IEEE Transactions on Quantum Engineering

Received 27 June 2022; revised 16 December 2022; accepted 17 December 2022; date of publication 9 January 2023; date of current version 29 March 2023.

Digital Object Identifier 10.1109/TQE.2022.3231194

Quantum Resources Required to Block-Encode a Matrix of Classical Data

B. DAVID CLADER[1], ALEXANDER M. DALZELL[2,3], NIKITAS STAMATOPOULOS[1], GRANT SALTON[2,3,4], MARIO BERTA[2,3,5,6], AND WILLIAM J. ZENG[1]

[1]Goldman Sachs & Co., New York, NY 10282 USA
[2]AWS Center for Quantum Computing, Pasadena, CA 91125 USA
[3]California Institute of Technology, Pasadena, CA 91125 USA
[4]Amazon Quantum Solutions Lab, Seattle, WA 98170 USA
[5]Department of Computing, Imperial College London, SW7 2BX London, U.K.
[6]Institute for Quantum Information, RWTH Aachen University, 52062 Aachen, Germany

Corresponding author: William J. Zeng (e-mail: William.Zeng@gs.com)

ABSTRACT We provide a modular circuit-level implementation and resource estimates for several methods of block-encoding a dense $N \times N$ matrix of classical data to precision ϵ; the minimal-depth method achieves a T-depth of $\mathcal{O}(\log(N/\epsilon))$, while the minimal-count method achieves a T-count of $\mathcal{O}(N \log(\log(N)/\epsilon))$. We examine resource tradeoffs between the different approaches, and we explore implementations of two separate models of quantum random access memory. As a part of this analysis, we provide a novel state preparation routine with T-depth $\mathcal{O}(\log(N/\epsilon))$, improving on previous constructions with scaling $\mathcal{O}(\log^2(N/\epsilon))$. Our results go beyond simple query complexity and provide a clear picture into the resource costs when large amounts of classical data are assumed to be accessible to quantum algorithms.

INDEX TERMS Block encoding, quantum circuit synthesis, quantum circuits, quantum computing, quantum random access memory (QRAM), quantum resources.

I. INTRODUCTION

A. MOTIVATION

A commonly used access model for quantum algorithms requiring classical data is a so-called "*block-encoding*" of the classical data into a unitary operator. Costs associated with these algorithms are often quoted in terms of the asymptotic scaling of the number of queries to the block-encoding oracle. However, if we are to make assessments about the potential of quantum random access memory (QRAM)-based algorithms, we must take into account the cost of each block-encoding query as well. In this work, we provide a detailed implementation and resource comparison of different methods one might use to block-encode a dense matrix A representing classical data.

Block-encodings for $N \times N$ matrices[1] are useful because they can, in principle, be implemented in exponentially small time (i.e., with quantum circuits of depth only polylog(N) [1]), albeit still with space cost of $\mathcal{O}(N^2)$ qubits. The block-encoding framework has proven useful for quantum algorithms for a variety of applications. Together with insights from adiabatic quantum computing, block-encoding has been crucial to the discovery of quantum linear system solvers with provably optimal scaling with respect to relevant parameters [2], [3], [4], [5]. The framework has led to faster algorithms for phase estimation [6], quantum gradients [7], and improved Hamiltonian simulation and regression techniques [1], [8]. Furthermore, block-encoding has been used to analyze quantum optimization algorithms [9], [10] and related algorithms for portfolio optimization [11]. Many of these algorithms make use of the quantum singular value transform (QSVT) algorithm [2], [12], which uses a technique known as quantum signal processing (QSP) [8], [13] that performs polynomial transformations on the block-encoded matrix.

B. MODEL

A unitary matrix U_A block-encodes the matrix $A \in \mathbb{R}^{N \times N}$ when the top-left block of U_A is proportional to A, i.e.,

$$U_A = \begin{pmatrix} A/\alpha & \cdot \\ \cdot & \cdot \end{pmatrix} \tag{1}$$

[1]Block-encodings for nonsquare matrices are a straightforward extension of the square case. We show how to do this in Appendix A1.

where $\alpha \geq ||A||$ is a normalization constant. We use $||\cdot||$ to denote the operator norm. The other blocks in U_A are irrelevant, but they must be encoded such that U_A is unitary. Note that we focus on real matrices A but the extension to complex matrices is straightforward.

There is a variety of techniques available to block-encode a matrix [1], [2]. For applications where we wish to operate on dense classical data, such as is often encountered in machine learning or finance applications, a particularly relevant method is the QRAM input model and its many proposed implementations [14], [15], [16], [17], [18]. More concretely, we refer to QRAM as the quantum circuit that allows query access to classical data in superposition

$$\sum_j \alpha_j |j\rangle |0\rangle \xrightarrow{\text{QRAM}} \sum_j \alpha_j |j\rangle |b_j\rangle \tag{2}$$

where j is the address in superposition with amplitude α_j and $|b_j\rangle$ is the classical data loaded into a quantum state. In this work, we discuss two different circuit-level approaches to accomplish the QRAM operation, which are optimized for different objectives. The first is a select-swap (SS) model, which is particularly efficient in terms of T-gate utilization [19]. The second is a bucket-brigade (BB) model [14], which has reduced susceptibility to errors when operated on potentially faulty hardware [18]. We also provide a variant of the SS-QRAM approach that performs a slightly different version of (2), where a more general single-qubit state can be loaded controlled by a flag qubit.

The block-encoding constructions presented in our work compile all circuits into Clifford gates and T gates, where attempts are made to minimize either the total number of qubits, the total number of T gates—henceforth called the T-count—or the minimum number of layers of parallel T gates—henceforth called the T-depth. Our estimates, as overviewed in the next section, illustrate the tradeoffs between these three metrics. We focus on T gates since, in many proposals for fault-tolerant quantum computation based on quantum error-correcting codes such as the surface code, Clifford gates are transversal and can be performed essentially for free (or in some cases by simply updating the Pauli frame for the decoder completely in software). Meanwhile, T gates require the expensive process of magic state distillation [20], [21]. Indeed, the motivating model for our work is the one in which the logical quantum circuits we describe are carried out with lattice surgery [22], [23], [24], [25] on many physical qubits encoded with the surface code. We additionally assume that logical CNOT gates can be performed between arbitrary qubits in constant time and that the fanout-CNOT operation, i.e., the product of many CNOTs with the same control but different targets, can be performed in constant time [23]. As such, our calculations can be seen as optimistic and could be underestimated if there are additional hardware-specific constraints on logical topology and connectivity.

TABLE I Quantum Resources Required to Block-Encode an $N \times N$ Matrix to Precision ϵ, Where We Assume That $N = 2^n$

Resource	Minimum Depth	Minimum Count
# Qubits	$4N^2 - 3N + 2n - 1$	$N(t+1) + 3n - t + 1$
T-Depth	$10n + 8R_y - 4$	$8N + 16n + 4R_y nt - 8$
T-Count	$(4R_y + 32)N^2 - 24N - 4R_y - 32n - 8$	$8(2t+3)N - 16t(n+1) + 4R_y nt - 24$
# Qubits	[1e3, 3e5, 7e7]	[4e2, 7e3, 1e5]
T-Depth	[5e2, 7e2, 8e2]	[2e4, 7e4, 2e5]
T-Count	[7e4, 2e7, 7e9]	[3e4, 2e5, 2e6]

The top part of the table contains parameterized estimates, while the bottom shows realistic values obtained for chosen values of the parameters. The parameter R_y is the number of T gates needed to synthesize a single qubit rotation about the Y-axis by an arbitrary angle, and t is the number of bits precision to store the classical data, where both t and R_y scale as $\mathcal{O}(\log(\log(N)/\epsilon))$. In the bottom part of the table, we computed the costs to block-encode random real matrices of sizes $[16 \times 16, 256 \times 256, 4096 \times 4096]$ up to precision $\epsilon = 0.01$.

C. OVERVIEW OF RESULTS

The main contributions of our work are summarized as follows.

1) Clifford+T circuits with minimal T-depth, or alternatively, with minimal T-count, to perform a block-encoding unitary U_A as given in (1) for $N \times N$ matrices A.
2) Resource calculations for the block-encoding, including all constant factors, expressed in terms of a small number of tunable parameters.
3) Clifford+T circuits for QRAM and state preparation subroutines, along with associated resource calculations.
4) Conceptual improvements to state preparation procedure yielding quadratic speedup in T-depth complexity.

Our results for the circuit resource costs in terms of the number of qubits, T-count, and T-depth are shown in the top part of Table 1, including constant factors. We note that the cost of making the T-depth exponentially smaller than the time needed even to write down all of the matrix entries is a large overhead of $\mathcal{O}(N^2)$ ancilla qubits and total T-count. It is possible to reduce the qubit and T-count to $\mathcal{O}(N)$ at the cost of making T-depth grow also to $\mathcal{O}(N)$.

In addition to parameterized resource estimates, we compute the exact costs to block-encode matrices of sizes $[16 \times 16, 256 \times 256, 4096 \times 4096]$ with precision $\epsilon = 0.01$. To estimate a matrix norm, needed for the scale factor α, we took the entries to be uniformly distributed random numbers on the interval [5.0, 105.0] as representative of the type of data we might choose to block-encode (e.g., financial market data). Example numbers are shown in the bottom part of Table 1.

D. BLOCK-ENCODING STRATEGY

In order to make the definition from (1) precise, we provide quantum circuits that perform a unitary $\tilde{U}_A$ on $n + \ell$ qubits,

IEEE Transactions on Quantum Engineering

where $N = 2^n$ for which

$$\begin{aligned} &\left\| A - \alpha(\langle 0|_\ell \otimes I)\tilde{U}_A(|0\rangle_\ell \otimes I) \right\| \\ &\quad = \alpha \left\| (\langle 0|_\ell \otimes I)(U_A - \tilde{U}_A)(|0\rangle_\ell \otimes I) \right\| \leq \epsilon \end{aligned} \tag{3}$$

where the parameter ℓ denotes the number of ancillas, I denotes the identity on n qubits, and $\|\cdot\|$ denotes the operator norm.[2] Subscripts on bras and kets are included for clarity to indicate the associated number of qubits in the state. Following standard convention, we call $\tilde{U}_A$ a (α, ℓ, ϵ)-block-encoding of A. The normalization factor for the block-encoding we study is $\alpha = ||A||_F$, where $||\cdot||_F$ denotes the Frobenius norm.[3]

All of our constructions follow the prescription laid out in [2], [26], and [27], which calls for forming U_A as the product of a pair of controlled-state preparation unitaries U_L and U_R. In this prescription

$$U_A = U_R^\dagger U_L \tag{4}$$

where controlled on an n-qubit register in the state $|j\rangle_n$, U_R prepares the n-qubit state $|\psi_j\rangle_n$, and U_L prepares the state $|\phi_j\rangle_n$ with the assistance of ℓ' QRAM ancilla qubits.[4] That is, we have

$$\begin{aligned} U_R\,|0\rangle_n\,|0\rangle_{\ell'}\,|j\rangle_n &= |\psi_j\rangle_n\,|0\rangle_{\ell'}\,|j\rangle_n \\ U_L\,|0\rangle_n\,|0\rangle_{\ell'}\,|k\rangle_n &= |k\rangle_n\,|0\rangle_{\ell'}\,|\phi_k\rangle_n \end{aligned} \tag{5}$$

We then choose

$$\begin{aligned} |\psi_j\rangle_n &= \sum_{k=0}^{N-1} \frac{A_{jk}}{||A_{j,\cdot}||}\,|k\rangle_n \\ |\phi_k\rangle_n &= \sum_{j=0}^{N-1} \frac{||A_{j,\cdot}||}{||A||_F}\,|j\rangle_n \quad \text{(independent of k)} \end{aligned} \tag{6}$$

where $A_{j,\cdot}$ denotes the jth row of A and $\|A_{j,\cdot}\|$ is the standard Euclidean norm of that vector. Since $|\phi_k\rangle_n$ is independent of k, U_L is simply state preparation rather than controlled-state preparation. However, in the q-norm version of the block-encoding presented in Appendix A2, $|\phi_k\rangle_n$ will depend on k. It is easily verified that U_A is an exact block-encoding of A, i.e.,

$$(\langle 0|_\ell \otimes \langle j|_n)U_A(|0\rangle_\ell \otimes |k\rangle_n) = A_{jk} \quad \text{with } \ell = n + \ell'. \tag{7}$$

We refer to Fig. 1 for an overview of this reduction from block-encoding to controlled-state preparation. To implement the controlled-state preparation unitaries U_R and U_L, we combine a QRAM-like data-loading step with a protocol for state preparation of n-qubit states.

[2]If N is not a power of 2, we can pad the matrix with zeros such that this is the case.

[3]One can use a different norm (the q-norm), which we discussed in Appendix A2.

[4]The unitary U_L additionally swaps the two n-qubit registers, cf., Fig. 1.

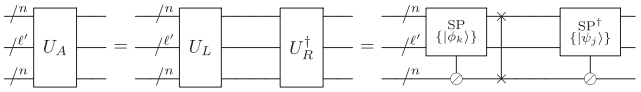

FIGURE 1. Reduction from block-encoding to controlled-state preparation. The block-encoding unitary U_A is the product of two controlled-state preparation procedures, U_L and $U_R^\dagger$. Given a family of quantum states $\{|\phi_k\rangle\}_{k=0}^{N-1}$, the controlled-SP gate denotes the unitary that performs $|0\rangle_n \mapsto |\phi_k\rangle_n$ on the first register conditioned on the final register being $|k\rangle$ and using $\ell' = \ell - n$ ancillas on the second register that begin and end in the state $|0\rangle_{\ell'}$. One can obtain the A_{jk} matrix element from the unitary by evaluating the $\langle 00j| \cdot |00k\rangle$ matrix element, as in (7). The ⊘ indicates that the register acts as a control and may produce nontrivial action on the target for any control setting (in contrast to • and ∘). The controlled-state-preparation gates can be implemented either with the circuit from Fig. 11 or the circuit from Fig. 15, discussed in Section III.

E. STATE PREPARATION

The state preparation problem has been studied in a series of prior works [19], [26], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39]. To prepare an n-qubit state, one basic approach is to perform a sequence of n single-qubit rotations on successive qubits, where the angle of rotation is controlled on previously rotated qubits. That is, for $p = 0, \ldots, n-1$, a rotation is performed on qubit p by one of 2^p possible rotation angles determined by the setting of qubits $0, \ldots, p-1$. Thus, there are $\sum_{p=0}^{n-1} 2^p = N - 1$ angles (where $N = 2^n$ is the dimension of the Hilbert space) that might be used at some point in the procedure. In [19], it was shown that given an n-qubit state $|\psi\rangle_n$, this approach can produce a state $|\tilde{\psi}\rangle_n$ for which

$$\| \, |\psi\rangle_n - |\tilde{\psi}\rangle_n \, \| \leq \epsilon \tag{8}$$

in T-depth $\mathcal{O}(\log^2(N/\epsilon))$ and using $\mathcal{O}(N)$ total qubits. This scaling comes from the need to do the $n = \log(N)$ controlled-rotations in series,[5] and to perform each controlled rotation by 1) leveraging a $\mathcal{O}(n)$-depth controlled-swap network to copy in a $\mathcal{O}(\log(1/\epsilon))$-bit description of the relevant rotation angle, and 2) for each bit applying an ϵ-accurate Clifford+T gate decomposition of the corresponding controlled-rotation, costing $\mathcal{O}(\log(1/\epsilon))$ gates per bit.[6] In Section III-C, we implement a version of this approach, which we call the "*fixed-precision*" method.

The extension to controlled-state preparation calls for creating one of $N = 2^n$ n-qubit states $\{|\phi_k\rangle\}_{k=0}^{N-1}$ controlled on an n-qubit register. As there are now $N(N-1)$ rotation angles that might be needed, the first step is to load the correct subset (which depends on the setting of the control k) of $N-1$ angles into an ancilla register. This loading step can be understood as a more general form of the QRAM query from (2). In Section II, we provide details of this initial loading step

[5]All logarithms in this article have taken base 2.

[6]A slightly more efficient approach applies the controlled rotation using a controlled-adder and a preprepared Fourier state rather than gate syntheses for each bit separately [19], [40].

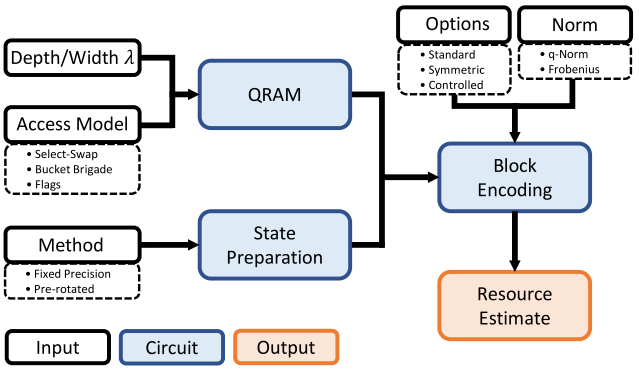

FIGURE 2. Schematic demonstrating our resource estimation procedure for block-encoding. The user specifies a QRAM access model, a parameter λ that allows one to trade depth for width, the method for state preparation routine, along with the block-encoding normalization factor and whether one wants a standard block-encoding, a controlled version, and/or a symmetric encoding. Note that not all possible combinations are allowed, e.g., the state-preparation method *prerotated* is only available together with the access model *flags* and the choice $\lambda = n$. Details are provided in the main text.

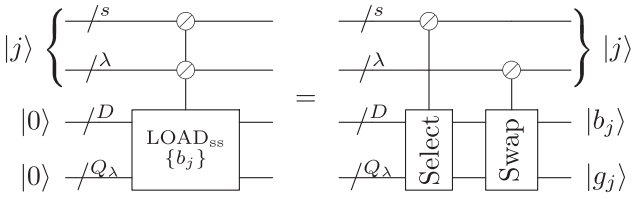

FIGURE 3. SS circuit with variable depth/width, which coherently loads classical data into a D-qubit data register. The $n = s + \lambda$ qubit control register is divided up into s qubits for the select control and λ qubits for the swap control. The size of the ancilla register is $Q_\lambda = (2^\lambda - 1)D$. Select iterates through all 2^s settings of the first s bits and writes the corresponding $2^\lambda D$ bits of classical data to the final two registers, costing $\mathcal{O}(2^s)$ depth. Swap moves the correct D bit register into the first of the 2^λ positions in $\mathcal{O}(\lambda)$ depth. When select is implemented with unary iteration to minimize T-count and T-depth, additional $s - 1$ ancilla qubits are needed (not shown) [41]. Implementation details for select and swap are given in Fig. 17.

and illustrate two separate circuit-level approaches compatible with the fixed-precision approach: 1) the SS approach, which is based on [19], and 2) the BB approach, which is based on [14], [15], and [18]. Both approaches allow the loading step to be accomplished in as little as $\mathcal{O}(\log(N))$ T-depth. The SS approach has a better constant factor, while the BB approach has enhanced natural noise resilience [18] that may lead to better overall performance when quantum error correction overhead is considered. After loading, a normal state preparation procedure is performed, followed by unloading of the angles to reset the ancillas to $|0\rangle$. As QRAM is a common subroutine in quantum algorithms, our estimates in Section II could be useful beyond the application of block-encoding.

F. IMPROVED STATE PREPARATION

In addition to the state preparation procedure presented above, we also present a distinct approach to state preparation that we call the "*prerotated*" approach. Prerotated state preparation may be of independent interest, as it quadratically improves the T-depth for the task of state preparation and controlled-state preparation from $\mathcal{O}(\log^2(N/\epsilon))$ to $\mathcal{O}(\log(N/\epsilon))$. The prerotated approach achieves logarithmic T-depth by 1) preapplying all possible single-qubit rotations onto $N - 1$ ancilla qubits in parallel using $\mathcal{O}(\log(1/\epsilon))$ depth, and 2) enacting each of the $\log(N)$ controlled-rotations in series using a constant-depth controlled-swap network that injects the appropriate ancilla into the state register. Our controlled-swap networks require only constant depth per controlled rotation because we wait to uncompute garbage until after all $\log(N)$ controlled-rotations have been completed. Note that the fixed-precision method can implement step 2), but not step 1). The prerotated method utilizes both steps and also requires a flag mechanism for uncomputing the ancillas that were not injected. The results of the prerotated approach are summarized in Table 5. A similar idea of prerotation for state preparation was also explored in [37], but their construction is not garbage-free. Additionally, the methods in [38] and [39] also have certain similarities and produce garbage-free states, but in both cases, the T-depth scales as at least $\mathcal{O}(\log(N)\log(\log(N)/\epsilon))$, which is still essentially quadratically worse than our $\mathcal{O}(\log(N/\epsilon))$.

G. VARIATIONS AND EXTENSIONS

We also consider a few variations of the basic block-encoding. The available variations are illustrated in Fig. 2 and the summarized as follows.

1) We demonstrate how to implement a mechanism suggested in [18], [19], and [41] that trades T-depth for T-count, parameterized by an integer $\lambda \in \{0, 1, \ldots, \log(N)\}$. The T-count for block-encoding is minimized at $\mathcal{O}(N)$ using the SS-QRAM with $\lambda = 0$, as reported in Table 1. However, the corresponding T-depth is also $\mathcal{O}(N)$, which is exponentially worse than the minimal depth case. Our results for general λ for the BB and the SS approaches appear later in Table 4. We report exact expressions including constant prefactors and subleading terms for all cases, which allows us to find a minimal T-count construction. These ideas are discussed in Sections II-B and II-C.
2) We explain how our construction can be adapted to perform a controlled-block-encoding of A, i.e., the operation $|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U_A$. These controlled block-encodings are important, e.g., when composing block-encodings of multiple matrices [2] and are discussed in Section IV-D.
3) We construct block-encodings of symmetrized matrices

$$\bar{A} = \begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix} \quad \text{where} \quad \bar{A} \in \mathbb{R}^{(M+N)\times(M+N)} \tag{9}$$

IEEE Transactions on Quantum Engineering

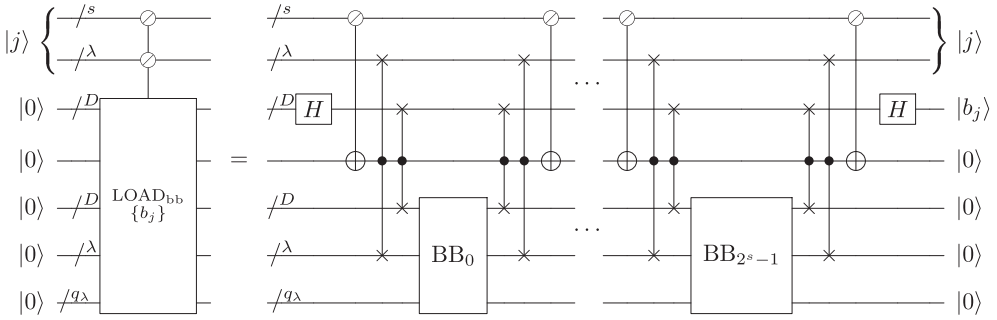

FIGURE 4. BB circuit with variable circuit depth/width, which coherently loads classical data into a D-qubit data register. The circuit iterates through all 2^s settings of the first s bits of the address register, with the first and last iteration shown. In each iteration, a multiply controlled Toffoli gate (where some bits are controlled on $|0\rangle$ and some on $|1\rangle$, denoted generically with $\oslash$) computes whether the s bits agree with the address setting $|j\rangle$. If so, the remaining $\lambda = n - s$ address qubits and the D-qubit bus (third register) in the state $|+\rangle^{\otimes D}$ are swapped into ancilla registers, where the BB circuit (denoted by BB_j, see Fig. 18 and [18]) is performed to load the state $H^{\otimes D}|b_j\rangle_D$ into the bus. No garbage is produced since during all other iterations the ancillas are in $|0\rangle$, which is a $+1$ eigenvector of the BB circuit. When the sequence of multiply controlled Toffoli gates is implemented with unary iteration to minimize T-count and T-depth, additional $s-1$ ancilla qubits are needed (not shown) [41], which, combined with the $q_\lambda = (2^\lambda - 1)(2D+1) - 1$ ancillas needed for the BB, gives a total of $Q_\lambda = (2D+1)2^\lambda + n - D - 2$ ancillas for the $\mathrm{LOAD}_{\mathrm{bb}}$ operation.

which are useful in some applications, and which can be block-encoded more efficiently than the general construction owing to their structure. Symmetrized block-encodings are discussed in Appendix A1.

4) We give a variation of the symmetrized encoding for which the normalization factor is the q-norm, which can be smaller than the Frobenius norm; the resource counts are similar. This variation is discussed in Appendix A2.

H. OUTLOOK

While implementation of QRAM-based algorithms with a large amount of classical data on actual hardware is quite a ways off, we hope our detailed circuit layouts and resource counts challenge the hardware community to think through these requirements and investigate whether or not specialized hardware can meet them with reduced resources. Just as modern computer random access memory is distinct from processing elements and permanent memory, it would be desirable for future quantum computers to have hybrid architectures in which the QRAM elements are distinct from the quantum processor and optimized for the requirements at hand. We hope that our results provide a blueprint for these requirements, as having an efficient means to load classical data into a quantum computer could open up many applications in business, finance, and beyond.

I. OUTLINE

The remainder of this article is structured as follows. In Section II, we give circuits and resource estimates for performing the QRAM operation of (2), as well as a more general data-loading operation that we use in our minimal depth block-encoding construction. In Section III, we elaborate on the conceptual approach to state preparation and controlled-state preparation and give circuits and resource estimates for these tasks. We present two distinct approaches, which we call fixed-precision and prerotated, where the former relies on the SS-QRAM or BB-QRAM, while the latter relies on the generalized QRAM operation. In Section IV, we compute and report the overall resource estimates for block-encoding. In Section V, we give an error analysis for finite precision implementations. Finally, Section VI, concludes this article. Various technical details and variants are deferred to Appendices A–C.

II. DATA LOADING AND QRAM IMPLEMENTATIONS

A. OVERVIEW

As overviewed in the previous section, block-encoding relies on controlled-state preparation, which necessitates an initial data loading step resembling the QRAM query of (2). We present multiple options for QRAM implementations, as depicted in the upper-left block of the flowchart in Fig. 2. We explain each of these options and provide non-Clifford resource counts. Since the QRAM operation is an important primitive in many quantum algorithms, this resource analysis could be of interest independent from our block-encoding estimates.

Note that in this section and throughout the article, we make extensive use of the symbol $\oslash$ when depicting $(a+b)$-qubit gates in which the a-qubit register acts as a control. That is, when the gate can be decomposed as

$$\sum_{j=0}^{2^a-1} |j\rangle\langle j|_a \otimes U_j \tag{10}$$

for some set of b-qubit unitaries $\{U_j\}$, we draw $\oslash$ on the a-qubit register in the circuit diagram. When U_j is identity for all j besides $j = |1\rangle^{\otimes b}$, we replace $\oslash$ with $\bullet$, and when U_j is identity for all j besides $j = |0\rangle^{\otimes b}$, we replace it with $\circ$. A similar usage of $\oslash$ appears in [19].

B. SELECT-SWAP DATA LOADING

The SS-QRAM access model allows one to trade circuit depth for width via a tunable integer parameter λ, where $0 \leq$

$\lambda \leq n$ [19]. The model also allows one to utilize qubits prepared in arbitrary initial states (the so-called "dirty" qubits) at the cost of a constant factor increase to the circuit depth. While dirty qubits could lead to resource savings when considering a full architectural implementation of an algorithm, we do not make explicit use of dirty qubits; we assume that the memory is initialized to the $|0\rangle$ state for all circuits in this article, except where otherwise stated.

The SS formalism—introduced in [19]—realizes the QRAM data-loading operation of (2) for a D-bit data register, except that it leaves additional garbage states in a $(\Lambda - 1)D$-qubit ancilla register, where $\Lambda = 2^{\lambda}$. That is, SS realizes the unitary LOAD_{ss} operation defined by the equation

$$\text{LOAD}_{\text{ss}}\left(\sum_{j=0}^{N-1}\alpha_j\,|j\rangle_{\lambda}\,|0\rangle_D\,|0\rangle_{(\Lambda-1)D}\right) = \sum_{j=0}^{N-1}\alpha_{\lambda}\,|j\rangle_n\,|b_j\rangle_D\,|g_j\rangle_{(\Lambda-1)D} \tag{11}$$

where $|b_j\rangle_D$ is the D-bit classical data associated with address j, and $|g_j\rangle_{(\Lambda-1)D}$ is a garbage state that contains shuffled and possibly phase-flipped versions of classical data at other addresses. Note that the garbage could be uncomputed by copying the data into an ancilla register and then running LOAD_{ss} in reverse, but this will not be necessary in our application.

The idea behind SS is as follows. The $n = s + \lambda$ qubit control register is divided into s qubits for the select control and λ qubits for the swap control, as shown schematically in Fig. 3. This division corresponds to a partition of the 2^n D-bit entries in the classical database into 2^s subsets of size $2^{\lambda} = \Lambda$. The first portion of LOAD_{ss} is a select subroutine, which implements unary iteration [41] through the s high bits of the control register and, controlled on each setting, writes the corresponding subset of classical data into Λ D-qubit registers. The depth of select is exponential in the number of control bits s. The next portion is the Swap subroutine, which, controlled on the λ low bits of the control, swaps the appropriate ancilla register into the top register, leaving the other $\Lambda - 1$ registers in some garbage state, as defined by the following equation:

$$\text{Swap}\left(\sum_{j=0}^{\Lambda-1}\alpha_j\,|j\rangle_n\,|\xi_0\rangle_D\,|\xi_1\rangle_D\cdots|\xi_{\Lambda-1}\rangle_D\right) = \sum_{j=0}^{\Lambda-1}\alpha_j\,|j\rangle_n\,|\xi_j\rangle_D\,|h_j\rangle_{(\Lambda-1)D} \tag{12}$$

where the states $|\xi_i\rangle_D$ are arbitrary and $|h_j\rangle_{(\Lambda-1)D}$ is a garbage state. The depth of Swap is linear in the number of control bits λ. More details of these two subroutines are given in Appendix B1.

Note that during the select subroutine, fanout CNOT gates are used to write the classical data, where the presence or absence of a target on each qubit is determined by the classical data at compile time. Fanout CNOT is an architectural primitive for surface-code based quantum computers, so we assume their cost is minimal relative to non-Clifford gates [23]. If single-target CNOT gates are required by the architecture instead, the fanout CNOT can be decomposed into a logarithmic depth network of normal CNOT gates [19]. Moreover, the swap subroutine requires controlled-swap gates, which can be implemented fairly efficiently with a phase-incorrect parallel controlled-swap operation, as shown in Fig. 23 (the incorrect phases can be absorbed into the garbage states $|g_j\rangle$).

C. BUCKET-BRIGADE DATA LOADING

The BB-QRAM was initially proposed in [14]. The primary motivation behind studying this approach is that it has improved noise-resilience compared to the SS approach [14], [15], [16], [17], [18], [42], [43]. We will not reprise this argument here; we refer the reader to the references, especially [16] and [18]. If this noise-resilience can be realized in a physical machine, it can reduce the resources required to error correct the data-loading operation, leading to physical resource savings. Whether SS-QRAM or BB-QRAM is preferred will depend upon the underlying architecture and details on the overhead required to implement quantum error correction.

Unlike the SS-QRAM, the BB-QRAM does not leave garbage and the unitary LOAD_{bb} can be defined by

$$\text{LOAD}_{\text{bb}}\left(\sum_{j=0}^{N-1}\alpha_j\,|j\rangle_n\,|0\rangle_D\,|0\rangle_{Q_{\lambda}}\right) = \sum_{j=0}^{N-1}\alpha_j\,|j\rangle_n\,|b_j\rangle_D\,|0\rangle_{Q_{\lambda}} \tag{13}$$

where the last register contains Q_{λ} ancilla states that begin and end in $|0\rangle$. Similar to the swap portion of the SS-QRAM circuit (see Fig. 17), the BB-QRAM circuit is composed primarily of controlled-swap gates, with an arrangement that acts to minimize entanglement and enhance noise resilience. Additionally, as in the SS-QRAM case, we can trade circuit depth for width using a parameter λ [18], as shown in Fig. 4, where the implementation of the gates labeled BB_i is described in more detail in Appendix B2. However, reducing circuit width does not reduce the overall T-count as it does with the SS implementation, which is a drawback of this scheme. Note that only the portion of the circuit labeled BB_i will retain the noise resilience, and thus, the noise resilience is lost when we choose $\lambda = 0$. Finally, observe that our circuit for the BB-QRAM in Fig. 4 uses entirely qubits, rather than qutrits, as had been proposed in early work on the BB-QRAM; Hann et al. [18] showed that both the qubit and the qutrit version enjoy the same sort of noise resilience.

D. QRAM OPERATION WITH FLAGS

We introduce a third and final data-loading operation, which generalizes the previous approaches by allowing data of the form $\cos(\theta/2)\,|0\rangle + \sin(\theta/2)\,|1\rangle$ to be loaded for any θ rather

IEEE Transactions on Quantum Engineering

than just classical bits $|0\rangle$ or $|1\rangle$. Additionally, this load operation has a flag qubit and only acts nontrivially if the flag is set to 1, which is necessary to implement the minimal depth "prerotated" version of state preparation in Section III-D.

Suppose we have a set of $N = 2^n$ angles $\{\theta^{(j)}\}_{j=0}^{N-1}$. Let $R_y(\theta)$ denote the unitary rotation about the Y-axis by angle θ, i.e.,

$$\begin{aligned} R_y(\theta)\,|0\rangle &= \cos(\theta/2)\,|0\rangle + \sin(\theta/2)\,|1\rangle \\ R_y(\theta)\,|1\rangle &= -\sin(\theta/2)\,|0\rangle + \cos(\theta/2)\,|1\rangle\,. \end{aligned} \tag{14}$$

The data loading with flags operation, which we call LOADF, acts on four registers: an n-qubit "address" register, a single-qubit "flag" register, an N-qubit "angle" register, and an N-qubit ancilla register. The unitary LOADF operation is defined by the equation

$$\begin{aligned} &\mathrm{LOADF}\left(\sum_{j=0}^{N-1}\sum_{f=0}^{1}\alpha_{jf}\,|j\rangle_n\,|f\rangle_1\,|0\rangle_N\,|0\rangle_N\right) \\ &\quad= \sum_{j=0}^{N-1}\sum_{f=0}^{1}\alpha_{jf}\,|j\rangle_n\,|f\rangle_1\left[R_y(f\theta^{(j)})\,|0\rangle_1\right]|0\rangle_{N-1}\,|0\rangle_N\,. \end{aligned} \tag{15}$$

When the first address register is in the state $|j\rangle$, the unitary $R_y(\theta^{(j)})$ is applied to the first position of the angle register if and only if the flag bit f is set to 1.

Our implementation of LOADF involves applying doubly controlled-R_y gates, which can be done using the decomposition in Fig. 19 and adding an extra control. Indeed, we will need to apply a gate we call V, which enacts a unitary specified by the equation

$$\begin{aligned} &V\left(|f\rangle\,|0\rangle_N\,|x_0\rangle_1\,|x_1\rangle_1\ldots|x_{N-1}\rangle_1\right) \\ &\quad\otimes|x_0\rangle_1\,|x_1\rangle_1\ldots|x_{N-1}\rangle_1 = |f\rangle\left[\bigotimes_{k=0}^{N-1}R_y(fx_k\theta^{(k)})\,|0\rangle_1\right]. \end{aligned} \tag{16}$$

In other words, for each $k = 0, \ldots, N-1$, V performs a $R_y(\theta^{(k)})$ rotation on the kth qubit of the angle register if the flag bit is 1 and the kth qubit of the ancilla register is 1. This can be implemented in $\mathcal{O}(1)$ total T-depth using the doubly controlled-rotation decomposition of Fig. 19 with the parallel-Toffoli construction implicit from Fig. 20.

Since LOADF only applies $R_y(\theta^{(j)})$ and no other rotations, we must first prepare the state $|x_0\rangle_1\ldots|x_{N-1}\rangle_1$ for which $x_j = 1$ and $x_k = 0$ for all $k \neq j$, so that application of V applies the correct rotation. To prepare this state, we flip the first of the N ancilla bits to $|1\rangle$, and then run the inverse of the controlled-swap network from the $\mathrm{LOAD}_{\mathrm{ss}}$ operation in Fig. 3, which moves the $|1\rangle$ state into the jth ancilla. Application of V computes $R_y(f\theta^{(j)})\,|0\rangle$ into the jth position of the angle register while leaving all other angle registers in $|0\rangle$. Next, the controlled-swap network from $\mathrm{LOAD}_{\mathrm{ss}}$ is applied to move the jth angle register to the first position and the jth ancilla register back to the first position, so that an X gate on the first ancilla leaves the entire ancilla register back in $|0\rangle$. This implementation realizes the LOADF operation and is depicted in Fig. 5. A complete example for $n = 2$ is given in Fig. 6.

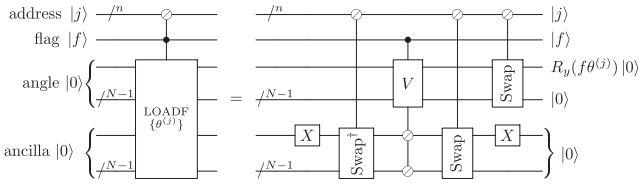

FIGURE 5. Circuit that implements LOADF, which loads a single-qubit state of the form $\cos(\theta/2)\,|0\rangle + \sin(\theta/2)\,|1\rangle$, where the angle θ depends on a control register. The controlled-swap gates on registers of size N have the action of swapping the register in position j to position 1; the adjoint of swap moves position 1 to position j. The gate V applies the rotation $R_y(\theta^{(k)})$ to the kth angle register controlled on the kth ancilla and flag qubits set to $|1\rangle$, for every k. The controlled-swap gates have depth $\mathcal{O}(n)$, and the V gate has depth $\mathcal{O}(\log(1/\delta))$ if we wish to perform each $R_y(\theta^{(k)})$ unitary up to precision δ. A complete example of LOADF for $n = 2$ is shown in Fig. 6.

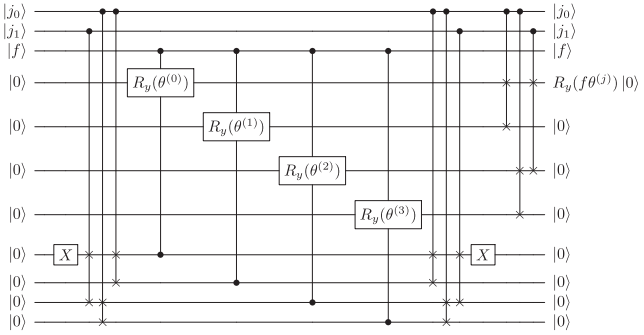

FIGURE 6. Example circuit for LOADF operation on $n = 2$. Each $R_y(\theta^{(k)})$ gate can be performed in T-depth $\mathcal{O}(1)$ plus the depth required to decompose the single-qubit rotation into Clifford + T, scaling as $\mathcal{O}(\log 1/\delta)$ if the target precision is δ.

The LOADF operation loads a single angle conditioned on a single flag. In general, we can load D states $R_y(\theta_r^{(j)})\,|0\rangle_1$ for $r = 1, \ldots, D$ by copying the flag, angle, and ancilla registers D times and running D copies of LOADF in parallel, where the controlled-swap operations for all D copies share a common control (the n-qubit address register).[7] The address qubits only ever act as controls for multiqubit controlled-swap gates, so increasing D does not increase the T-depth.

Note also that the LOADF operation requires controlled-swap gates with the target registers in states of the form $R_y(\theta)\,|0\rangle$, which does not admit the use of the phase-incorrect controlled swap that can be used for $\mathrm{LOAD}_{\mathrm{ss}}$ and $\mathrm{LOAD}_{\mathrm{bb}}$. Therefore, we utilize the parallel implementation shown in Fig. 20. This circuit shows a controlled swap between arbitrary two-qubit states with the assistance of two ancilla qubits. The Toffoli construction from [44] then allows one to implement the Toffoli with a T-depth of 1 and a T-count of 4 at the cost of a single extra clean ancilla qubit.

[7] For controlled-state preparation (see Fig. 15), we will need to load $N-1$ angles conditioned on $N-1$ separate flags, so we will take $D = N-1$.

TABLE II Resource Counts for LOAD and LOADF Operations, Which Coherently Load One Out of 2^n Possible D-Qubit Data Registers, Controlled by an n-Qubit Address Register

Resource	LOAD_{ss}	LOAD_{bb}	LOADF
# Qubits	$D \cdot 2^{\lambda} + 2n - \lambda - 1$	$(2D+1) \cdot 2^{\lambda} + 2n - 2$	$4D \cdot 2^n + n + D$
T-Depth	$4 \cdot 2^{n-\lambda} + 4\lambda - 4$	$(48\lambda - 36) \cdot 2^{n-\lambda} - 4$	$2n + 2R_y + 2$
T-Count	$4D \cdot 2^{\lambda} + 4 \cdot 2^{n-\lambda} - 4D - 4$	$16(D+1) \cdot 2^n - (8D + 8\lambda + 12) \cdot 2^{n-\lambda} - 4$	$(2R_y + 20)D \cdot 2^n - 12D$

LOAD loads D bits of classical data, and is accomplished using one of two models of QRAM implementation, SS and BB, each of which admits a depth/width tradeoff governed by integer parameter $0 \leq \lambda \leq n$. LOADF performs a more general data loading operation where each of the D data qubits is left in the state $\cos(\theta/2)\,|0\rangle + \sin(\theta/2)\,|1\rangle$ for some $\theta \in [0, 2\pi]$. The parameter R_y is the T-count of the single-qubit clifford+T gate sequence used to synthesize each single-qubit state, which scales as $\mathcal{O}(\log(1/\delta)))$ with the desired precision δ [see (46)]. We do not examine a depth/width tradeoff for LOADF as it is only used in our minimal depth construction.

E. QRAM RESOURCE ESTIMATES

We now present the non-Clifford resource estimates for the three data-loading operations implementations discussed above, summarized in Table 2. For LOAD, we report the resources required for the hybrid scheme that allows one to vary the circuit depth and width via the parameter $\lambda \in \{0, 1, \ldots, \log(N)\}$. For both LOAD_{ss} and LOAD_{bb}, the T-depth is minimized at $\mathcal{O}(n)$ when $\lambda = n$. Taking $D = \mathcal{O}(1)$, the LOAD_{ss} operation achieves minimum T-count of $\mathcal{O}(2^{n/2})$ when $\lambda \approx n/2$, while LOAD_{bb} requires $\Omega(2^n)$ T-count for all λ. For $\lambda = 0$, one has unary iteration [41], which yields the minimal-qubit-count but maximal-depth case.

Our counts can be verified using the circuit diagrams in Figs. 3–5, along with the following additional observations, which refer gate decompositions given in Appendix C.

1) The Select gate from Fig. 3 and the sequence of multiply controlled Toffoli gates in Fig. 4 are accomplished with unary iteration which requires $4(2^s - 1)$ (nonparallelizable) T gates and $s - 1$ additional ancilla qubits that are not depicted in those figures (see [41] for construction). We assume intermediate measurements and subsequent classically controlled Clifford gates are allowable; if not, additional $4(s - 1)$ T gates would be needed.
2) The Swap gate that appears in Figs. 3 and 5 is accomplished using $2^{\lambda} - 1$ controlled-swap gates, occurring over λ parallel layers. For LOAD_{ss}, these parallel controlled-swap gates are implemented with the construction in Fig. 23, costing total T-count $4(2^{\lambda} - 1)$ and T-depth 4λ. For LOADF (where $\lambda = n$), the goal is to give the minimal possible depth, so we choose to implement the parallel controlled-swap gates using the construction in Fig. 20 (costing 2^{n-1} ancilla qubits), where we perform each of the Toffolis using the depth-1 count-4 construction of [44] (costing 2^{n-1} additional ancilla qubits and requiring intermediate measurements). The total T-depth for the Swap gate is n and the total T-count is $4(2^n - 1)$. The final two swap gates of Fig. 5 are performed in parallel.
3) The costs of the BB_j gates of Fig. 4 are evaluated by inspection of Fig. 18 and quoted in the caption of that figure.
4) The controlled rotations appearing within the circuit for the gate V of Fig. 5 are implemented with Fig. 19, which introduce a set of parallel Toffoli gates and single-qubit rotations. Each of these parallel Toffoli gates is implemented the same way as parallel controlled-swap gates, reusing the same 2^n ancillas, contributing depth 1 and count $4 \cdot 2^n$. Note that if it is known that the flags are set to 1, these Toffolis become CNOTs, which are Clifford gates. Each single-qubit rotation costs T-depth and count equal to R_y, where R_y is the number of T gates in the gate synthesis of a single qubit rotation unitary, for which the leading term is $3\log(1/\delta)$ when the target precision is δ (see (46) of the error analysis in Section V).
5) For LOADF, Fig. 5 is for $D = 1$; to generalize to larger D, the flag, angle, and ancilla registers are copied D times, and all gates are performed in parallel, multiplying the T-count by a factor of D and leaving the depth unchanged.

III. QUANTUM STATE PREPARATION

A. OVERVIEW

In Section I-D, we described how block-encoding is reduced to controlled-state preparation. Here, we give explicit circuits for state preparation and controlled-state preparation. In Section III-B, we describe the conceptual approach to state preparation coming from prior literature. In Section III-C, we give circuits for what we call the "fixed-precision" version of state preparation and controlled-state preparation. Then, in Section III-D, we give circuits for what we call the "prerotated" version of state preparation and controlled-state preparation. Both versions have the same conceptual framework but differ on other aspects. The fixed-precision version of controlled-state preparation utilizes either the SS-QRAM

IEEE Transactions on Quantum Engineering

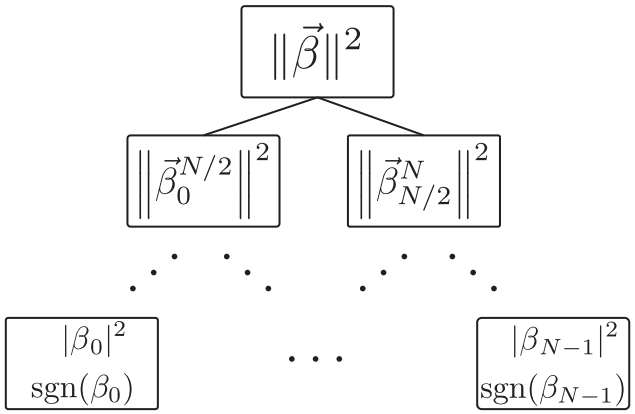

FIGURE 7. Illustration of the binary tree data structure that is used to create the quantum state $|\psi\rangle = ||\vec{\beta}||^{-1} \sum_{j=0}^{N-1} \beta_j |j\rangle$.

or BB-QRAM circuit from Section II as a subroutine and is compatible with the depth/width tradeoff parameterized by λ. Choosing $\lambda = 0$ with the SS-QRAM leads to our minimal T-count construction. Meanwhile, the prerotated version is used to achieve minimal T-depth. Indeed, the T-depth for preparing an N-dimensional state to error ϵ scales as $\mathcal{O}(\log(N/\epsilon))$ for the prerotated construction, an asymptotic improvement over $\mathcal{O}(\log^2(N/\epsilon))$ from [19].

B. ROTATION ANGLES AND BINARY TREE DATA STRUCTURE

Prior to the specific circuit-level instantiations, we describe our general approach to state preparation, which has appeared throughout the literature [19], [26], [29], [30]. The task is to construct a circuit that creates an n-qubit quantum state $|\psi\rangle_n = ||\vec{\beta}||^{-1} \sum_{j=0}^{N-1} \beta_j |j\rangle_n$ given a list of its coefficients $\{\beta_j\}$, where $N = 2^n$ is a power of two and zero padding of the classical data can ensure this, and $||\vec{\beta}||$ denotes the Euclidean vector norm to ensure normalization. We use the notation $\vec{\beta}$ to indicate the vector of values $[\beta_0, \beta_1, \ldots, \beta_{N-1}]$ and $\vec{\beta}_u^v = [\beta_u, \beta_{u+1}, \ldots, \beta_{v-1}]$ to denote the vector of values between the high (v) and low (u) indices with $v > u$.

We assume knowledge of the real amplitudes β_j. Given these amplitudes, we can construct a classical binary tree data structure as follows. The tree has depth n, and at the leaves we store the values $|\beta_j|^2$ and $\mathrm{sgn}(\beta_j)$ in the leaf nodes of the tree.[8] Then, each internal node stores the sum of the two child nodes. This proceeds to the root of the tree, which stores the sum of the squares of all N amplitudes, i.e., $||\vec{\beta}||^2$. We show an example of such a tree in Fig. 7. The binary tree data structure allows efficient updates if single amplitudes change; that is, if an amplitude is updated, only $\log(N)$ of the $2N - 1$ nodes need to be recomputed. Thus, the data structure can be efficiently maintained if matrix entries arrive in an online fashion [26].

[8]If we wish to allow β_j to be complex, we can store a complex phase in place of $\mathrm{sgn}(\beta_j)$.

Using the data in the binary tree, we construct a circuit that prepares the state $|\psi\rangle$. The state is initialized to the $|0\rangle_n$ state. In step 1, a Y-axis rotation is applied on the first of the n qubits by the angle $\theta_1 = 2\cos^{-1}(||\vec{\beta}_0^{N/2}||/||\vec{\beta}||)$, as in (14), yielding the state

$$|\psi_1\rangle_n = [\cos(\theta_1/2)\,|0\rangle + \sin(\theta_1/2)\,|1\rangle]\,|0\rangle_{n-1}\,. \tag{17}$$

This angle can be computed from the values stored at the root of the binary tree and its children. In step 2, a rotation is applied to the second qubit, either by the angle $\theta_2 = 2\cos^{-1}(||\vec{\beta}_0^{N/4}||/||\vec{\beta}_0^{N/2}||)$ or by the angle $\theta_3 = 2\cos^{-1}(||\vec{\beta}_{N/2}^{3N/4}||/||\vec{\beta}_{N/2}^{N}||)$, where the angle θ_2 is used conditioned on the first qubit being in state $|0\rangle$ and θ_3 is used conditioned on the first qubit being in state $|1\rangle$. Note that θ_2 and θ_3 can each be computed from values stored at one level-2 node and its children. This creates the state

$$\begin{aligned}|\psi_2\rangle_n = {} & [\cos(\theta_1/2)\,|0\rangle\,(\cos(\theta_2/2)\,|0\rangle \\ & + \sin(\theta_2/2)\,|1\rangle)]\,|0\rangle_{n-2} \\ & + [\sin(\theta_1/2)\,|1\rangle\,(\cos(\theta_3/2)\,|0\rangle \\ & + \sin(\theta_3/2)\,|1\rangle)]\,|0\rangle_{n-2}\,.\end{aligned} \tag{18}$$

Collecting terms, we can rewrite this state as

$$|\psi_2\rangle_n = [\alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle + \alpha_{10}\,|10\rangle + \alpha_{11}\,|11\rangle]\,|0\rangle_{n-2} \tag{19}$$

where

$$\begin{aligned}\alpha_{00} &= \cos(\theta_1/2)\cos(\theta_2/2) = ||\vec{\beta}_0^{N/4}||/||\vec{\beta}|| \\ \alpha_{01} &= \sin(\theta_1/2)\sin(\theta_2/2) = ||\vec{\beta}_{N/4}^{N/2}||/||\vec{\beta}|| \\ \alpha_{10} &= \sin(\theta_1/2)\cos(\theta_3/2) = ||\vec{\beta}_{N/2}^{3N/4}||/||\vec{\beta}|| \\ \alpha_{01} &= \sin(\theta_1/2)\sin(\theta_3/2) = ||\vec{\beta}_{3N/4}^{N}||/||\vec{\beta}||.\end{aligned} \tag{20}$$

In general, for any $w \in \{1, \ldots, n\}$ and for any w-bit binary string $y \in \{0, 1\}^w$, we can define $\alpha_y = ||\vec{\beta}_{y2^{n-w}}^{(y+1)2^{n-w}}||/||\vec{\beta}||$, where for the purpose of multiplication on the right-hand side, we interpret y the integer associated with its binary string. Note that the values stored at the 2^w nodes of level w of the binary tree are precisely $|\alpha_y|^2$ for $y \in \{0, 1\}^w$. Extrapolating from the pattern in steps 1 and 2, we assert that the state after step w is given by

$$|\psi_w\rangle_n = \sum_{y\in\{0,1\}^w} \alpha_y\,|y\rangle_w\,|0\rangle_{n-w}\,. \tag{21}$$

This formula will hold if we implement the transition from $|\psi_w\rangle_n$ to $|\psi_{w+1}\rangle_n$ as follows:

$$\begin{aligned}|\psi_w\rangle_n &= \sum_{y\in\{0,1\}^w} \alpha_y\,|y\rangle_w\,|0\rangle_{n-w} \\ &\to \sum_{y\in\{0,1\}^w} \alpha_y\,|y\rangle_w \left(\frac{\alpha_{y0}}{\alpha_y}\,|0\rangle_1 + \frac{\alpha_{y1}}{\alpha_y}\,|1\rangle_1\right)|0\rangle_{n-w-1}\end{aligned}$$

$$= \sum_{y\in\{0,1\}^{w+1}} \alpha_y |y\rangle_{w+1} |0\rangle_{n-w-1}$$
$$= |\psi_{w+1}\rangle \ . \tag{22}$$

That is, we apply a rotation to qubit $w+1$ by an angle θ_{1y}, where

$$\theta_{1y} = 2\cos^{-1}(\alpha_{y0}/\alpha_y) \tag{23}$$

which depends on y.[9] At the final iteration, the state becomes

$$|\psi_n\rangle_n = \sum_{y\in\{0,1\}^n} \alpha_y |y\rangle = \sum_{y\in\{0,1\}^n} \frac{|\beta_y|}{||\vec{\beta}||} |y\rangle$$
$$\equiv \sum_{j=0}^{N-1} \frac{|\beta_j|}{||\vec{\beta}||} |j\rangle \tag{24}$$

where in the final equivalence, we switched from binary notation, denoted by the variable y to decimal notation denoted by the variable j. This is exactly the state that we set out to prepare, up to the sign of the amplitude. To set the correct sign, we apply a (-1) phase to any state $|j\rangle$ for which the jth leaf in the binary tree indicates a sign of (-1). In practice, this is accomplished by loading the sign bit into an ancilla register and applying a Pauli-Z gate, and then unloading the sign bit.[10]

The binary tree data structure in Fig. 7 has $2N-1$ nodes. All nodes must store the value $||\vec{\beta}_u^v||^2$ and the leaf node must also store a single bit for the sign. However, we notice that in the above construction, all that matters is the angle of rotation, and there are only $N-1$ distinct angles, which are each inferred from the values at two of the nodes. (For example, $\theta_1 = 2\cos^{-1}(||\vec{\beta}_0^{N/2}||/||\vec{\beta}||)$.) Thus, in our resource analysis, we assume that these angles are classically precomputed to avoid the need for any arithmetic on the quantum computer. Ultimately, the information about these angles enters the circuit through the coherent data-loading operations discussed in Section II.

For controlled-state preparation, we are tasked with preparing one of N different arbitrary states, depending on the setting of a control register. To do so, we must compute the $N-1$ angles for each of these N states, giving $N(N-1)$ distinct angles, which can be organized into N separate binary tree data structures.

C. FIXED-PRECISION CIRCUIT

The high-level protocol described in the previous section calls for applying n single-qubit rotations by an angle that depends on the setting of other qubits. Our "fixed-precision" state preparation protocol, which we describe in this section, performs each of these rotations by loading a binary representation of the correct angle up to some fixed precision (i.e., an approximation of the angle with a prespecified number of bits), performing a controlled-R_y rotation by that angle, and then uncomputing the binary description of the angle.

The angle-loading step is essentially a QRAM query since a different angle must be loaded for each control setting as in (2). However, in this application, it is allowable to leave garbage in an ancilla register as long as the garbage is eventually uncomputed. Our implementation of the angle-loading step assumes that the initial state has a t-bit description of all $N-1$ angles stored in $N-1$ t-qubit ancilla registers. The circuit consists entirely of controlled-swap gates that shuffle these $N-1$ ancillas to move the correct angle into the first position, leaving the other $N-2$ registers in a garbage state that is entangled with the data. These circuits are very similar to the swap portion of the SS networks described in Section II-B.

Given the set of $N-1$ angles $\{\theta_1, \ldots, \theta_{N-1}\}$ needed to create the state $|\psi\rangle_n$, for each j, let $\bar{\theta}_j$ denote the integer multiple of $2\pi/2^t$ that is nearest to θ_j. Then, let $|\bar{\theta}_j\rangle_t$ be a computational basis state corresponding to the binary representation of the t-bit integer $2^t\bar{\theta}_j/(2\pi)$. Moreover, let $\{s_0, \ldots, s_{N-1}\}$ denote the sign bits for each of the N computational basis states. Define $|\bar{\Theta}\rangle_{(N-1)t} = |\bar{\theta}_1\rangle_t |\bar{\theta}_2\rangle_t |\bar{\theta}_3\rangle_t \ldots |\bar{\theta}_{N-1}\rangle_t$ and $|\vec{s}\rangle_N = |s_0\rangle_1 |s_1\rangle_1 \ldots |s_{N-1}\rangle_1$. Then, we assume that the state preparation protocol acts on an initial state of $n+D$ qubits, with $D = (N-1)t + N$, given by

$$|0\rangle_n |\bar{\Theta}\rangle_{(N-1)t} |\vec{s}\rangle_N \ . \tag{25}$$

This is a computational basis state and thus can be prepared in a single Clifford layer by applying X gates on the appropriate qubits determined by the classical data. In the case of controlled-state preparation, a different set of $N-1$ angles and N sign bits needs to be loaded depending on the setting of an n-qubit control register; in this case, the operation LOAD_{ss} or LOAD_{bb} defined in Sections II-B and II-C with $D = (N-1)t + N$ is used to load in the D bits of angular and sign data. Recall that the state we want to create is $|\psi\rangle_n = ||\vec{\beta}||^{-1} \sum_{j=0}^{N-1} \beta_j |j\rangle$. For each basis state $|j\rangle$, a different sequence of n angles and one sign bit is actually applied. Accordingly, we define $n+1$ controlled-swap networks denoted by $S_1, S_2, S_3, \ldots, S_n, S_\pm$ (note that S_1 will always be the identity operation and can be omitted from the circuit). These can each be written in the form

$$S_p = \sum_{j=0}^{N-1} |j\rangle\langle j|_n \otimes S_p^{(j)} \tag{26}$$

where $S_p^{(j)}$ acts on the D-qubit ancilla register such that the product $S_p^{(j)} S_{p-1}^{(j)} \ldots S_1^{(j)}$ has the action of swapping the t-bit description of the angle associated with the pth rotation for

[9] Writing the angle as θ_{1y} ensures that when the subscript is interpreted as an integer written in binary, the angle at step 1 is θ_1, the angle at step 2 is one of $\{\theta_2, \theta_3\}$, the angle at step 3 is one of $\{\theta_4, \theta_5, \theta_6, \theta_7\}$, etc.

[10] We note that complex amplitudes could also be considered; in that case, one would load the complex phase into an ancilla register and then apply a controlled rotation by that angle about the Z-axis.

IEEE Transactions on Quantum Engineering

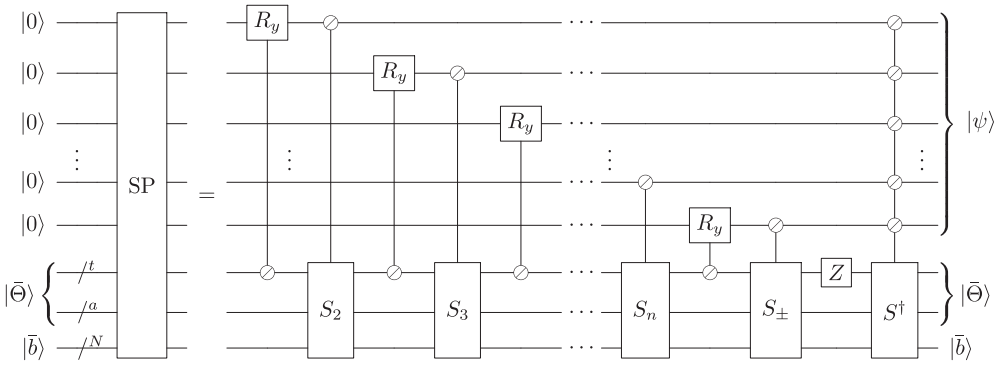

FIGURE 8. Fixed-precision state preparation circuit, which approximately prepares the state $|\psi\rangle_n$ into the first n qubits. The protocol requires an ancilla register initialized in a computational basis state $|\bar{\Theta}\rangle_{(N-1)t}\,|\bar{b}\rangle_N$ containing information about the $N-1$ rotation angles and N sign bits (note from the figure that $a=(N-2)t$). The circuit alternates between swapping a t-bit description of the next angle into the first ancilla register (the operation that accomplishes this is denoted by S_p for $p=2,\ldots,n$) and rotating a single qubit by the angle stored in that register (denoted by R_y in the figure). The implementation of the t-qubit-controlled-R_y gate is given in Fig. 9. The operation S_1 is omitted because it is the identity, and the other S_p operations can each be completed in constant T-depth; an example of this implementation for $n=3$ is shown in Fig. 10. The Z gate acts only on the first qubit of the t-qubit register, which contains the appropriate sign bit after application of $S_\pm$.

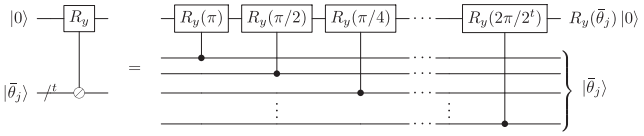

FIGURE 9. Implementation of t-qubit controlled-R_y from Figs. 8 and 10. We assume the angle $\bar{\theta}_j \in [0,\pi]$ is an integral multiple of $(2\pi)/2^t$ and that the input state $|\bar{\theta}_j\rangle$ contains the binary representation of $\bar{\theta}_j 2^t/(2\pi)$. The controlled-$R_y$ rotation by angle $\bar{\theta}_j$ is accomplished by performing t singly controlled-R_y gates by fixed angles π, $\pi/2$, $\pi/4$,..., $(2\pi)/2^t$ in succession.

$|j\rangle$ into the first t-qubit ancilla register, and $S_\pm^{(j)} S_n^{(j)} \ldots S_1^{(j)}$ has the action of swapping the sign bit for $|j\rangle$ into the first 1-qubit register. Importantly, it will be the case that S_p is controlled only on the first $p-1$ bits of $|j\rangle$.

To prepare $|\psi\rangle_n$, these swap networks are interleaved with controlled single-qubit rotations: for each $p=1,\ldots,n$, after the gate S_p, a controlled-R_y rotation is implemented on qubit p controlled by the t-bit description of the angle stored in the first register. This rotation is implemented as shown in Fig. 9 with t singly controlled-R_y rotations by increasingly finer angles. After the gate $S_\pm$, a Z gate is applied to the first ancilla qubit to apply the correct sign. Finally, the angles are restored to their initial positions by performing the gate $S^\dagger = S_1^\dagger S_2^\dagger \ldots S_p^\dagger S_\pm^\dagger$. This protocol realizes the unitary SP depicted in Fig. 8 and defined by the equation

$$\mathrm{SP}\left(|0\rangle_n\,|\bar{\Theta}\rangle_{(N-1)t}\,|\vec{s}\rangle_N\right) = |\psi\rangle_n\,|\bar{\Theta}_1\rangle_{(N-1)t}\,|\vec{s}\rangle_N\,. \qquad (27)$$

The controlled-swap networks in the SS-QRAM circuit have depth $\mathcal{O}(p)$ when there are $\mathcal{O}(p)$ controls. A straightforward way to implement S_p is to use these networks first to do the reverse of S_{p-1} in depth $\mathcal{O}(p)$ to swap out the $(p-1)$th angle, and second to swap in the correct angle in depth $\mathcal{O}(p)$. This would suggest the T-depth of performing $S_1,\ldots,S_n,S_\pm$ is $\sum_{p=0}^{n}\mathcal{O}(p)=\mathcal{O}(n^2)$. However, we give an

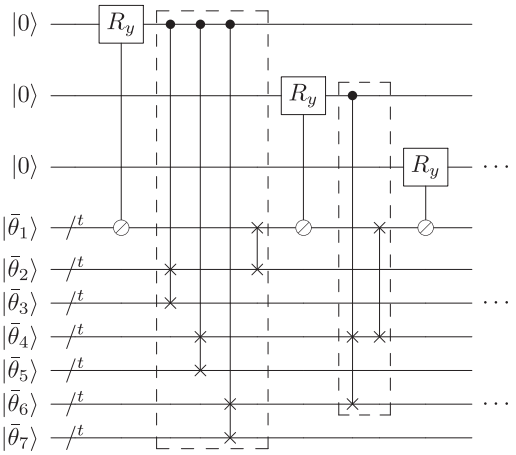

FIGURE 10. Example three-qubit fixed-precision state preparation circuit. For simplicity of presentation, the sign bits and the inverse operation $S^\dagger$ are omitted. The two boxes correspond to implementations of S_2 and S_3 from Fig. 8. The key point is that all controlled swaps in the implementation of S_p have a single common control, which allows them all to be implemented with T-depth 4, independent of n, as shown in Fig. 23.

optimization that reduces the depth of each S_p to $\mathcal{O}(1)$, and thus, an overall depth of $\mathcal{O}(n)$; this is seen in the example implementations of S_2 and S_3 for $n=3$ shown in Fig. 10. The main idea is to avoid undoing the work already accomplished by S_{p-1}, and note that each S_p can be controlled on just one of the n data qubits.

The full block-encoding circuit in Fig. 1 requires a controlled-state preparation not state preparation. Here, there are N different states $|\phi_k\rangle$, each with their own angle data $\bar{\Theta}^{(k)}$ and sign data $\vec{s}^{(k)}$. To perform controlled-state preparation, one simply loads this data conditioned on a control register in state $|j\rangle$ using either $\mathrm{LOAD}_{\mathrm{ss}}$ or $\mathrm{LOAD}_{\mathrm{bb}}$ defined in Section II. Then the state preparation circuit from Fig. 8 is performed followed by the inverse of the LOAD operation to clear the data registers. This is shown in Fig. 11.

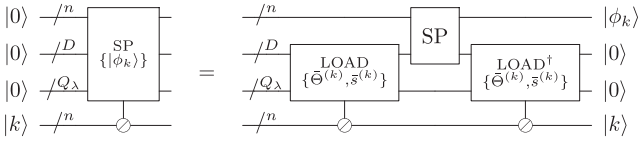

FIGURE 11. Controlled-state preparation circuit diagram for the fixed-precision approach to state preparation. For each of the N possible settings of the bottom register, a different n-qubit state $|\phi_k\rangle$ is prepared in the top register. This is accomplished with three subroutines. The LOAD subroutine, which can be either LOAD_{ss} from Fig. 3 or LOAD_{bb} from Fig. 4, loads in a t-bit description of the $N-1$ angles $|\tilde{\Theta}^{(k)}\rangle_{(N-1)t}$ and the N sign bits $|\tilde{s}^{(k)}\rangle_N$ into the second register of size $D=(N-1)t+N$, with the assistance of Q_λ ancillas. The SP subroutine is given in Fig. 8 and prepares the state $|\phi_k\rangle_n$. Finally, the reverse of the LOAD operation is performed to reset the second register and the ancillas to $|0\rangle$. The instances of controlled-state preparation that appear in the block-encoding unitary U_A of Fig. 1 can be accomplished with this circuit.

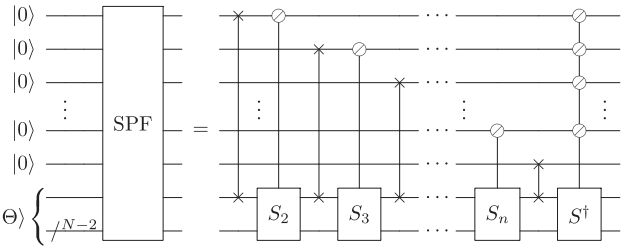

FIGURE 12. Prerotated state preparation circuit, which approximately prepares the state $|\psi\rangle_n$ into the first n qubits with garbage. That is, if $|\psi\rangle_n=\sum_j \frac{\beta_j}{||\vec{\beta}||}|j\rangle_n$, the state that is prepared is $\sum_j \frac{\beta_j}{||\vec{\beta}||}|j\rangle_n|g_j\rangle_{N-1}$. Each S_p gate can be performed in $\mathcal{O}(1)$ T-depth, for a total T-depth of $\mathcal{O}(\log(N))$. The entanglement between the n-qubit data register and $(N-1)$-qubit garbage register can be uncomputed using a flag mechanism as described in the main text with additional $\mathcal{O}(\log(N/\epsilon))$ cost, where ϵ is the error on the state prepared.

D. PREROTATED CIRCUIT

In this section, we present an alternative approach to state preparation which achieves smaller T-depth than the fixed-precision version, both practically and asymptotically. The asymptotic T-depth scaling is $\mathcal{O}(\log(N/\epsilon))$, with ϵ the error on the state prepared. The prerotated version takes the idea of precomputing the angles and signs for state preparation one step further and encodes them as the amplitude of a single qubit. For a given quantum state $|\psi\rangle_n$, we assume that we have classically computed the $N-1$ angles $\{\theta_r\}_{r=1}^{N-1}$ and N sign bits $\{s_j\}_{j=0}^{N-1}$ stored in the binary tree data structure as discussed previously. Let

$$|\theta_r\rangle_1=\cos(\theta_r/2)\,|0\rangle_1+\sin(\theta_r/2)\,|1\rangle_1 \tag{28}$$

if $1\le r<N/2$ and

$$|\theta_r\rangle_1=(-1)^{s_{2r-N}}\cos(\theta_r/2)\,|0\rangle_1+(-1)^{s_{2r-N+1}}\sin(\theta_r/2)\,|1\rangle_1 \tag{29}$$

if $N/2\le r<N$. These added signs allow us to load the sign information in the final $N/2$ qubits. For each r, let V_r be a Clifford+T gate decomposition of an R_y rotation by some angle that prepares $|0\rangle\mapsto|\theta_r\rangle$ up to error δ.[11] Note that if we have classically precomputed the angle θ_r, we can also classically compute a gate sequence V_r that gives a δ approximation for $R_y(\theta_r)$ in classical time $\text{polylog}(1/\delta)$ [45]. We assume that the input state is the product state on $n+N-1$ qubits given by

$$|0\rangle_n|\Theta\rangle_{N-1}=|0\rangle_n\left(\bigotimes_{r=1}^{N-1}|\theta_r\rangle_1\right) \tag{30}$$

which also acts as the definition of $|\Theta\rangle_{N-1}$. This product state can be prepared by applying each of the V_r to $N-1$ ancillas initially in $|0\rangle$ in parallel. In the case of controlled-state preparation, later, we will use the LOADF operation to prepare a different initial state for each setting of a control register.

Given the state in (30) as input, we perform a nearly identical state preparation circuit to that shown in the previous section. The only differences are that the application of the sign bit with $S_\pm$ and a Z gate can be omitted, as the sign bit is built into the state $|\theta_r\rangle$, and that the nt non-Clifford controlled-R_y rotations are replaced by n single-qubit swap gates (which are Clifford gates). That is, rather than use the angle qubit as a control for a rotation, we inject the angle into the data with a simple swap gate, as shown in Fig. 12. For any computational basis state $|j\rangle$, n of the $N-1$ states $|\theta_r\rangle$ are injected into the data. Once the state preparation procedure is complete, we reverse the swap circuit to return all the angles $|\theta_r\rangle_1$ back to their initial positions with the exception that the n angles that were injected are replaced by $|0\rangle_1$. Let $f_{r|j}=1$ if angle θ_r is an ancestor of the jth leaf in the binary tree, i.e., if $|\theta_r\rangle$ was injected into the state in the computational path associated with $|j\rangle$; otherwise, $f_{r|j}=0$. Then, we can state the action of our protocol, denoted by the unitary SPF and depicted in Fig. 12, by the following equation:

$$\text{SPF}\left(|0\rangle_n|\Theta\rangle_{N-1}\right)=\sum_j\frac{\beta_j}{||\vec{\beta}||}|j\rangle_n|g_j\rangle_{N-1} \tag{31}$$

where

$$|g_j\rangle_{N-1}=\bigotimes_{r=1}^{N-1}|(1-f_{r|j})\theta_r\rangle_1\ . \tag{32}$$

The state given by (31) has the correct coefficients for each basis state $|j\rangle_n$ but has $|0\rangle_1$ states in place of $|\theta_r\rangle_1$ for the n angles that were swapped into one of the first n registers. In other words, there is garbage leftover that is entangled with the data. In some applications, this garbage might be allowable. However, one can also uncompute the garbage and disentangle the two parts of the state, which we do by

[11] Any $|\theta_r\rangle$ can be written as $R_y(\theta)|0\rangle$ for some θ. Let H_r be a Clifford+T decomposition of $R_y(\theta/2)$ up to error $\delta/2$ so that $V_r=H_r^*H_r$ approximates $R_y(\theta)$ up to error δ, where H_r^* denotes the complex conjugate of H_r. Decomposing in this way allows us to give a δ-approximate Clifford+T decomposition for the controlled-$R_y(\theta)$ gate by using H_r and $H_r^\dagger$ in place of $R_y(\theta/2)$ and $R_y(-\theta/2)$ in Fig. 19.

IEEE Transactions on Quantum Engineering

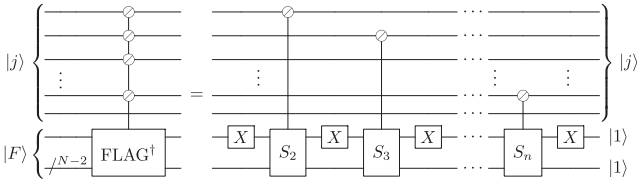

FIGURE 13. Circuit for the adjoint of the FLAG gate used to disentangle the garbage and data registers in the prerotated state-preparation protocol. We present FLAG† rather than FLAG to draw attention to the similarity between FLAG and the first half of SPF from Fig. 15. Controlled on the data qubits in state $|j\rangle$, FLAG switches n of the $N-1$ flag qubits in the bottom register $|F\rangle = \bigotimes_{r=1}^{N-1} |1 - f_{r|j}\rangle_1$ from $|1\rangle \mapsto |0\rangle$ at locations corresponding to the positions of the n angles that are used in the synthesis of amplitude $|j\rangle$.

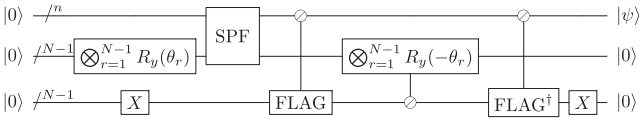

FIGURE 14. Circuit for garbage-free state preparation with prerotated method. An arbitrary n-qubit state $|\psi\rangle$ is prepared with the assistance of $2(N-1)$ ancilla qubits. The circuit for the SPF gate is given in Fig. 12 and the FLAG gate is given in Fig. 13. The X gate denotes a Pauli-X on all $N-1$ qubits. The controlled-R_y gate denotes $N-1$ completely parallel controlled $R_y(\theta_r)$ gates, with each flag qubit (third register) acting as a control for a rotation on a different angle qubit (second register). To prepare $|\psi\rangle$ to precision ϵ, the single-qubit rotations must be synthesized to precision $\epsilon/\log(N)$, requiring a gate sequence of T-depth $\mathcal{O}(\log(\log(N)/\epsilon))$, and meanwhile, the SPF and FLAG gates incur $\mathcal{O}(\log(N))$ T-depth for a total T-depth of $\mathcal{O}(\log(N/\epsilon))$.

computing the "flag" bits $1 - f_{r|j}$ into ancilla registers, using them as a control to apply a controlled-V_r operation controlled on that bit, and then uncomputing the flag bits. The $N-1$ flag bits can be computed into $N-1$ ancillas using the unitary FLAG, depicted in Fig. 13 and defined by the equation

$$\mathrm{FLAG}\left(\sum_{j=0}^{N-1}\alpha_j\,|j\rangle_n\,|1\rangle_{N-1}\right) = \sum_{j=0}^{N-1}\alpha_j\,|j\rangle_n \bigotimes_{r=1}^{N-1}|1-f_{r|j}\rangle_1\,. \tag{33}$$

The circuit for FLAG is very similar to the SPF circuit, and, in fact, FLAG can be run in parallel with the $S^\dagger$ portion of the SPF gate. The full prerotated garbage-free state-preparation protocol, including uncomputation of garbage with flags, is depicted in Fig. 14.

To perform controlled-state preparation, one must first load one of N initial states $|\Theta^{(k)}\rangle_{N-1}$ into an ancilla register, depending on the setting of an n-qubit control register in the state $|k\rangle$. This is accomplished by the LOADF operation depicted in Fig. 5. In particular, we perform LOADF with $D = N-1$, which is equivalent to $N-1$ copies of LOADF to load each of the states $\{|\theta_r^{(k)}\rangle\}_{r=1}^{N-1}$. Each of these LOADF operations has its own flag qubit, which we initialize to $|1\rangle_1$, but all share the same n-qubit control and use their own Q_F-qubit ancilla space. Once $|\Theta^{(k)}\rangle_{N-1}$ is loaded, application of SPF yields the correct state with garbage. To disentangle the garbage and reset all ancillas to $|0\rangle$, we follow a three step

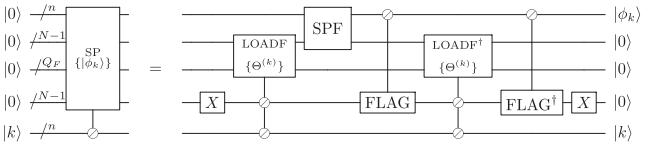

FIGURE 15. Controlled-state preparation circuit diagram for the prerotated approach to state preparation. For each of the N possible settings of the bottom register, a different n-qubit state $|\phi_k\rangle$ is prepared in the top register, which is accomplished with three subroutines. First, an $X^{\otimes(N-1)}$ gate sets the $N-1$ flag qubits (fourth register) to 1, and $N-1$ parallel copies of LOADF (see Fig. 5) load the state $|\Theta^{(k)}\rangle$ into the second register, conditioned on the last register being $|k\rangle$ (note that since all the flags are set to 1, the doubly controlled rotations that appear in the circuit for LOADF can be replaced with singly controlled rotations in this instance). Each of these copies uses the same n-qubit control, but its own ancilla space of Q_F qubits, so the total ancilla count is $Q_F = (N-1)(2N-1)$. Second, the state is computed into the first register using the SPF operation (Fig. 12), which leaves garbage in the second register. Third, the garbage is uncomputed by setting the flags for the angles that were injected into the circuit to 0 using the FLAG gate (see Fig. 13), running LOADF in reverse, and then returning the rest of the flags to $|0\rangle$ with FLAG†. The depth of LOADF is $\mathcal{O}(\log(N/\epsilon))$ and the depth of SPF and FLAG is $\mathcal{O}(\log(N))$ for a total depth of $\mathcal{O}(\log(N/\epsilon))$, where ϵ is the error on the state $|\phi_k\rangle$. The instances of controlled-state preparation that appear in the block-encoding unitary U_A of Fig. 1 can be accomplished with this circuit.

TABLE III Resource Counts for Two Approaches to State Preparation as Depicted in Figs. 8 and 14

Resource	Fixed-Precision	Pre-rotated
# Qubits	$(t+1)\cdot 2^n + n - t$	$4\cdot 2^n + n - 6$
T-Depth	$2tnR_y + 8n$	$3n + 4R_y - 3$
T-Count	$8(t+1)(2^n-1)$ $+ 2tnR_y - 8tn$	$(4R_y+16)\cdot 2^n$ $- 4R_y - 16n - 16$

The parameter R_y is the number of T gates needed to synthesize a single qubit rotation about the Y-axis by an arbitrary angle, and t is the number of bits precision to store the classical data, where both t and R_y scale as $\mathcal{O}(\log(1/\epsilon))$, where ϵ is the error on the state prepared by the protocol.

process: first, we apply FLAG, which flips flag bits from 1 to 0 in positions that were injected into the data; second, we run the reverse of LOADF, which sends $|\theta_r\rangle_1 \mapsto |0\rangle_1$ in all the positions where the control bit is 1, which is precisely the positions where it is not already $|0\rangle_1$; third, we apply FLAG again, followed by an X gate to return all flag bits to $|0\rangle_1$. This process is depicted in Fig. 15.

E. STATE PREPARATION RESOURCE ESTIMATES

Here, we summarize the non-Clifford resources required for state preparation. As with the QRAM estimates, we utilize the phase-incorrect controlled-swap gates for the fixed-precision version and the phase-correct version for the prerotated case. The phase-correct version requires additional ancilla qubits not shown in these circuits. Details of the controlled-swap circuits for both cases are summarized in Appendix C. The state preparation resource counts are provided in Table 3. For controlled-state preparation, one simply prepends the state preparation routine with a LOAD operation followed by a LOAD† operation (LOADF in the case of the prerotated approach) as in Figs. 11 and 15.

Our counts can be verified using the circuit diagrams in Figs. 8 and 14, along with the following additional observations, which reference gate decompositions from Appendix C.

1) Each of the n multiply controlled-R_y gates in Fig. 8 is accomplished by t singly controlled-R_y rotations in series, by the fixed rotation angles $\pi, \pi 2^{-1}, \pi 2^{-2}, \ldots, \pi 2^{-t+1}$. These are implemented with the construction from Fig. 19, which involves two applications of a single-qubit rotation. These single-qubit rotations are synthesized with a Clifford+T gate sequence with T-count denoted by the quantity R_y in the table. The total T-depth and T-count is thus $2nR_y$.
2) Each S_p gate with $p = 2, 3, \ldots, n, \pm$ in Figs. 8, 12, and 13 is a parallel controlled-swap gate with a single mutual control and many target pairs. For fixed-precision, each S_p is implemented with the construction from Fig. 23 along with the phase-incorrect decomposition in Fig. 22 yielding T-depth 4. The total number of controlled-swap gates that occur during S_2,...,S_n is $(2^n - n - 1)t + (2^n - 1)$, where the first term comes from shuffling the t-bit angle data and the second term comes from shuffling the sign bit data. For prerotated, the parallel controlled swaps are implemented with the construction from Fig. 20 (costing $(N-2)/2$ ancillas), where Toffolis are implemented via the depth-1 count-4 construction of [44] (costing another $(N-2)/2$ ancillas). The number of controlled swaps is $2^n - n - 1$.
3) In Fig. 14, the FLAG gate can be performed in parallel with the $S^\dagger$ gate (last gate) of Fig. 12. Note that each requires $N-2$ ancillas (see previous bullet). The depth of the controlled-swap portion of the circuit is $2(n-1)$ for SPF and $n-1$ for FLAG.

Finally, we remark here that one could also utilize the swap networks of the BB QRAM to perform the S_p gates in the state preparation routine, which could confer some amount of natural noise resilience. However, due to the interleaved R_y rotations, the parallelism that was previously exploited to ensure log-depth scaling is now broken, so the BB state preparation circuit can only achieve a minimum T-depth of $\mathcal{O}(n^2)$ depth. Furthermore, the constant factors are higher for qubit and T-count. For these reasons, we do not consider a BB style state preparation approach, and all of our resource estimates use the SS versions presented here.

IV. BLOCK-ENCODING RESOURCE ESTIMATES

A. OVERVIEW

We now have all the necessary ingredients to estimate the full resources required to block-encode a dense matrix of classical data using the circuit shown in Fig. 1. This is the portion labeled "block-encoding" in Fig. 2. A variety of choices can be made with respect to exactly how the matrix is block-encoded. These options are shown in the upper right side of Fig. 2. We outline the controlled version in the next section. We provide methods to implement the other options symmetric and q-Norm in Appendix A.

For the standard Frobenius encoding, the block-encoding procedure reduces to two applications of controlled-state preparation requiring both a QRAM-like data-loading operation and state preparation routine (see Figs. 11 and 15). We can make one optimization to reduce the resource count: note that the state $|\phi_k\rangle$ defined in (6) is independent of the control register k. Therefore, this state can be prepared with standard state preparation. We provide the resource counts for the fixed-precision case in Table 4 and prerotated block-encoding in Table 5 for the two versions of QRAM that we consider.

TABLE IV Block-Encoding Resource Requirements for Fixed-Precision Implementation

Resource	Select-Swap	Bucket-Brigade
# Qubits	$(t+1)\cdot 2^{n+\lambda} - t\cdot 2^{\lambda} + 3n - \lambda + 1$	$(2t+2)\cdot 2^{n+\lambda} - 2t\cdot 2^{\lambda} + 2^{\lambda} + 3n - 2$
T-Depth	$8\cdot 2^{n-\lambda} + 4tnR_y + 16n + 8\lambda - 8$	$(96\lambda - 72)\cdot 2^{n-\lambda} + 4R_y nt + 16n - 8$
T-Count	$8(t+1)(2^{n+\lambda} + 2^n) - 8t\cdot 2^{\lambda} + 8\cdot 2^{n-\lambda} + 4tnR_y - 16tn - 8t - 24$	$16(t+1)\cdot 2^{2n}(2 - 2^{-\lambda}) + 2^{n-\lambda}(-16\lambda + 16t - 24) - (16t - 48)\cdot 2^n + 4tnR_y - 16tn - 16t - 24$

Taking $\lambda = n$ yields the minimal depth circuit at the cost of needing $\mathcal{O}(N^2)$ T-count for both QRAM implementations. The SS-QRAM can achieve a T-count of $\mathcal{O}(N)$ by taking $\lambda = 0$.

TABLE V Block-Encoding Resource Requirements for Prerotated Method

Resource	Pre-rotated
# Qubits	$4\cdot 2^{2n} - 3\cdot 2^n + 2n - 1$
T-Depth	$10n + 8R_y - 4$
T-Count	$(4R_y + 32)2^{2n} - 24\cdot 2^n - 4R_y - 32n - 8$

To compare to Table IV, take $\lambda = n$. Loading the classical data in prerotated form with flag qubits allows us to achieve asymptotic depth scaling of $\mathcal{O}(\log N + R_y) \sim \mathcal{O}(\log(N/\epsilon))$.

B. FIXED-PRECISION RESOURCES

For the fixed-precision case, we leave the resource counts in terms of the parameter $\lambda \in \{0, 1, \ldots, n\}$ that allows one to trade T-depth for circuit width and overall T-count. For the SS-QRAM, the minimal T-count is achieved by choosing $\lambda = 0$, which yields a T-count and T-depth proportional to $\mathcal{O}(N)$ on $\mathcal{O}(N)$ total qubits. The minimal T-depth of $\mathcal{O}(\log(N))$ is achieved when $\lambda = n$ at the expense of requiring a T-count of $\mathcal{O}(N^2)$.

The fixed-precision BB approach cannot achieve the same scaling with respect to T-count; in all cases the count is at least $\Omega(N^2)$. The minimal-depth case can achieve the same asymptotic scaling as SS, but we note that the constant factors for all resources are higher. Whether this approach can achieve an overall physical resource reduction will depend upon the quantum architecture, error correcting code, and

IEEE Transactions on Quantum Engineering

error requirements, which are all beyond the scope of this article.

C. PREROTATED RESOURCES

For the prerotated case, we only consider the minimal-depth circuit, that is, we make no attempt to trade width for depth. Note that any manifestation of the prerotated idea would need to satisfy a T-count lower bound of $\Omega(N^2)$ due to the need to have controlled-Ry rotations for all $N(N-1)$ angles somewhere in the circuit. However, the benefits of the prerotated approach can be seen by both an improvement in asymptotic scaling and constant factor improvements to both T-depth and qubit count. This technique allows us to achieve T-depth of $\mathcal{O}(\log N + R_y) \sim \mathcal{O}(\log(N/\epsilon))$, which we contrast with the fixed-precision approach that scales as $\mathcal{O}(R_y t \log N) \sim \mathcal{O}(\log N \log^2(1/\epsilon))$.

The number of qubits required for this approach is approximately four per classical matrix entry (not counting additional qubits needed for routing operations). One qubit is required for the data and one for the flag, and an additional two ancilla are needed to perform the parallel controlled-swap operations (see Fig. 20). One could exploit the fact that the parallel controlled-swap gate can use dirty qubits for the ancilla to reduce this to just a single extra ancilla for the T-depth one Toffoli gate [46]. However, the first round of controlled-swap gates in the SS-QRAM circuit operates in parallel across all QRAM data registers, so there are no available qubits to do this. One could split the first set of controlled-swap gates into two rounds to utilize the other data qubits as dirty ancilla, but we choose the shortest possible depth approach and counted the cost of the additional ancilla qubit in our resource analysis.

Note that since all the flags are set to 1 at the beginning of the controlled-state preparation protocol, the Toffolis that appear within the gate V of LOADF in Fig. 5 can be replaced with Clifford CNOTs, saving T-depth 2 and T-count $2N(N-1)$.

D. CONTROLLED BLOCK-ENCODINGS

A controlled block-encoding is useful in certain applications, for example, in solving linear systems of equations [4]. In particular, we wish to implement the unitary

$$CU_A = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U_A \tag{34}$$

where U_A is the standard form of the block-encoded matrix A given in (1). One possible implementation is to add zeros to the QRAM and just select these values if the appropriate control bit is 0, but this approach is highly inefficient in qubit count for our assumed case of N being an exact power of two since the number of qubits in the QRAM must be doubled. Instead, we simply select zeros by adding a single binary tree of all zeros, which will be selected using a single register controlled swap to replace the loaded register if the control bit is one, as shown in Fig. 16. This method requires one extra qubit for the control, D extra qubits for the $|0\rangle$ state in QRAM, and one extra controlled swap between D qubit registers.

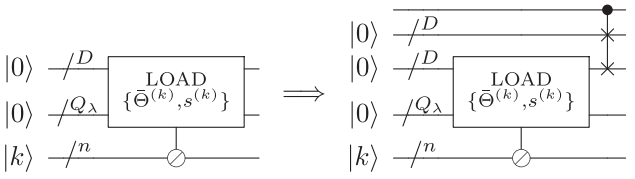

FIGURE 16. Possible modification to the LOAD operation that allows a controlled-block-encoding $CU_A = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U_A$ to be efficiently performed. The LOAD operation can be either LOAD_{ss} or LOAD_{bb} from Section II, and brings D bits of classical data into the third register, where $D = (N-1)t + N$. Controlled on the first register, the data are swapped into the second register, which acts as the input to state preparation, as in Fig. 11. A similar idea could be implemented for the LOADF operation with respect to Fig. 15. If the control is $|0\rangle$, the input to state preparation is the trivial state. The cost of this construction is D ancilla qubits and D controlled-swap gates with a common control, which can be performed in $\mathcal{O}(1)$ T-depth and $\mathcal{O}(D)$ T-count (in cases that the LOAD operation does not produce garbage, the LOAD ancillas could be reused, in which case no additional ancillas are needed for controlled block encoding). Adding additional controls onto the controlled-swap gate yields a multiply controlled block-encoding with an arbitrary number of controls, without any additional ancillas.

V. FINITE-PRECISION ERROR ANALYSIS

A. OVERVIEW

The unitary implemented by our circuit is $\tilde{U}_A$, which is different from the exact block-encoding unitary U_A in two respects: 1) the rounding error from the representation of the angles θ using t-qubit registers and 2) the error from the gate synthesis of the R_y rotations. Both 1) and 2) affect the fixed-precision block-encoding; only 2) affects the prerotated block-encoding.

B. ROUNDING ERROR

For the fixed-precision case, rounding errors cause the ideal angles $\theta \in [0, \pi]$ to be approximated by the angle $\tilde{\theta}$ that is the nearest exact multiple of $2\pi/2^t$, i.e., $|\theta - \tilde{\theta}| \leq \pi 2^{-t}$. We can, therefore, represent $1 - (\tilde{\theta}/2\pi)$ exactly in binary with t bits $(\theta_1, \ldots, \theta_{t-1})$, where $\theta_i \in \{0, 1\}$ and

$$1 - \frac{\tilde{\theta}}{2\pi} = 2^{-\theta_1} 2^{-2\theta_2} 2^{-3\theta_3} \ldots 2^{-t\theta_t}. \tag{35}$$

The rotation $R_y(\theta)$ sends $|0\rangle \mapsto \cos(\theta/2)|0\rangle + \sin(\theta/2)|1\rangle$; we need only cover the space $\theta \in [0, \pi]$, since we may assume the matrix entries are all positive (signs are applied later), and hence, $\sin(\theta/2) \in [0, 1]$ is sufficient. Note that $R_y(\eta)$ has eigenvalues $e^{\pm i\eta/2}$ and hence $||R_y(\eta) - I|| \leq |e^{i\eta/2} - 1| \leq \eta/2$, which implies that

$$\begin{aligned} ||R_y(\theta) - R_y(\tilde{\theta})|| &= ||R_y(\theta) R_y^\dagger(\tilde{\theta}) - I|| \\ &= ||R_y(\theta - \tilde{\theta}) - I|| \\ &\leq |\theta - \tilde{\theta}|/2 \leq \pi 2^{-t-1}. \end{aligned} \tag{36}$$

We now argue that the above-mentioned implies that the *controlled*-rotations are close to their rounded versions. The controlled rotations in the circuit for U_A perform a different

rotation angle, depending on the setting of several control qubits. Suppose that, for some integer p, we have a collection of 2^p unitaries V_i and approximations $\tilde{V}_i$ such that $||V_i - \tilde{V}_i|| \leq \delta$ for all $i \in [2^p]$. Let CV and $C\tilde{V}$ be the operations that, controlled on a p-qubit register being in the state i, perform the operations V_i and $\tilde{V}_i$, respectively. It is then easy to verify that $||CV - C\tilde{V}|| \leq \delta$, as CV and $C\tilde{V}$ are each block diagonal matrices with the same block structure. Hence, as long as all angles are correct up to error $\pi 2^{-t-1}$, the controlled-$R_y(\theta)$ operation in the circuit is $(\pi 2^{-t-1})$-close to controlled-$R_y(\tilde{\theta})$.

Let $\tilde{U}'_A$ denote the unitary for which all controlled-$R_y(\theta)$ gates are replaced by an exact implementation of controlled-$R_y(\tilde{\theta})$. As the circuit has $2\log(N)$ controlled rotations ($\log(N)$ each for $U_R^\dagger$ and U_L), by the triangle inequality, we have that

$$\begin{aligned} ||U_A - \tilde{U}'_A|| &\leq 2\log(N)||R_y(\theta) - R_y(\tilde{\theta})|| \\ &\leq \pi \log(N) 2^{-t}. \end{aligned} \tag{37}$$

C. GATE SYNTHESIS ERROR

In the fixed-precision approach, the unitary $\tilde{U}'_A$ exactly implements controlled-$R_y(\tilde{\theta})$ by loading the bits $(\theta_1, \ldots, \theta_t)$ of $\tilde{\theta}$ into t ancilla registers, and then exactly performing controlled-$R_y(\pi 2^{-j+1})$ operations (with one control qubit) for $j = 1, \ldots, t$ with these ancilla registers acting as the controls. As seen in Fig. 19, a controlled-$R_y(\pi 2^{-j+1})$ is accomplished by decomposition into two CNOTs and two $R_y(\pi 2^{-j})$ operations. The actual circuit $\tilde{U}_A$ differs from $\tilde{U}'_A$ only in that these $R_y(\pi 2^{-j})$ operations are performed approximately using a decomposition of the $R_y(\pi 2^{-j})$ gate into Clifford+T. Denote the unitary enacted by this decomposition by $\tilde{R}_y(\pi 2^{-j})$. The decomposition error [45] is then bounded as

$$||R_y\left(\pi 2^{-j}\right) - \tilde{R}_y\left(\pi 2^{-j}\right)|| \leq \delta_{\text{decomp}} \tag{38}$$

as long as we choose a gate sequence with T-count (note T-depth $= T$-count for a single-qubit operation) at least equal to R_y with

$$R_y = 3\log(1/\delta_{\text{decomp}}) + \mathcal{O}(\log(\log(1/\delta_{\text{decomp}}))). \tag{39}$$

Again using the triangle inequality, we find that replacing the $4t\log(N)$ appearances of a $R_y(\pi 2^{-j})$ gate with the approximate $\tilde{R}_y(\pi 2^{-j})$, we incur error bounded as

$$||\tilde{U}_A - \tilde{U}'_A|| \leq 4t\log(N)\delta_{\text{decomp}}. \tag{40}$$

In the prerotated approach, there are $2N - 2$ controlled-R_y rotations (actually, they are doubly controlled R_y rotations), but nearly all of them (the ones that do not get injected) are exactly undone. Moreover, by choosing a gate decomposition of $R_y(\theta)$ of the form $H_\theta^* H_\theta$ for some gate sequence H_θ (where H_θ^* denotes complex conjugate), the construction of Fig. 19 with H_θ and $H_\theta^\dagger$ in place of $R_y(\theta/2)$ and $R_y(-\theta/2)$ guarantees that the action on the target is *exactly* trivial when the control is set to $|0\rangle$. Ultimately, the error on the state prepared by the protocol is exactly the same as it would be in the fixed-precision method using the decomposition $H_\theta^* H_\theta$ for all the controlled-rotations (rather than a product of t gate decompositions). As there are $2\log(N)$ rotations, and thus, $4\log(N)$ applications of some sequence H_θ, the overall error is $4\log(N)\delta_{\text{decomp}}$. Note that there is no rounding error for this approach.

D. OVERALL BLOCK-ENCODING ERROR

For the fixed-precision approach, from (37) and (40) and the triangle inequality, we have that

$$||U_A - \tilde{U}_A|| \leq 4t\log(N)\delta_{\text{decomp}} + \pi\log(N)2^{-t}. \tag{41}$$

Further, note that by the definition of block-encoding and submultiplicativity of the spectral norm, we have

$$\begin{aligned} &\|A - \alpha(\langle 0| \otimes I)\tilde{U}_A(|0\rangle \otimes I)\| \\ &\quad = \alpha\|(\langle 0| \otimes I)(U_A - \tilde{U}_A)(|0\rangle \otimes I)\| \\ &\quad = \alpha\|(\langle 0| \otimes I)(U_A - \tilde{U}_A)(|0\rangle \otimes I)\| \\ &\quad \leq \alpha\|\langle 0| \otimes I\| \cdot \|U_A - \tilde{U}_A\| \cdot \||0\rangle \otimes I\| \\ &\quad = \alpha\|U_A - \tilde{U}_A\| \\ &\quad \leq 4t\alpha\log(N)\delta_{\text{decomp}} + \pi\alpha\log(N)2^{-t}. \end{aligned} \tag{42}$$

If we wish for this error to be smaller than ϵ, it suffices to choose $t = \log(\alpha\pi/\epsilon) + \log(\log(N)) + 1$, and $\delta_{\text{decomp}} = \epsilon/(8t\alpha\log(N))$, which is achieved with

$$\begin{aligned} R_y = \lceil &3\log(\alpha/\epsilon) + 3\log(\log(N)) + 9 \\ &+ \mathcal{O}(\log(\log(\alpha/\epsilon))) + \mathcal{O}(\log(\log(\log(N))))\rceil \end{aligned} \tag{43}$$

$$t = \lceil \log(\alpha/\epsilon) + \log(\pi) + \log(\log(N)) + 1\rceil. \tag{44}$$

These equations should be substituted in the fixed-precision resource counts, whenever one encounters them, if one wishes to have resources in terms of the final block-encoding error rate ϵ.

For the prerotated approach, the same analysis gives

$$\|A - \alpha(\langle 0| \otimes I)\tilde{U}_A(|0\rangle \otimes I)\| \leq 4\alpha\log(N)\delta_{\text{decomp}} \tag{45}$$

meaning it suffices to choose $\delta_{\text{decomp}} = \epsilon/(4\alpha\log(N))$, which is achieved with a gate sequence of length

$$\begin{aligned} R_y = \lceil &3\log(\alpha/\epsilon) + 3\log(\log(N)) + 6 \\ &+ \mathcal{O}(\log(\log(\alpha/\epsilon))) + \mathcal{O}(\log(\log(\log(N))))\rceil. \end{aligned} \tag{46}$$

Our numerical estimates use these formulas for estimating exact resources, but we ignore the $\mathcal{O}(\cdot)$ terms which are all doubly logarithmic or smaller.

VI. CONCLUSION

It is well known that loading classical data into a quantum computer is a challenging problem. However, for many problems of practical interest, one assumes access to such classical data without necessarily considering the full cost to encode the data in the quantum computer. Our results

provide concrete resource counts and system sizes required to perform this task for two different QRAM models.

Combining QRAM with a state preparation routine, we provide detailed circuit descriptions and resource estimates for a commonly used algorithmic primitive: a block-encoding. Our modular implementation also allows one the freedom to consider resource counts for different QRAM models that allow for differing optimizations. We provide two such choices: one that minimizes T-count or T-depth and the other that minimizes the impact of noise. Our results address the practical feasibility of quantum algorithms that require large amounts of classical data, and we note that in fault-tolerant implementations of these algorithms, QRAM implementations can be seen as a bottleneck.

The details of our circuits also elucidate the ingredients that would be necessary in a circuit architecture optimized for block-encoding classical data. The QRAM circuits we describe use a large number of controlled-swap gates that we assume can be applied in parallel. Realizing such parallelism requires applying T gates across many qubits at once and to achieve polylog(N) depth (which is necessary in any scenario where an exponential speedup is sought), our constructions require $\mathcal{O}(N^2)$ parallel T gates. Even if a processor with so many qubits was available, it could be challenging to implement these parallel operations at large N due to the overhead required for magic-state-distillation and decoding latencies, which could significantly decrease the rate at which layers can be applied. Furthermore, we assume that fanout-CNOT gates with arbitrarily long range can be applied in one time step. In surface-code lattice-surgery architectures, this parallelism is possible, but it requires additional communication qubits for performing the required lattice-surgery operations, thereby increasing the qubit resources beyond what we have considered here. Therefore, although we do not prove any rigorous lower bounds on block-encoding, our resource estimates might be viewed as maximally optimistic for block-encoding of dense classical data, and they would increase as architectural constraints are added.

APPENDIX A
ALTERNATIVE BLOCK-ENCODING STRATEGIES
A. BLOCK-ENCODING OFF-DIAGONAL HERMITIAN MATRICES

In cases where the matrix A is not square or when one requires a Hermitian block-encoding, rather than block-encode A directly, one can instead block-encode the matrix

$$\bar{A} = \begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix} \quad \text{where} \quad \bar{A} \in \mathbb{R}^{(M+N)\times(M+N)}. \tag{47}$$

Block-encoding $\bar{A}$ ensures that the matrix is Hermitian and square at the cost of one extra qubit.

We assume that $M \geq N$ without loss of generality (we could block encode the transpose to ensure this, if necessary). For the Frobenius encoding, we also take $M = 2^m$ and $N = 2^n$, which can be enforced by padding the matrix. This implies that $\ell = m + 1$. We now define the $2^{2\ell}$-dimensional operators U_L and U_R as the following:

$$\begin{aligned} U_R\,|i\rangle_\ell\,|0\rangle_\ell &= |\psi_i\rangle_{2\ell} \\ U_L\,|i\rangle_\ell\,|0\rangle_\ell &= |\phi_i\rangle_{2\ell} \end{aligned} \tag{48}$$

where

$$|\psi_j\rangle = \sum_{k\in[N]} \frac{A_{j,k}}{||A_{j,\cdot}||}\,|j, M+k\rangle \tag{49a}$$

$$|\psi_{M+k}\rangle = \sum_{j\in[M]} \frac{||A_{j,\cdot}||}{||A||_F}\,|M+k, j\rangle \tag{49b}$$

$$|\phi_j\rangle = \sum_{k\in[N]} \frac{A_{j,k}}{||A_{j,\cdot}||}\,|M+k, j\rangle \tag{50a}$$

$$|\phi_{M+k}\rangle = \sum_{j\in[M]} \frac{||A_{j,\cdot}||}{||A||_F}\,|j, M+k\rangle \tag{50b}$$

with $j \in [M]$ and $k \in [N]$. We use the notation $[M] = [1, 2, \ldots, M]$, $[N] = [1, 2, \ldots, N]$, and $||A_{j,\cdot}||$ denotes the norm of the jth row of A such that $\sum_{j\in[M]} ||A_{j,\cdot}||^2 = ||A||_F^2$. From these definitions, it is straightforward to show that

$$\begin{aligned} \langle\psi_j|\phi_{j'}\rangle &= \langle\psi_{M+k}|\phi_{M+k'}\rangle = 0 \\ \langle\psi_j|\phi_{M+k}\rangle &= \frac{A_{j,k}}{||A||_F}, \quad \langle\psi_{M+k}|\phi_j\rangle = \frac{A^T_{k,j}}{||A||_F} \end{aligned} \tag{51}$$

and

$$\begin{aligned} &(\langle 0|_\ell \otimes I_{M+N}) U_R^\dagger U_L (I_{M+N} \otimes |0\rangle_\ell) \\ &= \sum_{k\in M+[N]} \sum_{j\in[M]} |k\rangle\langle j| \frac{A^T_{k,j}}{||A||_F} \\ &\quad + \sum_{j\in[M]} \sum_{k\in M+[N]} |j\rangle\langle k| \frac{A_{j,k}}{||A||_F} \end{aligned} \tag{52}$$

where the notation $M + [N] = [M + 1, \ldots M + N]$, and I_{M+N} is the projector onto the space spanned by basis states $|j\rangle$ with $j \in [M + N]$. This shows that the unitary $U_R^\dagger U_L$ is the block encoding of (47) as desired.

Both U_R and U_L are controlled-state preparation operators and can be implemented as described in the main text. The upshot of this construction, compared to a direct application of the construction in the main text to the off-diagonal Hermitian matrix, is that the normalization factor for the block-encoding is $||A||_F$ rather than $2||A||_F$.

B. Q-NORM BLOCK-ENCODING

For some algorithms, it might be advantageous to use a slightly modified block-encoding that we call the q-norm block-encoding. This is a $(\mu_p(A), \lceil\log(N+1)\rceil, \epsilon)$ block-encoding of A where

$$U^{(q)} = (U_R^{(q)})^\dagger U_L^{(q)} = \begin{pmatrix} A/\mu_p(A) & \cdot \\ \cdot & \cdot \end{pmatrix} \tag{53}$$

such that

$$||A - \mu_p(A)(\langle 0|_\ell \otimes I_{N+1})U(|0\rangle_\ell \otimes I_{N+1})|| \leq \epsilon \tag{54}$$

with $\mu_p(A) = \sqrt{S_{2p}(A)S_{2(1-p)}(A^T)}$, $p \in [0, 1]$, and $S_q(A) = \max_j ||A_{j,\cdot}||_q^q$ is the qth power of the maximum q-norm of any row. As in the main text, we take $N = 2^n$, which can be enforced by padding the matrix, if necessary. This restriction ensures that certain required rotations can be implemented without the need for basis state permutations or alternative encodings [47]. Just as shown in (48), we define two operators $U_R^{(q)}$ and $U_L^{(q)}$ but now with the superscript (q) to denote that this is the q-norm version. These are $2^{2\ell+1}$ dimensional operators that prepare the states $U_R^{(q)}|i\rangle_\ell|0\rangle_{\ell+1} = |\psi_i^{(q)}\rangle_{2\ell+1}$ and $U_L^{(q)}|i\rangle_\ell|0\rangle_{\ell+1} = |\phi_i^{(q)}\rangle_{2\ell+1}$ where

$$|\psi_j^{(q)}\rangle = \sum_{k\in[N]} \frac{\text{sgn}(A_{j,k})|A_{j,k}|^p}{\sqrt{||A_{j,\cdot}||_{2p}}} \times \left[\cos\chi_j|j,k\rangle + \sin\chi_j|j,N+k\rangle\right] \tag{55}$$

$$|\phi_k^{(q)}\rangle = \sum_{j\in[N]} \frac{|A_{j,\cdot}|^{1-p}}{\sqrt{||A_{\cdot,k}||_{2(p-1)}}} \times [\cos\chi_k|j,k\rangle + \sin\chi_k|N+j,k\rangle] \tag{56}$$

with $j \in [N]$ and $k \in [N]$ for the control registers as before, sgn the signum function, and

$$\cos\chi_j = \sqrt{\frac{||A_{j,\cdot}||_{2p}}{S_{2p}(A)}}$$
$$\cos\chi_k = \sqrt{\frac{||A_{\cdot,k}||_{2(p-1)}}{S_{2(1-p)}(A^T)}}. \tag{57}$$

From these definitions, one can show that

$$(\langle 0|_{\ell+1} \otimes I_N)(U_R^{(q)})^\dagger U_L^{(q)}(I_N \otimes |0\rangle_{\ell+1}) = \sum_{j\in[N]}\sum_{k\in[N]} |j\rangle\langle k| \frac{A_{j,k}}{\mu_p(A)}. \tag{58}$$

As with the Frobenius case shown in the main text, the block-encoding procedure is reduced to showing how to create circuits to prepare the states given in (55). With the encoding presented here, this task is nearly identical to the Frobenius case with just minor changes. Specifically, one must perform a controlled-rotation on an ancilla qubit by the angle χ_j controlled by the index j, which requires LOAD and its adjoint, as well as a controlled-rotation. This controlled-rotation can be performed either in fixed precision or using prerotated states with flags. In addition, both controlled-state preparations require a complete LOAD and LOAD† operation (recall that for the Frobenius case, the second state preparation was independent of the control register).

C. SYMMETRIZED Q-NORM BLOCK-ENCODING

We can also perform a symmetrized block-encoding for the q-norm. A similar block-encoding was given in [1] and [2] but with different state definitions. We prefer the block-encodings as follows, as they allow for simpler circuit implementations. We obtain a $(\mu_p(A), \lceil\log(M+N+1)\rceil, \epsilon)$ block-encoding of $\bar{A}$ such that

$$||\bar{A} - \mu_p(\bar{A})(\langle 0|_\ell \otimes I_{M+N+1})U(|0\rangle_\ell \otimes I_{M+N+1})|| \leq \epsilon \tag{59}$$

with $\mu_p(A) = \sqrt{S_{2p}(A)S_{2(1-p)}(A^T)}$, $p \in [0, 1]$, and $S_q(A) = \max_j ||A_{j,\cdot}||_q^q$ is the qth power of the maximum q-norm of any row. The states to prepare in this case are

$$|\psi_j^{(q)}\rangle = \sum_{k\in[N]} \frac{\text{sgn}(A_{j,k})|A_{j,k}|^p}{\sqrt{||A_{j,\cdot}||_{2p}}} \times \left[\cos\chi_j|j,M+k\rangle + \sin\chi_j|j,3M+k\rangle\right] \tag{60a}$$

$$|\psi_{M+k}^{(q)}\rangle = \sum_{j\in[M]} \frac{|A_{j,k}|^{1-p}}{\sqrt{||A_{\cdot,k}||_{2(1-p)}}} \times \left[\cos\chi_{M+k}|M+k,j\rangle + \sin\chi_{M+k}|M+k,2M+j\rangle\right] \tag{60b}$$

$$|\phi_j^{(q)}\rangle = \sum_{k\in[N]} \frac{\text{sgn}(A_{j,k})|A_{j,k}|^p}{\sqrt{||A_{j,\cdot}||_{2p}}} \times \left[\cos\chi_j|M+k,j\rangle + \sin\chi_j|3M+k,j\rangle\right] \tag{61a}$$

$$|\phi_{M+k}^{(q)}\rangle = \sum_{j\in[M]} \frac{|A_{j,\cdot}|^{1-p}}{\sqrt{||A_{\cdot,k}||_{2(p-1)}}} \times \left[\cos\chi_{M+k}|j,M+k\rangle + \sin\chi_{M+k}|2M+j,M+k\rangle\right] \tag{61b}$$

with $j \in [M]$ and $k \in [N]$ for the control registers. From these states, one can show

$$(\langle 0|_{\ell+1} \otimes I_{M+N})(U_R^{(q)})^\dagger U_L^{(q)}(I_{M+N} \otimes |0\rangle_{\ell+1}) = \sum_{k\in M+[N]}\sum_{j\in[M]} |k\rangle\langle j| \frac{A_{k,j}^T}{\mu_p(A)} + \sum_{j\in[M]}\sum_{k\in M+[N]} |j\rangle\langle k| \frac{A_{j,k}}{\mu_p(A)}. \tag{62}$$

As noted above, these state definitions vary slightly from those given in [1] and [2], as the subspaces rotated by the angle χ_i are different. Our description allows for a much simpler circuit implementation, as one just needs to implement a controlled-rotation followed by the state preparation procedure outlined in the main text.

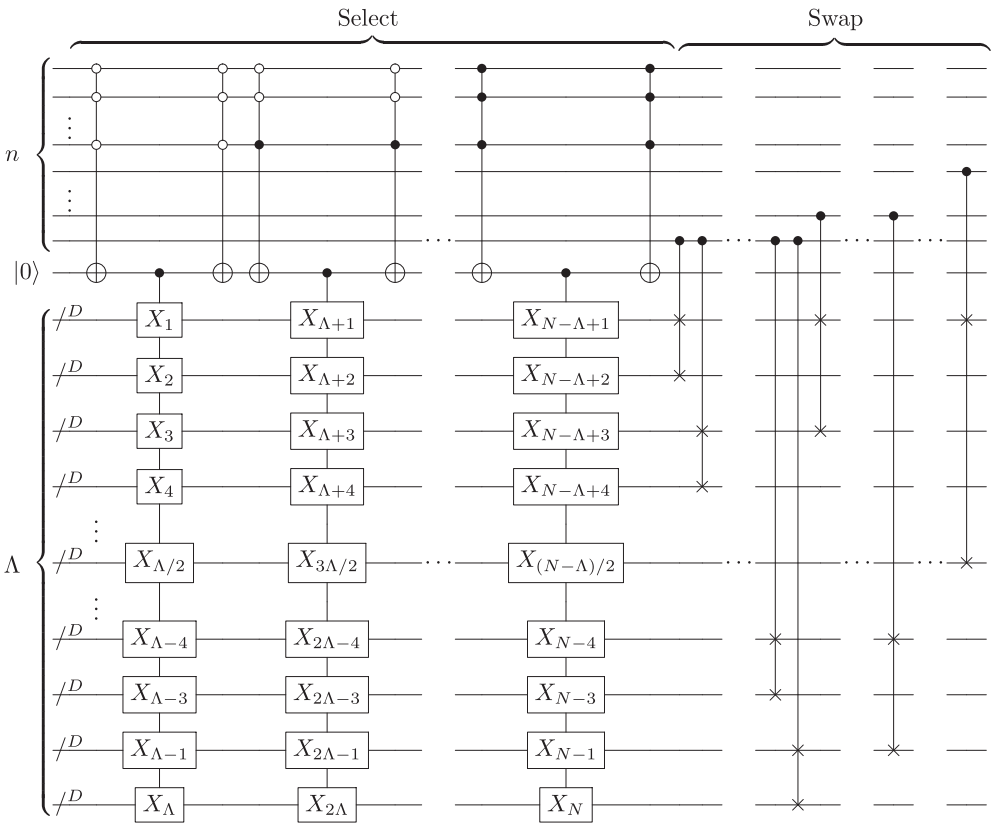

FIGURE 17. SS implementation with variable circuit depth and width. The $n = s + \lambda$ qubit control register is divided into s qubits for the select control and λ qubits for the swap control. The select circuit requires 2^s fanout CNOT gates as well as 2^{s+1} multicontrolled Toffoli gates that can be optimally decomposed using the unary iteration procedure, which requires s additional ancillas (not shown) [41]. The swap circuit requires $\Lambda - 1$ controlled-swap gates between D-qubit registers.

APPENDIX B
QRAM DETAILS

A. SELECT-SWAP

The SS circuit implementation [19] is shown in Fig. 17. The first portion, labeled "Select," utilizes a unary iteration [41] to select from Λ out of N possible classical data-loading operations of D-qubit registers. For fixed-precision, this requires a fanout CNOTgate to load the classical data, which is Clifford and can be implemented efficiently in surface-code architectures [23]. For prerotated gates, this presents an issue, as we must control on both the select control register as well as the flag register, requiring one to always use $\mathcal{O}(ND)$ Toffoli gates. For this reason, we only use the prerotated gates in the minimal-depth cases.

The "Swap" portion uses the remaining qubits in the control register to swap the desired state into the top register. This circuit realizes the data-loading with garbage query

$$\mathrm{LOAD}_{\mathrm{ss}}\left(\sum_{j=0}^{2^n-1}\alpha_j\,|j\rangle_n\,|0\rangle_D\,|0\rangle_{(\Lambda-1)D}\right)$$
$$=\sum_{j=0}^{2^n-1}\alpha_j\,|j\rangle_n\,|b_j\rangle_D\,|g_j\rangle_{(\Lambda-1)D} \tag{63}$$

as discussed in (11) in the main text. The SS architecture enables one to achieve an optimal T-count of $\mathcal{O}(N)$ by taking $\Lambda = \mathcal{O}(\sqrt{N})$ [19].

B. BUCKET-BRIGADE

The original formulation [14] of the BB idea assumed access to qutrits. Most modern quantum hardware implements qubits, so we analyze the qubit version of the BB-QRAM, as laid out and analyzed in [18]. While it is difficult to draw an m address qubit version of the BB-QRAM, the four-qubit version is sufficient to understand how the circuit can be generalized. This circuit is shown in Fig. 18. The address qubits are routed from the address space into individual routers. The bottom qubit in the address space is routed to the level-0 or $L0$ router. The next qubit is routed to the $L1$ router in either the bottom or top branch, depending on whether the bottom address qubit is a $|0\rangle$ or $|1\rangle$, respectively. The rest of the address qubits follow in the same manner, being sent to the appropriate router depending upon the value of the preceding address qubits. Once all address qubits have been routed, the circuit swaps the bus register (top of figure) into the appropriate memory location, where Z (identity) gates are applied to qubits in the register if the corresponding classical bit is a 1 (0). The routing portion of the circuit is then reversed to

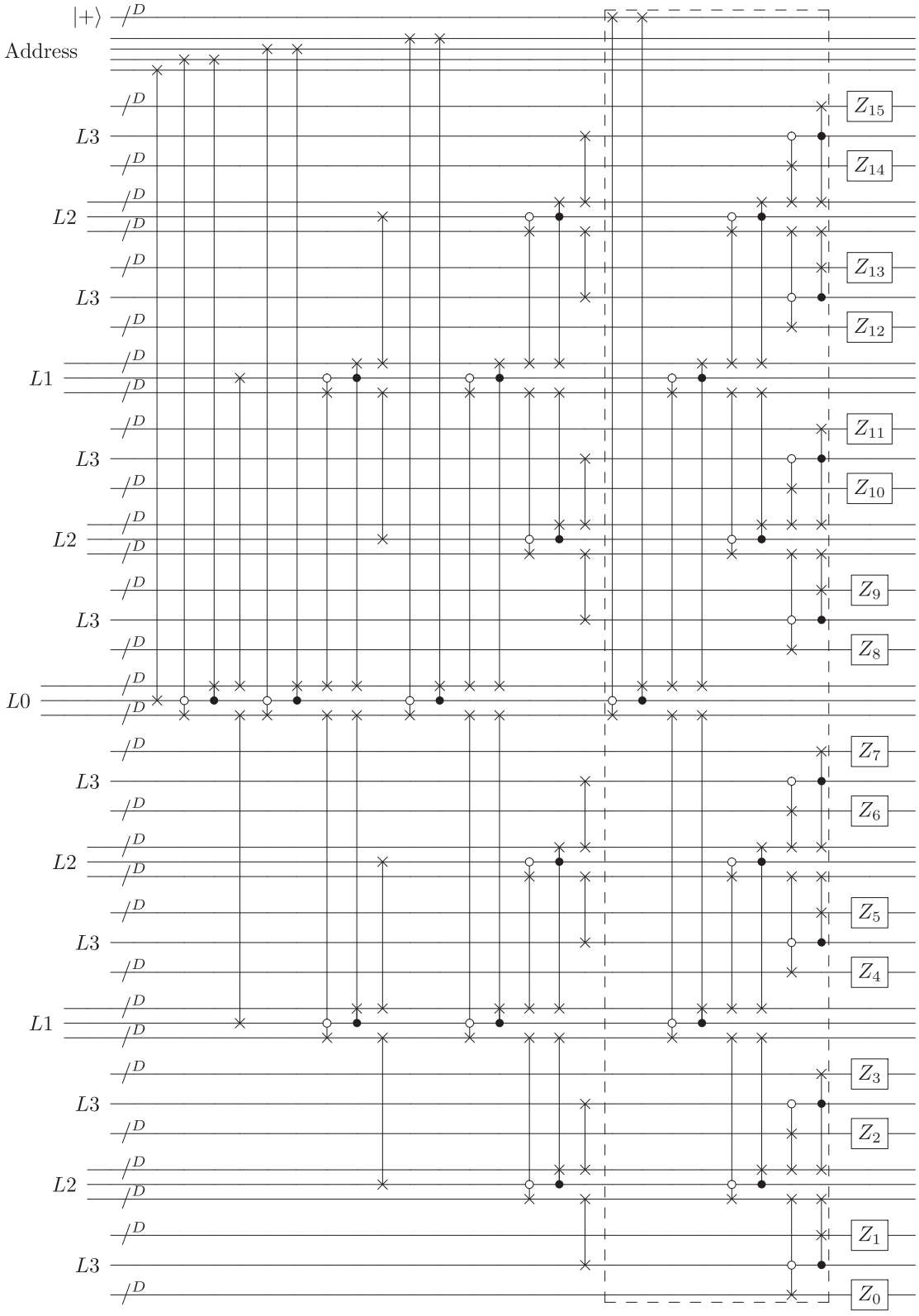

FIGURE 18. BB-QRAM with an address space of four bits shown here since it can help visualize the general case. Note that when single qubits are swapped into a D-qubit register (e.g., the first gate in the circuit), it is assumed that they occupy the first position. As discussed in [18, Fig. 10], additional parallelization is possible (but hard to depict in the drawing), which reduces the depth of a QRAM with n address qubits from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$. An example of this effect is seen in the figure by noting that the region in the dashed box can be shifted left by three layers. The total T-count of the n-qubit version of this circuit is $16(D+1)(2^n-1)$ over T-depth of $48(n-1)$ assuming that controlled-swap gates are implemented with the phase-incorrect construction of Fig. 22. The total number of qubits is $n+D+(2D+1)(2^n-1)-1$ (where we save a qubit by omitting the first swap gate and identifying the first address qubit directly with the $L0$ router setting).

IEEE Transactions on Quantum Engineering

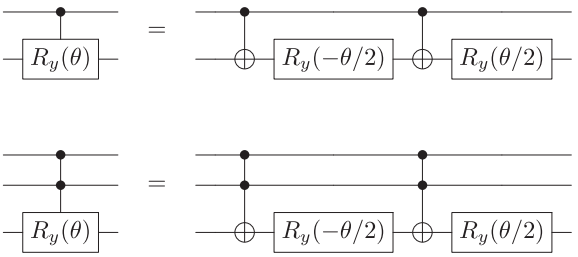

FIGURE 19. Decomposition of controlled-R_y rotation and controlled-controlled-R_y rotation used for resource counts. Since $R_y(\theta)$ is equivalent to $R_y(\theta + 2\pi)$ up to a global phase, controlled-$R_y(\theta)$ is equivalent to controlled-$R_y(\theta + 2\pi)$ up to a controlled-phase, i.e., a Z gate. Thus, for angles $\theta \in [\pi, 2\pi]$, a Z gate should additionally be applied to the first qubit in the top diagram, and a CZ gate should be applied between the top two qubits of the bottom diagram.

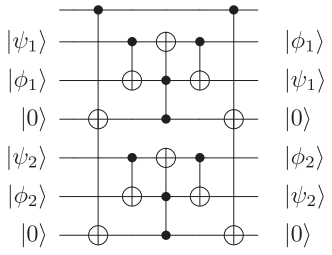

FIGURE 20. Phase-correct parallel controlled swap between two qubit registers in arbitrary states with a single control use for prerotated resource estimates. Two clean ancillas are required (see [19] for how dirty ancillas can be used at the expense of two additional Toffoli gates). With an additional clean ancilla (not shown), the Toffoli gates can be constructed with a T-depth of 1 and a T-count of 4 [44], [46]. The output state is shown assuming the input control is $|1\rangle$. If the control is $|0\rangle$, the states are not swapped. To perform parallel Toffolis (instead of controlled swaps), simply omit the CNOT gates in the second and fourth layers.

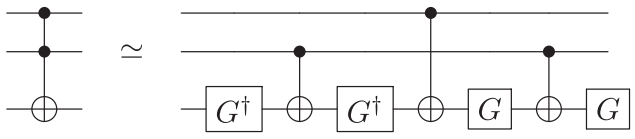

FIGURE 21. Toffoli gate decomposed into a T-depth 4 circuit used for fixed-precision resource estimates. The gate $G = S^\dagger HTHS$. The decomposition is exact up to a phase that takes $|100\rangle \to -|100\rangle$.

swap the bus and address qubits back to their initial locations (not shown). This formulation assumes that the router qubits are initialized to the $|0\rangle$ state and the bus register in initialized to the $|+\rangle_D$ state. Similarly to the SS-QRAM, a version that can utilize routing qubits in an arbitrary initial state is given in [18] at the cost of additional queries. We do not consider that case here.

A naive counting of the circuit depth of the BB-QRAM leads to a depth complexity of $\mathcal{O}(m^2)$ controlled swaps. However, this complexity does not take into account additional parallelism that can be exploited. The dashed box in Fig. 18 can be shifted to the left such that the first two controlled swaps can be done in parallel with the eight controlled

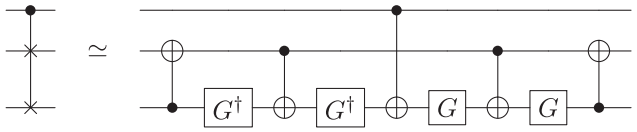

FIGURE 22. Controlled-swap gate decomposed into a T-depth and T-count 4 circuit used for fixed-precision resource estimates. The gate $G = S^\dagger HTHS$. The decomposition is exact up to a phase that takes $|100\rangle \to -|100\rangle$.

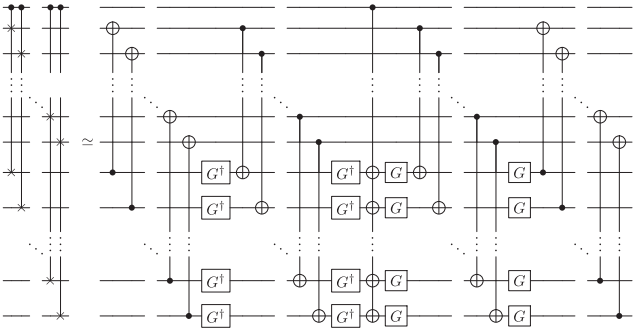

FIGURE 23. Controlled swap between multiqubit registers decomposed into a set of T gates, a fanout CNOT, and $G = S^\dagger HTHS$ gates [19], [48]. The G (or equivalently T) gates and CNOT gates can all be implemented in parallel. The decomposition on the right adds a phase to certain basis states, so it can only be used in situations where that phase is irrelevant [48]. We only use this version for fixed-precision resource estimates. The entire circuit has a T-depth of 4. The T-count is $4t$, where t is the size of the registers being swapped.

swaps being used to route qubits from $L1$ to $L2$. This parallelism persists as the address space increases and, as a result, each additional address qubit will only add a total depth of six controlled swaps to the circuit, leading to a total depth of $\mathcal{O}(m)$ controlled-swap gates for the circuit.

APPENDIX C
GATE DECOMPOSITIONS

For completeness, we show the gate decompositions that we use for the resource counts, which can also be found in [19], [48], and [49]. For the fixed-precision state preparation, the phases of the swap gates are not important, which can provide some resource savings. We denote circuits that apply the correct bit transformations but add incorrect phases by $\simeq$, whereas exact decompositions have the $=$ sign. We consistently use the phase-incorrect versions of the swaps presented here for the fixed-precision resource estimates, which allows us to achieve the minimal T-count. We consistently use the phase-correct version of the swaps for the prerotated case since the phase is important in that case. We use the T-depth $= 1$ version of the Toffoli [44], [46] for our resource counts. This version requires an additional ancilla but it achieves minimal T-depth, which was the motivation behind the state preparation with flags procedure that we introduced in the main text.

ACKNOWLEDGMENT

The authors would like to thank D. Bader, T. Bohdanowicz, P. Burchard, C. Hann, H. Katzgraber, R. Krishnakumar, C. Lin, S. Roy, M. Schuetz, and J. Tarantino for helpful discussions. They would also like to thank E. Campbell for early collaboration during an initial phase of this project.

REFERENCES

[1] S. A. Chakraborty, A. Gilyén and S. Jeffery, "The power of block-encoded matrix powers: Improved regression techniques via faster hamiltonian simulation," in *Proc. 46th Int. Colloq. Automata, Lang., Program.* (ser. Leibniz Int. Proc. Informatics), C. Baier, I. Chatzigiannakis, P. Flocchini, and S. Leonardi, Eds., Dagstuhl, Germany, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, pp. 33:1–33:14, doi: 10.4230/LIPIcs.ICALP.2019.33.

[2] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, "Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics," in *Proc. 51st Annu. ACM SIGACT Symp. Theory Comput.*, 2019, pp. 193–204, doi: 10.1145/3313276.3316366.

[3] Y. Subaşi, R. D. Somma, and D. Orsucci, "Quantum algorithms for systems of linear equations inspired by adiabatic quantum computing," *Phys. Rev. Lett.*, vol. 122, Feb. 2019, Art. no. 060504, doi: 10.1103/PhysRevLett.122.060504.

[4] P. C. S. Costa, D. An, Y. R. Sanders, Y. Su, R. Babbush, and D. W. Berry, "Optimal scaling quantum linear systems solver via discrete adiabatic theorem," *PRX Quantum*, vol. 3, no. 4, 2021, Art. no. 040303, doi: 10.1103/PRXQuantum.3.040303.

[5] D. An and L. Lin, "Quantum linear system solver based on time-optimal adiabatic quantum computing and quantum approximate optimization algorithm," *ACM Trans. Quantum Comput.*, vol. 3, no. 2, Mar. 2022, Art. no. 5. doi: 10.1145/3498331.

[6] P Rall, "Faster coherent quantum algorithms for phase, energy, and amplitude estimation," *Quantum*, vol. 5, p. 566, Oct. 2021, doi: 10.22331/q-2021-10-19-566.

[7] A. Gilyén, S. Arunachalam, and N. Wiebe, "Optimizing quantum optimization algorithms via faster quantum gradient computation," in *Proc. 30th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2019, pp. 1425–1444, doi: 10.1137/1.9781611975482.87.

[8] G. H. Low and I. L. Chuang, "Hamiltonian simulation by qubitization," *Quantum*, vol. 3, p. 163, Jul. 2019, doi: 10.22331/q-2019-07-12-163.

[9] I. Kerenidis and A. Prakash, "A quantum interior point method for LPs and SDPs," *ACM Trans. Quantum Comput.*, vol. 1, no. 1, Oct. 2020, Art. no. 5, doi: 10.1145/3406306.

[10] B. Augustino, G. Nannicini, T. Terlaky, and L. F. Zuluaga, "Quantum interior point methods for semidefinite optimization," 2021, *arXiv:2112.06025*, doi: 10.48550/arXiv.2112.06025.

[11] I. Kerenidis, A. Prakash, and D. Szilágyi, "Quantum algorithms for portfolio optimization," in *Proc. 1st ACM Conf. Adv. Financial Technol.*, 2019, pp. 147–155, doi: 10.1145/3318041.3355465.

[12] J. M. Martyn, Z. M. Rossi, A. K. Tan, and I. L. Chuang, "Grand unification of quantum algorithms," *PRX Quantum*, vol. 2, Dec. 2021, Art. no. 0040203, doi: 10.1103/PRXQuantum.2.040203.

[13] G. H. Low, T. J. Yoder, and I. L. Chuang, "Methodology of resonant equiangular composite quantum gates," *Phys. Rev. X*, vol. 6, Dec. 2016, Art. no. 041067, doi: 10.1103/PhysRevX.6.041067.

[14] V. Giovannetti, S. Lloyd, and L. Maccone, "Quantum random access memory," *Phys. Rev. Lett.*, vol. 100, Apr. 2008, Art. no. 160501, doi: 10.1103/PhysRevLett.100.160501.

[15] V. Giovannetti, S. Lloyd, and L. Maccone, "Architectures for a quantum random access memory," *Phys. Rev. A*, vol. 78, Nov. 2008, Art. no. 052310, doi: 10.1103/PhysRevA.78.052310.

[16] O. D. Matteo, V. Gheorghiu, and M. Mosca, "Fault-tolerant resource estimation of quantum random-access memories," *IEEE Trans. Quantum Eng.*, vol. 1, 2020, Art. no. 4500213, doi: 10.1109/TQE.2020.2965803.

[17] A. Paler, O. Oumarou, and R. Basmadjian, "Parallelizing the queries in a bucket-brigade quantum random access memory," *Phys. Rev. A*, vol. 102, Sep. 2020, Art. no. 032608, doi: 10.1103/PhysRevA.102.032608.

[18] C. T. Hann, G. Lee, S. Girvin, and L. Jiang, "Resilience of quantum random access memory to generic noise," *PRX Quantum*, vol. 2, Apr. 2021, Art. no. 020311, doi: 10.1103/PRXQuantum.2.020311.

[19] G. H. Low, V. Kliuchnikov, and L. Schaeffer, "Trading T-gates for dirty qubits in state preparation and unitary synthesis," 2018, *arXiv:1812.00954*, doi: 10.48550/arXiv.1812.00954.

[20] E. Knill, "Fault-tolerant postselected quantum computation: Schemes," 2004, *arXiv:quant-ph/0402171*, doi: 10.48550/arXiv.quant-ph/0402171.

[21] S. Bravyi and A. Kitaev, "Universal quantum computation with ideal Clifford gates and noisy ancillas," *Phys. Rev. A*, vol. 71, Feb. 2005, Art. no. 022316, doi: 10.1103/PhysRevA.71.022316.

[22] C. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, "Surface code quantum computing by lattice surgery," *New J. Phys.*, vol. 14, no. 12, Dec. 2012, Art. no. 123011, doi: 10.1088/1367-2630/14/12/123011.

[23] D. Litinski and F. v. Oppen, "Lattice surgery with a twist: Simplifying clifford gates of surface codes," *Quantum*, vol. 2, p. 62, May 2018, doi: 10.22331/q-2018-05-04-62.

[24] D. Litinski, "A game of surface codes: Large-scale quantum computing with lattice surgery," *Quantum*, vol. 3, p. 128, Mar. 2019, doi: 10.22331/q-2019-03-05-128.

[25] C. Chamberland and E. T. Campbell, "Universal quantum computing with twist-free and temporally encoded lattice surgery," *PRX Quantum*, vol. 3, no. 1, 2022, Art. no. 010331, doi: 10.1103/PRXQuantum.3.010331.

[26] I. Kerenidis and A. Prakash, "Quantum recommendation systems," 2016, *arXiv:1603.08675*, doi: 10.48550/arXiv.1603.08675.

[27] S. Chakrabarti, A. M. Childs, T. Li, and X. Wu, "Quantum algorithms and lower bounds for convex optimization," *Quantum*, vol. 4, p. 221, Jan. 2020, doi: 10.22331/q-2020-01-13-221.

[28] L. K. Grover, "Synthesis of quantum superpositions by quantum computation," *Phys. Rev. Lett.*, vol. 85, no. 6, 2000, Art. no. 1334, doi: 10.1103/PhysRevLett.85.1334.

[29] L. Grover and T. Rudolph, "Creating superpositions that correspond to efficiently integrable probability distributions," 2002, *arXiv: quant-ph/0208112*, doi: 10.48550/arXiv.quant-ph/0208112.

[30] P. Kaye and M. Mosca, "Quantum networks for generating arbitrary quantum states," in *Proc. Opt. Fiber Commun. Conf. Int. Conf. Quantum Inf.*, 2001, Art. no. PB28, doi: 10.48550/arXiv.quant-ph/0407102.

[31] A. N. Soklakov and R. Schack, "Efficient state preparation for a register of quantum bits," *Phys. Rev. A*, vol. 73, Jan. 2006, Art. no. 012307, doi: 10.1103/PhysRevA.73.012307.

[32] M. Mottonen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa, "Transformation of quantum states using uniformly controlled rotations," 2004, *arXiv: quant-ph/0407010*, doi: 10.48550/arXiv.quant-ph/0407010.

[33] M. Plesch and I. C. V. Brukner, "Quantum-state preparation with universal gate decompositions," *Phys. Rev. A*, vol. 83, Mar. 2011, Art. no. 032302, doi: 10.1103/PhysRevA.83.032302.

[34] Y. R. Sanders, G. H. Low, A. Scherer, and D. W. Berry, "Black-box quantum state preparation without arithmetic," *Phys. Rev. Lett.*, vol. 122, Jan. 2019, Art. no. 020502, doi: 10.1103/PhysRevLett.122.020502.

[35] J. Bausch, "Fast black-box quantum state preparation," 2020, *arXiv:2009.10709*, doi: 10.48550/arXiv.2009.10709.

[36] S. Wang et al., "Fast black-box quantum state preparation based on linear combination of unitaries," *Quantum Inf. Process.*, vol. 20, no. 8, 2021, Art. no. 270, doi: 10.48550/arXiv.2105.06230.

[37] I. F. Araujo, D. K. Park, F. Petruccione, and A. J. da Silva, "A divide-and-conquer algorithm for quantum state preparation," *Sci. Rep.*, vol. 11, no. 1, 2021, Art. no. 6329, doi: 10.48550/arXiv.2008.01511.

[38] X. Sun, G. Tian, S. Yang, P. Yuan, and S. Zhang, "Asymptotically optimal circuit depth for quantum state preparation and general unitary synthesis," 2021, *arXiv:2108.06150*, doi: 10.48550/arXiv.2108.06150.

[39] X.-M. Zhang, T. Li, and X. Yuan, "Quantum state preparation with optimal circuit depth: Implementations and applications," *Phys. Rev. Lett.*, vol. 129, Nov. 2022, Art. no. 230504, doi: 10.1103/PhysRevLett.129.230504.

[40] C. Gidney, "Halving the cost of quantum addition," *Quantum*, vol. 2, p. 74, Jun. 2018, doi: 10.22331/q-2018-06-18-74.

[41] R. Babbush et al., "Encoding electronic spectra in quantum circuits with linear T complexity," *Phys. Rev. X*, vol. 8, Oct. 2018, Art. no. 041015, doi: 10.1103/PhysRevX.8.041015.

[42] F.-Y. Hong, Y. Xiang, Z.-Y. Zhu, L.-Z. Jiang, and L.-N. Wu, "Robust quantum random access memory," *Phys. Rev. A*, vol. 86, Jul. 2012, Art. no. 010306, doi: 10.1103/PhysRevA.86.010306.

[43] S. Arunachalam, V. Gheorghiu, T. Jochym-O'Connor, M. Mosca, and P.V. Srinivasan, "On the robustness of bucket brigade quantum RAM," *New J. Phys.*, vol. 17, no. 12, Dec. 2015, Art. no. 123010, doi: 10.1088/1367-2630/17/12/123010.

[44] C. Jones, "Low-overhead constructions for the fault-tolerant Toffoli gate," *Phys. Rev. A*, vol. 87, Feb. 2013, Art. no. 022328, doi: 10.1103/PhysRevA.87.022328.

[45] N. J. Ross and P. Selinger, "Optimal ancilla-free Clifford T approximation of z-rotations," *Quantum Inf. Comput.*, vol. 16, no. 11-12, pp. 901–953, Sep. 2016, doi: 10.26421/QIC16.11-12-1.

[46] P. Selinger, "Quantum circuits of T-depth one," *Phys. Rev. A*, vol. 87, no. 4, Apr. 2013, Art. no. 042302, doi: 10.1103/PhysRevA.87.042302.

[47] J. J. Vartiainen, M. Möttönen, and M. M. Salomaa, "Efficient decomposition of quantum gates," *Phys. Rev. Lett.*, vol. 92, Apr. 2004, Art. no. 177902, doi: 10.1103/PhysRevLett.92.177902.

[48] A. Barenco et al., "Elementary gates for quantum computation," *Phys. Rev. A*, vol. 52, pp. 3457–3467, Nov. 1995, doi: 10.1103/PhysRevA.52.3457.

[49] M. Amy, D. Maslov, M. Mosca, and M. Roetteler, "A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 32, no. 6, pp. 818–830, Jun. 2013, doi: 10.1109/TCAD.2013.2244643.