involve variables that do not appear in any product, so they cannot be absorbed in the same way.

If all $A''_i = 0$, we are done with $\delta = 0$ and $\epsilon = \epsilon'$. Otherwise pick an index $j \geq 2d$ with $A''_j = 1$ and relabel such that $j = 2d$. Now let

$$y_{2d} = \sum_{i=2d}^{n-1} A''_ix_i + \epsilon',$$

and keep $y_i = x_i$ for the other unpaired variables. This is invertible because y_{2d} contains x_{2d}. All remaining linear terms and the constant are now collected in the single variable y_{2d}, which gives $\delta = 1$ and $\epsilon = 0$. This last linear term cannot be removed. If it is present, q equals 1 on exactly 2^{n-1} inputs, while a sum of products plus a constant never does, and an affine transformation does not change this number.

All steps are invertible affine transformations, so their composition is of the form $y = Mx + c$ with $M \in GL(n, 2)$, which proves the lemma. □

We will now show how this can be used to implement the phase circuits D_k.

Lemma 22. *Fix $k \geq 2$, and let $s \geq 1$ be an integer. Every D_k in Definition 6 has an exact catalytic implementation over G using $O(sn)$ catalytic qubits and no clean qubits, with depth*

$$O\left(\log(n+1) + \frac{n^{k-1}}{s\log(n+1)}\right)$$

and size

$$O\left(\frac{n^k}{\log(n+1)}\right).$$

In particular, depth $O(\log(n+1))$ requires at most $O(n^k/\log^2(n+1))$ catalytic qubits.

Proof. We start by decomposing the polynomial $p(x)$, where we write $x_T = \prod_{i \in T} x_i$. Assign each nonzero monomial to one of its subsets of size $k-2$, so that every monomial occurs in exactly one group. This gives

$$p(x) = \sum_{T \in \mathcal{T}} x_T q_T(x), \qquad g := |\mathcal{T}| \leq \binom{n}{k-2} = O(n^{k-2}),$$

where the $q_T(x)$ are homogeneous quadratic polynomials involving only variables outside T. When $k = 2$, take $T = \emptyset$ and $x_T = 1$. If $g = 0$, the circuit is the identity. Any $q_T(x)$ consists naively of $O(n^2)$ monomials, so a naive implementation of $p(x)$ requires evaluating $O(n^k)$ different monomials. However, because $q_T(x)$ is a quadratic form, Lemma 21 gives the construction of a basis transformation that reduces the number of monomials in $q_T(x)$ to at most $\lfloor n/2 \rfloor + 1$. The additional term is either linear or constant. Let $y_T = M_T x + c_T$, for $M_T \in GL(n, 2)$ and $c_T \in \mathbb{F}_2^n$ and,

$$\tilde{q}_T(y_T) = \sum_{i=0}^{r_T - 1} y_{2i,T} y_{2i+1,T} + \delta_T y_{2r_T,T} + \epsilon_T,$$

with $r_T \leq n/2$, such that $q_T(x) = \tilde{q}_T(y_T)$ as in Lemma 21. The linear term is omitted when $2r_T = n$.

Then if one has access to y_T instead of x, the term $x_T \tilde{q}_T(y_T)$ consists of at most $r_T + 1$ monomials instead of $O(n^2)$.

We cannot write y_T into a clean register, since all work qubits are catalytic. Writing it into a catalytic register gives $z + y_T$ for an unknown z, and applying the phase to that register would give $x_T \tilde{q}_T(z + y_T)$, which depends on z. Instead we add y_T directly into the work blocks of the phase construction of Lemma 20. That construction gives every monomial its own block of k catalytic qubits. It repeatedly applies a map F that adds the variables of the monomial to its block, $|\tau\rangle \mapsto |\tau + v\rangle$. The proof of Lemma 18 only uses that the label v is the same in every round, not that it consists of input bits. For $T = \{i_1, \ldots, i_{k-2}\}$ and the monomial $x_T y_{2i,T} y_{2i+1,T}$, use the label

$$v = (x_{i_1}, \ldots, x_{i_{k-2}}, y_{2i,T}, y_{2i+1,T}),$$

whose last two entries are affine functions of x. Each factor of x_T occupies a separate coordinate. The linear term $\delta_T x_T y_{2r_T,T}$ and the constant term $\epsilon_T x_T$ use the corresponding shorter lists, padded with zeros to length k.

For a term with $d \geq 1$ factors, use Lemma 18 with $t = k$, $r = 1$, and $a = 1$. The local phase Q acts on the first d coordinates. For each fixed x, the same vector v is added in every round, so (3) gives the required product independently of the initial work string. No independence assumption on the entries of v is needed. If $k = 2$ and $\epsilon_T = 1$, the resulting constant phase -1 is applied directly using $(XZ)^2 = -I$ on one input qubit.

For this addition, write O for the phase blocks of one group and W for the additional catalytic qubits. Let C_T be the circuit that adds all labels of group T to their blocks:

$$C_T \left|x\right\rangle_I \left|\tau\right\rangle_O \left|w\right\rangle_W = \left|x\right\rangle_I \left|\tau + L_T x + c'_T\right\rangle_O \left|w\right\rangle_W ,$$

where every row of L_T is either a unit vector for an index in T or a row of M_T, with zero rows for the padding, and c'_T collects the corresponding entries of c_T. The matrix L_T has n columns and $O(n)$ rows. Its linear addition is represented by the invertible matrix

$$\begin{pmatrix} I & 0 \\ L_T & I \end{pmatrix}$$

on $O(n)$ qubits. For sufficiently large n and an integer s' with $1 \leq s' \leq n/\log^2 n$, Theorem 12 implements this addition in depth $O(n/(s' \log n))$ using $O(s'n)$ additional catalytic qubits.

All C_T read the same input. To run them in parallel, give each active group an additional n-qubit register with arbitrary basis label u_T. First add x to every u_T, apply L_T from each copy to its phase blocks, add x to every u_T again, and apply L_T once more. The copies return to u_T, and the phase labels become

$$\tau + L_T(u_T + x) + L_T u_T = \tau + L_T x.$$

The unknown content of each copy enters twice and cancels because the map is linear. Now add c'_T by local X gates. The affine constant is added once, after the two linear additions. Each linear addition restores its additional catalytic qubits, so this implements C_T exactly and restores the copies and all other work qubits outside the phase blocks.

For b active groups, each addition of x to the copies has depth $O(\log(b+1))$ and size $O(nb)$ by Lemma 19. The linear circuits act on disjoint registers and run in parallel.

We now allocate the available workspace between these groups. For sufficiently large n, choose

$$s' := \min\{s, \lfloor n/\log^2 n \rfloor\}, \qquad b := \min\{g, \lfloor s/s' \rfloor\}.$$

Process at most b groups at a time, reusing the work registers between batches. Since $bs' \le s$, the phase blocks, the copies, and the additional registers for the linear circuits together use $O(sn)$ catalytic qubits.

We run the construction of Lemma 20 on the polynomial

$$p(x) = \sum_{T \in \mathcal{T}} x_T \tilde{q}_T(y_T),$$

which has $O(n^{k-1})$ products in the indicated coordinates, with F implemented by the maps C_T for the active groups. The phase blocks use $O(kn)$ catalytic qubits per group. Each round applies F, the local Q^{-1} circuits, F again, the local Q circuits, and the binary linear maps $\mathcal{A}$, in this order. Each application of F is completed before the next local operation begins. There are $2^k - 1$ rounds, after which every work register is restored and can be reused for the next batch.

Since k is fixed, $b \le g = O(n^{k-2})$ and $n/(s' \log n) \ge \log n$, a batch has depth $O(n/(s' \log n))$, including the additions to the copies. The total depth is

$$O\left(\left\lceil \frac{g}{b} \right\rceil \frac{n}{s' \log n} \right) = O\left(\log n + \frac{gn}{s \log n} \right).$$

Indeed, when $s \le \lfloor n/\log^2 n \rfloor$, we have $s' = s$ and $b = 1$: the groups are processed one at a time, using the available workspace for each linear circuit. For larger s, each group has depth $O(\log n)$, and additional workspace allows more groups to run in parallel. Using $g = O(n^{k-2})$ gives the depth bound.

Each linear addition has total width $O(s'n)$ and depth $O(n/(s' \log n))$, hence size $O(n^2/\log n)$. The additions to the copies and the operations on the phase blocks contribute $O(n)$ gates per group per round. Summing over the groups and the $2^k - 1$ rounds gives size $O(gn^2/\log n) = O(n^k/\log n)$.

In particular, choosing s of order $n^{k-1}/\log^2 n$ allows all groups to run in parallel. The workspace is dominated by the maps C_T, giving

$$\text{depth } O(\log n), \qquad \text{workspace } O\left(\frac{n^k}{\log^2 n} \right).$$

The remaining finitely many values of n can be implemented directly. Writing $\log(n+1)$ in the bounds makes them valid for all $n \ge 1$. □

5.1.4 Final circuit

The degree-k terms in the diagonal normal form give the phases in Definition 6. There are fewer terms of smaller degree, and Lemma 20 suffices to implement them within the same bounds.

Theorem 23 (Catalytic implementation of diagonal elements)**.** *Fix $k \geq 2$, and let $s \geq 1$ be an integer. Let D be given by* (1)*, with its nonzero coefficients specified. There is a circuit $\mathcal{C}_D$ over G using $O(sn)$ catalytic qubits and no clean qubits such that*

$$\mathcal{C}_D = D \otimes I_W.$$

Its depth and size are, respectively,

$$O\left(\log(n+1) + \frac{n^{k-1}}{s\log(n+1)}\right) \qquad \text{and} \qquad O\left(\frac{n^k}{\log(n+1)}\right).$$

In particular, there is an implementation of depth $O(\log(n+1))$ using $O(n^k/\log^2(n+1))$ catalytic qubits, with the same size bound.

Proof. Apply Lemma 22 to the degree-k terms. There are only $O(n^{k-1})$ remaining monomials, so Lemma 20, with the same parameter s, implements them in depth

$$O\left(\log(n+1) + \frac{n^{k-2}\log(n+1)}{s}\right)$$

and size $O(n^{k-1})$, using $O(sn)$ catalytic qubits. Since $\log^2(n+1) = O(n)$, these costs are within the bounds for the degree-k terms. Both circuits act as the identity on the work register, so they can reuse it. Their product is $D \otimes I_W$, and their depths and sizes add. Choosing s of order $n^{k-1}/\log^2(n+1)$ gives the logarithmic-depth implementation. □

5.1.5 Lower bounds

The following bounds show that logarithmic depth is necessary and that the workspace in Theorem 23 at this depth is optimal in the worst case, for fixed k.

Lemma 24. *Fix $k \geq 2$ and let s be an integer with $1 \leq s \leq n^{k-1}/\log^2(n+1)$. Suppose every n-qubit diagonal unitary in* (1) *has an exact catalytic implementation over the finite gate set G, of depth at most h and using at most sn catalytic qubits. Then*

$$h \geq \lceil \log_2 n \rceil, \qquad h = \Omega\left(\frac{n^{k-1}}{s\log(n+1)}\right).$$

In particular, the depth in Theorem 23 is optimal up to constant factors, and depth $O(\log(n+1))$ requires $\Omega(n^k/\log^2(n+1))$ catalytic qubits in the worst case.

Proof. The logarithmic depth is necessary in the worst case, even with an arbitrarily large catalytic register. Indeed,

$$D = \prod_{j=2}^{n} \mathrm{CZ}_{1,j} \in C_2 \subseteq C_k \qquad \text{satisfies} \qquad DX_1D^\dagger = X_1 \prod_{j=2}^{n} Z_j.$$

A layer of disjoint two-qubit gates at most doubles the support of an operator. A circuit implementing $D \otimes I_W$ must therefore have depth at least $\lceil \log_2 n \rceil$.

A counting argument gives a corresponding comparison for workspace. For fixed k, uniqueness of the coefficients in (1) gives

$$2^{\sum_{d=1}^{\min(k,n)}(k-d+1)\binom{n}{d}} = 2^{\Theta(n^k)}$$

distinct diagonal unitaries. The constants in this argument may depend on k. A layer on w qubits over the finite gate set G has at most $\exp(O(w\log(w+1)))$ possible choices. Thus, to implement every such diagonal element exactly with depth at most h and width at most w, we need

$$hw\log(w+1) = \Omega(n^k).$$

With sn catalytic qubits, the total width is $w = (1+s)n \leq 2sn$, which gives the stated depth bound. For $s \leq \mathrm{poly}(n)$ we have $\log(n+sn+1) = O(\log(n+1))$, so this matches the depth in Theorem 23. At depth $h = O(\log(n+1))$, it gives the worst-case lower bound $sn = \Omega(n^k/\log^2(n+1))$. If $w > n^k$ this is immediate, and otherwise $\log(w+1) = O(\log n)$.

□

5.2 Semi-Clifford circuits

The diagonal construction also applies to unitaries that become diagonal after multiplication by Clifford operators, known as semi-Clifford unitaries.

Definition 7 (Semi-Clifford unitary)**.** An n-qubit unitary is semi-Clifford if conjugation by it maps some maximal abelian subgroup of P_n to another maximal abelian subgroup of P_n.

Equivalently, a unitary U is semi-Clifford when there are Clifford unitaries $\mathcal{C}_1, \mathcal{C}_2$ and a diagonal unitary D such that

$$U = \mathcal{C}_1 D \mathcal{C}_2,$$

as shown in [ZCC08, Proposition 1]. We call the Clifford unitaries $\mathcal{C}_1$ and $\mathcal{C}_2$ in $U = \mathcal{C}_1 D \mathcal{C}_2$ the Clifford factors of the decomposition. If $U \in C_k$, then $D = \mathcal{C}_1^\dagger U \mathcal{C}_2^\dagger$ also belongs to C_k, since multiplication on either side by a Clifford unitary preserves membership in C_k [ZCC08, Proposition 3]. Any overall phase of D can be absorbed into $\mathcal{C}_1$, so we may write D in the form (1).

Corollary 25 (Catalytic implementation of semi-Clifford gates)**.** *Fix $k \geq 2$, and let $s \geq 1$ be an integer. Let an n-qubit semi-Clifford unitary $U \in C_k$ be given by a decomposition $U = \mathcal{C}_1 D \mathcal{C}_2$, with the two Clifford factors and the coefficients of D specified. There is a catalytic implementation over G using $O(sn)$ catalytic qubits and no clean qubits, with depth*

$$O\left(\log(n+1) + \frac{n^{k-1}}{s\log(n+1)}\right)$$

and size

$$O\left(\frac{n^k}{\log(n+1)}\right).$$

It acts as $U \otimes I_W$ up to one global phase independent of the input and catalyst.

In particular, there is an implementation of depth $O(\log(n+1))$ using $O(n^k/\log^2(n+1))$ catalytic qubits, with the same size bound.

Proof. For sufficiently large n, Lemma 13, with parameter

$$s' := \min\{s, \lfloor n/\log^2 n \rfloor\},$$

gives a circuit for each Clifford factor of depth

$$O\left(\log n + \frac{n}{s \log n}\right)$$

using $O(s'n)$ catalytic qubits. Since gates in each layer act on disjoint qubits, multiplying the depth $O(n/(s' \log n))$ by the total width $O(s'n)$ gives size $O(n^2/\log n)$. For the remaining finitely many values of n, the Clifford factors can be implemented directly with constant size and depth. Writing $\log(n+1)$ in place of $\log n$ makes the bounds valid for all $n \geq 1$.

Together with the circuit for D from Theorem 23, these circuits implement U by applying $\mathcal{C}_2$, D, and $\mathcal{C}_1$ in this order. Each circuit implements the required unitary and acts as the identity on the work register, up to a global phase independent of the input and work state. The three circuits can thus share a work register large enough for the largest workspace requirement, leaving any unused qubits untouched. Their depths and sizes add. For fixed $k \geq 2$, the bounds for D cover the costs of both Clifford factors, giving the claimed bounds. Choosing s of order $n^{k-1}/\log^2(n+1)$ gives the logarithmic-depth implementation. □

For a diagonal factor with only N nonzero coefficients, Lemma 20 gives another implementation. With $O(sn)$ catalytic qubits, its depth and size are

$$O\left(\frac{n}{s \log(n+1)} + \left(1 + \frac{N}{sn}\right) \log(n+1)\right) \qquad \text{and} \qquad O\left(k^2 2^k N + \frac{n^2}{\log(n+1)}\right).$$

These bounds can be smaller when the normal form has few nonzero coefficients.

If all N monomials are implemented in parallel and each Clifford factor has depth $O(\log(n+1))$, the common work register requires only

$$O\left(\max\left\{kN, \frac{n^2}{\log^2(n+1)}\right\}\right)$$

catalytic qubits. The maximum suffices because the three circuits use the register in turn.

Acknowledgements

Folkertsma acknowledges funding from the Quantum Software Consortium. Mertz was partially supported by grant 24-10306S of GA ČR and by Charles Univ. project UNCE 24/SCI/008. Strelchuk acknowledges support from the Wellcome Leap as part of the Q4Bio Program and the Royal Society University Research Fellowship. Subramanian acknowledges support from the Royal Society through a University Research Fellowship.

References

[AG04] Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5):052328, 2004. DOI: 10.1103/PhysRevA.70.052328. arXiv: quant-ph/0406196.

[AM25] Aryan Agarwala and Ian Mertz. Bipartite matching is in catalytic logspace. In *IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society, 2025.

[BBC+95] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical Review A*, 52(5):3457–3467, 1995. DOI: 10.1103/PhysRevA.52.3457. arXiv: quant-ph/9503016.

[BCH+20] Michael Beverland, Earl Campbell, Mark Howard, and Vadym Kliuchnikov. Lower bounds on the non-clifford resources for quantum computations. *Quantum Science and Technology*, 5(3):035009, May 2020. DOI: 10.1088/2058-9565/ab8963. URL: https://doi.org/10.1088/2058-9565/ab8963.

[BCK+14] Harry Buhrman, Richard Cleve, Michal Koucký, Bruno Loff, and Florian Speelman. Computing with a full memory: catalytic space. In *ACM Symposium on Theory of Computing (STOC)*, pages 857–866, 2014. DOI: 10.1145/2591796.2591874.

[BFM+25] Harry Buhrman, Marten Folkertsma, Ian Mertz, Florian Speelman, Sergii Strelchuk, Sathyawageeswar Subramanian, and Quinten Tupker. Quantum catalytic space. In *TQC*, volume 350 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 11:1–11:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025. DOI: 10.4230/LIPICS.TQC.2025.11.

[CDK+26a] Srijan Chakraborty, Samir Datta, Aryan Kusre, Partha Mukhopadhyay, and Amit Sinhababu. Maximum matching and related problems in catalytic logspace. *Electronic Colloquium on Computational Complexity (ECCC)*, TR26, TR26-080, 2026. URL: https://eccc.weizmann.ac.il/report/2026/080.

[CKL01] Dong Pyo Chi, Jinsoo Kim, and Soojoon Lee. Initialization-free generalized deutsch-jozsa algorithm. *Journal of Physics A: Mathematical and General*, 34(25):5251, June 2001. DOI: 10.1088/0305-4470/34/25/307. URL: https://doi.org/10.1088/0305-4470/34/25/307.

[CDK+26b] Petr Chmel, Aditi Dudeja, Michal Koucký, Ian Mertz, and Ninad Rajgopal. Frontier space-time algorithms using only full memory. In *IEEE Conference on Computational Complexity (CCC)*, volume 383 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 29:1–29:23, 2026. DOI: 10.4230/LIPICS.CCC.2026.29.

[CGM+25] James Cook, Surendra Ghentiyala, Ian Mertz, Ted Pyne, and Nathan Sheffield. The structure of in-place space-bounded computation. *Computing Research Repository (CoRR)*, abs/2510.12005, 2025.

[CLM+25] James Cook, Jiatu Li, Ian Mertz, and Edward Pyne. The structure of catalytic space: capturing randomness and time via compression. In *ACM Symposium on Theory of Computing (STOC)*, pages 554–564. Association for Computing Machinery (ACM), 2025.

[CM25] James Cook and Ian Mertz. Tree evaluation is in space O(log n · log log n). *SIAM Journal on Computing (SICOMP)*, Special Section STOC 2024:130–160, 2025. DOI: 10.1137/24M1690126.

[CP26] James Cook and Edward Pyne. Efficient catalytic graph algorithms. In *Innovations in Theoretical Computer Science Conference (ITCS)*, volume 362 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 43:1–43:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2026. DOI: 10.4230/LIPICS.ITCS.2026.43.

[CGK17] Shawn X. Cui, Daniel Gottesman, and Anirudh Krishna. Diagonal gates in the Clifford hierarchy. *Physical Review A*, 95(1):012329, 2017. DOI: 10.1103/PhysRevA.95.012329. arXiv: 1608.06596 [quant-ph].

[Dic01] Leonard Eugene Dickson. *Linear Groups with an Exposition of the Galois Field Theory*. B. G. Teubner, Leipzig, 1901.

[DPT+25] Dean Doron, Edward Pyne, Roei Tell, and R. Ryan Williams. When connectivity is hard, random walks are easy with non-determinism. In *ACM Symposium on Theory of Computing (STOC)*, pages 1108–1117. Association for Computing Machinery (ACM), 2025. DOI: 10.1145/3717823.3718303.

[DCM26] Zhenyu Du, Siyuan Cheng, and Xiongfeng Ma. Low-depth random unitaries without ancillae, 2026. arXiv: 2609.06528 [quant-ph]. URL: https://arxiv.org/abs/2609.06528.

[FMS+25] Marten Folkertsma, Ian Mertz, Florian Speelman, and Quinten Tupker. Fully characterizing lossy catalytic computation. In *Innovations in Theoretical Computer Science Conference (ITCS)*, volume 325 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 50:1–50:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025.

[Gid15] Craig Gidney. Constructing large controlled nots. https://algassert.com/circuits/2015/06/05/Constructing-Large-Controlled-Nots.html, 2015. Accessed 2026.

[Gid17] Craig Gidney. Factoring with $n+2$ clean qubits and $n-1$ dirty qubits, 2017. arXiv: 1706.07884 [quant-ph].

[GKZ25] David Gosset, Robin Kothari, and Chenyi Zhang. Multi-qubit toffoli with exponentially fewer t gates. *arXiv preprint arXiv:2510.07223*, 2025.

[HC18] Luke Heyfron and Earl T. Campbell. An efficient quantum compiler that reduces T count, 2018. arXiv: 1712.01557 [quant-ph]. URL: https://arxiv.org/abs/1712.01557.

[JST+20] Jiaqing Jiang, Xiaoming Sun, Shang-Hua Teng, Bujiao Wu, Kewen Wu, and Jialin Zhang. Optimal space-depth trade-off of CNOT circuits in quantum logic synthesis. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 213–229, 2020. DOI: 10.1137/1.9781611975994.13. arXiv: 1907.05087v3 [quant-ph]. Full version, revised 2022.

[KG25] Tanuj Khattar and Craig Gidney. Rise of conditionally clean ancillae for efficient quantum circuit constructions. *Quantum*, 9:1752, 2025. DOI: 10.22331/q-2025-05-21-1752.

[Kou16] Michal Koucký. Catalytic computation. *Bulletin of the EATCS (B.EATCS)*, 118, 2016.

[KMP+25] Michal Koucký, Ian Mertz, Ted Pyne, and Sasha Sami. Collapsing catalytic classes. In *IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society, 2025.

[KMS26] Michal Koucký, Ian Mertz, and Sasha Sami. Understanding robust catalytic computing. *Computing Research Repository (CoRR)*, abs/2605.09648, 2026. DOI: 10.48550/ARXIV.2605.09648.

[LPT24] Jiatu Li, Edward Pyne, and Roei Tell. Distinguishing, predicting, and certifying: on the long reach of partial notions of pseudorandomness. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, to appear, 2024.

[LTH+25] Yu Li, Guojing Tian, Xiaoyu He, and Xiaoming Sun. Preparation of hamming-weight-preserving quantum states with log-depth quantum circuits, 2025. arXiv: 2508.14470 [quant-ph]. URL: https://arxiv.org/abs/2508.14470.

[MN18] Dmitri Maslov and Yunseong Nam. Use of global interactions in efficient quantum circuit constructions. *New Journal of Physics*, 20:033018, 2018. DOI: 10.1088/1367-2630/aaa398.

[Mer23] Ian Mertz. Reusing space: techniques and open problems. *Bulletin of the EATCS (B.EATCS)*, 141:57–106, 2023.

[MLW+09] M. Müller, I. Lesanovsky, H. Weimer, H. P. Büchler, and P. Zoller. Mesoscopic Rydberg gate based on electromagnetically induced transparency. *Physical Review Letters*, 102:170502, 2009. DOI: 10.1103/PhysRevLett.102.170502.

[PMH08] Ketan N. Patel, Igor L. Markov, and John P. Hayes. Optimal synthesis of linear reversible circuits. *Quantum Information and Computation*, 8(3–4):282–294, 2008. DOI: 10.26421/QIC8.3-4-4.

[PSL05] Dong Pyo Chi, Jeong San Kim, and Soojoon Lee. Quantum algorithms without initializing the auxiliary qubits. *Phys. Rev. Lett.*, 95:080504, 8, August 2005. DOI: 10.1103/PhysRevLett.95.080504. URL: https://link.aps.org/doi/10.1103/PhysRevLett.95.080504.

[RV25] Maxime Remaud and Vivien Vandaele. Ancilla-free quantum adder with sublinear depth. In *Reversible Computation: 17th International Conference, RC 2025, Odense, Denmark, July 3–4, 2025, Proceedings*, pages 137–154, Odense, Denmark. Springer-Verlag, 2025. ISBN: 978-3-031-97062-7. DOI: 10.1007/978-3-031-97063-4_11. URL: https://doi.org/10.1007/978-3-031-97063-4_11.

[Sha26] Yakov Shalunov. Improved bounds on the space complexity of circuit evaluation. In *IEEE Conference on Computational Complexity (CCC)*, volume 383 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 3:1–3:13, 2026. DOI: 10.4230/LIPICS.CCC.2026.3.

[TT21] Yasuhiro Takahashi and Seiichiro Tani. Power of uninitialized qubits in shallow quantum circuits. *Theoretical Computer Science*, 851:129–153, 2021. DOI: 10.1016/j.tcs.2020.11.039. arXiv: 1608.07020 [quant-ph].

[Urs23] John Urschel. Representing the special linear group with block unitriangular matrices. *PNAS Nexus*, 2(10):pgad311, 2023. DOI: 10.1093/pnasnexus/pgad311. URL: https://doi.org/10.1093/pnasnexus/pgad311.

[Wil25] R. Ryan Williams. Simulating time with square-root space. In *ACM Symposium on Theory of Computing (STOC)*, pages 13–23. Association for Computing Machinery (ACM), 2025. DOI: 10.1145/3717823.3718225.

[YLN10] Chui-Ping Yang, Yu-xi Liu, and Franco Nori. Phase gate of one qubit simultaneously controlling n qubits in a cavity. *Physical Review A*, 81:062323, 2010. DOI: 10.1103/PhysRevA.81.062323.

[ZCC08] Bei Zeng, Xie Chen, and Isaac L. Chuang. Semi-Clifford operations, structure of $\mathcal{C}_k$ hierarchy, and gate complexity for fault-tolerant quantum computation. *Physical Review A*, 77(4):042313, 2008. DOI: 10.1103/PhysRevA.77.042313. arXiv: 0712.2084 [quant-ph].