Sparse Block-Encodings for Linear Combinations of Ladder Operators

(Dated: August 14, 2024)

In this work, we detail the construction of quantum circuit oracles that create block-encodings for observables described as a linear combination of products of ladder operators acting on fermionic, antifermionic, and bosonic modes. We refer to this constuction as LOBE (Ladder Operator Block-Encoding) and show how it can be used to simulate Hamiltonians involving interactions between these different types of particles. Our work builds off of similar sparse-oracle constructions in the literature, but generalizes prior works to establish a clear connection with block-encoding methods that are commonly refered to as LCU (Linear Combination of Unitaries). In addition to extnding LCU to more general observables that are given as a Linear combination of products of ladder operators, we also demonstrate how these oracles can be extended to include bosonic ladder operators. To our knowledge, this is the first block-encoding construction that allows for interactions between fermions, antifermions, and bosons, paving the way for simulation of more complicated quantum systems such as those that arise in high-energy physics.

I. INTRODUCTION

Wop, wop, wop, wop, wop, Dot, fuck 'em up
Wop, wop, wop, wop, wop, I'ma do my stuff [1]

II. THEORY

Give background of ladder operators and constructions of realistic Hamiltonians/Observables from ladder operators

A. Encoding

Here we'll discuss how we encode the physical states we are interested in terms of qubits/registers.
Question: do we want to also cost-out the unary encoding or just the "compact" encoding that we've been primarily working with?

B. Ladder Operators

@Gus, you probably have much better language to define all of this stuff. I just needed to write something down so I could reference it in the circuit construction. Don't hesitate to scrap anything in here.

1. Feromons and Antiferomons

Define action of fermionic ladder operators.

Fermions (and antifermions) obey the Pauli-exclusion principle citation and therefore the occupation of a (anti)fermionic mode can only be occupied ($|1\rangle$) or unoccupied ($|0\rangle$). Fermionic (and antifermionic) ladder operators only act non-trivially on the qubits encoding the mode that the ladder operator acts on and we define their action as follows.

The fermionic creation operator is given by:

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

where b_i denotes a fermionic ladder operator on the ith mode, the † indicates a creation operator, and $|n_{b_i}\rangle$ is the occupation of the ith fermionic mode. An antifermionic creation operator is defined as above with the symbol d to denote that the operator acts on antifermions.

For a fermionic creation operator, if the mode being acted upon is unoccupied, then the creation operator "creates" a fermion in that mode and applies a phase determined by the parity of the occupation of the previous modes.

Therefore the ordering of the modes in the encoding has an implication on the action of the operator that must be accounted for. Since fermionic modes can only be either occupied or unoccupied, then if the mode is already occupied the operator zeroes the amplitude of the quantum state, thereby "destoying" that portion of the quantum state.

The fermionic annihilation operator is given by:

$$b_i \left| n_{b_i} \right\rangle = \begin{cases} (-1)^{\sum_{j<i} b_j} \left| 0 \right\rangle & when \left| n_{b_i} \right\rangle = \left| 1 \right\rangle \\ 0 & when \left| n_{b_i} \right\rangle = \left| 0 \right\rangle \end{cases} \tag{2}$$

and the antifermionic annihilation operator is likewise defined for d instead of b.

The action of the annihilation operators is similar (and opposite) to the creation operators. If the mode is already occupied, then the annihilation operator "annihilates" the fermion at that mode by setting the occupation to zero and applies a phase based on the parity of the occupation of the preceeding modes. If the mode is unoccupied before the operator is applied, then the annihilation operator zeroes the amplitude.

2. Bosos

$$a_i^\dagger \left| n_{a_i} \right\rangle = \begin{cases} \sqrt{n_{a_i} + 1} \left| n_{a_i} + 1 \right\rangle & when \left| n_{a_i} \right\rangle \neq \left| \Omega \right\rangle \\ 0 & when \left| n_{a_i} \right\rangle = \left| \Omega \right\rangle \end{cases} \tag{3}$$

where a_i denotes a bosonic ladder operator on the ith mode, the † indicates a creation operator, $\left| n_{a_i} \right\rangle$ is the occupation of the ith bosonic mode, and Ω is the maximum allowable bosonic occupation.

$$a_i \left| n_{a_i} \right\rangle = \begin{cases} \sqrt{n_{a_i}} \left| n_{a_i} - 1 \right\rangle & when \left| n_{a_i} \right\rangle \neq \left| 0 \right\rangle \\ 0 & when \left| n_{a_i} \right\rangle = \left| 0 \right\rangle \end{cases} \tag{4}$$

3. Commutation Rules

C. Observables

1. Products of Ladder Operators (Terms)

We define a *term* (T) as a product of ladder operators that can act on fermionic, antifermionic, and bosonic modes:

$$T = \prod_{m=0}^{M-1} c_m \tag{5}$$

where M is the number of ladder operators in the term and $c_m \in \{b_i, b_i^\dagger, d_i, d_i^\dagger, a_i, a_i^\dagger\}$.

The ladder operators (c_m) can be reordered arbitrarily with the introduction of additional terms due to the commutation rules described in II B 3. In this work, we will have a preference for *normal ordering* of the operators to obey the following structure:

$$T = \left(\prod_i ((a_i^\dagger)^S)(A_i)^R((a_i)^P) \right) \left(\prod_i ((d_i^\dagger)^{\delta_{d_i}^\dagger}(D_i)^{\delta_{D_i}}(d_i)^{\delta_{d_i}}) \right) \left(\prod_i ((b_i^\dagger)^{\delta_{b_i}^\dagger}(B_i)^{\delta_{B_i}}(b_i)^{\delta_{b_i}}) \right) \tag{6}$$

where δ takes the value 0 or 1 to denote if the operator is active in the term and the values P, R, and S are integers $\in [0, \Omega]$ and denote the exponent of the bosonic ladder operators acting on that particular bosonic mode. The operators A_i, D_i, and B_i denote occupation operators acting on the ith bosonic, antifermionic, and fermionic modes respectively.

Describe number/occupation operators acting on fermions/antifermions and bosons and rewrite previous equation for T including these operators.

2. Linear Combinations of Terms

We can write Hamiltonians (or observables) in the form of linear combinations of terms:

$$H = \sum_{l=0}^{L-1} \alpha_l T_l \tag{7}$$

where L is the total number of terms and α_l is a real-valued coefficient associated with the term T_l.

III. LADDER OPERATOR BLOCK-ENCODING (LOBE)

In this section of the text, we'll do the following:

A. Defining "Block-Encoding"

B. Prior Works

1. Linear Combination of Unitaries

2. Sparse Block-Encoding of Pairing Hamiltonians

stuff from Liu et al Might ask you to fill this out @Gus

C. Rescaling of Coefficients Due to Bosonic Terms

The inclusion of bosonic ladder operators in our models requires rescaling the coefficients of our Hamiltonian such that they are always normalized and appropriately weighted. As shown in Eqs. 3 and 4, bosonic ladder operators acquire a coefficient on the state that is proportional to the square-root of the occupation of the state. From these definitions, it is clear that if we were to apply a block-encoding that contains bosonic ladder operators onto a quantum state, the output quantum state may not be normalized.

To remedy this, we can rescale the coefficients that are picked up by the operators such that the states are always normalized. In the case of bosonic ladder operators, we can rescale Eqs. 3 and 4 by dividing by the square-root of the maximum allowable occupation number plus one such that any coefficient acquired is < 1:

$$
\begin{aligned}
a_i^\dagger \left| n_{a_i} \right\rangle &= \sqrt{\frac{n_{a_i} + 1}{\Omega + 1}} \left| n_{a_i} + 1 \right\rangle \quad when \left| n_{a_i} \right\rangle \neq \left| \Omega \right\rangle \\
a_i \left| n_{a_i} \right\rangle &= \sqrt{\frac{n_{a_i}}{\Omega + 1}} \left| n_{a_i} - 1 \right\rangle \qquad when \left| n_{a_i} \right\rangle \neq \left| 0 \right\rangle
\end{aligned}
\tag{8}
$$

We do not redefine these equations for the cases where the state is annihilated since they will be the same as in Eqs. 3 and 4 These rescaled coefficients will be accounted for within the action of the *Select* oracle as described in Subsubsection III D 2.

Since multiple bosonic operators may be present within a single term (T_l), then the overall coefficient of the output state will be rescaled by a factor of:

$$\lambda_{k_l} = (\Omega + 1)^{k_l/2} \tag{9}$$

where k_l is the number of bosonic operators included in the term T_l.

Different terms within the Hamiltonian may have different numbers of bosonic operators and therefore will have differing rescaling coefficients of λ_{k_l}. In order to guarantee that this rescaling is consistent across the different terms, we can classically preprocess the coefficients in the Hamiltonian such that after the block-encoding is applied, all output states have the same bosonic rescaling factor. If we define $K \equiv \max_l k_l$, then we can define the preprocessed coeffieints as:

$$\alpha_l \rightarrow \frac{\alpha_l}{(\Omega + 1)^{(K - k_l)/2}} \equiv \alpha_l^* \tag{10}$$
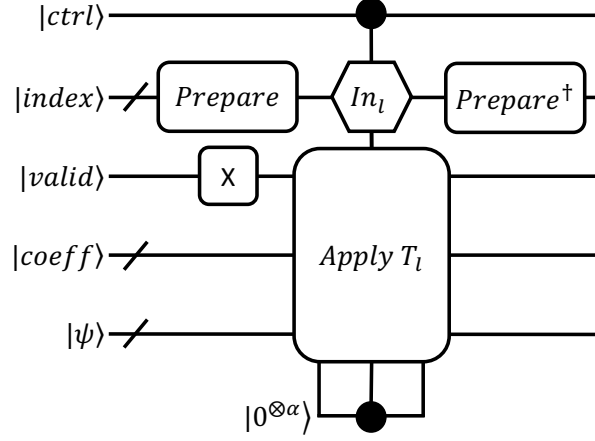
FIG. 1. **Ladder Operator Block-Encoding.**

where the coefficients α_l^* are the rescaled coefficients of the terms.

This preprocessing means that after the block-encoding is applied, all output states will be rescaled by a coefficient of $(\Omega + 1)^{(K)/2}$. Therefore, the rescaled Hamiltonian that we block-encode is given by:

$$H^* = \frac{H}{(\Omega + 1)^{K/2}} \tag{11}$$

where H^* is the rescaled Hamiltonian.

D. Circuit Construction

In Figure 1, we define the LOBE circuit in terms of generic oracles. Disregarding the (optional) control qubit ($|ctrl\rangle$), the LOBE circuit makes use of 5 qubit registers: $|index\rangle$, $|valid\rangle$, $|coeff\rangle$, $|\psi\rangle$, and $|0^{\otimes\alpha}\rangle$.

The register denoted $|index\rangle$ is referred to as the index register and is used to index the terms in the Hamiltonian as is done in LCU constructions. The integer representations of the computational basis states of the index register corresponds to the indices l in Eq. 7.

The register denoted $|valid\rangle$ consists of a single qubit and is referred to as the validation qubit. It serves the same purpose as in [2] which is to denote whether or not the term at the current index (T_l) will annihilate the quantum state. If the term will annihilate the state, then the validation qubit remains in the $|1\rangle$ state such that the branch of the wavefunction stays outside the desired subspace of the block-encoding. If the term will not annihilate the state, then the validation qubit gets flipped to the $|0\rangle$ state for the term T_l.

The register denoted $|coeff\rangle$ is referred to as the coefficient register and is used to apply the coefficients associate with the term T_l. These coefficients include both the coefficients of the terms in the linear combination (α_l) as well as the coefficients associated with the bosonic ladder operators. One qubit is used to apply the α_l coefficient while a separate qubit will be required for each bosonic operator (defined as a creation, annihilation, or occupation operator acting with a positive integer-valued exponent) in the term.

The register denoted $|\psi\rangle$ is referred to as the system register and is used to encode the state of the system. This register can be broken up into three subsequent registers: the fermionic system $|\psi_b\rangle$, the antifermionic system ($|\psi_d\rangle$), and the bosonic system ($|\psi_a\rangle$). The encoding studied in this work is outlined in more detail in Subsection II A.

Finally, the register denoted $|0^{\otimes\alpha}\rangle$ is referred to as the clean ancillae register. This register includes ancillae qubits that are promised to begin in the $|0\rangle$ state and are returned to this state at the end of the block-encoding circuit.

The LOBE circuit is structured similarly to an LCU circuit in that it involves sequential application of a *Prepare* oracle, a *Select* oracle, and then uncomputing the *Prepare* oracle using the daggered circuit. In addition to these oracles, a single X gate is applied to the $|valid\rangle$ qubit in order to prepare this qubit in the $|1\rangle$ state. This single X gate is *not* repeated on subsequent applications of the block-encoding.

1. Prepare Oracle

The *Prepare* oracle is applied to the index register to prepare a superposition over the computational basis states of this register.

Uniform State Preparation

The simplest implementation of the *Prepare* oracle is to prepare a uniform superposition over the computational basis states using a series of Hadamard gates on each qubit in the register. The rescaling condition for this implementation is that the coefficients must be rescaled such that they all have magnitude ≤ 1. Let the largest coefficient magnitude be given by $\alpha^* = \max |\alpha_l|$. Then, this rescaling imposes a constant rescaling factor of α^* and we refer to these rescaled coefficients as $\tilde{\alpha}_l$. An additional rescaling factor of $2^{\lceil \log_2 L \rceil}$ from the uniform superposition results in an overall rescaling of the Hamiltonian by a constant factor of:

$$\lambda_{usp} = 2^{\lceil \log_2 L \rceil} \alpha^* \tag{12}$$

such that the rescaled Hamiltonian is given by:

$$\bar{H} = \frac{H^*}{\lambda_{usp}} \tag{13}$$

The quantum resource requirements of this implementation of *Prepare* are negligible as no ancillae are required and only Clifford operations (Hadamards) are performed. However, this implementation of *Prepare* requires that the rescaled coefficients of the terms ($\tilde{\alpha}_l$) be incorporated within the *Select* oracle which will be described in Subsubsection III D 2.

Arbitrary State Preparation

An alternative implementation of *Prepare* is to use the same implementation traditionally used in LCU circuits. The coefficients are first rescaled by their $L1 - norm$:

$$\lambda_{asp} = \sum_{l=0}^{L-1} |\alpha_l| \tag{14}$$

and the rescaled Hamiltonian is given by:

$$\bar{H} = \frac{H^*}{\lambda_{asp}} \tag{15}$$

It should be noted that $\lambda_{asp} \leq \lambda_{usp}$ with equality when the coefficients of the terms all have equal magnitude (confirm this) and give some analytical proof for this bound.

Then, a state preparation routine is applied that performs the following operation:

$$\left| 0^{\otimes \lceil \log_2 L \rceil} \right\rangle \rightarrow_{Prepare} \sum_{l=0}^{L-1} \sqrt{|\alpha_l|/\lambda_{asp}} \left| l \right\rangle \tag{16}$$

This results in a weighted superposition over the computational basis states where the squared amplitudes of the basis states are equal to the magnitude of the associated coefficients.

For Hamiltonians that have structure among the coefficients of the terms, implementations of the *Prepare* oracle can be constructed that leverage this structure. In certain cases, this can drastically reduce the cost of *Prepare* such as is done for the Fermi-Hubbard model in [3]. However, when a certain structure cannot be assumed, the Grover-Rudolph algorithm from [4] gives a formulaic routine to generate these *Prepare* circuits.

Gover-Rudolph requires $L - 1$ rotations, with 1 being uncontrolled and the others controlled. The implementation also requires several series of multiplexing. I think we can make the implementation use fewer Toffolis by doing some smart multiplexing. The total number of Toffolis (if my assumption is correct) is given by:

$$N_{toff} = \lfloor \frac{L}{2} \rfloor \tag{17}$$

and the total number of ancillae is given by:

$$Q_{asp} = \lceil \log_2 L \rceil - 2 \tag{18}$$

check this. It should be noted that these ancillae, which will be returned to $|0\rangle$ after the oracle call completes, will likely be reused in the *Select* oracle and therefore will not contribute to the total qubit requirement of the block-encoding.
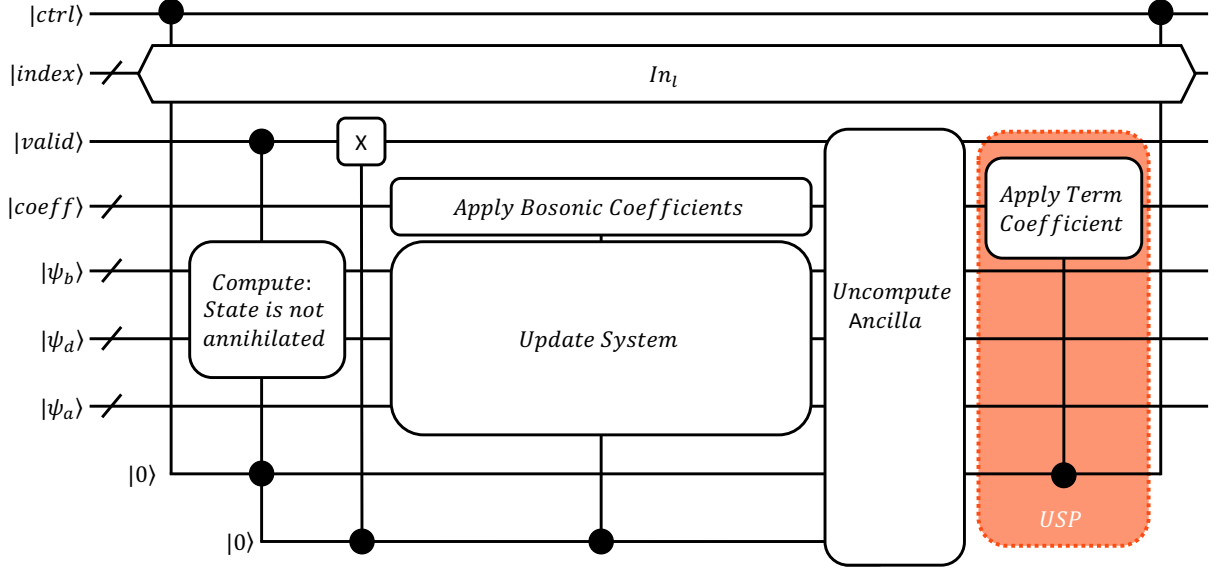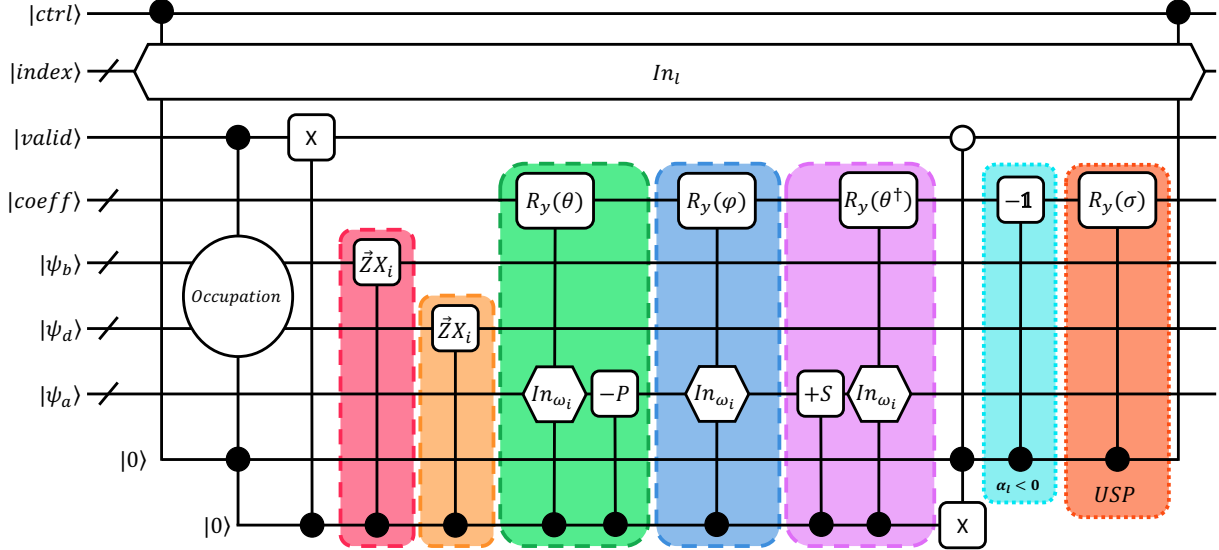
FIG. 2. **Ladder Operator Select Oracle.**

FIG. 3. **Ladder Operator Select Oracle.**

Overall, after the bosonic coefficient rescaling is applied and one of these *Prepare* oracle compilations is chosen and the coefficients rescaled again, the overall rescaling of the Hamiltonian that is block-encoded is given by:

$$\bar{H} = \frac{H}{(\Omega + 1)^{K/2}\lambda} \tag{19}$$

with λ being either λ_{usp} or λ_{asp}.

2. Select Oracle

The *Select* oracle is shown in Figure 1 as the oracle within the blue shaded region. As is done in standard LCU circuits, the *Select* oracle is designed such that it applies the term T_l onto the system when the index register is in the computational basis state $|l\rangle$. In Figure 1, the hexagonal box represents the multiplexing operation given in Figure

7 of [3] which creates a coherent for-loop over the computational basis states using $L - 1$ left-elbows and $\lceil \log_2 L \rceil$ ancillae qubits.

In Figure 2, we depict a broad schematic for applying a single term within the *Select* oracle in LOBE and in Figure 3 we give a more explicit compilation. The entire *Select* oracle can be constructed by applying the circuit for each term (T_l) when the ancilla qubit of the multiplexing operator over the index register is in the corresponding computational basis state.

Working from left to right, the first operation shown in Figure 3 is the left-elbow of the multiplexor which puts the ancilla qubit in the $|1\rangle$ state when the index register is in the computational basis state $|l\rangle$. Next, a left-elbow - controlled on the validation qubit being in the $|1\rangle$ state, the index register being in the $|l\rangle$ state, and the system being in a state that will not be annihilated - is applied to produce a new ancilla qubit. To perform a check on if the state will be annihilated, we control based on the occupation of the fermionic and antifermionic modes. The controls on the fermionic and antifermionic modes are determined by the operators that are present within the current term. For each fermionic (antifermionic) creation operator acting on mode i, a 0-control is added on i^{th} fermionic (antifermionic) mode within $|\psi_b\rangle$ ($|\psi_d\rangle$). Likewise, for each fermionic (antifermionic) annihilation operator acting on mode i, a 1-control is added on i^{th} fermionic (antifermionic) mode within $|\psi_b\rangle$ ($|\psi_d\rangle$). For each occupation operator acting on a fermionic (or antifermionic) mode, we include a 1-control on the ith mode since the state gets "zeroed-out" if the mode is unoccupied. If any of these conditions are not satisfied, then the state should be "zeroed-out" and the ancilla qubit will be in the $|0\rangle$ state for that branch of the wavefunction. The quantum states that should be "zeroed-out" due to the bosonic operators will be handled when applying the bosonic operators to the system.

After these checks are performed, the bottom ancilla qubit will be in the $|1\rangle$ state if and only if the three conditions are met, indicating that the term T_l should be applied onto the system. If the ancilla is in the $|1\rangle$ state, the validation qubit is first flipped from $|1\rangle$ to $|0\rangle$ to put the appropriate branch of the wavefunction into the encoded subspace. Then the term T_l is applied to the system registers as portrayed in the shaded boxes with the dashed borders in Figure 3.

In Figure 3, the first shaded box (red) shows the update on the fermionic system corresponding to a single creation or annihilation operator acting on the ith mode. Since both the creation and annihilation operators flip the occupation and apply a phase dependent on the parity of the occupation of the preceeding modes, they can both be applied using the same circuit. To apply the appropriate phase, Pauli Z gates are applied on all the fermionic modes for $j < i$. To flip the occupation of the mode that the ladder operator acts on, a Pauli X gate is applied to the ith fermionic qubit. This series of gates is repeated for all fermionic creation or annihilation operators. Many of these gates could be compiled out of the circuit, however, they are all Clifford operations and therefore will not significantly contribute to the cost of the block-encoding. Fermionic occupation (number) operators will not affect the occupation of the fermionic modes and therefore do not need to be accounted for when updating the state of the system. The second shaded box (orange) in Figure 3 portays the application of an antifermionic ladder operator and is constructed in an analogous way, but acts on the corresponding antifermionic modes.

Next, the third shaded box (green) in Figure 3 depicts the action of a bosonic annihilation operator acting on the system. This box is repeated for each bosonic annihilation operator within the current term. We begin by accounting for the coefficients associated with applying the annihilation operators onto the system. To achieve this, we multiplex over the possible occupation states of the corresponding bosonic mode and apply a rotation on a qubit in the coefficient register. Let k count the number of bosonic operators that have already been applied in this term (T_l) and P denote the exponent on the annihilation operator. A controlled-Ry rotation is applied on the kth qubit in the coefficient register with an angle given by:

$$\theta(\omega_i, P) \equiv 2\cos^{-1}\left(\sqrt{\frac{\prod_{p=0}^{P-1}(\omega_i - p)}{(\Omega + 1)^P}}\right) \tag{20}$$

where ω_i is the occupation of the ith bosonic mode that we are currently multiplexed on.

This rotates the kth qubit in the coefficient register such that the amplitude in the $|0\rangle$ state (encoded subspace) is given by:

$$R_y\left(\theta(\omega_i, P)\right)|0\rangle = \sqrt{\frac{\prod_{p=0}^{P-1}(\omega_i - p)}{(\Omega + 1)^P}}|0\rangle + \beta|1\rangle \tag{21}$$

where β is a complex coefficient that normalizes the state of the qubit. When multiple bosonic ladder operators exist within a single term, these rotations must be applied on separate qubits such that the coefficient of the all-zero state of the coefficient register is the product of these amplitudes.

For states that should be "zeroed-out" due to these operators, the coefficient picked up will be zero, therefore the state will be completely outside the encoded subspace. Alternatively, since we are multiplexing over the occupation states of the bosonic mode, one could perform classical logic to determine if the occupation (ω_i) is less than the exponent (P). If the occupation is less than the exponent, the rotation gate is equivalent to a CNOT gate which will clearly flip the state outside of the encoded subspace.

Next, the occupation of the ith bosonic mode is updated by decrementing occupation mode by a value of P. We note that for states that will be "zeroed-out", these decrementers may raise the occupation from $|0\rangle$ to $|\Omega\rangle$, however since these states are already outside of the encoded subspace, we do not need to ensure that the occupation states are consistent. The circuit for decrementing (or incrementing) the occupation state is shown in Figure **??**.

The fourth shaded box (blue) depicts the application of a bosonic occupation operator and is likewise repeated for each bosonic occupation operator in the term. For these operators, we do not need to update the occupation of the modes as the occupation states should remain unchanged. Therefore we only need to apply the bosonic coefficients which we can do in a similar way to what is done for the bosonic annihilation operators. The angle of the rotation for these gates is given by:

$$\phi(\omega_i, R) \equiv 2\cos^{-1}\left(\frac{\prod_{r=0}^{R-1}(\omega_i - r)}{(\Omega + 1)^R}\right) \tag{22}$$

where R is the exponent of the occupation operator. As was done for the bosonic annihilation operators, these rotations are applied on the kth qubit in the coefficient register such that the amplitude of the all-zero subspace of the coefficient register picks up a factor of:

$$\frac{\prod_{r=0}^{R-1}(\omega_i - r)}{(\Omega + 1)^R} \tag{23}$$

Next, the fifth shaded box (purple) shows the application of a bosonic creation operator this box is repeated for each bosonic creation operator in the term. These operations are analogous to the bosonic annihilation operators. First, the occupation of the ith bosonic mode is incremented by S using the circuits shown in Figure **??**. Then, a multiplexor is applied on the Ω occupation states of the ith bosonic mode and Pauli-Y rotations are applied on the kth qubit in the coefficient register with an angle given by:

$$\theta^\dagger(\omega_i, S) \equiv 2\cos^{-1}\left(\sqrt{\frac{\prod_{s=0}^{S-1}(\omega_i - s)}{(\Omega + 1)^S}}\right) \tag{24}$$

where S is the exponent of the bosonic creation operator. Since the occupation states are updated prior to the rotations (as opposed to afterwards in the annihilation operators), the coefficient of the rotations is computed using the same function of the occupation (ω_i) and the exponent (P or S) as is used for the annihilation operators.

Once the state of the system is updated and the bosonic coefficients have been applied, the bottom ancilla qubit can be uncomputed (reset to $|0\rangle$). This can be achieved by applying a Toffoli onto this qubit with a zero-control on the validation qubit and a one-control on the ancilla qubit used for multiplexing over the index register. It's not quite clear to me if we can use a right-elbow here instead of a Toffoli since we are changing the states of fermionic/antifermionic modes after the left-elbow. Obviously it's only one Toff so it's not a big deal, but I wanna try to do some derivations for this.

Next, if the sign of the coefficient of the current term is -1, we apply a $-\mathbb{1}$ - controlled on the index register being in the state $|l\rangle$ - to the state, regardless of which *Prepare* oracle is used. This operation is shown in the sixth shaded box (cyan) with the dotted border to indicate that this operation may or may not be present depending on the sign of the term. This can be achieved in many ways, but we show an implementation using only Clifford gates in Figure **??**.

Finally, if the *uniform state perparation* protocol is used for the *Prepare* oracle, then a controlled Pauli-Y rotation is applied onto the final qubit in the coefficient register. This is shown in the seventh shaded box (red-orange) with the dotted border to indicate that this operation may or may not be present depending on which *Prepare* oracle is used. This rotation is used to account for the rescaled coefficient of the term ($\tilde{\alpha}_l$) and is achieved by applying a rotation with an angle given by:

$$\sigma(\tilde{\alpha}_l) \equiv 2\cos^{-1}|\tilde{\alpha}_l| \tag{25}$$

If the *arbitrary state preparation* protocol is used, then this rotation is not needed as the coefficient of the term is already accounted for and one fewer qubit is needed in the coefficient register.

Alternative Bosonic Operator Implementation

This needs to be confirmed, but should be much more efficient.

Let any series of bosonic operators acting on the ith mode be described by:

$$(a_i^\dagger)^R (a_i)^S \tag{26}$$

where R and S are nonzero integers in the range $[0, \Omega]$.

We can describe the action of the rescaled operator on the state of the ith bosonic mode with initial occupation ω as:

$$(a_i^\dagger)^R (a_i)^S |\omega\rangle = \prod_{r=0}^{R-1} \left(\sqrt{\frac{\omega - S + r + 1}{\Omega + 1}}\right) \prod_{s=0}^{S-1} \left(\sqrt{\frac{\omega - s}{\Omega + 1}}\right) |\omega + R - S\rangle \tag{27}$$

This operation can implemented in the block-encoding via a quantum circuit as follows with all operations controlled on the ancilla qubit indicating if the current term should be applied:

(i) Apply a Pauli-X gate on the kth qubit in the coefficient register to rotate it completely outside of the encoded subspace

(ii) Multiplex over the occupation of the ith bosonic mode for occupation states in the range: $S \leq \omega_i \leq \min(\Omega, \Omega - S + R)$. Any state beginning with an occupation less than S will be "zeroed-out" and therefore can be left outside of the encoded subspace and the multiplexor can ignore these states. Likewise, any state that should end with an occupation greater than Ω will also be "zeroed-out" and therefore the multiplexor can terminate early.

(iii) For each value of ω_i in the multiplexor, apply a rotation on the kth qubit in the coefficient register of angle θ given by Eq. 28. The purpose here will be to rotate the qubit back into the encoded subspace with an amplitude in the $|0\rangle$ state equal to the amplitude in Eq. 27. Need to figure out if this should be an Ry rotation or something else or perhaps a rotation of $\pi - \theta$, etc... regardless I'm confident this is do-able.

(iv) If $S > R$, apply a decrementer to the ith bosonic mode, reducing the occupation by a value of $S - R$.

(v) If $S < R$, apply an incrementer to the ith bosonic mode, increasing the occupation by a value of $R - S$.

(vi) If $S = R$, don't do anything

$$\theta(\omega_i, S, R) = 2\cos^{-1}\left(\prod_{r=0}^{R-1} \left(\sqrt{\frac{\omega - S + r + 1}{\Omega + 1}}\right) \prod_{s=0}^{S-1} \left(\sqrt{\frac{\omega - s}{\Omega + 1}}\right)\right) \tag{28}$$

Overall, this implementation would mean that only one multiplexor would be required for each bosonic mode that is being acted upon, instead of one for each bosonic operator in the term. Additionally, the multiplexor will have a reduced number of occupation modes to multiplex over, thereby reducing the number of Toffolis required for the multiplexor. With this implementation, K is redefined as the maximum number of bosonic modes being acted upon in a single term (instead of the maximum number of bosonic operators within a term) which will reduce the number of qubits needed in the coefficient register. Lastly, only one incrementer/decrementer circuit is required since the occupation state will only need to be updated once for each bosonic mode being acted upon.

E. Analytical Cost Analysis

The minimum number of qubits required for the index register is given by:

$$Q_{index} = \lceil \log_2 L \rceil \tag{29}$$

If we let K denote the maximum number of bosonic operators within a single term, then the minimum number of
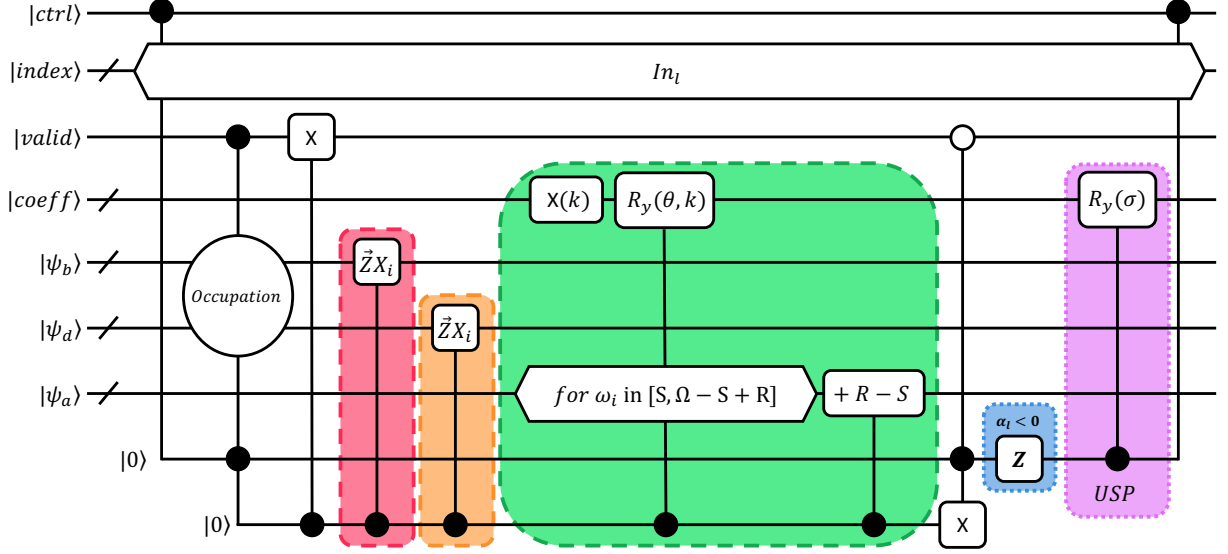
FIG. 4. **Ladder Operator Select Oracle - Normal Ordering.**

qubits in the coefficient register is given by:

$$Q_{coeff} = K + 1 \tag{30}$$

Using the qubit-efficient encoding described in Subsection II A, the minimum number of qubits required for the system registers is given by:

$$
\begin{aligned}
Q_{\psi_b} &= \Lambda \\
Q_{\psi_d} &= \Lambda \\
Q_{\psi_a} &= \Lambda \lceil \log_2 (\Omega + 1) \rceil
\end{aligned} \tag{31}
$$

There are numerous space-time tradeoffs that one can make to either reduce the number of ancillae qubits at the cost of more gates or vice versa. In this work, we opt for compilations that minimize the number of non-Clifford operations at the expense of more ancillae qubits. With this choice, the number of clean ancillae required is given by:

$$\alpha = \lceil \log_2 L \rceil + (B + 1) + \lceil \log_2 (\Omega + 1) \rceil \tag{32}$$

where $\lceil \log_2 L \rceil$ qubits are used for multiplexing over the index register, $\lceil \log_2 (\Omega + 1) \rceil$ qubits are used for multiplexing over the bosonic occupation registers, and B denotes the maximum number of fermionic and antifermionic ladder operators within a single term. And we assume that any left-elbow with N controls and one ancilla qubit storing the quantum boolean can be decomposed into $N - 1$ left-elbows, each using 2 controls at the expense of $N - 1$ total ancillae.

Therefore, the total qubit requirement under these compilation choices - disregarding the control qubit - is given by:

$$
\begin{aligned}
N &= Q_{index} + 1 + Q_{coeff} + Q_{\psi_b} + Q_{\psi_d} + Q_{\psi_a} + \alpha \\
&= 2 \lceil \log_2 L \rceil + 2\Lambda + (\Lambda + 1) \lceil \log_2 (\Omega + 1) \rceil + B + K + 3
\end{aligned} \tag{33}
$$

F. Example

Step-by-step example (intention is to move this to an appendix)

IV. RESULTS / NUMERICAL BENCHMARKING

here we'll "benchmark" (aka numerically compute the cost - number of qubits and gates - of creating a lobe block-encoding) for some systems

ideas of systems to benchmark on:
- Fermi-Hubbard
- Something with just bosons
- Something with fermions, antifermions, and bosons

V. CONCLUSIONS

Future Directions The application of the bosonic coefficients is a potential area of improvement for more efficient implementations. Primarily, if one could coherently compute $\cos^{-1}\sqrt{\omega_i}$ either in-place or out-of-place, it may be possible to use techniques such as hamming weight phasing to apply the appropriate coefficient without the need for multiplexing over all possible occupation states. Alternatively, if the coefficient qubits begin outside of the encoded subspace (as opposed to inside), it may be possible to multiplex only over the $\Omega - R$ occupation states that will not be "zeroed-out". This would reduce the number of Toffolis by R (S/T) for each bosonic ladder operator.

[1] K. Lamar, *Not Like Us*, 1st ed. (Interscope Records, 2024).
[2] D. Liu, W. Du, L. Lin, J. P. Vary, and C. Yang, An efficient quantum circuit for block encoding a pairing hamiltonian, arXiv preprint arXiv:2402.11205 (2024).
[3] 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).
[4] L. Grover and T. Rudolph, Creating superpositions that correspond to efficiently integrable probability distributions, arXiv preprint quant-ph/0208112 (2002).