Direct Block-Encodings for Second-Quantized Operators

(Dated: February 21, 2025)

In this work, we describe and analyze a framework for block-encoding second-quantized ladder operators which act upon fermionic and bosonic modes. We refer to this framework as LOBE (Ladder Operator Block-Encoding) and show how it can be used to reduce the spacetime quantum resources required to block-encode various second-quantized operators. Specifically, we compare LOBE to block-encoding frameworks which require expanding the operators in the Pauli basis. We numerically benchmark these constructions using models arising in quantum field theories including ϕ^4 theory and the full Yukawa model. The numerical quantum resource estimates presented in this work show that LOBE produces block-encodings with fewer non-Clifford operations, fewer block-encoding ancillae and overall number of qubits, and lower rescaling factors for many different classes of operators and Hamiltonians. Additionally, the LOBE constructions have better asymptotic scaling with respect to several key paramters including the maximum occupation of bosonic modes, the total number of fermionic and bosonic modes, and the locality of the operators.

I. INTRODUCTION

The simulation of many-body quantum systems is a promising potential application for quantum computers [1]. In digital quantum simulation, quantum algorithms are composed using a series of unitary operations. Accessing the information of non-unitary operators - such as the Hamiltonian of a quantum system - within a quantum algorithm is a necessary subroutine for conducting such simulations. This encoding task has been pursued through various means, resulting in methods such as Trotterization [2–6] and Block-Encoding [7–9].

Block-Encoding describes a general strategy for encoding a non-unitary matrix within a chosen subspace (block) of a larger unitary operator. Two general frameworks for constructing block-encodings of different operators - sparse block-encodings [7, 10, 11] and Linear Combinations of Unitaries (LCU) [12] - have allowed for the exploration of explicitly compiled block-encodings of several systems.

Understanding the spacetime quantum resources - the number of qubits (space), the number of operations (time), and the rescaling factor (overhead) - required for quantum simulation algorithms is important for understanding the feasability of simulating different systems. These quantum resource estimates are crucial as they allow us to gauge the practical usefulness of quantum computers, particularly those that have experimentally demonstrated quantum error correction [13, 14].

Many previous works have numerically investigated quantum simulation algorithms for purely fermionic systems, with a particular emphasis on the simulation of molecules in quantum chemistry [15–23]. Another interesting set of quantum systems to simulate are those that are derived from quantum field theories [24, 25] @Gus, pls add citation for all models (quartic, static, phi, yukawa), which have applications in areas such as high-energy physics [26]. These systems often include interactions between fermions, antifermions, and bosons and several works have produced quantum resource estimates for simulating such systems [27–29].

In this work, we provide a framework for constructing block-encodings of second-quantized operators, which we refer to as Ladder Operator Block-Encoding (LOBE). This framework directly block-encodes operators comprised of creation and annihilation operators which act upon fermionic and bosonic modes. It does not require the use of operator transformations that expand fermionic [30–32] and bosonic [33, 34] ladder operators in the Pauli operator basis. This framework therefore avoids the potential overhead caused by these operator transformations and leads to block-encodings with lower spacetime requirements in many cases.

We give numerical quantum resource estimates for implementing block-encodings of several classes of operators and several Hamiltonians that arise in quantum field theories. This includes purely fermionic systems, purely bosonic systems, and systems which include fermions, antifermions, and bosons. We analyze the numerical spacetime quantum resources for LOBE - in comparison to techniques which require mapping ladder operators onto the Pauli basis - and find that LOBE results in constructions with better asymptotic scaling and lower numerical quantum resources for many of the systems examined.

This work is organized as follows. In Section II, we review the action of ladder operators on quantum states in second-quantization. In Section III, we review block-encodings and discuss frameworks for constructing block-encodings of different operators. In Section IV, we describe the LOBE framework, show compiled block-encodings for several classes of second-quantized operators, and give analytical spacetime costs of the associated constructions. In Section V, we provide numerical quantum resource estimates for block-encodings of various classes of operators and Hamiltonians. In Section VI, we summarize the results presented in this work and discuss future directions. Additionally, a glossary that defines the terminology and variables used throughout this work is given in Appendix A.

II. SECOND-QUANTIZATION

In many models of quantum physics, such as quantum field theory and quantum chemistry, quantum operators are described in second-quantization [35]. In second-quantization, multiparticle state vectors are written in terms of fermionic, antifermionic, and bosonic modes which can be occupied by different numbers of particles. In the occupation number basis, this can be written as $|n\rangle = |n_{I-1}, \ldots, n_1, n_0\rangle$ where $n_i \in \mathbb{Z}$ is the number of particles in the i^{th} mode.

The space spanned by the second-quantized state vectors is called the *Fock space* ($\mathcal{F}$) and the state vectors are referred to as *Fock states*. The Fock space is a direct sum of n-particle sectors for all n. [36]:

$$\mathcal{F} = \oplus_n \mathcal{H}_n. \tag{1}$$

In quantum field theories, field operators, rather than wavefunctions, are the main objects. These field operators act on second-quantized states to create and annihilate particles in the field. This action is often described in terms of ladder operators.

In second-quantization, ladder operators are quantum operators that act on fermionic, antifermionic, and bosonic modes to either increase (create) or decrease (annihilate) the number of particles occupying that mode. Many observables of interest for these systems, such as the Hamiltonian, can be efficiently expressed as sums or products of ladder operators acting on different modes.

A. Fermionic Ladder Operators

Fermions obey the Pauli-exclusion principle [37]. Therefore, the occupation of a (anti)fermionic mode can only be either occupied or unoccupied. Fermionic ladder operators only act non-trivially on the mode that the ladder operator acts upon. This non-trivially action is defined by:

$$b_i^\dagger |n_{i_b}\rangle = \begin{cases} (-1)^{\sum_{j<i} n_{j_b}} |1\rangle & \text{when } |n_{i_b}\rangle = |0\rangle \\ 0 & \text{when } |n_{i_b}\rangle = |1\rangle \end{cases} \tag{2}$$

where b_i denotes a fermionic ladder operator on the i^{th} mode, the † indicates a creation operator, and $|n_{i_b}\rangle$ is the Fock state of the i^{th} fermionic mode.

For a creation operator, if the mode being acted upon is unoccupied, then the creation operator "creates" a fermion in that mode. If the mode is already occupied before the creation operator is applied, then the operator "zeroes-out" the quantum state, which is indicated by setting the amplitdue of the state to zero.

The fermionic ladder operators also cause a potential sign flip determined by:

$$p(\psi) = (-1)^{\sum_{j<i} n_{j_b}} \tag{3}$$

If the parity of the occupation of the fermionic modes with index $j < i$ is odd, then the sign of the output state is flipped. Therefore the ordering of the modes in the encoding has an affect on the action of the operator which must be accounted for.

In the above definition, it is implied that the ladder operator acts coherently on a superposition state of the corresponding fermionic mode. The desired action on such states is defined by linearity following the expansion of the state in the fock basis. Throughout this work, if the action of the operator is defined by the action on particular basis states (as is done above), then it is implied that the operator will act linearly on superpositions of these basis states.

The action of the fermionic annihilation operator is defined by:

$$b_i |n_{i_b}\rangle = \begin{cases} p(\psi) |0\rangle & \text{when } |n_{i_b}\rangle = |1\rangle \\ 0 & \text{when } |n_{i_b}\rangle = |0\rangle \end{cases} \tag{4}$$

The action of the annihilation operators is opposite to that of the creation operators. If the mode is unoccupied before the operator is applied, then the annihilation operator "zeroes-out" the state. If the mode is already occupied, then the annihilation operator "annihilates" the fermion in that mode by updating the occupation of the mode to zero.

The fermionic ladder operators can be reordered arbitrarily with the introduction of additional terms due to the commutation rules. These commutation rules are given by:

$$\begin{aligned} \{b_i, b_j^\dagger\} &= \delta_{ij} \\ [b_i, b_j] &= [b_i^\dagger, b_j^\dagger] = 0 \end{aligned} \tag{5}$$

In addition to fermions, antifermions are often included in quantum field theories. The properties and commutation rules for antifermions are the same as fermions and the sign flip for antifermions includes the parity of the occupation of the fermionic modes. Therefore, for simplicity we will treat antifermionic modes simply as fermionic modes that are indexed after all native fermionic modes for the remainder of this work.

B. Bosonic Ladder Operators

For bosonic modes, there is no physical limitation on the occupancy of each mode: $n_{i_a} \in [0, 1, 2, \ldots)$. Hence the Hilbert space of a single bosonic mode is countably infinite dimensional. For discretized computations, a cutoff on the bosonic occupancy (Ω) is chosen to make the dimension of the Hilbert finite: $n_{i_a} \in [0, 1, 2, \ldots, \Omega]$. The cutoff on the bosonic occupancy can introduce error as some physically allowable states become computationally inaccessible. In practice, simulations of bosonic systems often proceed by increasing Ω until the error induced by this cutoff is either negligible or well understood.

With the imposed bosonic cutoff, the action of bosonic creation operator is defined by:

$$a_i^\dagger |n_{i_a}\rangle = \begin{cases} \sqrt{n_{i_a}+1}\,|n_{i_a}+1\rangle & \text{when } |n_{i_a}\rangle \neq |\Omega\rangle \\ 0 & \text{when } |n_{i_a}\rangle = |\Omega\rangle \end{cases} \quad (6)$$

where a_i denotes a bosonic ladder operator on the i^{th} mode, the † indicates a creation operator, $|n_{i_a}\rangle$ is the occupation of the i^{th} bosonic mode, and Ω is the maximum number of bosons allowed in a single mode.

The bosonic creation operator increases the occupation of the bosonic mode being acted upon by 1. The operator will also multiply the amplitude of the state by the square-root of the updated occupancy of the mode. If the occupancy is already at the (artificially restricted) maximum allowable occupancy, then the operator will zero-out the state.

Likewise, the bosonic annihilation operators lower the occupancy of the mode being acted upon and multiply the amplitude by the square-root of the occupancy of the mode prior to being acted upon:

$$a_i |n_{i_a}\rangle = \begin{cases} \sqrt{n_{i_a}}\,|n_{i_a}-1\rangle & \text{when } |n_{i_a}\rangle \neq |0\rangle \\ 0 & \text{when } |n_{i_a}\rangle = |0\rangle \end{cases} \quad (7)$$

Similarly, if the occupancy of the state is zero before the operator is applied, then the state is zeroed-out.

The commutation rules for bosonic ladder operators are given by:

$$\begin{aligned} [a_i, a_j^\dagger] &= \delta_{ij} \\ [a_i, a_j] &= [a_i^\dagger, a_j^\dagger] = 0 \end{aligned} \quad (8)$$

Additionally, bosonic ladder operators commute with fermionic (and antifermionic) ladder operators.

C. Observables

Many observables of interest for a particular system, such as the Hamiltonian, can be expressed in terms of linear combinations of products of ladder operators that act on both fermionic and bosonic modes.

In this work, we will default to expressing operators in their *mode ordered* form, which we define as an adaptation of normal ordering. We say a term (T) is *mode ordered* when all operators acting on a single mode are grouped next to one another and all creation operators appear to the left of all annihilation operators:

$$T = \Big(\prod_i (b_i^\dagger)^{\delta_{b_i}^\dagger} (b_i)^{\delta_{b_i}}\Big)\Big(\prod_i (a_i^\dagger)^{R_i} (a_i)^{S_i}\Big) \quad (9)$$

where i indexes the respective fermionic and bosonic modes, δ takes the value 0 or 1 to denote if the individual ladder operator is included in ther term, and the values R_i and S_i are integers in the range $[0, \Omega]$ which denote the exponent of the bosonic ladder operators acting on the i^{th} bosonic mode.

When ordering an operator, the commutation rules (Eq. 5 and Eq. 8) must be taken into account. These commutation relations can also introduce more terms when reordering.

As an example, second-quantized hermitian operators are often expressed in the form of linear combinations of products of ladder operators:

$$H = \sum_{l=0}^{L-1} \alpha_l T_l \quad (10)$$

where L is the total number of terms and α_l are positive, real-valued I *think* we can have complex coefficients here, no? coefficients associated with the terms T_l.

III. BLOCK-ENCODING

Quantum algorithms are restricted to being composed by a series of unitary operators. However, it is often necessary to access information about various non-unitary operators within a quantum algorithm. "Block-Encoding" refers to a type of access model where the information regarding non-unitary operators is encoded in a labeled subspace (block) of a larger unitary operator.

If A represents some $N_A \times N_A$ non-unitary operator, then a block-encoding of A in matrix form is given by:

$$U_A = \begin{pmatrix} \bar{A} & * \\ * & * \end{pmatrix} \quad (11)$$

where U_A is a unitary operator, $\bar{A}$ is a rescaled form of A such that the L2 norm ($||\bar{A}||$) is less than or equal to 1, and matrix entries $*$ denote matrix elements that ensure U_A is unitary.

The action of U_A on an arbitrary quantum state ($|\psi\rangle$) with dimension N_A can be defined as:

$$U_A |\psi\rangle |0\rangle_{\text{anc}} = \bar{A} |\psi\rangle |0\rangle_{\text{anc}} + \beta_\psi |\perp\rangle \quad (12)$$

where $|\rangle_{\text{anc}}$ is a register of block-encoding ancillae used to generate U_A and β_ψ is a complex coefficient that normalizes the output quantum state. The portion of the state given by $|\perp\rangle$ represents the branch of the wavefunction that is orthogonal to any state in the encoded subspace. In the above equations, the encoded subspace of the block-encoding is chosen (without loss of generality) to be the all-zero state of the block-encoding ancillae.

It is important to note that β_ψ is dependent on the initial state of the system register ($|\psi\rangle$). The dependence of β on $|\psi\rangle$ can be seen in a simple case where $|\psi\rangle$ is an eigenstate of $\bar{A}$. If $|\lambda_k\rangle$ is an eigenstate of $\bar{A}$ with eigenvalue k, then Eq. 12 leads to:

$$U_A |\lambda_k\rangle |0\rangle_{\text{anc}} = k |\lambda_k\rangle |0\rangle_{\text{anc}} + \sqrt{1-|k|^2}\,|\perp\rangle \quad (13)$$

where $\sqrt{1-|k|^2}$ is clearly dependent upon the eigenvalue associated with the eigenstate.

One way to interpret the action of the block-encoding is that it produces a probabilistic application of $\bar{A}$. After the block-encoding is applied, if a measurement of the block-encoding ancillae results in all of those qubits being in the zero state, then $\bar{A}$ has been applied to the quantum state. The probability for this measurement result to occur is given by $\mathrm{P}(0) = ||\bar{A}\,|\psi\rangle\,||^2$.

The rescaling factor (λ) of a block-encoding is given by:

$$A = \lambda \bar{A} \tag{14}$$

This rescaling factor can have important implications in the cost of quantum algorithms. Smaller rescaling factors generally lead to more efficient algorithms though the exact impact on the cost is algorithm-dependent. For example, the success probability of applying $\bar{A}$ as described above decreases as λ increases. If the block-encoding is used to construct a quantum walk operator [8, 9] and used in Quantum Phase Estimation [8, 19, 21], then the output eigenvalue estimates are scaled by the rescaling factor which will affect both the error and the precision of the estimate.

In the following subsections, we will describe some commonly used frameworks for constructing block-encodings of different operators. In particular, we discuss both the Sparse-Oracle and Linear Combination of Unitaries (LCU) frameworks for generating block-encodings. Then we discuss methods to combine block-encodings of operators to produce block-encodings for a linear combination of operators or a product of operators.

A. Sparse-Oracle Framework

The Sparse-Oracle framework [10, 11, 38–42] assumes the presence of oracles that provide both the location and the values of the nonzero matrix elements. When an operator is sparse within some known basis, the Sparse-Oracle framework can build off of this structure to reduce the cost of the block-encoding construction.

Let A be a matrix with a maximum of s nonzero entries in a single row and each matrix element (A_{ij}) have magnitude ≤ 1.

Lin et. al [7] defines three oracles (D_s, O_A, and O_c) to produce a Sparse-Oracle block-encoding. The "diffusion operator" (D_s) is defined by:

$$D_s \left|0^{\log_2 s}\right\rangle = \frac{1}{\sqrt{s}} \sum_{l=0}^{s-1} |l\rangle \tag{15}$$

Where s is increased to the nearest power of 2 by treating some zero-valued elements as nonzero. This oracle can be implemented by a tensor product of Hadamards: $H^{\otimes \log_2 s}$.

The "column oracle" (O_c) is defined by:

$$O_c\,|j\rangle\,|l\rangle = |j\rangle\,|c(j,l)\rangle \tag{16}$$

where $c(j,l)$ is a function that returns the row-index of the l^{th} nonzero matrix element in the j^{th} column.

Lastly, the "value oracle" (O_A) is defined by:

$$O_A\,|0\rangle\,|j\rangle\,|i\rangle = \big(A_{ij}\,|0\rangle + \beta_{ij}\,|1\rangle\,\big)\,|j\rangle\,|i\rangle \tag{17}$$

where $\beta_{ij} \equiv \sqrt{1 - |A_{ij}|^2}$.

With these three oracles, a block-encoding for A is given by:

$$U_A = D_s O_c O_A D_s \tag{18}$$

with a rescaling factor of:

$$\lambda_{\text{SO}} = 2^n \max_{ij} |A_{ij}| \tag{19}$$

A proof that U_A block-encodes A is given in Camps et. al [27].

The factor of $\max_{ij} |A_{ij}|$ in Eq. 19 comes from the constraint that all elements of A have magnitude at most 1. The factor of 2^n comes from the sparsity of A. If s is not a power of two, then this factor can be reduced to s by replacing the "diffusion operator" with the generalized *Uniform State Preparation* (USP) protocol which is discussed in more detail in Appendix B. Implementing USP typically requires more nontrivial operations than the diffusion operator, so a trade-off exists between reducing the rescaling factor and reducing the time complexity.

A framework for compiling quantum circuits that implement these oracles for general matrices (D_s, O_A, and O_c) is given in [27, 43] and Sparse-Oracle block-encodings for particular systems have been explored in several works [28, 43, 44].

B. Linear Combination of Unitaries Framework

An alternative framework proposed by Childs et. al [12] called Linear Combination of Unitaries (LCU) is *specifically* designed for generating block-encodings of operators that can be written in the following form:

$$A = \sum_{l=0}^{L-1} \alpha_l U_l \tag{20}$$

where α_l are positive, real-valued coefficients and each U_l is a unitary operator. For terms with negative or imaginary coefficients, the sign and imaginary phase can be absorbed into the operator: $-i\alpha_l U_l = \alpha_l(-iU_l)$. Likewise, a term with a complex coefficient can be treated as two distinct terms, each with a real coefficient.

To generate an LCU block-encoding, two oracles, *Prepare* and *Select*, are required. The *Prepare* oracle serves a similar role to the O_A oracle in that it loads the coefficients or the values of the matrix elements onto the quantum computer. The action of the *Prepare* oracle is to map the all-zero state of a block-encoding ancilla register

to a particular superposition state where the squared amplitudes of the computational basis states are weighted to encode the coefficients of the terms in the operator:

$$\textbf{Prepare}: \left|0^{\otimes \lceil \log_2 L \rceil}\right\rangle_{\text{index}} \rightarrow \sum_{l=0}^{L-1} \sqrt{|\alpha_l|/\lambda}\, |l\rangle_{\text{index}} \quad (21)$$

where λ is given by:

$$\lambda_{\text{LCU}} = \sum_{l=0}^{L-1} |\alpha_l| \quad (22)$$

The block-encoding ancilla register is often referred to as the *index register* since the computational basis states of this register ($|l\rangle$) index the L terms in Eq. 20.

The desired action of the *Select* oracle is to apply the l^{th} unitary (U_l) onto the system register when the index regsiter is in the computation basis state $|l\rangle$:

$$\textbf{Select}: \begin{cases} \text{Apply } U_l \text{ on } |\psi\rangle & \text{when } |\text{index}\rangle \text{ is } |l\rangle \text{ for } l < L \\ \text{Undefined} & \text{Otherwise} \end{cases} \quad (23)$$

and the action of the *Select* oracle is not strictly defined for the computational basis states outside of the range $[0, L)$.

With these two oracles, a block-encoding for A is given by:

$$U_A = (\textbf{Prepare}^\dagger)(\textbf{Select})(\textbf{Prepare}) \quad (24)$$

with an overall rescaling factor of λ_{LCU} (Eq. 22). A proof that U_A block-encodes A is given in Section 7.3 of Lin et. al [7].

A block-encoding for A could also be generated using the Sparse-Oracle framework, but would result in a rescaling factor of $\lambda_{\text{SO}} = 2^{\lceil \log_2 L \rceil} \max_l |\alpha_l|$. A proof that $\lambda_{\text{LCU}} \leq \lambda_{\text{SO}}$ is given in Appendix D.

1. Implementing *Prepare*

The matrix representation for a valid implementation of the *Prepare* oracle in the computational basis is given by:

$$\text{Prepare} = \begin{bmatrix} \sqrt{|\alpha_0|/\lambda_{\text{LCU}}} & * & \dots & * \\ \sqrt{|\alpha_1|/\lambda_{\text{LCU}}} & * & \dots & * \\ \vdots & \vdots & \ddots & \vdots \\ \sqrt{|\alpha_{L-1}|/\lambda_{\text{LCU}}} & * & \dots & * \end{bmatrix} \quad (25)$$

From this definition, it is clear that there are infinitely many unitaries that implement the *Prepare* oracle since only the first column of the operator is fixed.

The Grover-Rudolph algorithm [45] gives a formulaic routine to generate quantum circuits that implement the *Prepare* oracle. The cost of implementing Grover-Rudolph scales exponentially with the number of qubits in the register it acts upon. However, the size of the index register is logarithmic in the number of terms (L), therefore the number of operations required to implement Grover-Rudolph is linear in L.

For operators that have structure among the coefficients of the terms, implementations of *Prepare* can be constructed that leverage this structure to reduce the number of operations required to implement the oracle. In certain cases, this can drastically reduce the cost such as is done for the Fermi-Hubbard model in [19].

2. Implementing *Select*

Without any assumptions on the structure of A, the *Select* oracle can be implemented by the operator:

$$U_{\text{Select}} = \sum_{l=0}^{L-1} |l\rangle\langle l| \otimes U_l + \sum_{l \geq L} |l\rangle\langle l| \otimes \mathbb{1} \quad (26)$$

In this above form, U_{Select} is restricted to apply the identity operator for computational basis states of the index register outside of the range $[0, L)$. However, if this constraint is relaxed or if there is structure in the description of the operator, then a more efficient implementation of *Select* can be constructed as is done in [19].

As an aside, if the implementation of the *Select* oracle is self-inverse, then the LCU block-encoding is also self-inverse. This structure makes LCU block-encodings particularly well-suited for being applied in algorithms based on Qubitization [9]. The construction for *Select* in Eq. 26 is naturally self-inverse if the unitaries themselves are self-inverse, which is true when the operator is decomposed in the Pauli basis.

C. Block-Encoding Linear Combinations of Non-Unitary Operators

A natural question to ask is if the structure of an LCU block-encoding can be generalized to a linear combination of non-unitary operators:

$$A = \sum_{l=0}^{L-1} \alpha_l O_l \quad (27)$$

where O_l are operators and we again restrict the coefficients α_l to be positive and real-valued for clarity.

Let the set of unitary operators $\{U_l\}$ represent block-encodings of the operators O_l:

$$U_l |\psi\rangle |0\rangle_{\text{anc}} = \bar{O}_l |\psi\rangle |0\rangle_{\text{anc}} + \beta_{\psi,l} |\perp\rangle \quad (28)$$

where each block-encoding uses the same encoded subspace and has a rescaling factor λ_l such that $\lambda_l \bar{O}_l = O_l$. Additionally, let the operator $\tilde{A}$ be defined by the linear combination of the block-encoding unitaries:

$$\tilde{A} = \sum_l \tilde{\alpha}_l U_l \quad \tilde{\alpha}_l = \frac{\alpha_l \lambda_l}{\max_{l'} \lambda_{l'}} \quad (29)$$

Then an LCU block-encoding of $\tilde{A}$ will also give a block-encoding of A with an overall rescaling factor of:

$$\lambda = \sum_{l=0}^{L-1} |\alpha_l \lambda_l| \tag{30}$$

Is this paragraph helpful/necessary or should it be removed? The fact that $U_{\tilde{A}}$ block-encodes A can be seen from the linearity of the operators themselves:

$$\begin{aligned} U_{\tilde{A}} &= \begin{pmatrix} \bar{\tilde{A}} & * \\ * & * \end{pmatrix} \propto \sum_l \begin{pmatrix} \tilde{\alpha_l} U_l & * \\ * & * \end{pmatrix} \\ &= \sum_l \left(\begin{pmatrix} \alpha_l O_l & * \\ * & * \end{pmatrix} \begin{matrix} * \\ * \end{matrix} \right) \end{aligned} \tag{31}$$

It is clear from this form that the block-encodings $\{U_l\}$ must all have the same encoded subspace as they would otherwise not yield the desired linear combination in the encoded subspace of $U_{\tilde{A}}$. It is also worth noting that the block-encodings $\{U_l\}$ may all use the same block-encoding ancillae as long as the implementation of the *Select* oracle ensures that the block-encoding ancillae begin in the all-zero state in the subspace in which each block-encoding unitary is applied.

In this work, we will refer to a block-encoding of this form as a Linear Combination of Operators (LCO). As noted in prior works [7, 39, 41, 42, 46], an LCO can be thought of as generalization of LCU to produce a block-encoding for a linear combination of matrices. Alternatively, it may be enlightening to consider an LCU block-encoding as a special case of an LCO block-encoding wherein the operators in the linear combination are all unitary and thus block-encode themselves.

D. Block-Encoding Products of Non-Uniatry Operators

Likewise, it is natural to ask if a block-encoding for a product of operators can be easily produced, assuming access to block-encodings for the individual operators. As noted by Gilyén et. al [42], a block-encoding for a product of operators can be constructed by the product of the block-encodings of the individual operators.

Let A represent a product of operators B and C:

$$A = BC \tag{32}$$

with block-encodings U_B and U_C. Then, the product of U_B and U_C results in a block-encoding of A:

$$\begin{aligned} &U_B U_C \left|\psi\right\rangle \left|0\right\rangle_{\text{anc,C}} \left|0\right\rangle_{\text{anc,B}} = \\ &U_B \left(\bar{C} \left|\psi\right\rangle \left|0\right\rangle_{\text{anc,C}} + \beta_{\psi,C} \left|\perp^*\right\rangle\right) \left|0\right\rangle_{\text{anc,B}} = \\ &\bar{B}\bar{C} \left|\psi\right\rangle \left|0\right\rangle_{\text{anc,C}} \left|0\right\rangle_{\text{anc,B}} + \beta_{\psi,C,B} \left|\perp\right\rangle \end{aligned} \tag{33}$$

with an overall rescaling factor of $\lambda = \lambda_B \lambda_C$.

It is important to note that in general the block-encodings for the individual operators (U_B and U_C) require separate block-encoding ancillae. This constraint ensures that each block-encoding unitary acts on an ancilla register beginning in the all-zero state to adhere to the form of Eq. 12.

IV. COMPILING LADDER OPERATOR BLOCK-ENCODINGS

In this work, we focus on generating block-encodings for second-quantized operators that act on both fermionic and bosonic modes. In this section, we provide a framework for compiling controlled block-encodings of products and linear combinations of fermionic and bosonic ladder operators.

The framework for generating these block-encodings primarily relies on infering the action of the operators on the fermionic and bosonic modes upon which the operator acts nontrivially. The compilations presented below are optimized primarily to reduce the number of non-Clifford operations and secondarily to reduce the number of block-encoding ancillae.

Although controlled applications of block-encodings are not always required in quantum algorithms, control qubits are included throughout this section to demonstrate that controlling the block-encoding does not significantly increase the cost of the block-encoding. For applications that do not require a controlled application of the block-encoding, the control qubits can simply be omitted and the cost associated with the uncontrolled block-encoding will be reduced accordingly.

A. Encoding

The Jordan-Wigner transformation [47] maps fermionic ladder operators to Pauli operators acting on qubits. Under this transformation, Fock states of fermionic modes are represented by qubit states using the following map:

$$\left|\psi_{I_b-1} \ldots \psi_{1_b} \psi_{0_b}\right\rangle \rightarrow \left|q_{I_b-1} \ldots q_{1_b} q_{0_b}\right\rangle \tag{34}$$

where $\psi_{i_b} = q_{i_b} \in [0, 1]$ depending on if the i^{th} fermionic mode is occupied ($|1\rangle$) or unoccupied ($|0\rangle$). In this work, we use the same qubit encoding to represent the occupation of fermionic modes. The number of qubits required for the fermionic system is equal to the number of fermionic modes.

The encoding scheme for bosons must be able to represent a number of bosons in the range $\psi_{i_a} \in [0, \Omega]$ for each bosonic mode. We choose to represent the occupancy of a bosonic mode in binary notation following [29]:

$$\left|\psi_{i_a}\right\rangle \rightarrow \left|q_{i_a}^{w-1} \ldots q_{i_a}^1 q_{i_a}^0\right\rangle \tag{35}$$

where the subscript denotes the index of the bosonic mode and the superscripts denote the index of the qubit

storing the w^{th} least-significant bit of the binary representation of ψ_{i_a}. Due to the binary encoding, the number of qubits needed to represent each mode is $w = \lceil \log_2(\Omega+1) \rceil$. Fock states of multiple bosonic modes can be expressed as follows:

$$\begin{aligned} &|\psi_{I_a-1}\rangle \otimes \ldots \otimes |\psi_{1_a}\rangle \otimes |\psi_{0_a}\rangle \rightarrow \\ &\left|q_{I_a-1}^{w-1} \ldots q_{I_a-1}^{1} q_{I_a-1}^{0}\right\rangle \otimes \ldots \\ \otimes &\left|q_1^{w-1} \ldots q_1^1 q_1^0\right\rangle \otimes \left|q_0^{w-1} \ldots q_0^1 q_0^0\right\rangle \end{aligned} \tag{36}$$

In this work, we assume that the maximum bosonic occupancy for each bosonic mode is identical. However, this encoding scheme and the block-encoding constructions presented below can be easily adapted to allow for bosonic modes to have different bosonic cutoffs.

B. Fermionic Ladder Operators

Here, we construct block-encodings for the individual fermionic creation ($b_j^\dagger$) and annihilation (b_j) operators. As these are block-encodings of non-unitary operators, we define the associated unitary operators to be of the form of Eq. 12:

$$U_{b_j^\dagger} |\psi\rangle |0\rangle_{\text{anc}} = b_j^\dagger |\psi\rangle |0\rangle_{\text{anc}} + \beta |\perp\rangle \tag{37}$$

The L2 norm of the fermionic ladder operators in the Fock basis is 1 and therefore do not need to be rescaled.

The definition of the fermionic creation operator (Eq. 2) results in two implications. If the mode is *unoccupied*, then the mode should become occupied and a potential phase flip should be applied, determined by the occupancy of the preceeding fermionic modes. If the mode is occupied, then state should be zeroed-out. Therefore, the non-trivial action of the unitary will be dependent on the occupation of the associated fermionic mode:

$$U_{b_j^\dagger} |\psi_j\rangle |0\rangle_{\text{anc}} = \begin{cases} p(\psi) |1\rangle |0\rangle_{\text{anc}} & \text{when } |\psi_j\rangle \text{ is } |0\rangle \\ |\perp\rangle & \text{when } |\psi_j\rangle \text{ is } |1\rangle \end{cases} \tag{38}$$

where $p(\psi)$ denotes the potential phase flip induced by the parity of the occupation of the preceeding fermionic modes as defined by Eq. 3.

As noted in Section III, when the action of an operator is defined by the action on individual basis states (as is done above), it is implied that the operator will act linearly on a superposition of basis states. This implication will be assumed throughout the remainder of this section.

An implementation for $U_{b_j^\dagger}$ is given in subfigure 1a. The initial Toffoli gate flips the ancilla qubit to indicate that the state has been zeroed-out when the control qubit is on ($|1\rangle$) and the fermionic mode is occupied ($|1\rangle$). The sign of the output state corresponding to $p(\psi)$ can be applied appropriately using a series of controlled Pauli Z operators applied to each of the fermionic modes with index $i < j$: $\vec{Z}_i$. The occupation of the fermionic mode on which the ladder operator acts is updated using a controlled Pauli X operator: X_j.

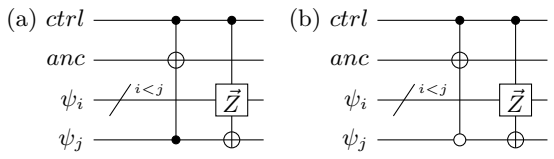

FIG. 1. **Fermionic Ladder Operator Block-Encodings** In (a), a block-encoding for the fermionic creation operator acting on the j^{th} mode is given. In (b), a block-encoding for the fermionic annihilation operator acting on the j^{th} mode is given. For a creation (annihilation) operator, the branch of the wavefunction will be flipped outside of the encoded subspace if the mode is occupied (unoccupied). The state is updated by applying Pauli Z gates to the preceeding fermionic modes which result in the output state having the appropriate sign based on $p(\psi)$. Then a Pauli X gate is applied to flip the occupation of the j^{th} mode.

A block-encoding for the fermionic annihilation operator, U_{b_j}, can be constructed similarly to $U_{b_j^\dagger}$ and is shown in subfigue 1b. The fermionic creation operator ($b^\dagger$) only acts nontrivially when the mode it acts upon is *unoccupied*. Inversely, the fermionic annihilation operator (b) will only act nontrivially if the mode it acts upon is *occupied*. Therefore, the block-encoding ancilla is flipped outside of the encoded subspace if the control qubit is on ($|1\rangle$) and the fermionic mode is unoccupied ($|0\rangle$).

As noted in Appendix C, each Toffoli gate can be implemented with four T gates using one clean ancilla. These block-encodings have an optimal rescaling factor ($\lambda = 1$), require one block-encoding ancilla and one clean ancillae, and use four T gates.

C. Products of Fermionic Ladder Operators

As discussed in subsection III D, a block-encoding for a product of operators can be constructed using a product of the unitaries that block-encode each operator. In this subsection, we construct block-encodings for a product of fermionic ladder operators that use fewer quantum resources than are required by this naive strategy.

Consider the action of the fermionic number operator ($b_i^\dagger b_i$):

$$b_i^\dagger b_i |\psi_{b_i}\rangle = \begin{cases} |1\rangle & \text{when } |\psi_{b_i}\rangle \text{ is } |1\rangle \\ 0 & \text{when } |\psi_{b_i}\rangle \text{ is } |0\rangle \end{cases} \tag{39}$$

If the i^{th} mode is unoccupied, the state will be zeroed-out. If the i^{th} mode is occupied, then the state is left unchanged.

This action can be encoded using the circuit shown in subfigure 2a. The Toffoli gate flips the block-encoding ancilla outside of the encoded subspace if the control is

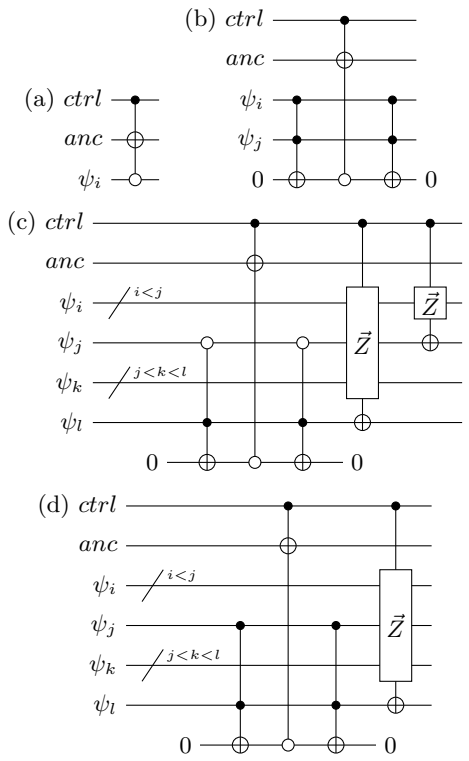

FIG. 2. **Block-Encoding Products of Fermionic Ladder Operators** In (a), a block-encoding for the fermionic number operator acting on the i^{th} mode ($b_i^\dagger b_i$) is given. In (b), a block-encoding for the product of two fermionic number operators acting on the i^{th} and j^{th} modes ($b_i^\dagger b_i b_j^\dagger b_j$) is given. In (c), a block-encoding for the operator $b_j^\dagger b_l$ with $i \neq l$ is given. In (d), a block-encoding for the operator $b_j^\dagger b_j b_l$ with $i \neq l$ is given.

on and the i^{th} mode is unoccupied. Otherwise, the state of the system and the block-encoding ancilla are left unchanged.

Block-encoding this operator as the product of the block-encodings for b_i and $b_i^\dagger$ would require two block-encoding ancillae and use 8 T gates. Meanwhile, this construction has an optimal rescaling factor ($\lambda = 1$), requires one block-encoding ancilla and one clean ancilla, and uses four T gates. The spacetime quantum resources quoted in this work assume that an N-controlled Toffoli gate acting on a clean ancilla is decomposed using the strategies discussed in Appendix C.

A block-encoding circuit for the operator $b_j^\dagger b_l$ is given in subfigure 2c. For this operator, we can infer that the state will be zeroed-out *unless* both the j^{th} mode is unoccupied and the l^{th} mode is occupied. If the control qubit is on and these two conditions are not both true, then the block-encoding ancilla is flipped outside of the encoded subspace. The state of the system is updated based on the two operators in the order in which they would act upon the quantum state: $\vec{Z}X_l$ (b_l) then $\vec{Z}X_j$ ($b_j^\dagger$). This block-encoding circuit has an optimal rescaling factor ($\lambda = 1$), requires one block-encoding ancilla and two clean ancillae, and uses 8 T gates.

This construction can be generalized to an arbitrary product of ladder operators. Let B represent the number of active modes in an operator. We define the number of "active modes" as the number of unique modes upon which an operator is applied nontrivially. For example, the operator $b_i b_j^\dagger b_k b_l^\dagger b_l$ has 4 active modes: i, j, k, l. Each active mode will contribute a new control condition on the state of the corresponding mode. Therefore, a block-encoding circuit for a general product of fermionic ladder operators with B active modes will have an optimal rescaling factor ($\lambda = 1$), require one block-encoding ancilla and B clean ancillae, and use $4B$ T gates.

D. Linear Combinations of Fermionic Ladder Operators

As discussed in subsection III C, a block-encoding for a linear combination of operators can be constructed using the LCO framework. In this subsection, we show how to construct block-encodings of linear combinations of products of fermionic ladder operators that use fewer quantum resources than are required by an LCO construction. In particular, we give a generalized construction for a product of fermionic ladder operators plus its hermitian conjugate, however, we note that the strategies we present here are not restricted to a linear combination of hermitian conjugates.

Consider a linear combination of an individual fermionic ladder operator with its hermitian conjugate: $b_j^\dagger + b_j$. The action of this operator on the two possible occupation states of the j^{th} fermionic mode is given by:

$$(b_j^\dagger + b_j)\left|\psi_{b_j}\right\rangle = \begin{cases} p(\psi)\left|1\right\rangle & \text{when } \left|\psi_{b_j}\right\rangle \text{ is } \left|0\right\rangle \\ p(\psi)\left|0\right\rangle & \text{when } \left|\psi_{b_j}\right\rangle \text{ is } \left|1\right\rangle \end{cases} \tag{40}$$

An LCO construction using the block-encodings for these two ladder operators presented in the previous section would have a rescaling factor of $\lambda = 2$, require two block-encoding ancillae and two clean ancillae, and use 12 T gates. However, by considering the action of this operator (Eq. 40) on the active mode, a more efficient block-encoding can be constructed. A Pauli X gate can be applied to the j^{th} mode to flip the occupation and a string of Pauli Z gates can be applied on each fermionic mode with index $i < j$ to apply the potential phase flip induced by $p(\psi)$. This results in the same decomposition one would arrive at using the Jordan-Wigner transformation ($\vec{Z}X_j$). A circuit diagram for this block-encoding is shown in subfigure 3a. This block-encoding circuit has an optimal rescaling factor ($\lambda = 1$), requires zero block-encoding ancillae and zero clean ancillae, and uses zero non-Clifford operations.

For a two-body fermionic operator and its hermitian conjugate ($b_j b_l + b_l^\dagger b_j^\dagger$) with $j \neq l$ we arrive at a construction that differs from the Jordan-Wigner transfor-

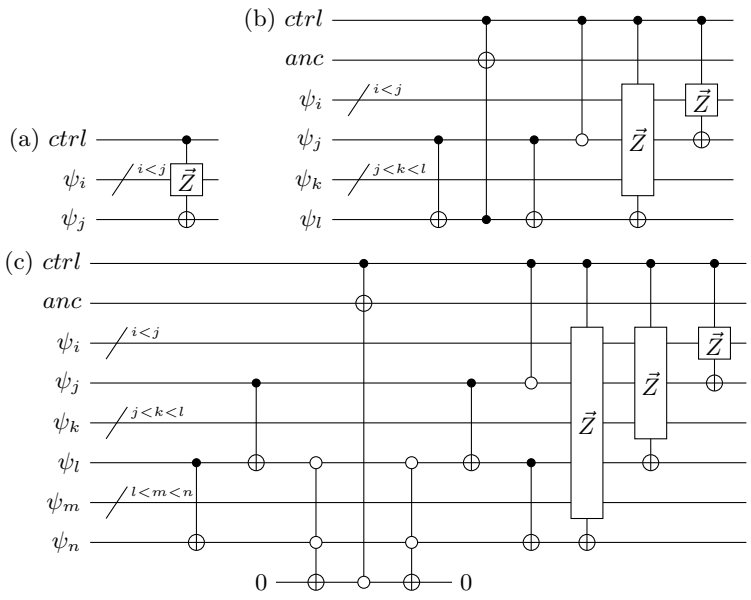

FIG. 3. **Block-Encoding Product of Fermionic Operators Plus Hermitian Conjugate** In (a), a block-encoding for the operator $b_j + b_j^\dagger$ is given. In (b), a block-encoding for the operator $b_j b_l + b_l^\dagger b_j^\dagger$ is given. In (c), a block-encoding for the operator $b_j b_l b_n + b_n^\dagger b_l^\dagger b_j^\dagger$ is given.

mation. We begin by swapping the operators in the second term using the anticommutation relations to ensure that the order in which the corresponding operators appear in each term is the same: $b_j b_l - b_j^\dagger b_l^\dagger$. In the circuit, the order in which the system is updated based on the ladder operators is fixed so it must be consistent for both terms. This reordering means that the potential phase flips caused by each ladder operator will be consistent for both terms since the order at which the operators are applied onto the system is consistent. The action of the joined operator on the possible occupation states of the j^{th} and l^{th} fermionic modes is given by:

$$(b_j b_l - b_j^\dagger b_l^\dagger) \left|\psi_{b_l} \psi_{b_j}\right\rangle = \begin{cases} -p(\psi) \left|11\right\rangle & \text{when } \left|\psi_{b_l} \psi_{b_j}\right\rangle \text{ is } \left|00\right\rangle \\ p(\psi) \left|00\right\rangle & \text{when } \left|\psi_{b_l} \psi_{b_j}\right\rangle \text{ is } \left|11\right\rangle \\ 0 & \text{when } \left|\psi_{b_l} \psi_{b_j}\right\rangle \text{ is } \left|01\right\rangle \\ 0 & \text{when } \left|\psi_{b_l} \psi_{b_j}\right\rangle \text{ is } \left|10\right\rangle \end{cases} \tag{41}$$

From this perspective, it is clear that the action of this operator on the system is determined by the parity of the two active fermionic modes.

In subfigure 3b, we give a circuit diagram for block-encoding this two-body operator. The parity of the active modes can be computed using a CNOT gate controlled on the j^{th} mode, targeting the l^{th} mode. If the control qubit is on and $\left|\psi_{b_l}\right\rangle$ is in the $\left|1\right\rangle$ state (odd parity), then the block-encoding ancilla is flipped to push that branch of the wavefunction outside of the encoded subspace. After this operation, the parity can be uncomputed, returning the qubit storing the occupation of the l^{th} mode to its original state. Next, a CZ gate that is 0-controlled on the j^{th} fermionic mode and 1-controlled on the control qubit applies the desired sign flip corresponding to the term $-b_j^\dagger b_l^\dagger$. Lastly, a series of $\vec{Z}X$ operators are applied to each active mode in the order at which the operators would be applied onto the system (right to left).

For an operator of the form $b_j b_l^\dagger + b_l b_j^\dagger$, we can construct a similar block-encoding with a slight modification. In this case, we simply flip the occupation of the l^{th} mode using a Pauli X gate before we compute the parity of the modes. The circuit diagram for block-encoding this operator is given in subfigure 4a.

Likewise, if a number operator is included in a term, this can be accounted for by including a 1-control on that mode when determining if the block-encoding ancilla should be flipped and excluding that mode from any parity computations. An example circuit diagram for an operator including a number operator is given in subfigure 4b.

The construction of these block-encoding circuits can be generalized to any operator that is a product of fermionic operators plus its hermitian conjugate. An example circuit diagram for this generalized construction is shown in Figure 5. If C represents the number of active modes excluding modes where a number operator is acting on the mode, then the CZ gate is present only if (C choose 2) is odd. These block-encoding circuits will all have optimal rescaling factors ($\lambda = 1$), require one block-encoding ancilla and $B - 1$ clean ancillae, and use $4(B - 1)$ T gates.

As mentioned, the strategy employed here to construct block-encodings of a linear combination of fermionic ladder operators is not restricted to hermitian conjugates.

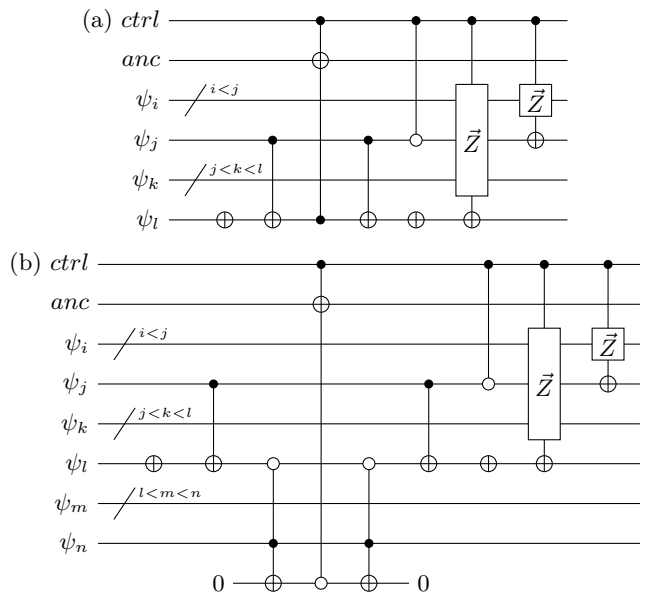

FIG. 4. **Block-Encoding Product of Fermionic Operators Plus Hermitian Conjugate (modifications)** In (a), a block-encoding for the operator $b_j b_l^\dagger + b_l b_j^\dagger$ is given. In (b), a block-encoding for the operator $b_j b_l^\dagger b_n^\dagger b_n + b_n^\dagger b_n b_l b_j^\dagger$ is given.

Consider the action of the operator $b_i b_j b_k^\dagger + b_j^\dagger b_i^\dagger b_k^\dagger$ on the occupation states of the fermionic modes. This operator will zero-out the state *unless* both $|\psi_{b_i}\rangle \oplus |\psi_{b_j}\rangle$ and $|\psi_{b_k}\rangle$ are $|0\rangle$. The implementation of the block-encoding for this operator is given in Figure 6. This block-encoding circuit has an optimal rescaling factor ($\lambda = 1$), requires one block-encoding ancilla and two clean ancillae, and uses 8 T gates.

E. Bosonic Ladder Operators

In this section, we aim to define a family of unitaries that generate block-encodings of the bosonic creation ($a_i^\dagger$) and annihilation (a_i) operators. Based on the defined action of the bosonic creation operator (Equation 6), a block-encoding unitary following the form of Eq. 12 can be defined as follows:

$$U_{a_i^\dagger} |\psi_{i_a}\rangle |0\rangle_{\text{anc}} = \begin{cases} \sqrt{\omega_i + 1/\Omega}\, |\omega_i + 1\rangle |0\rangle_{\text{anc}} + \beta |\perp\rangle & \text{when } \omega_i < \Omega \\ |\perp\rangle & \text{when } \omega_i = \Omega \end{cases} \tag{42}$$

where ω_i is the occupation number of the i^{th} bosonic mode and the operator is rescaled by a factor of $\sqrt{\Omega}$.

The desired action of the block-encoding circuit for the bosonic creation operator is to increase the occupation of the bosonic mode by 1 and rotate the block-encoding ancilla such that it has a coefficient of $\sqrt{\omega_i + 1/\Omega}$ in the $|0\rangle$ state when the initial occupation of the i^{th} bosonic mode is w_i. This can be achieved by applying an incrementer to the the bosonic mode which increases the occupation by 1 mod Ω in each branch of the wavefunction. The desired amplitudes of the block-encoding ancilla can be obtained using a series of R_y gates that are controlled on the corresponding bosonic occupation. An example circuit diagram is given in subfigure 7a. A block-encoding for the bosonic annihilation operator can be constructed similarly, but with the block-encoding ancilla rotated prior to the occupation of the mode being decremented (subfigure 7b).

The angles of the R_y gates are determined classically by the following function:

$$\theta(\omega_i) = \begin{cases} 2\cos^{-1}\left(\sqrt{\omega_i/\Omega}\right) & \text{when } \omega_i < \Omega \\ \pi & \text{when } \omega_i = \Omega \end{cases} \tag{43}$$

where ω_i here in comparison with $\omega_i + 1$ in Eq. 42 is due to the occupation of the state being updated prior to the controlled rotations.

An implementation of a controlled incrementer circuit is given in [48] which requires $\lceil \log_2 \Omega \rceil - 1$ clean ancillae and $3\lceil \log_2 \Omega \rceil$ T gates. The series of uniformly controlled rotations can be decomposed via the protocol given in Möttönen et. al [49]. A discussion of different schemes to construct a *controlled* set of uniformly controlled rotations is given in Appendix G. The resource estimates quoted in the remainder of this work assume the decomposition that requires (at most) $\Omega + 3$ uncontrolled rotations and $4\lceil \log_2 \Omega \rceil$ T gates (Figure **??**). In total, this block-encoding has an optimal rescaling factor ($\sqrt{\Omega}$), requires one block-encoding ancilla and $\lceil \log_2 \Omega \rceil$ clean ancillae, and uses $7\lceil \log_2 \Omega \rceil$ T gates and (at most) $\Omega + 3$ arbitrary rotations.

F. Products of Bosonic Ladder Operators

In this subsection, we discuss constructing block-encodings for a product of bosonic ladder operators acting on the same mode. Unlike fermions, multiple bosons can occupy the same bosonic mode and a series of bosonic ladder operators can be applied to the state without necessarily zeroing-out the state. Products of bosonic ladder operators acting on different modes can be block-encoded using the methods described in subsection III D.

Consider the desired action of a block-encoding for series of R bosonic creation operators acting on the i^{th} bosonic mode:

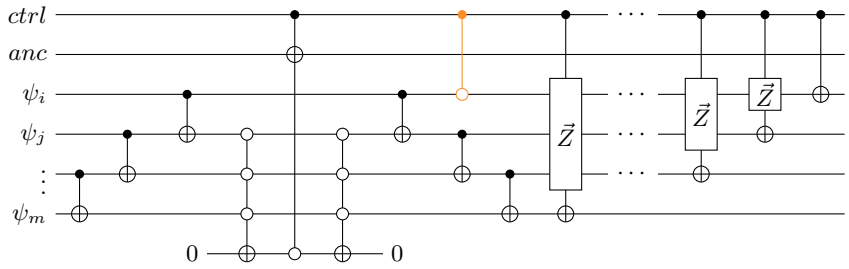

FIG. 5. **Generalized Block-Encoding for Product of Fermionic Operators Plus Hermitian Conjugate** A block-encoding for the operator $b_i b_j ... b_m + b_m^\dagger ... b_j^\dagger b_i^\dagger$ is given. The CZ gate highlighted in orange is present only if the swapping on the order of the operators in the second term induces a negative sign on the second term (an odd number of swaps are required to reorder the ladder operators). Block-encodings for similar operators such as those that include number operators or different arrangements of the creation and annihilation operators can be accounted for using the modifications shown in Figure 4.

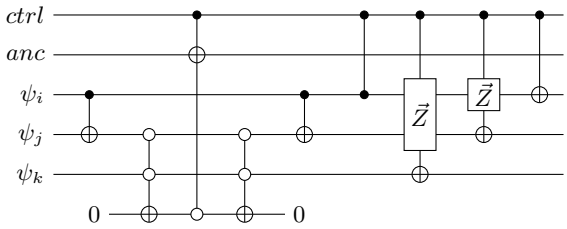

FIG. 6. **Block-Encoding for Linear Combination of Non-Conjugate Fermionic Operators** A block-encoding for the operator $b_i b_j b_k^\dagger + b_j^\dagger b_i^\dagger b_k^\dagger$ is given.

$$U_{(a_i^\dagger)^R} |\psi_{i_a}\rangle |0\rangle_{\text{anc}} = \begin{cases} \prod_{r=1}^{R} \sqrt{(\omega_i + r)/\Omega} \, |\omega_i + R\rangle |0\rangle_{\text{anc}} + \beta |\perp\rangle & \text{when } \omega_i \leq \Omega - R \\ |\perp\rangle & \text{when } \omega_i > \Omega - R \end{cases} \tag{44}$$

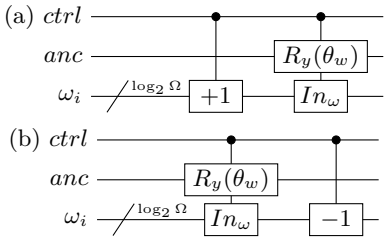

FIG. 7. **Bosonic Ladder Operator Block-Encoding** In (a), a block-encoding for the bosonic creation operator $(a_i^\dagger)$ is given. In (b), a block-encoding for the bosonic annihilation operator (a_i) is given.

A block-encoding for this operator can be achieved by updating the occupation of the bosonic mode by $+R$ and then performing a single series of rotations. A circuit diagram for this construction is shown in subfigure 8a.

The angles of the R_y rotations are determined classically by the following function:

$$\theta(\omega_i, R) = \begin{cases} 2\cos^{-1}\left(\prod_{r=0}^{R-1} \sqrt{(\omega_i - r)/\Omega}\right) & \text{when } \omega_i \leq \Omega - R \\ \pi & \text{when } \omega_i > \Omega - R \end{cases} \tag{45}$$

where $\omega_i - r$ here in comparison with $\omega_i + r$ in Eq. 44 is due to the occupation of the state being updated prior

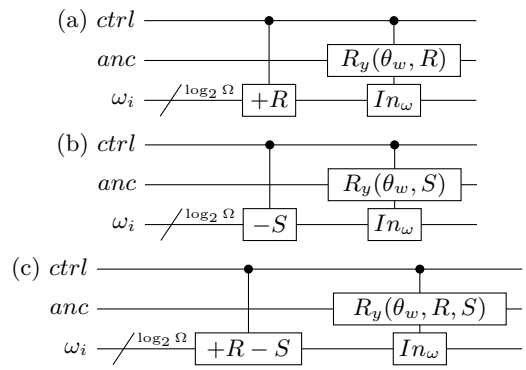

FIG. 8. **Block-Encoding Product of Bosonic Ladder Operators** In (a), a block-encoding for the operator $(a_i^\dagger)^R$ is given. In (b), a block-encoding for the operator $(a_i)^S$ is given. In (c), a block-encoding for the operator $(a_i^\dagger)^R(a_i)^S$ is given.

to the controlled rotations.

A block-encoding for a bosonic annihilation operator being applied S times can be achieved using a similar construction shown in subfigue 8b. The occupation of the mode is first decreased by S and then the uniformly controlled rotations are applied to pick up the corresponding coefficient on the block-encoding ancilla. The function to determine the rotation angles is given by:

$$\theta(\omega_i, S) = \begin{cases} 2\cos^{-1}\left(\prod_{s=1}^{S}\sqrt{(\omega_i+s)/\Omega}\right) & \text{when } \omega_i \geq S \\ \pi & \text{when } \omega_i < S \end{cases} \tag{46}$$

Likewise, a block-encoding for an operator of the form $(a_i^\dagger)^R(a_i)^S$ can be constructed using a similar form (subfigue 8c). The occupation of the mode is updated by a value of $+R-S$ and then the uniformly controlled rotations are applied to pick up the corresponding coefficient on the block-encoding ancilla. The function to determine the rotation angles is given by:

$$\theta(\omega_i, R, S) = \begin{cases} 2\cos^{-1}\left(\prod_{r=0}^{R-1}\sqrt{(\omega_i-r)/\Omega}\prod_{s=1}^{S}\sqrt{(\omega_i-R+s)/\Omega}\right) & \text{when } S \leq \omega_i \leq \Omega - R \\ \pi & \text{Otherwise} \end{cases} \tag{47}$$

Incrementing a quantum register by a classical value can be implemented in multiple ways with different implementations having different space-time tradeoffs. The resource estimates presented in this work assume the compilation given in Figure 24, which requires (at most) $3\lceil\log_2 \Omega\rceil$ T gates. In total, these block-encoding circuits have a rescaling factor of $\lambda = \Omega^{(R+S)/2}$, require one block-encoding ancilla and $\lceil\log_2 \Omega\rceil$ clean ancillae, and use (at most) $7\lceil\log_2 \Omega\rceil$ T gates and $\Omega + 3$ arbitrary rotations.

G. Linear Combinations of Bosonic Ladder Operators

In this subsection, we discuss constructing block-encodings for a linear combination of a product of bosonic ladder operators with its hermitian conjugate.

Consider the block-encodings for the operators $a_i^\dagger$ and a_i. An efficient block-encoding of the operator $a_i^\dagger + a_i$ can be constructed using the symmetry between the two individual block-encodings. To apply the creation operator, the occupancy is increased by 1 and then the rotations are applied, whereas for the annihilation operator, the rotations are applied first and then the occupancy is decreased by 1. Since the desired rotation angles are independent of which operator is being applied (Eq. 43), we can apply these rotations once for both operators. However, one additional block-encoding ancilla is required to index between the two operators. An example circuit diagram for this construction is shown in subfigue 9a. A block-encoding for the generalized operator $\left((a_i^\dagger)^R(a_i)^S + (a_i^\dagger)^S(a_i)^R\right)$ can be constructed similarly and is shown in subfigure 9b.

Likewise, this construction can be generalized to a linear combination of a product of bosonic operators acting on different modes plus its hermitian conjugate. An example circuit diagram for this construction is shown in Figure 10. For an operator acting on B bosonic modes, these block-encodings have a rescaling factor of $\lambda = 2\Omega^{P/2}$ where P is the sum of the exponents of the operators in a term: $P = \sum_{b=0}^{B-1}(R_b + S_b)$. Additionally, they require $B+1$ block-encoding ancillae and $\lceil\log_2 \Omega\rceil+1$ clean ancillae, and use (at most) double check this $12B\lceil\log_2 \Omega\rceil - 4(B-1)$ T gates and $B(\Omega+3)$ arbitrary rotations.

H. Terms with Fermionic and Bosonic Ladder Operators

In the previous subsections, we discussed strategies for compiling block-encodings of different operators acting on either fermionic or bosonic modes. In this subsection, we discuss block-encodings for operators that contain operators that act on both fermionic and bosonic operators.

Consider the action of the operator: $b_i a_j + a_j^\dagger b_i^\dagger$ on the respective fermionic and bosonic modes. When the

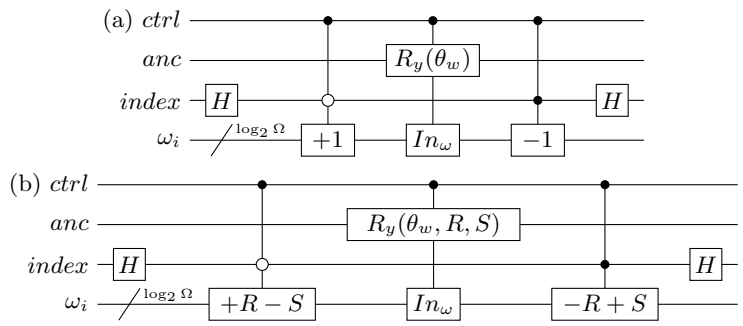

FIG. 9. **Block-Encoding Product of Bosonic Ladder Operators Plus Hermitian Conjugate** In (a), a block-encoding for the operator $(a_i^\dagger + a_i)$ is given. In (b), a block-encoding for the operator $\left((a_i^\dagger)^R(a_i)^S + (a_i^\dagger)^S(a_i)^R\right)$ is given.

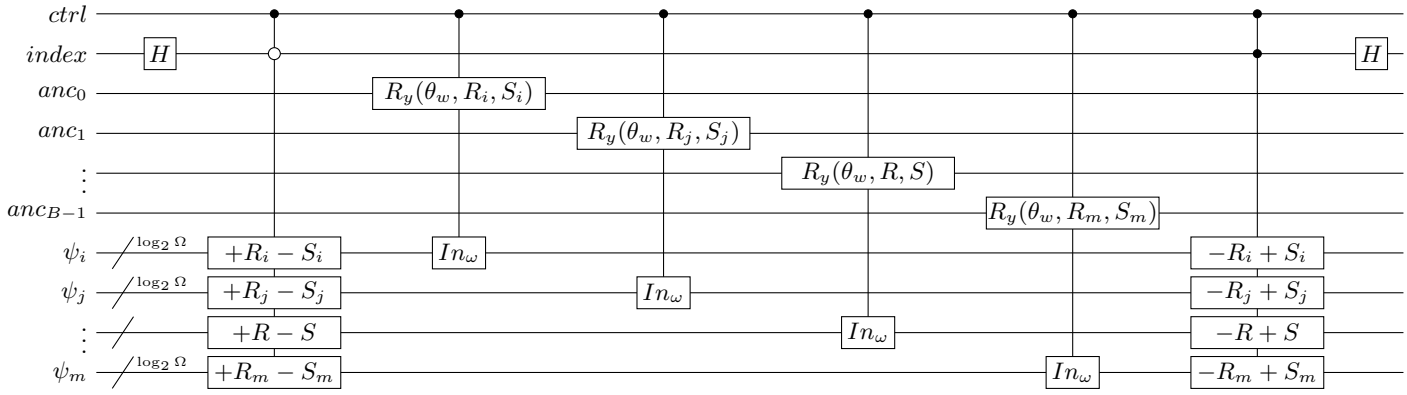

FIG. 10. **Generalized Block-Encoding Product of Bosonic Ladder Operators Plus Hermitian Conjugate** A block-encoding for the operator $((a_i^\dagger)^{R_i} + a_i^{S_i})((a_j^\dagger)^{R_j} + a_j^{S_j})...((a_m^\dagger)^{R_m} + a_m^{S_m}) + h.c.$ is given.

fermionic mode is occupied, only the operator $b_i a_j$ will act nontrivially. Meanwhile if the fermionic mode is unoccupied, only the operator $a_j^\dagger b_i^\dagger$ will act nontrivially. Therefore, the occupation of the fermionic mode can be used to dictate which bosonic operator should be applied to the system. After the appropriate bosonic operator is applied, the fermionic system is updated. A circuit diagram for this block-encoding in given in subfigue 11a. This block-encoding circuit will have a rescaling factor of $\lambda = \sqrt{\Omega}$, requires one block-encoding ancillae and $\lceil \log_2 \Omega \rceil + 1$ clean ancillae, and uses double check this $12\lceil \log_2 \Omega \rceil - 4$ T gates and (at most) $\Omega + 3$ arbitrary rotations.

This strategy can be employed for other combinations of bosonic and fermionic ladder operators. In the full Yukawa model the operator $b_i^\dagger b_j^\dagger a_k^\dagger + a_k b_j b_i$ is used to model the process of a boson being annihilated to form a fermion and antifermion and a fermion and antifermion being annihilated to form a boson. A block-encoding for this operator is shown in subfigue 11b. When constructing the circuit, the occupation of either fermionic mode can be used to determine which bosonic operator is applied to the system. This block-encoding circuit will have a rescaling factor of $\lambda = \sqrt{\Omega}$, requires two block-encoding ancillae and $\lceil \log_2 \Omega \rceil + 1$ clean ancillae, and uses double check this $12\lceil \log_2 \Omega \rceil$ T gates and (at most) $\Omega + 3$ arbitrary rotations.

A similar block-encoding can be constructed for the operator $b_i^\dagger b_j^\dagger a_k^\dagger a_l^\dagger + a_l a_k b_j b_i$, which appears in the full Yukawa model. An example circuit diagram for this block-encoding is shown in subfigue 11c. This block-encoding circuit will have a rescaling factor of $\lambda = \Omega$, requires three block-encoding ancillae and $\lceil \log_2 \Omega \rceil + 1$ clean ancillae, and uses double check this $24\lceil \log_2 \Omega \rceil - 8$ T gates and (at most) $2\Omega + 6$ arbitrary rotations.

V. RESULTS

In this section, we numerically evaluate the quantum resources required for block-encoding various operators using LOBE. The spacetime costs that we analyze include: the number of T gates, the number of non-Clifford single qubit rotations, the number of block-encoding ancillae, the maximum number of qubits required, and the rescaling factor imposed on the resulting block-encoding.

We compare the LOBE constructions presented in Section IV to techniques that first expand the ladder operators in the Pauli basis and then block-encode the resulting linear combinations of Pauli operators using LCU.

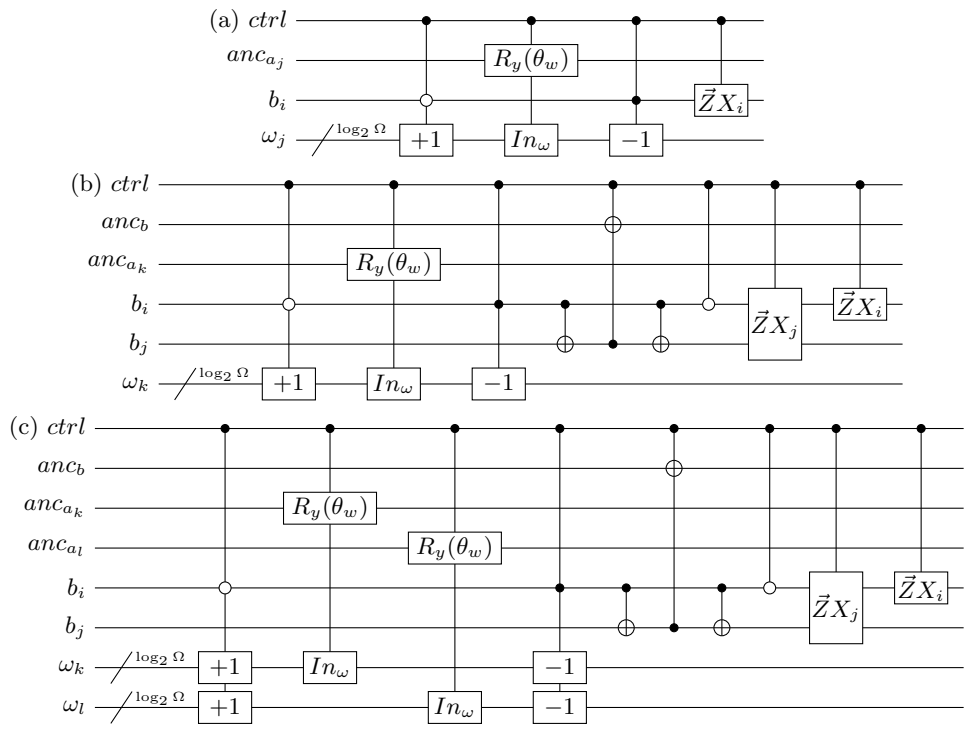

FIG. 11. **Block-Encoding Terms** In (a), a block-encoding for the operator $b_i^\dagger a_j^\dagger + a_j b_i$ is given. In (b), a block-encoding for the operator $b_i^\dagger b_j^\dagger a_k^\dagger + a_k b_j b_i$ is given. In (c), a block-encoding for the operator $b_i^\dagger b_j^\dagger a_k^\dagger a_l^\dagger + a_l a_k b_j b_i$ is given.

The Jordan-Wigner transformation [47] is used to expand fermionic ladder operators in the basis of Pauli operators, while the Standard Binary encoding [34] is used to expand bosonic ladder operators.

The first Pauli-based method we compare against will be referred to as "Pauli Expansion". In this method, the respective Pauli transformations are applied to all ladder operators within a term and the product is fully expanded. This results in a single linear combination of Pauli operators, which is then block-encoded using the LCU framework.

The second Pauli-based method we compare against will be referred to as "Piecewise Pauli". In this method, each product of ladder operators acting on a single mode is expanded in the Pauli basis and then block-encoded using the LCU framework. These block-encodings of the ladder operators acting on each mode are then combined together to produce a block-encoding of the full operator using the techniques described in subsections III C and III D.

In the following subsections, we benchmark these three different block-encoding methods for various types of operators. First, we analyze the associated spacetime costs for block-encoding several classes of operators which appear in many second-quantized models. Then, we present the spacetime costs for block-encoding Hamiltonians associated with various models that arise in quantum field theory. These models include Hamiltonians derived from two non-relativistic models - the quartic harmonic oscillator and the static massive Yukawa - and two fully relativistic models - ϕ^4 theory and the full massive Yukawa.

For all block-encodings, we will assume that all terms are mode ordered (Eq. 9) and that the identity term is removed. The identity term is removed since it artificially inflates the rescaling factor and only results in a constant shift in the spectrum of the Hamiltonian, however, if required it can be included with a negligible increase in spacetime quantum resources.

The results presented in this section are generated using a software library called LOBE which has been made open-source []. The circuits are implemented using Cirq [50] and block-encodings are numerically verified for circuits of up to 18 qubits. OpenParticle [51] is used to construct and manipulate the operators in terms of fermionic, antifermionic, and bosonic ladder operators. The Symmer software library [] is used for various subroutines, including expanding the ladder operators in the Pauli basis. The software library [52] is used to determine the rotation angles required for arbitrary state-preparation using Grover-Rudolph.

A. Components

In this subsection, we numerically benchmark the spacetime cost of LOBE constructions for three different sets of operators. We compare the cost of the LOBE constructions with the two methods outlined above that

require expanding the ladder operators in the Pauli basis.

The first set of operators we examine are described by a product of fermionic annihilation operators acting on different modes plus its hermitian conjugate: $b_0 b_1 \dots b_{B-1} + h.c.$. The numerical spacetime costs of the three block-encoding methods for these operators are shown as a function of the number of active modes (B) in Figure 12.

Notably, the spacetime costs for LOBE and "Pauli Expansion" are identical for $B = 1$ and $B = 2$. However, the number of required T gates scales exponentially with B for "Pauli Expansion", yet scales linearly for LOBE. Additionally, the LOBE constructions only require a single block-encoding ancilla, independent of B,while both Pauli-based methods require a number of block-encoding ancillae that scale linearly with B.

The second set of block-encodings we consider is those that encode a single bosonic annihilation operator (a) with an increasing bosonic occupation cutoff (Ω). Since there is only a single operator, both Pauli methods result in the same construction therefore we refer to these constructions by "Pauli (LCU)". The numerical spacetime costs and rescaling factors of the Pauli (LCU) and LOBE block-encodings are shown in Figure 13.

The LOBE constructions result in block-encodings with fewer required resources for all metrics as compared to the Pauli (LCU) method. Notably, the number of required T gates scales logarithmically with Ω for LOBE, yet scales roughly linearly for the Pauli (LCU) method. Additionally, the number of block-encoding ancillae is constant (1) regardless of Ω for LOBE, yet scales logarithmically for the Pauli (LCU) method. Lastly, the LOBE construction results in a rescaling factor that matches the L2 norm (optimal), while the Pauli (LCU) construction results in a larger rescaling factor.

The third set of operators we analyze are given as a linear combination of a product of bosonic annihilation operators acting on different modes plus its hermitian conjugate: $a_0 a_1 \dots a_{B-1} + h.c.$. The numerical spacetime costs of the various block-encoding constructions for these operators with $\Omega = 3$ are shown in Figure 14.

Notably, the overall time-complexity scales linearly with B for the LOBE and "Piecewise Pauli" constructions, while it scales exponentially for "Pauli Expansion". For the space complexity metrics, all methods scale linearly with B, yet the LOBE constructions have both the smallest prefactor and numerical values. Finally, the LOBE construction results in the lowest rescaling factor of all constructions.

In certain cases - such as when the number of active modes is small or the bosonic occupation cutoff is low - the "Pauli Expansion" construction can lead to the lowest spacetime costs. However, when the number of active modes is large or the bosonic occupation cutoff is high, the LOBE constructions lead to the lowest spacetime costs. For all components analyzed here, the "Piecewise Pauli" constructions have similar asymptotic scalings compared to LOBE, but result in larger numerical spacetime costs. For this reason, the "Piecewise Pauli" construction will be omitted when benchmarking full systems.

B. Quartic Harmonic Oscillator Results

The quartic harmonic oscillator [53–55] is an extension of the standard harmonic oscillator with an additional non-linearity of the form $g\varphi^4$, where g is an arbitrary constant that dictates the strength of the interaction, and ϕ is a bosonic field. This leads to the Hamiltonian $H = \frac{1}{2}\dot{\phi}^2 + \frac{m}{2}\phi^2 + gm^3\phi^4$, where ϕ is composed of a single mode. m is the mass of the bosonic particle, and $\dot{\phi}$ represents the time derivate of the field.

In a second-quantized, dimensionless form, the Hamiltonian can be written as:

$$H = a^\dagger a + g\left(a + a^\dagger\right)^4 \tag{48}$$

where there is only one bosonic mode.

After expanding the product, normal ordering all terms, and removing the constant offset, this Hamiltonian can be written as a linear combination of 8 terms consisting of three pairs of operators plus their hermitian conjugates and two operators that are their own hermitian conjugates:

$$\begin{aligned} H =&(12g+1)a^\dagger a + 6g{a^\dagger}^2 a^2 + 6g\left({a^\dagger}^2 + a^2\right) \\ &+4g\left({a^\dagger}^3 a + a^\dagger a^3\right) + g\left({a^\dagger}^4 + a^4\right) \end{aligned} \tag{49}$$

In Figure V B, we show the scaling of the spacetime costs associated with both the LOBE and Pauli Expansion block-encodings as a function of the bosonic occupation cutoff (Ω). For small values of Ω, the Pauli Expansion block-encodings result in lower spacetime costs, however, the LOBE constructions have favorable scaling and therefore a crossover point is seen.

For the time-complexity, this crossover occurs at $\Omega = 15$ for the number of T gates and $\Omega = 7$ for the number of non-Clifford rotations. For the space-complexity, this crossover occurs at $\Omega = 7$ for the number of block-encoding ancillae and $\Omega = 31$ for the maximum number of qubits. Finally, for the rescaling factor, this crossover occurs at $\Omega = 31$.

C. Massive Static Yukawa Results

The next simplest model that we consider is a non-relativistic approximation to the Yukawa model, which is commonly referred to as the massive static Yukawa model [56]. This model is taken as the limit of the full Yukawa model of infinitely heavy fermions, and bosons at rest relative to the fermions that emit/absorb them.

The second-quantized Hamiltonian for this model is given by:

$$H = C_f b^\dagger b + C_b a^\dagger a + g b^\dagger b\left(a + a^\dagger\right) \tag{50}$$

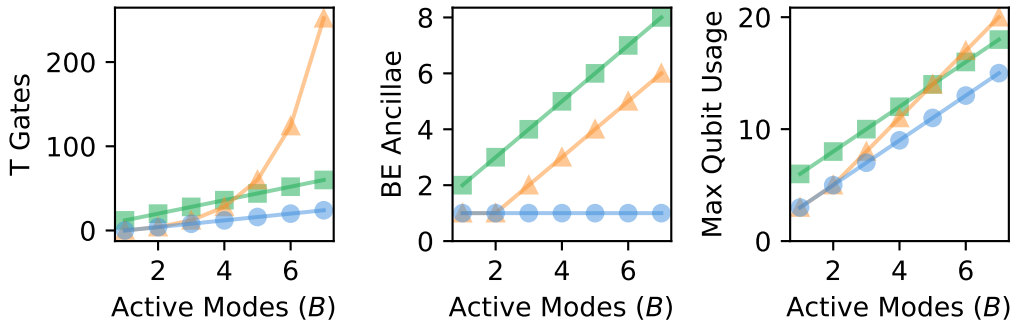

FIG. 12. **Spacetime Cost to Block-Encode** $O = b_0 b_1 \dots b_{B-1} + h.c.$ The number of T gates (left), block-encoding ancillae (middle), and maximum number of qubits used (right) are shown as a function of the number of active modes (B). Results for "Pauli Expansion" are shown as the orange triangles, results for "Piecewise Pauli" are shown as the green squares, and results for LOBE are shown as the blue circles. For this operator, all block-encodings use zero non-Clifford rotations. Both "Pauli Expansion" and LOBE have optimal rescaling factors ($\lambda = 1$) while "Piecewise Pauli" has a rescaling factor of $\lambda = 2$.

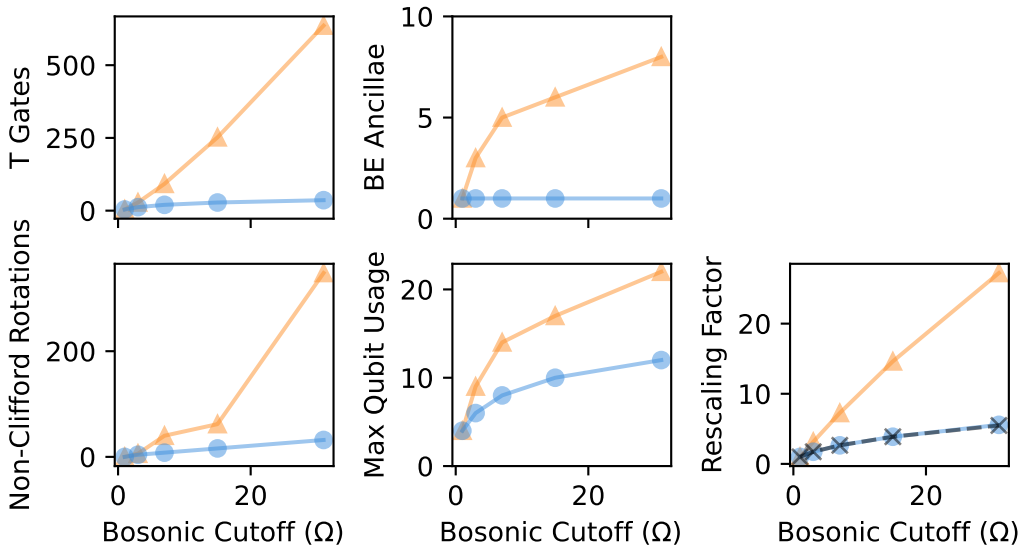

FIG. 13. **Spacetime Cost to Block-Encode Bosonic Annihilation Operator** The number of T gates (upper-left), number of non-Clifford rotations (lower-left), block-encoding ancillae (upper-middle), maximum number of qubits used (lower-middle), and rescaling factor (lower-right) are shown as a function of the bosonic occupation cutoff (Ω). Results for the Pauli (LCU) method are shown as the orange triangles and results for LOBE are shown as the blue circles. The optimal rescaling factor, which is given by the L2 norm of the matrix representing the bosonic annihilation operator with fixed Ω, is shown as the dashed black crosses.

where there is only one fermionic mode and one bosonic mode.

As this model is restricted to one fermionic and one bosonic mode, we compare the spacetime costs of the different block-encoding methods as a function of the bosonic occupation cutoff (Ω). Similar to the quartic harmonic oscillator, the Pauli Expansion method results in lower spacetime costs when the bosonic cutoff is low. However, the LOBE constructions have better asymptotic scaling with resect to Ω and crossover points exist for all metrics, above which LOBE results in lower cost constructions.

For the time-complexity, this crossover point occurs at $\Omega = 3$ for the number of T gates and the LOBE constructions require fewer non-Clifford rotations at all values of Ω. For the space-complexity, this crossover point occurs at $\Omega = 7$ for the number of block-encoding ancillae and $\Omega = 15$ for the maximum number of qubits. Finally, for the rescaling factor, this crossover point occurs at $\Omega = 15$.

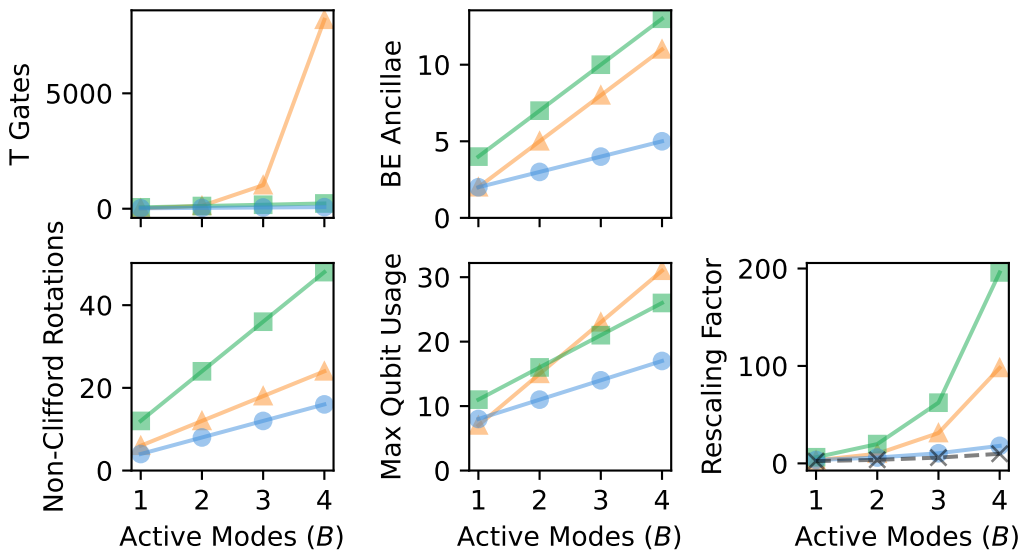

FIG. 14. **Spacetime Cost to Block-Encode** $O = a_0 a_1 \dots a_{B-1} + h.c.$ The number of T gates (upper-left), number of non-Clifford rotations (lower-left), block-encoding ancillae (upper-middle), maximum number of qubits used (lower-middle), and rescaling factor (lower-right) are shown as a function of the number of active modes (B). The bosonic cutoff is fixed to $\Omega = 3$ for all data points. Results for "Pauli Expansion" are shown as the orange triangles, results for "Piecewise Pauli" are shown as the green squares, and results for LOBE are shown as the blue circles. The optimal rescaling factor, which is given by the L2 norm of the matrix representing the operator with $\Omega = 3$, is shown as the dashed black crosses.

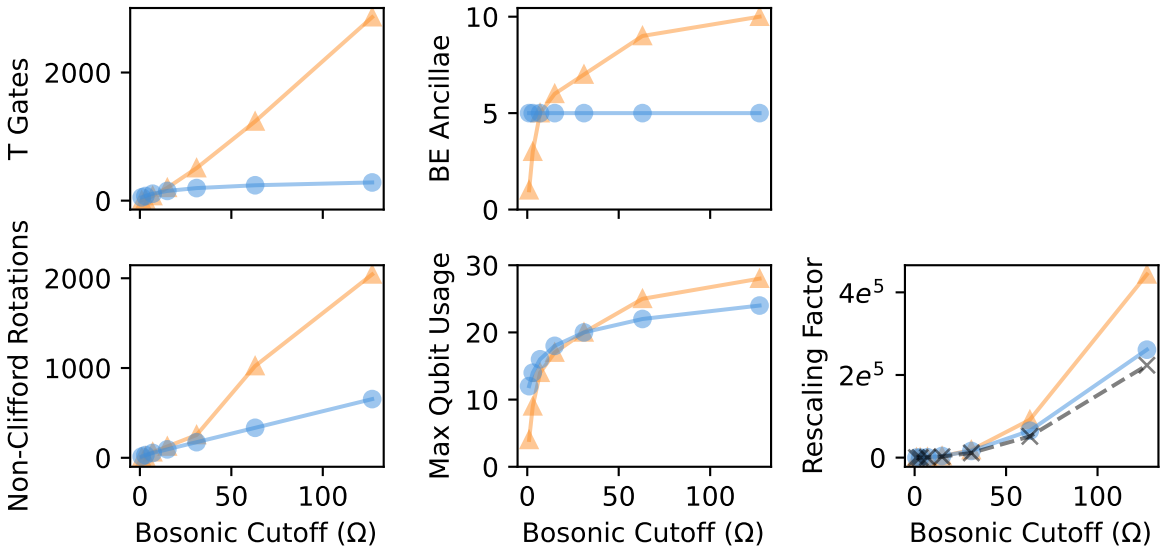

FIG. 15. **Quartic Harmonic Oscillator** The number of T gates (upper-left), number of non-Clifford rotations (lower-left), block-encoding ancillae (upper-middle), maximum number of qubits used (lower-middle), and rescaling factor (lower-right) are shown as a function of the bosonic occupation cutoff (Ω). The parameter g is set to 1 for all data points. Results for the Pauli Expansion method are shown as the orange triangles and results for LOBE are shown as the blue circles. The optimal rescaling factor, which is given by the L2 norm of the Hamiltonian, is shown as the dashed black crosses.

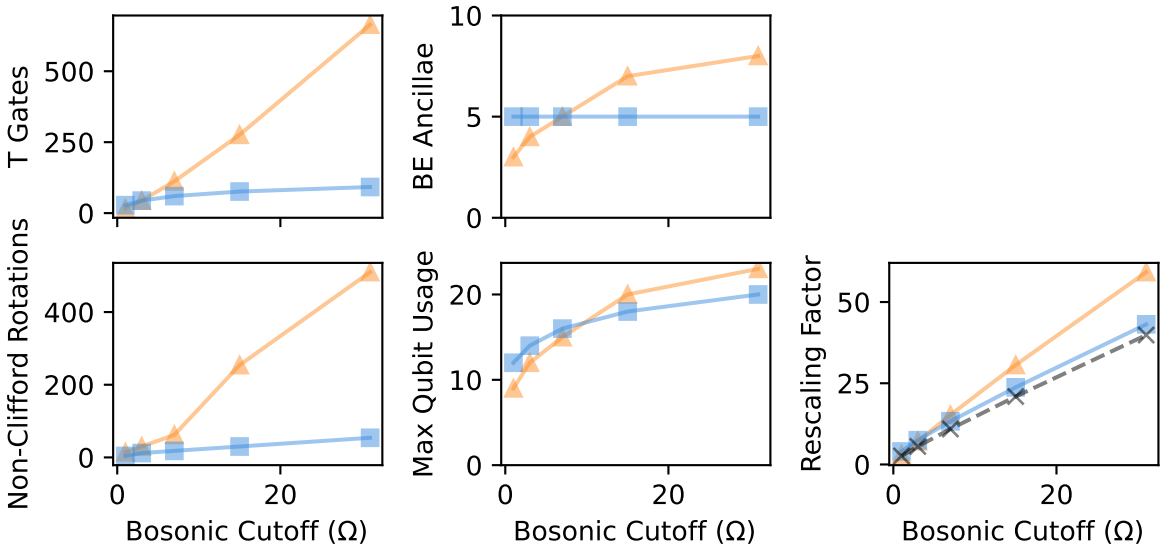

FIG. 16. **Static Massive Yukawa** The number of T gates (upper-left), number of non-Clifford rotations (lower-left), block-encoding ancillae (upper-middle), maximum number of qubits used (lower-middle), and rescaling factor (lower-right) are shown as a function of the bosonic occupation cutoff (Ω). The parameters C_f, C_b, and g are set to 1 for all data points. Results for the Pauli Expansion method are shown as the orange triangles and results for LOBE are shown as the blue circles. The optimal rescaling factor, which is given by the L2 norm of the Hamiltonian, is shown as the dashed black crosses.

D. ϕ^4 Results

Any field theory is defined at the level of a Lagrangian. For ϕ^4 theory is given as

$$\mathcal{L} = \frac{1}{2}\left(\partial_\mu \phi\right)^2 - \frac{m^2}{2}\phi^2 - g\phi^4. \tag{51}$$

To obtain a Hamiltonian from a Lagrangian, a Legendre transformation is performed, in which an explicit set of coordinates must be chosen. The set of coordinates used in this paper, which lead to the simplest forms of the corresponding Hamiltonians, are front form (lightfront) coordinates [57]. A discussion on lightfront coordinates will be given in appendix I.

The ϕ^4 Hamiltonian can be written as:

$$H = \sum_i c_i a_i^\dagger a_i + \sum_{ijkl} c_{ijkl}\left(a_i^\dagger a_j^\dagger a_k^\dagger a_l + h.c.\right) + \sum_{ijkl} c_{ijkl} a_i^\dagger a_j^\dagger a_k a_l \tag{52}$$

where the values of the coefficients are omitted for brevity, but can be determined numerically.

Unlike the Quartic Harmonic Oscillator and the Static Massive Yukawa model, ϕ^4 theory is defined based on a discretization of momentum modes. In lightfront field theories, it is common to refer to the resolution of the model, which dictates the size of the momentum grid or the total number of momentum modes. For those unfamiliar, the resolution can be thought of analogously to the molecular orbital basis in quantum chemistry simulations. A further discussion on the physical meaning of resolution will be given in appendix I.

Since the previous results establish that LOBE constructions are generally preferred when the bosonic cutoff is large, here we chose to look at the spacetime cost as a function of the total number of momentum modes. In Figure V D, we show the scaling of the spacetime costs associated with both the LOBE and Pauli Expansion block-encodings as a function of the number of modes. For these estimates, the bosonic cutoff is fixed to $\Omega = 3$.

When the number of modes is small (low resolution), the LOBE and Pauli Expansion constructions require similar spacetime costs, with the Pauli Expansion block-encodings requiring one fewer block-encoding ancilla when there are only two modes. For higher resolutions (more modes), the LOBE constructions require signficantly fewer T gates and non-Clifford rotations, in addition to requiring fewer block-encoding ancillae, using fewer total qubits, and having smaller rescaling factors. Notably, when 7 momentum modes are used, the time cost for the LOBE construction is approximately two orders of magnitude smaller than the Pauli Expansion construction.

E. Full Yukawa Results

Here, we examine the spacetime cost to block-encode the full Yukawa model which includes interactions between fermionic, antifermionic, and bosonic modes. This is a theory of interacting fermions and bosons which can be used as a model of the strong nuclear force between hadrons. The Lagrangian for this model is given as:

$$\mathcal{L} = \bar{\psi}\left(i\gamma^\mu \partial_\mu - m\right)\psi + \frac{1}{2}\partial_\mu\phi\partial^\mu\phi - \frac{1}{2}\mu^2\phi^2 - g\bar{\psi}\psi\phi, \tag{53}$$

where m_f is the mass of the fermion, m_b is the mass of the boson, and g describes the strength of the interaction. Unlike ϕ^4 theory, the Yukawa model involves multiple types of fields including a fermionic (Dirac) field (ψ) and the conjugate (antifermionic) Dirac ($\bar{\psi}$) field, in addition to the bosonic field (ϕ).

The Hamiltonian associated with this Lagrangian can be written in second-quantization as:

$$\begin{aligned} H = &\sum_i c_i b_i^\dagger b_i + \sum_i c_i d_i^\dagger d_i + \sum_i c_i a_i^\dagger a_i + \\ &\sum_{ijk} c_{ijk}\left(b_i^\dagger b_j a_k^\dagger + h.c.\right) + \sum_{ijk} c_{ijk}\left(d_i^\dagger d_j a_k^\dagger + h.c.\right) + \\ &\sum_{ijk} c_{ijk}\left(b_i^\dagger d_j^\dagger a_k + h.c.\right) + \sum_{ijkl} c_{ijkl} b_i^\dagger b_j a_k^\dagger a_l + \\ &\sum_{ijkl} c_{ijkl} d_i^\dagger d_j a_k^\dagger a_l + \sum_{ijkl} c_{ijkl}\left(b_i^\dagger d_j^\dagger a_k a_l + h.c.\right) \end{aligned} \tag{54}$$

where the values of the coefficients can be determined by classical preprocessing.

In Figure V E, the spacetime costs for both the Pauli Expansion and LOBE block-encodings are shown as a function of the number of momentum modes. For this model, the LOBE constructions result in fewer required quantum resources for all metrics. Notably, the number of T gates and number of non-Clifford rotations required for LOBE is approximately two orders of magnitude smaller than the Pauli Expansion construction when the number of modes is 12.

VI. CONCLUSIONS

In this work, we detail a framework which we refer to as LOBE (Ladder Operator Block-Encoding) which constructs block-encoding circuits for operators written in second-quantization. We give explicit compilations for operators written as linear combinations of products of ladder operators acting on both fermionic and bosonic modes. With this framework, many quantum operators of interest - such as Hamiltonians - can be block-encoded directly in their second-quantized form. This avoids expanding operators in the Pauli basis prior to block-encoding which introduces a signficant overhead.

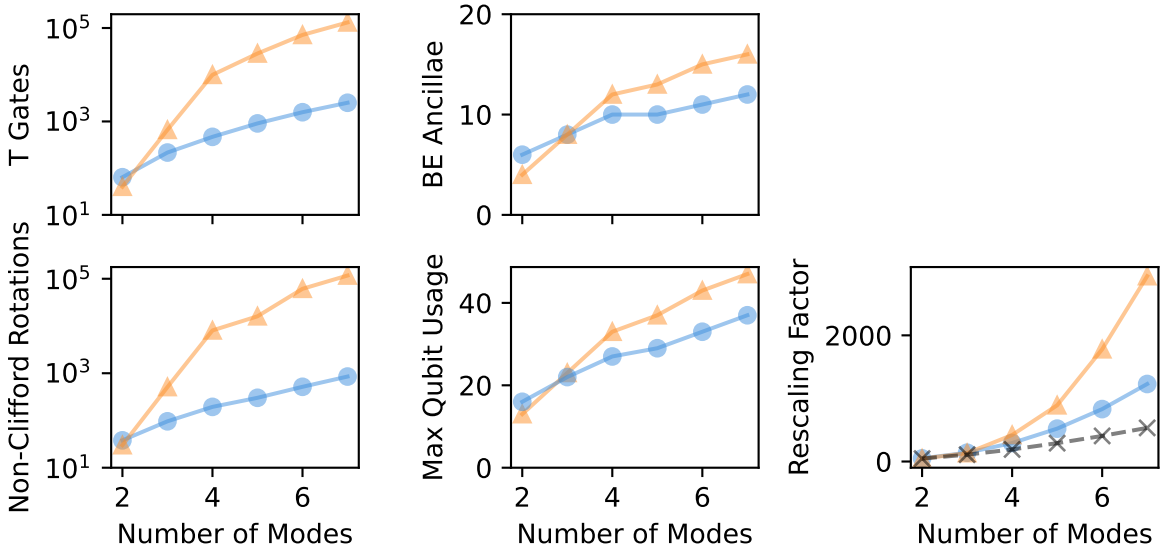

FIG. 17. ϕ^4 The number of T gates (upper-left), number of non-Clifford rotations (lower-left), block-encoding ancillae (upper-middle), maximum number of qubits used (lower-middle), and rescaling factor (lower-right) are shown as a function of the number of momentum modes. The bosonic cutoff is fixed to $\Omega = 3$ and the parameters g and m_b are set to 1 for all data points. Results for the "Pauli - Expansion" method are shown as the orange triangles and results for LOBE are shown as the blue circles. The optimal rescaling factor, which is given by the L2 norm of the Hamiltonian, is shown as the dashed black crosses.

Additionally, we provide analytical and numerical costs for the relevant quantum resources required by LOBE for various operators of interest. We compare our constructions to those that require the Jordan-Wigner and Standard Binary transformations to express ladder operators in the Pauli basis. Our numerical results show that in most relevant cases, the LOBE framework produces block-encodings that require signficantly fewer T gates, non-Clifford rotations, block-encoding ancillae and total number of qubits and result in block-encodings with smaller rescaling factors. In addition, the LOBE framework has better asymptotic scaling with respect to several parameters such as the maximum occupation of bosonic modes, the number of momentum modes, and the locality of the operator. Notably, the LOBE constructions result in exponentially fewer non-Clifford operations for block-encoding products of ladder operators with respect to the locality as compared to a naive expansion of the operator in the Pauli basis.

VII. ACKNOWLEDGEMENTS

Wop, wop, wop, wop, wop, Dot, fuck 'em up
Wop, wop, wop, wop, wop, I'ma do my stuff [58]

[1] R. P. Feynman, Simulating physics with computers, in *Feynman and computation* (cRc Press, 2018) pp. 133–153.
[2] M. Suzuki, Generalized trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems, Communications in Mathematical Physics **51**, 183 (1976).
[3] N. Hatano and M. Suzuki, Finding exponential product formulas of higher orders, in *Quantum annealing and other optimization methods* (Springer, 2005) pp. 37–68.
[4] S. Lie, *Theorie der transformationsgruppen*, Vol. 3 (Teubner, 1893).
[5] H. F. Trotter, On the product of semi-groups of operators, Proceedings of the American Mathematical Society **10**, 545 (1959).
[6] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu, Theory of trotter error with commutator scaling, Physical Review X **11**, 011020 (2021).
[7] L. Lin, Lecture notes on quantum algorithms for scientific computation, arXiv preprint arXiv:2201.08309 (2022).
[8] D. Poulin, A. Kitaev, D. S. Steiger, M. B. Hastings, and M. Troyer, Quantum algorithm for spectral measurement with a lower gate count, Physical review letters **121**, 010501 (2018).
[9] G. H. Low and I. L. Chuang, Hamiltonian simulation by qubitization, Quantum **3**, 163 (2019).
[10] D. W. Berry and A. M. Childs, Black-box hamiltonian simulation and unitary implementation, arXiv preprint arXiv:0910.4157 (2009).
[11] A. M. Childs, Universal computation by quantum walk,

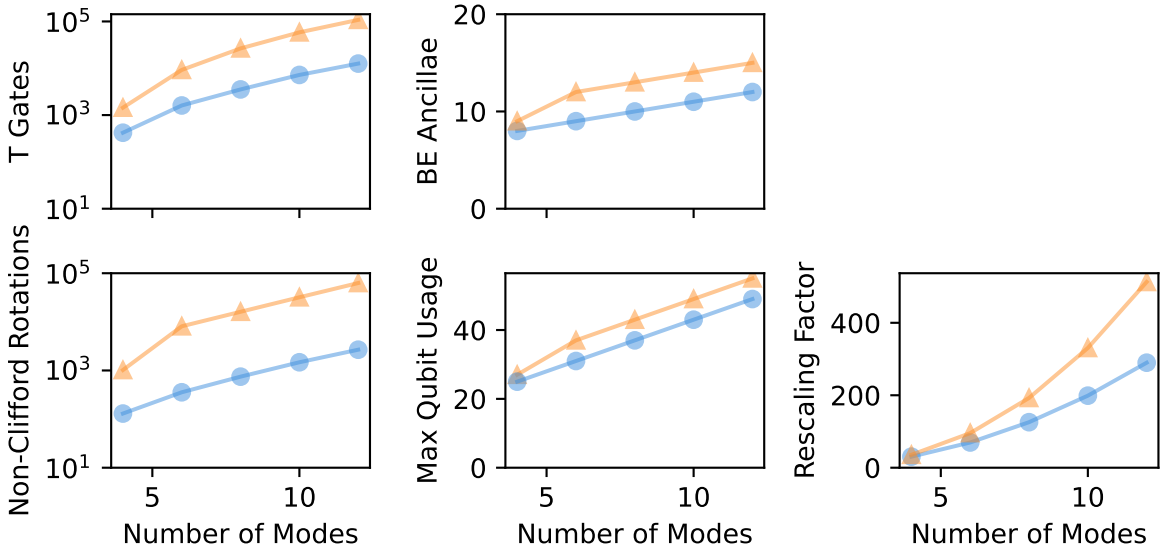

FIG. 18. **Full Yukawa** The number of T gates (upper-left), number of non-Clifford rotations (lower-left), block-encoding ancillae (upper-middle), maximum number of qubits used (lower-middle), and rescaling factor (lower-right) are shown as a function of the number of momentum modes. The bosonic cutoff is fixed to $\Omega = 3$ and the parameters m_f, m_b, and g are set to 1 for all data points. Results for the "Pauli - Expansion" method are shown as the orange triangles and results for LOBE are shown as the blue circles. The optimal rescaling factor, which is given by the L2 norm of the Hamiltonian, is shown as the dashed black crosses.

Physical review letters **102**, 180501 (2009).

[12] A. M. Childs and N. Wiebe, Hamiltonian simulation using linear combinations of unitary operations, arXiv preprint arXiv:1202.5822 (2012).

[13] D. Bluvstein, S. J. Evered, A. A. Geim, S. H. Li, H. Zhou, T. Manovitz, S. Ebadi, M. Cain, M. Kalinowski, D. Hangleiter, *et al.*, Logical quantum processor based on reconfigurable atom arrays, Nature **626**, 58 (2024).

[14] R. Acharya, L. Aghababaie-Beni, I. Aleiner, T. I. Andersen, M. Ansmann, F. Arute, K. Arya, A. Asfaw, N. Astrakhantsev, J. Atalaya, *et al.*, Quantum error correction below the surface code threshold, arXiv preprint arXiv:2408.13687 (2024).

[15] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, Simulated quantum computation of molecular energies, Science **309**, 1704 (2005).

[16] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'brien, A variational eigenvalue solver on a photonic quantum processor, Nature communications **5**, 4213 (2014).

[17] R. Babbush, P. J. Love, and A. Aspuru-Guzik, Adiabatic quantum simulation of quantum chemistry, Scientific reports **4**, 6603 (2014).

[18] P. J. O'Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, *et al.*, Scalable quantum simulation of molecular energies, Physical Review X **6**, 031007 (2016).

[19] R. Babbush, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, A. Paler, A. Fowler, and H. Neven, Encoding electronic spectra in quantum circuits with linear t complexity, Physical Review X **8**, 041015 (2018).

[20] G. A. Quantum, Collaborators*†, F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, S. Boixo, M. Broughton, B. B. Buckley, *et al.*, Hartree-fock on a superconducting qubit quantum computer, Science **369**, 1084 (2020).

[21] J. Lee, D. W. Berry, C. Gidney, W. J. Huggins, J. R. McClean, N. Wiebe, and R. Babbush, Even more efficient quantum computations of chemistry through tensor hypercontraction, PRX Quantum **2**, 030305 (2021).

[22] I. D. Kivlichan, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, W. Sun, Z. Jiang, N. Rubin, A. Fowler, A. Aspuru-Guzik, *et al.*, Improved fault-tolerant quantum simulation of condensed-phase correlated electrons via trotterization, Quantum **4**, 296 (2020).

[23] E. T. Campbell, Early fault-tolerant simulations of the hubbard model, Quantum Science and Technology **7**, 015007 (2021).

[24] M. E. Peskin and D. V. Schroeder, *An Introduction to quantum field theory* (Addison-Wesley, Reading, USA, 1995).

[25] S. P. Jordan, K. S. Lee, and J. Preskill, Quantum algorithms for quantum field theories, Science **336**, 1130 (2012).

[26] C. W. Bauer, Z. Davoudi, A. B. Balantekin, T. Bhattacharya, M. Carena, W. A. De Jong, P. Draper, A. El-Khadra, N. Gemelke, M. Hanada, *et al.*, Quantum simulation for high-energy physics, PRX quantum **4**, 027001 (2023).

[27] D. Camps, L. Lin, R. Van Beeumen, and C. Yang, Explicit quantum circuits for block encodings of certain sparse matrices, SIAM Journal on Matrix Analysis and Applications **45**, 801 (2024).

[28] D. Liu, W. Du, L. Lin, J. P. Vary, and C. Yang, An efficient quantum circuit for block encoding a pairing hamil-

tonian, arXiv preprint arXiv:2402.11205 (2024).
[29] M. Rhodes, M. Kreshchuk, and S. Pathak, Exponential improvements in the simulation of lattice gauge theories using near-optimal techniques, arXiv preprint arXiv:2405.10416 (2024).
[30] P. Jordan and E. Wigner, über das paulische äquivalenzverbot. z phys 47: 631 (1928).
[31] S. B. Bravyi and A. Y. Kitaev, Fermionic quantum computation, Annals of Physics **298**, 210 (2002).
[32] J. T. Seeley, M. J. Richard, and P. J. Love, The bravyi-kitaev transformation for quantum computation of electronic structure, The Journal of chemical physics **137** (2012).
[33] R. D. Somma, Quantum computation, complexity, and many-body physics, arXiv preprint quant-ph/0512209 (2005).
[34] N. P. D. Sawaya, T. Menke, T. H. Kyaw, S. Johri, A. Aspuru-Guzik, and G. G. Guerreschi, Resource-efficient digital quantum simulation of d-level systems for photonic, vibrational, and spin-s hamiltonians, npj Quantum Information **6**, 49 (2020).
[35] J. J. Sakurai and J. Napolitano, *Modern Quantum Mechanics*, 3rd ed. (Cambridge University Press, 2020).
[36] M. D. Schwartz, *Quantum Field Theory and the Standard Model* (Cambridge University Press, 2013).
[37] W. Pauli, Über den zusammenhang des abschlusses der elektronengruppen im atom mit der komplexstruktur der spektren, Zeitschrift für Physik **31**, 765 (1925).
[38] D. W. Berry, A. M. Childs, and R. Kothari, Hamiltonian simulation with nearly optimal dependence on all parameters, in *2015 IEEE 56th annual symposium on foundations of computer science* (IEEE, 2015) pp. 792–809.
[39] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Simulating hamiltonian dynamics with a truncated taylor series, Physical review letters **114**, 090502 (2015).
[40] G. H. Low and I. L. Chuang, Optimal hamiltonian simulation by quantum signal processing, Physical review letters **118**, 010501 (2017).
[41] A. M. Childs, R. Kothari, and R. D. Somma, Quantum algorithm for systems of linear equations with exponentially improved dependence on precision, SIAM Journal on Computing **46**, 1920 (2017).
[42] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (2019) pp. 193–204.
[43] D. Camps and R. Van Beeumen, Fable: Fast approximate quantum circuits for block-encodings, in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, 2022) pp. 104–113.
[44] C. Sanavio, E. Mauri, and S. Succi, Explicit quantum circuit for simulating the advection-diffusion-reaction dynamics, arXiv preprint arXiv:2410.05876 (2024).
[45] L. Grover and T. Rudolph, Creating superpositions that correspond to efficiently integrable probability distributions, arXiv preprint quant-ph/0208112 (2002).
[46] D. Jennings, M. Lostaglio, S. Pallister, A. T. Sornborger, and Y. Subaşı, Efficient quantum linear solver algorithm with detailed running costs, arXiv preprint arXiv:2305.11352 (2023).
[47] P. Jordan and E. Wigner, Über das paulische Äquivalenzverbot, Zeitschrift für Physik **47** (1928).
[48] C. Gidney, Constructing large increment gates (2015).
[49] M. Mottonen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa, Transformation of quantum states using uniformly controlled rotations, arXiv preprint quant-ph/0407010 (2004).
[50] Q. A. team and collaborators, Cirq (2020).
[51] C. Gustin, Openparticle (2024).
[52] grover-rudolph (2024).
[53] C. M. Bender and T. T. Wu, Anharmonic oscillator, Phys. Rev. **184**, 1231 (1969).
[54] M. Girguś and S. D. Głazek, Spiral flow of quantum quartic oscillator with energy cutoff (2024), arXiv:2404.17446 [quant-ph].
[55] K. P. Wójcik, Application of a numerical renormalization group procedure to an elementary anharmonic oscillator (2012), arXiv:1210.1703 [quant-ph].
[56] S. D. Głazek, Elementary example of exact effective-hamiltonian computation, Phys. Rev. D **103**, 014021 (2021).
[57] P. A. M. Dirac, Forms of relativistic dynamics, Rev. Mod. Phys. **21**, 392 (1949).
[58] K. Lamar, *Not Like Us*, 1st ed. (Interscope Records, 2024).
[59] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*, Vol. 2 (Cambridge university press Cambridge, 2001).
[60] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, Elementary gates for quantum computation, Physical review A **52**, 3457 (1995).
[61] P. Selinger, Quantum circuits of t-depth one, Physical Review A—Atomic, Molecular, and Optical Physics **87**, 042302 (2013).
[62] C. Jones, Low-overhead constructions for the fault-tolerant toffoli gate, Physical Review A—Atomic, Molecular, and Optical Physics **87**, 022328 (2013).
[63] C. Gidney, Halving the cost of quantum addition, Quantum **2**, 74 (2018).

Appendix A: Glossary

In this Section, we explicitly define some of the technical phrases and symbols used throughout this work.

- *term* (T): An operator defined as a product of ladder operators.
- *Active Mode*: A fermionic or bosonic mode upon which a ladder operator is being non-trivially applied.
- *block-encoding ancillae*: A register of qubits that give additional degrees of freedom to produce a block-encoding for the operator in a larger Hilbert space.
- *"zeroed-out"*: When the coefficient of a state becomes zero due to the application of an operator.
- *encoded subspace*: The chosen subspace of the block-encoding ancillae that denotes the subspace

in which the non-unitary operator is encoded. Typically, this is the subspace where all block-encoding ancillae are in the $|0\rangle$ state.

- *clean ancillae*: A register of qubits that are promised to begin in the $|0\rangle$ and are returned to the $|0\rangle$ state at the end of a particular operation.
- *all-zero state*: A state of a register where all qubits are in the $|0\rangle$ state.
- L: The number of terms in the Hamiltonian.
- α_l: The coefficient of the l^{th} term in a linear combination. Assumed to be real and positive unless otherwise stated.
- Ω: The occupation cutoff for the bosonic modes. Ω gives the maximum number of bosons that can be present in a single mode.
- I: The number of fermionic or bosonic modes. The subscripts b and a will be used to denote the number of fermionic and bosonic modes respectively.
- b_i: Fermionic annihilation (creation - $b_i^\dagger$) operator acting on mode i.
- d_i: Antifermionic annihilation (creation - $d_i^\dagger$) operator acting on mode i.
- a_i: Bosonic annihilation (creation - $a_i^\dagger$) operator acting on mode i.
- n_{i_b}: The number of fermions (b) occupying the i^{th} mode. n_{i_d} and n_{i_a} give the occupancy for antifermions and bosons respectively.
- ω_i: The number of bosons occupying the i^{th} bosonic mode.
- N_A: The dimension of a matrix A.
- β_ψ: The amplitude of a state that is outside of the encoded subspace after a block-encoding unitary is applied to $|\psi\rangle$.
- λ: The rescaling factor imposed on the operator for a given block-encoding.
- w: The index of the w^{th} least-significant bit in a binary encoding.
- B: The number of active modes within a term. The with ladder operators acting on them within the term T_l.
- C: The number of active modes within a term *excluding* modes upon which a number operator is being applied.
- $S_{l,i}$: The exponent of bosonic annihilation operators acting on the i^{th} bosonic mode within the l^{th} term.

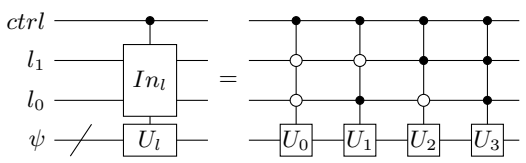

FIG. 19. **Uniformly Controlled Operations** A series of operations that are subsequently controlled on the computational basis states of a register are referred to as *uniformly controlled operations*.

- $R_{l,i}$: The exponent of bosonic creation operators acting on the i^{th} bosonic mode within the l^{th} term.
- P: The sum of the exponents of bosonic ladder operators acting on all modes within a single term.

Appendix B: Uniform State Preparation

Appendix C: Compiling Toffoli Gates

The Toffoli gate is a ubiquitous operation in quantum algorithms and is often a significant contributor to the number of non-Clifford resources. In many algorithms - including those described in this work - a series of Toffoli gates are used to perform an operation that is coherently controlled on the logical-AND of several variables (Figure 20). Therefore, even small reductions in the cost of compiling a Toffoli gate can have a significant impact on the overall spacetime cost of an algorithm. In an effort to keep this work self-contained, we review a specific compilation technique that can reduce the number of T gates required to synthesize a pair of N-controlled Toffoli gates.

In Nielsen and Chuang [59], a method for synthesizing a single Toffoli gate from 7 T gates is shown which is likely derived from the methods proposed by Barenco et al. [60]. Selinger [61] and Jones [62] reduce this cost for a single Toffoli gate from 7 T gates to 4 using one clean ancilla. To the best of our knowledge, this is the most T-gate efficient compilation for a single Toffoli gate.

Gidney [63] showed that a pair of Toffoli gates which subsequently compute and uncompute the logical-AND of two variables can be synthesized using 4 T gates, a measurement, and a classically conditioned operation (Figure 3 of [63]). This compilation scheme is sometimes referred to as "elbows" and is depicted in Figure 21.

Following the methods shown by Barenco et al. [60] and Gidney [63], a pair of N-controlled Toffoli gates that are used to compute and uncompute the logical-AND of the same variables can be composed using $4(N-1)$ T gates and $N-1$ clean ancilla using a series of elbows. It is important to note that this technique does not always apply to neighboring Toffoli gates, but sufficient conditions are given in Figure 5 of [63]. The spacetime costs for the block-encoding circuits used in this work assume

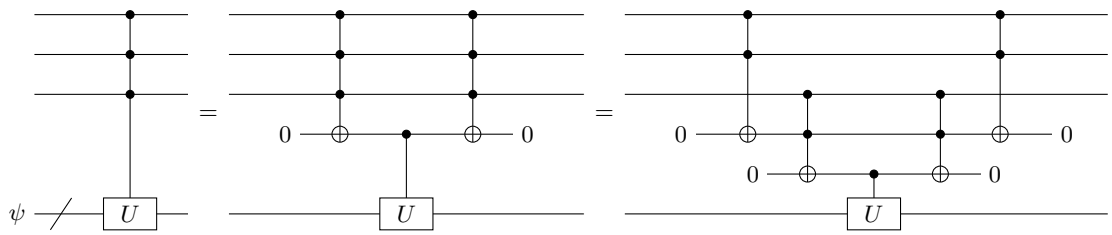

FIG. 20. **Multi-Controlled Operations** Operations with multiple controls are referred to as *multi-controlled operations*. An N-controlled operation can be decomposed into $2(N-2)$ Toffoli gates using $N-1$ clean ancillae.

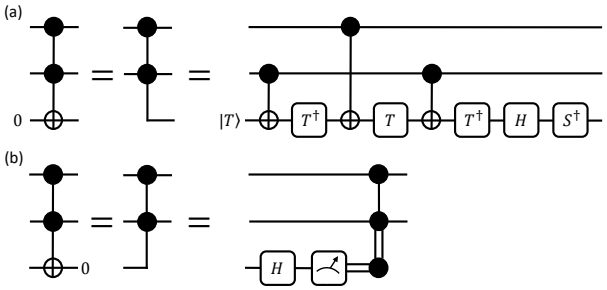

FIG. 21. **Elbows** In (a), a compilation scheme for a Toffoli gate acting on a clean ancillae is shown that uses four T gates. This is sometimes referred to as a "left-elbow". In (b), a compilation scheme for a Toffoli gate that uncomputes a clean ancillae is shown that uses a measurement and classically conditioned operation. This is sometimes referred to as a "right-elbow".

this strategy when applicable.

Additionally, there are several optimizations that can reduce the cost of neighboring pairs of elbows which are shown in Figure 6 of Babbush et al. [19]. These optimizations are particularly useful when compiling uniformly controlled operations (Figure 19). With these optimizations, the control structure for applying L uniformly controlled operations uses $4(L-1)$ T gates. The spacetime costs quoted throughout this work assume this strategy when applicable, including in cases such as Figures 10 and 11.

Appendix D: Proof of $\lambda_{ASP} \leq \lambda_{USP}$

Theorem 1 $\lambda_{ASP} \leq \lambda_{USP}$

Proof:
Define the following:

$$\lambda_{USP} \equiv L \max_l |\tilde{\alpha}_l| \tag{D1}$$

and

$$\lambda_{ASP} = \sum_l |\tilde{\alpha}_l| \tag{D2}$$

Case 1: All Coefficients are Equal

Assume $\tilde{\alpha}_i = \tilde{\alpha}_j \forall i,j$. Let $\tilde{\alpha}_i = \tilde{\alpha}$. Then, $\lambda_{USP} \equiv L \max_l |\tilde{\alpha}_l| = L|\tilde{\alpha}|$ and $\sum_{l=0}^{L-1} |\tilde{\alpha}_l| = L|\tilde{\alpha}|$. Thus, in this case, $\lambda_{USP} = \lambda_{ASP}$.

Case 2: Coefficients are Not All Equal

Assume that there exist at least two coefficients, $\tilde{\alpha}_i$ and $\tilde{\alpha}_j$ such that $\tilde{\alpha}_i \neq \tilde{\alpha}_j$. This implies that at least one coefficient, α_l, has the largest magnitude. Then, $\lambda_{ASP} = max_l|\tilde{\alpha}_l| + \sum_{k\neq l} |\tilde{\alpha}_k|$. Now, if $\lambda_{ASP} < \lambda_{USP}$, then $\lambda_{ASP} - \lambda_{USP} < 0$:

$$\begin{aligned}
\lambda_{ASP} - \lambda_{USP} &= max_l|\tilde{\alpha}_l| + \sum_{k\neq l} |\tilde{\alpha}_k| - Lmax_l|\tilde{\alpha}_l| \quad \text{(D3)} \\
&= -(L-1)max_l|\tilde{\alpha}_l| + \sum_{k\neq l} |\tilde{\alpha}_k| \\
&< -(L-1)max_l|\tilde{\alpha}_l| + \sum_{k\neq l} max_l|\tilde{\alpha}_l| \\
&= -(L-1)max_l|\tilde{\alpha}_l| + (L-1)max_l|\tilde{\alpha}_l| \\
&= 0
\end{aligned}$$

Therefore, $\lambda_{ASP} \leq \lambda_{USP}$.■

Appendix E: Addition by a Known Classical Value

Adding a (known) classical integer (m) to a quantum register encoded in binary is a required operation throughout this work:

$$|n\rangle \rightarrow |n+m\rangle \tag{E1}$$

In an effort to keep this work self-contained and pedagogical, we review some methods for constructing this operation in this section.

One option is to use a series of controlled incrementer ($+1$) circuits. An implementation of an incrementer circuit given by Gideny [48] is shown in Figure 22. If N is the number of qubits in the register being incremented, this implementation requires $4(N-1)$ T gates and $N-1$ clean ancillae. Naively, a series of m incrementers will result in increasing the value of the register by $m \mod 2^N$.

However, an incrementer circuit can also be used to perform addition by a power of 2 by acting on only the most-significant qubits. For example, adding the value 8

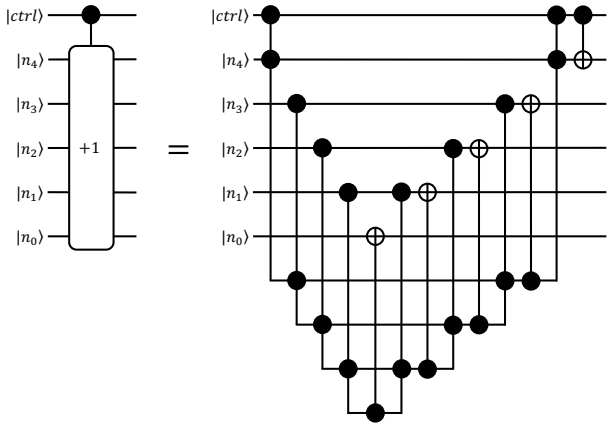

FIG. 22. **Controlled Incrementer** A circuit diagram implementing a controlled incrementer (mod 32) is shown. The controlled incrementer performs the operation $|x\rangle \rightarrow |x+1\rangle$ when the control qubit is in the $|1\rangle$ state. Decrementing the register by 1 can be achieved by applying Pauli-X gates on each qubit before and after the operation.

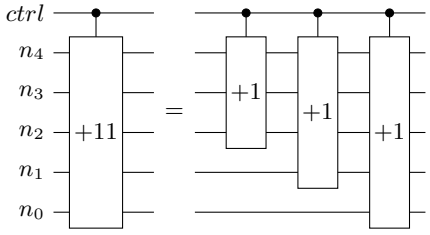

FIG. 23. **Addition via Incrementers** An implementation of addition by a classical value (11) (mod 32) using a series of incrementers is shown. An incrementer applied onto a register excluding the least-significant qubit implements a bit-shifted incrementer. This effectively increases the value of the register by 2. Addition by the classical value 11 can be constructed by bit-shifted incrementers adding the values +8, +2, and +1. Subtraction by the same value can be achieved by applying Pauli-X gates on each qubit before and after the operation.

can be achieved using an incrementer circuit that treats the 4^{th} least significant qubit as the least significant qubit in the incrementer circuit and disregards the 3 lesser qubits. A circuit adding any classical value can then be constructed based on the binary representation of the classical number. A circuit diagram for this construction is shown in Figure 23. The cost of this construction would require N incrementer circuits requiring $4\sum_{i=0}^{N-2} N-i-1$ T gates in total and $N-1$ clean ancillae.

Since we are performing modular addition, the same result can also be achieved by subtracting the value $2^N - m$. As an example, if the classical value is 31 and $N = 5$, then this can be accomplished using a single decrementer circuit which can be constructed by conjugating an incrementer circuit with Pauli X gates acting on the qubits encoding $|n\rangle$. The cost of these two methods can be classically determined and the more favorable option can

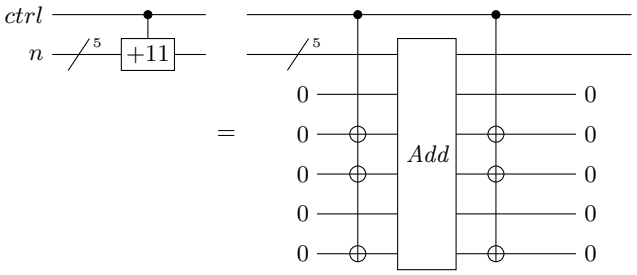

FIG. 24. **Time-Efficient Controlled Addition of** 11 Increasing the value of a quantum register by a known classical value can be implemented using clean ancillae and an uncontrolled quantum addition circuit. The known classical value is loaded into a clean ancilla register using a series of CNOT gates corresponding to the binary representation of the classical value. Then an uncontrolled quantum addition circuit is applied to the two registers. Finally, the loading of the classical value is uncomputed.

be chosen during compilation. The upper bound for the number of T gates, regardless of the classical value being added, is $N^2 + N$. This scaling was determined numerically, but i think it shouldn't be hard to derive.

Another option is to first load the classical value into an clean ancilla register, controlled on the control qubit, perform *uncontrolled* (modular) addition of the two quantum registers, then "unload" the classical value. If the control is off, then the classical value is not loaded and the uncontrolled addition simply adds the value 0 and leaves the original register unchanged. An example diagram for this construction depicting adding the value $m = 11$ to a register with $N = 5$ qubits is shown in Figure 24.

The loading (and unloading) of the classical value only requires CNOTs and therefore does not contribute any non-Clifford resouces, but it does require N clean ancillae. However, if the p least significant bits of m are zero, then only $N-p$ qubits are required to load m and the p least-significant qubits of N can be omitted from the addition. Uncontrolled addition of two registers can be performed using $4(N-1)$ T gates and $N-1$ clean ancillae using the construction for addition shown in Figure 1 of [63]. Therefore, in total, this compilation will require $4(N-p-1)$ T gates and $2(N-p)-1$ clean ancillae.

When m is a power of 2, then compilation using incrementer circuits uses the same number of T gates, but fewer clean ancillae. When m is not a power of 2, then the compilation using uncontrolled quantum addition uses fewer T gates (at the expense of more clean ancillae). Since m is known during compilation, an implementation can be chosen that results in the fewest required resources.

Another implementation can be chosen to reduce the number of clean ancillae, which uses the classical information about m to modify the circuit for *controlled* quantum addition. Controlled addition of two registers can be performed using $4(2N-3)$ T gates and $2N-1$ clean ancil-

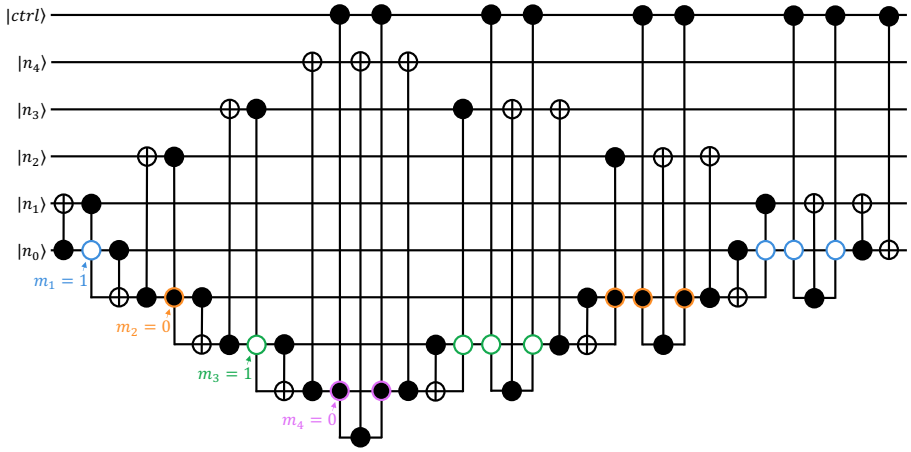

FIG. 25. **Space-Efficient Controlled Addition of 11** An implementation for increasing the value of a register by a known classical value is shown for the case when the known value is 11 and the number of qubits in the register is 5. The binary representation of 11 is 01011 with the left-most bit being the most-significant. The values of these M classical bits can be propagated into the control structure of the controlled quantum addition. If the value of the i^{th} bit of M is 0 (1), the corresponding control in the circuit is controlled on the $|1\rangle$ ($|0\rangle$) state.

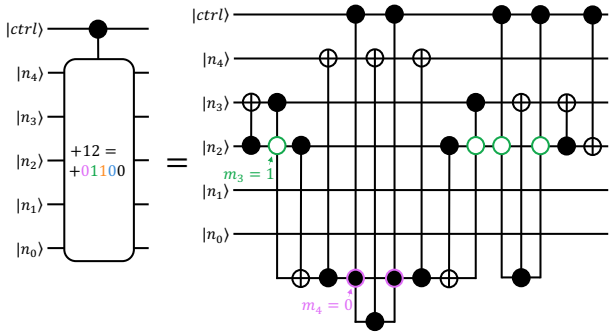

FIG. 26. **Space-Efficient Controlled Addition of 12** An implementation for increasing the value of a register by a known classical value is shown for the case when the known value is 12 (01100 in binary) and the number of qubits in the register is 5. When the least-significant bits of M are 0, the circuit can be bit-shifted, resulting in a lower cost implementation. In this case, the two least-significant bits are 0, so the circuit can be bit-shifted twice.

lae using the construction for addition shown in Figure 4 of [63]. This circuit can be modified by propagating the classical information about the binary encoding of m into the control structure of the adder circuit, thereby reducing the number of clean ancillae by N. An example diagram showing this propagation in the case where $m = 11$ and $N = 5$ is shown in Figure 25.

Similarly, if the p least-significant bits of m are known to be zero, the addition can be performed beginning with the first non-zero bit of m. An example circuit diagram for the case where $m = 12$ (01100 in binary) and $N = 5$ is shown in Figure 26. If the p least significant bits of m are zero, then this circuit uses $4(2(N-p)-3)$ T gates and $N-p-1$ clean ancillae. Since this work primarily focuses on reducing the number of T gates, the quantum resource estimates quoted in this work do not utilize this strategy.

Appendix F: Pauli Expansion of Fermionic and Bosonic Ladder Operators

In this Section, we review the two different operator transformations that are used in this work to express the ladder operators in the Pauli basis.

1. Jordan-Wigner Transformation

2. Standard Binary

In this work, the registers encoding the occupation of the bosonic modes store the value of the occupation in binary notation. The Standard Binary encoding [34] provides a method for expanding the bosonic ladder operators in the Pauli basis when this binary encoding is used.

The expansion in the Pauli basis begins by noting the following definitions of the bosonic creation and annihilation operators:

$$a^\dagger = \sum_{s=0}^{\Omega-1} \sqrt{s+1}|s+1\rangle\langle s| = |1\rangle\langle 0| + \sqrt{2}\,|2\rangle\langle 1| + \dots \tag{F1}$$

$$a = \sum_{s=0}^{\Omega-1} \sqrt{s+1}|s\rangle\langle s+1| = |0\rangle\langle 1| + \sqrt{2}\,|1\rangle\langle 2| + \dots \tag{F2}$$

(F3)

From here, each state in the expansion of a and $a^\dagger$ is converted into binary, e.g. $|3\rangle = |0\dots011\rangle$, where

the preceeding (most-significant) qubits are zeros and the total length of the register is given by $\lceil \log_2(\Omega+1) \rceil$.

Now, each outer product in Eq. F1 can be expanded by computing the outer product of the individual qubit values tensored together. For example, with $\Omega = 3$ the outer product of $|2\rangle$ and $|3\rangle$ is given by:

$$|2\rangle\langle 3| = |10\rangle\langle 11| = |1\rangle\langle 1| \otimes |0\rangle\langle 1| . \tag{F4}$$

Each single-qubit outer product can be mapped onto the Pauli basis using the following relations:

$$|0\rangle\langle 0| = \frac{1}{2}(I + Z) \tag{F5}$$

$$|0\rangle\langle 1| = \frac{1}{2}(X + iY) \tag{F6}$$

$$|1\rangle\langle 0| = \frac{1}{2}(X - iY) \tag{F7}$$

$$|1\rangle\langle 1| = \frac{1}{2}(I - Z) \tag{F8}$$

$$\tag{F9}$$

Finally, these single-qubit outer products can be tensored together to give the expansion of the multi-qubit outer products, thereby representing the ladder operators as a linear combination of Pauli operators.

An explicit example with $\Omega = 3$ is shown below:

$$a_0^\dagger = 0.683IX - 0.183ZX - 0.683\mathrm{i}IY \tag{F10}$$

$$+0.183\mathrm{i}ZY + 0.354XX - 0.354\mathrm{i}YX \tag{F11}$$

$$+0.354\mathrm{i}XY + 0.354YY \tag{F12}$$

Appendix G: Uniformly Controlled Rotations

Implementing a series of uniformly controlled rotations is a common subroutine used in this work. In this section, we discuss the cost and explicit circuit compilation for a series of uniformly controlled rotations around the same axis (but different angles) are applied on the same qubit:

$$\sum_{l=0}^{L-1} |l\rangle|\phi\rangle \to \sum_{l=0}^{L-1} |l\rangle R_a(\alpha_l)|\phi\rangle \tag{G1}$$

Möttönen et. al [49], provide a construction for *uncontrolled* uniformly controlled rotations. This construction is only defined when the number of rotations (L) is explicitly a power of 2, however, if fewer rotations are required, then this can be achieved by padding with zero-angle rotations.

In this construction, the rotation angles are classically preprocessed based on the Gray code (Eq. 3 of [49]):

$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_{L-1} \end{bmatrix} = M \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{L-1} \end{bmatrix} \tag{G2}$$

where M is a matrix transformation defined by:

$$M_{i,j} = L^{-1}(-1)^{b_j \cdot g_i} \tag{G3}$$

where b_j is the binary representation of the integer j, g_i is the Gray code representation of the integer i, and $b_j.g_i$ is the bitwise inner product of b_j and g_i.

However, in this work we require the use of a *controlled* series of uniformly controlled rotations. Naively, this can be implemented by controlling each of the arbitrary rotations in the construction given by Möttönen et al. [49]. An example circuit diagram for this construction is shown in subfigure 27a. Since each controlled rotation can be implemented by two uncontrolled rotations, this compilation strategy uses $2L$ uncontrolled arbitrary rotations.

An alternative approach which uses $4\log_2 L$ T gates, $L+3$ arbitrary rotations, and $\log_2 L$ clean ancillae is shown in subfigure 27b. In this construction, the temporary logical-AND of each qubit in the index register and the control qubit is computed using $\log_2 L$ Toffoli gates. CNOTs from these clean ancillae then conjugate each of the arbitrary rotations which are left uncontrolled. When the control is on, this fully recovers the construction given by Möttönen et. al.

However, when the control is off, the uncontrolled arbitrary rotations are still applied, resulting an undesired rotation of angle $\sum_i(\theta_i)$. This undesired rotation can then be undone using one 0-controlled rotation of angle $-\sum_i(\theta_i)$.

Appendix H: Grover-Rudolph State Preparation

In this section, we will describe the Grover-Rudolph state-preparation routine [45] in the context that it is used in this work. The Grover-Rudolph state-preparation algorithm constructs quantum circuits that prepare states of the form given by:

$$\left|0^{\otimes \lceil \log_2 L \rceil}\right\rangle \to_{Grover\text{-}Rudolph} \sum_{l=0}^{L} \sqrt{p(l)}\,|l\rangle \tag{H1}$$

where $p(l)$ is a probability distribution along the different indices (l) with the constraint that $\sum_l p(l) = 1$.

In the context of block-encodings, preparing such probability distributions can be used to construct the *Prepare* oracle (Eq. 21). The probability distribution in this case is defined by the normalized magnitudes of the coefficients of the terms in the linear combination: $p(l) = |\alpha_l|/\lambda$.

The Grover-Rudolph algorithm works by sequentially summing up the probability distribution to the left and right of a given index and then performing a rotation controlled on the current index.

For example, given the (noramlized) probabilities α_0, α_1, α_2, and α_3, the Grover-Rudolph algorithm proceeds as follows:

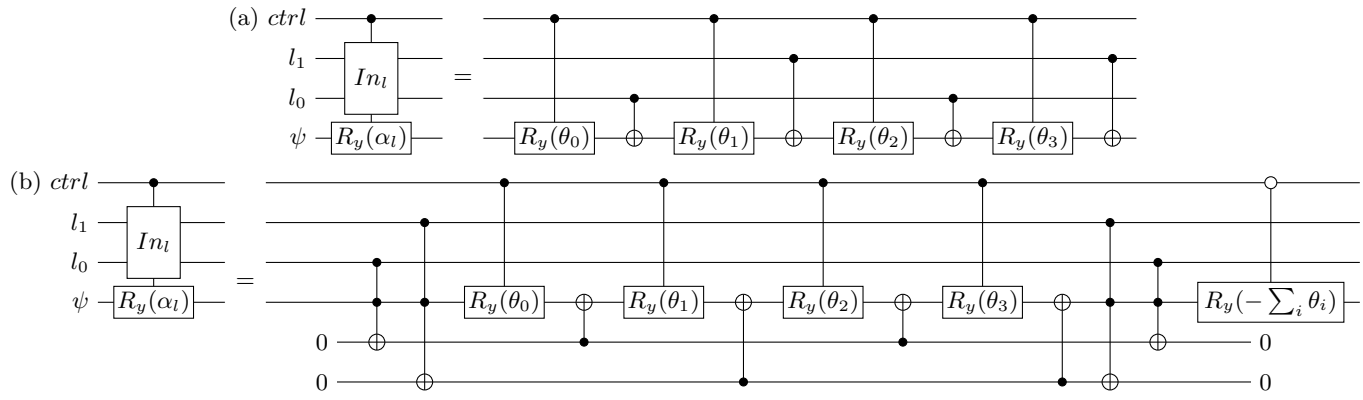

FIG. 27. **Controlled Uniformly Controlled Rotations** Two implementations for controlling a series of uniformly controlled rotations are shown. In (a), a naive implementation is shown which doubles the number of arbitrary rotations required. The implementation shown in (b) uses only one additional controlled rotation and $\log_2 L$ Toffoli gates, but requires $\log_2 L$ clean ancillae.

(i) Perform a Pauli-Y rotation on the top (leftmost) qubit in the register by an angle: $\theta = 2\cos^{-1}\left(\sqrt{\alpha_0+\alpha_1}\right)$.

(ii) Perform a Pauli-Y rotation on the second qubit in the register, controlled on the first qubit being in the state $|0\rangle$ by an angle: $\theta = 2\cos^{-1}\left(\sqrt{\frac{\alpha_0}{\alpha_0+\alpha_1}}\right)$

(iii) Perform a Pauli-Y rotation on the second qubit in the register, controlled on the first qubit being in the state $|1\rangle$ by an angle: $\theta = 2\cos^{-1}\left(\sqrt{\frac{\alpha_2}{\alpha_2+\alpha_3}}\right)$

The evolution of the quantum state is given by:

$$\begin{aligned}|00\rangle &\to_{(i)} \sqrt{\alpha_0+\alpha_1}\,|00\rangle + \sqrt{\alpha_2+\alpha_3}\,|10\rangle \\ &\to_{(ii)} \sqrt{\alpha_0}\,|00\rangle + \sqrt{\alpha_1}\,|01\rangle + \sqrt{\alpha_2+\alpha_3}\,|10\rangle \\ &\to_{(iii)} \sqrt{\alpha_0}\,|00\rangle + \sqrt{\alpha_1}\,|01\rangle + \alpha_2\,|10\rangle + \alpha_3\,|11\rangle\end{aligned} \tag{H2}$$

The Grover-Rudolph algorithm can be thought of as performing several series of uniformly controlled rotations. An example circuit diagram depicting this construction is shown in Figure 28. When L is the number of probabilities (or coefficients) to prepare and is a power of 2, this implementation uses $L-1$ uncontrolled rotations.

Appendix I: Quantum Field Theory Hamiltonians

The two quantum field theory models studied in section V are the ϕ^4 and Yukawa model. In performing a Legendre transformation from the Lagrangian to the Hamiltonian, $\mathcal{L} \to H$, one must choose an explicit coordinate system. Any field theory calculation done in a Hamiltonian approach benefits from using *front form* (lightfront) coordinates [57]. Lightfront coordinates are defined as $x^\mu = \left(x^+, x^-, x^1, x^2\right)$, where $x^+ \equiv x^0 + x^3$ acts as the space coordinate and $x^- \equiv x^0 - x^3$ acts as the time coordinate. In each model studied above, the transverse spacial coordinates x^1, x^2 are discarded, and the only independent coordinates are x^+ and x^-.

One main benefit of using this set of coordinates is that the Hamiltonian eigenvalue equation, whose eigenvectors are bound states of the theory, is simpler than in instant (standard) coordinates. This is because the Hamiltonian operator in instant form coordinates is $H = \sqrt{\vec{P}^2 - m^2}$, where the presence of the square root leads to an ambiguity. In front form coordinates, the Hamiltonian operator is given via the lightfront 'energy' $\hat{P}^- = \frac{\left(\vec{P}^\perp\right)^2 + m^2}{P^+}$, where the square root is absent.

In one lightfront space and time dimension, the Hamiltonian eigenvalue equation is given by $\hat{P}^-|\psi\rangle = \frac{m^2}{P^+}|\psi\rangle$. Any state can be expanded as

$$|\psi\rangle = \sum_n \int d[\mu_n]\,|\mu_n\rangle\,\langle\mu_n|\psi\rangle\,,$$

where $\{|\mu_n\rangle\}$ is a set of Fock states with n particles, i.e. $|f\rangle, |ff\rangle, |b\rangle, |fb\rangle, \ldots$.

The lightfront 'momentum' P^+ is a conserved quantity which every constituent's longitudinal momentum in a given Fock state $\in \{|\mu_n\rangle\}$ must sum to P^+. The 'modes' of the ladder operators correspond to discrete longitudinal momentum quantum numbers k, where the discretized approximation to a continuous constituent momentum $p^+ = \frac{\pi k}{L}$, where L truncates x^- to a finite region. Thus, the conservation of P^+ is analogous to $\sum k = K$, labeled the *resolution* []. Increasing the resolution leads to a larger Hilbert space of states, and thus the number of terms in the Hamiltonian that act non-trivially on this set of states also expands. This is why increasing K is a metric for 'number of terms in the Hamiltonian'.

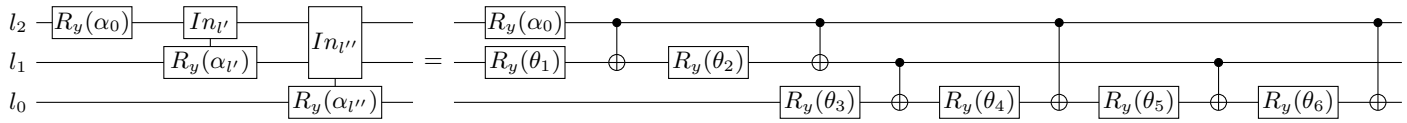

FIG. 28. **Grover-Rudolph Circuit Compilation.** An implementation of the Grover-Rudolph algorithm using several series of uniformly controlled rotations is shown when $L = 8$. This circuit requires $L - 1$ rotations when L is a power of 2 and the angles of the rotations are changed ($\theta_i \to \alpha_i$) based on classical preprocessing.