Quantum algorithms promise an immense improvement to our current information processing capabilities by utilizing interference phenomena in an exponentially large Hilbert space. However, the large size of the Hilbert space also poses a crucial challenge to the experimentalists, who strive to design protocols that efficiently navigate the Hilbert space using only a small number of semiclassical control fields. Here, we design a set of multi-qubit Rydberg blockade gates that provide a solution to this control challenge, by enabling the implementation of a vast variety of quantum algorithms with a gate-induced error-scaling exponentially slower than the circuit size. This exponential improvement is enabled by a technique from the quantum optician’s toolbox: Electromagnetically Induced Transparency (EIT). Specifically, we show that the proposed blockade gates facilitate a (i) error robust (ii) shallow depth and (iii) scalable implementation of the so-called block-encoding unitary, the building block of the Quantum Signal Processing (QSP) framework that unifies the most important subroutines of quantum algorithms. Our result highlights the advantages of studying concrete experimental implementations when discussing efficiency, and suggests a more useful alternative to the computational complexity paradigm based on gate counts. To showcase our approach, we construct explicit blueprints to implement QSP-based near-optimal Hamiltonian simulation on the Rydberg atom platform. Our protocols substantially improve both the scaling and the overhead of gate errors in comparison to those implementing the product-formula-based near-optimal simulation algorithm.

I. INTRODUCTION

Processing information by manipulating quantum mechanical degrees of freedom has been a holy grail of quantum science since the design of first quantum algorithms [1–5] that utilize superposition and entanglement. These algorithms promise a profound transformation of our current computational capabilities, by reducing the space and time required to accomplish important computational tasks, such as phase estimation [6], unstructured search [2], and Hamiltonian simulation [4].

Although the fundamental structure of quantum mechanics that justifies the promise of “Quantum Advantage” remains unclear, a recent framework called Quantum Signal Processing (QSP) [7, 8] and its generalization Quantum Singular Value Transformation (QSVT) [9, 10] managed to collect a plethora of otherwise disparate quantum algorithms, from phase estimation to Hamiltonian simulation, under one operational umbrella. Moreover, expressing these algorithms within the QSP framework yields exponentially improved scaling of the computational costs with respect to the error tolerance [9]. The successes of the QSP framework hint at the possibility of determining the aspects of Quantum Mechanics relevant to Quantum Advantage [11], and made QSP an important focus of activity in the field of theoretical quantum computation.

Clearly, the most important aspect of a unifying framework such as QSP is that it provides the basis for an overarching approach to the outstanding challenges of quantum information processing. In this work, we utilize QSP to address what we call the “control challenge”, which can be summarized in a single question: “What are the limits to efficiently control the time evolution in a large Hilbert space using only a exponentially small number of classical control fields?” [12].

A concrete and efficient implementation of block-encoding unitaries is paramount from the perspective of the control challenge. The block-encoding unitaries are building blocks of the QSP framework. They provide a common structure that enables us to answer to the control challenge in a comprehensive manner. Specific algorithms within the QSP framework can be instantiated with the help of a so-called signal operator $A$, encoded in a block-encoding unitary $U$ [13]. Formally,

$$U \equiv \left( \begin{array}{ccc} A & * & * \\ * & * & * \end{array} \right),$$

where the elements $*$ are unspecified, and the unitarity of $U$ implies $||A|| \leq 1$. For example, in the case of Hamiltonian simulation and phase estimation algorithms, $A$ is a Hamiltonian and the generator of the relevant unitary oracle, respectively. Given a suitable block-encoding unitary, QSP implements the target quantum algorithm by

\* sina_zeytinoglu@fas.harvard.edu
iterating $U$ interspersed with a set of single-qubit rotations $S_1, S_2$.

Here we demonstrate that the recently developed control framework of Linear Combination of Unitaries (LCU) leads to an efficient implementation block-encoding unitaries on the concrete experimental platform of Rydberg atoms [14, 15]. The most striking aspect of the proposed implementation protocol is its robustness to errors native to Rydberg atoms. In particular, we show that when the range of Rydberg interactions is large enough and the signal operator is $k$-local, the errors in our implementation scale independently of the system size. This is achieved by designing a gate which is controlled by electronic induced transparency technique. In addition, we provide non-trivial lower bounds for time and space complexity of implementing a block-encoding unitary on the Rydberg platform by formalizing the experimental constraints that apply to current Rydberg atom experiments. We note that although here we consider only Hermitian signal operators for simplicity, our methods generalize straightforwardly to the case of general signal operators. Moreover, our results highlights the necessity of studying detailed and concrete experimental implementations of the underlying unitaries when analysing the performance of quantum algorithms. Given the emergence of the Rydberg atoms as a leading experimental platform for accomplishing useful quantum information processing tasks [14], as well as emulating a variety of non-trivial Hamiltonian dynamics [19] and ground states [20–24], we expect that our contribution will serve as a conceptual bridge between the current experiments on the Rydberg atom platform and the cutting-edge of quantum algorithms design.

Before proceeding to a detailed discussion of how we arrived at the results listed above, we give a brief introduction to how the control challenge manifests itself in the context of the Rydberg atom platform, and discuss the main strategies that we used to respond to it in this work.

To begin with, we distinguish the classical and quantum degrees of freedom of the Rydberg atoms platform. The classical control of the dynamics is realized by shining laser pulses incident on individual Rydberg atoms, whose spatial positions are fixed by optical tweezers [24]. On the other hand, the quantum mechanical evolution of the system is governed by the local few-level systems realized by each atom, as well as the dipolar interactions between the atoms [14, 15]. In the absence of interactions the evolution of $n_{\text{site}}$ Rydberg atoms can be easily controlled and is described by a tensor product of single-qubit rotations, which we denote as $U_1 \otimes \cdots \otimes U_{n_{\text{site}}}$. Although no evolution of this form is sufficient for describing universal quantum computation, linear combinations of the members of the $n_{\text{site}}$-qubit Pauli group $P_i \in \{I, \sigma_x, \sigma_y, \sigma_z\}^{\otimes n_{\text{site}}}$ (i.e., $U_1^{(i)} \otimes \cdots \otimes U_{n_{\text{site}}}^{(i)} = P_i$) are sufficient to represent any evolution in the Hilbert space. Therefore, in this context, the control challenge can be responded to by discovering efficient control protocols that implement ever more complex evolutions derived from the simple unitaries $P_i$.

From the theoretical perspective, the recent development of two frameworks (i) Linear Combination of Unitaries (LCU) [24] and (ii) Quantum Signal Processing (QSP) [8] provides techniques to design any desired evolution. More specifically, the LCU method implements linear combinations of $P_i$ with the help of an “ancillary” register introduced in addition to the “system” degrees of freedom that we want to control. Entangling orthogonal states $|x_i\rangle$ of the ancillary register to a unique evolution described by $P_i$ allows for the implementation of any Hermitian operator $A = \sum_{i=1}^{N} |\alpha_i|^2 P_i$ ($\alpha_i \in \mathbb{R}$ and $\sum_{i} |\alpha_i|^2 \leq 1$) on the Rydberg system, block-encoded as in Eq. (1). On the other hand, QSP enables the implementation of the block-encoding unitary for any polynomial function $P(A)$ using only a single additional ancilla. In principle, these two frameworks enable the experimentalist to control the evolution of the system arbitrarily with enough computational resources.

In practice, however, the computational resources are scarce and any control protocol based on LCU and QSP frameworks should aim to be efficient with respect to resource requirements. Here, we evaluate the efficiency of a control protocol measuring the following resource costs: (i) the time it takes to complete the protocol, given by the depth of the control circuit, (ii) the number of ancillary qubits required, and (iii) the robustness of the control protocol to errors. In addition, it is also desired to implement scalable control protocols whose computational resource requirements scale slowly with the size of the controlled system.

In this work, we develop design strategies which utilize the unique properties of the Rydberg atom platform and allow us to implement control protocols for LCU- and QSP-based algorithms using near-minimal resource requirements. To summarize:

1. We design multi-qubit Rydberg gates to efficiently load classical data into the quantum processor and to apply conditional unitary operators. The error model of our gate set has strong state-dependent biases, which can be utilized to increase the robustness of proposed control protocols to native errors. To take into account the effects due to biased errors, we introduce an ”error-bounded gate count” (EBGC). Crucially, we show that the EBGC for implementing Eq. (1) is not only smaller than the conventional gate count, which is related to the circuit size, but it only depends on the largest support of the Pauli operators that make up $A$.

2. We design shallow control protocols to implement LCU-based block encoding of $A$. This analysis utilizes (i) the tree structure of the data describing $A$, (ii) the tunability and the geometry of dipolar interactions between Rydberg atoms, and (iii) constraints on the number of laser pulses incident on each atom. Our analysis imply that, assuming that
the maximum range of interactions is larger than the system size, the block-encoding unitary of any geometrically local $A$ can be implemented in constant depth.

3. We construct schemes which demonstrate that the multi-qubit Rydberg gates we use as the building blocks of our control protocols \cite{13,15,17,27,30} are scalable on a modular network, where each module communicates a single qubit of information to its neighbors. Remarkably, our FANIN and FANOUT protocols satisfy the lower bounds for circuit depth due to locality \cite{35,39}.

4. All of our protocols involve $O(n_{site})$ ancilla atoms, and therefore does not change the overall space complexity of the implementation.

In order to showcase the power of our approach, we give a concrete blueprint to implement the optimal simulation algorithm of Haah et al. \cite{31} for local Hamiltonians and the optimal Hamiltonian simulation algorithm of Ref. \cite{32}. In this context, the favorable scaling of EBGC for implementing block-encoding unitaries entails that the query-optimal QSP-based Hamiltonian simulation algorithm of Ref. \cite{32} also has optimal $O(n)$ EBGC complexity for $k$-local Hamiltonians. Lastly, we benchmark our blueprint using the one-dimensional disordered Heisenberg model \cite{31,33,34}, and show that the EBGC overhead of an implementation of the QSP-based simulation algorithm of Ref. \cite{32} on the Rydberg platform is more than one order of magnitude smaller than that of the near-optimal, product-formula-based Hamiltonian simulation of Ref. \cite{34}.

II. PAPER ORGANIZATION

Here we discuss the organization of the paper and highlight our technical results. In Sections \ref{S:LCU} and \ref{S:QSP} we introduce the LCU and QSP frameworks without any reference to their concrete implementation. In Section \ref{S:blocks} we discuss the tree-like classical data structure describing the block-encoded operator $A$, and how this structure entails the existence of shallow depth circuits for LCU-based block-encoding unitaries. Moreover, the tree structure can also be utilized to study the minimum depth of the protocols implemented on a particular platform (e.g., consisting of Rydberg atoms), by translating the physical constraints relevant that platform to constraints that determine which paths on the data-tree can be transverses simulatenously.

In Section \ref{S:specific} we introduce the specific laser pulse sequences that implement native multi-qubit Rydberg blockade gates, which serve as the building blocks for the proposed implementation of LCU-based block-encoding unitaries. Specifically, we introduce two multi-qubit gates: (i) the One-Hot amplitude encoding gate $V_{\text{OHE}}$ and (ii) the multi-qubit controlled Pauli operators $C_{x,P}$, where $x_i$ is a binary vector describing the control condition. While $V_{\text{OHE}}$ provides an efficient way of loading real valued amplitude data into the quantum processor, the $C_{x,P}$, efficiently loads the binary data describing each multi-qubit Pauli operator in the decomposition of $A$. Moreover, using the realistic error models for controlling the Rydberg system, we devise a way of calculating the error bounds for each control protocol. We codify the rules for calculating the errors associated with each protocol under an “error-bounded gate count” (EBGC) (see Section \ref{S:EBGC}). This error model also provides the guidelines to design protocols which are robust to native errors of the Rydberg atom platform. Our most important contribution to this end is the identification of a class of single-qubit controlled unitary operators which, thanks to their use of Electromagnetically induced transparency (EIT) \cite{30}, introduce errors only when the control condition is satisfied (see Section \ref{S:EIT}).

In Section \ref{S:advantages} we comment on the advantages of blockade mechanism itself, which utilizes always-on two-qubit interactions, compared to a setup where the multi-qubit gates are implemented by controlling the strength of two-qubit interactions themselves. In particular, we present a comparison of the computational resources required to implement the $V_{\text{OHE}}$ and $C_{x,P}$ gates using these two setups. Lastly, in Section \ref{S:conclusion} we present the formalization of constraints originating from the physical properties of Rydberg interactions and the experimental constraint that each qubit can be driven by a few laser pulses at a time. These constraints determine the number of $C_{x,P}$ gates traversing sub-trees of the data tree that can be applied in parallel.

Armed with these tools, we provide three different protocols for the implementation of the LCU-based block-encoding unitaries on the Rydberg atom platform (see Section \ref{S:protocols}). We also calculate the computational resources needed to implement each one of these protocols according to the discussion in Section \ref{S:EBGC}. Most importantly, we show that that the block-encoding of any $k$-local operator $A$ can be implemented using only $O(k)$ error-bounded gates. Moreover, in the case that the Pauli operators that make up $A$ are also geometrically local, the depth of implementation can also be made constant. Lastly, in Section \ref{S:discussion} we introduce a simple unitary that clarifies the connection between the LCU method and state-preparation algorithms.

Up to this point, we assume that the blockade radius can be taken to be infinite if needed. We treat the cost of scaling our protocols when the maximum blockade radius is finite in Section \ref{S:finiteness}. To this end, we discuss a generic technique for implementing the proposed native multi-qubit gates in a scalable way, and show that the circuit depth for implementing the ever larger multiqubit controlled Pauli operators matches the lower bound given by the Lieb-Robinson bounds \cite{35,39}.

In Section \ref{S:conclusion} we discuss the problem of Hamiltonian simulation. In Section \ref{S:comparison} we provide a comparison of the circuit depths and EBGCs associated with
three different near-optimal Hamiltonian simulation algorithms. Our calculation yield two important results. First, we show that the QSP-based Hamiltonian simulation of Ref. [22] can be implemented with an optimal EBGC scaling. Secondly, the error-bounded gate overhead of this algorithm is much smaller than that of other nearly optimal local-Hamiltonian simulation algorithms in Refs. [31, 34]. We conclude and give a brief opinion on the outlook of our work in Section X.

III. BLOCK ENCODING BY LCU: A QUANTUM CONTROL PERSPECTIVE

Here we discuss the method of LCU [26], which offers a generic and constructive strategy to implement block-encoding unitaries for linear combinations of multi-qubit Pauli operators. In order to assess the time and space complexities of the LCU method, we introduce the scaling variable \( N \) which denotes the number of Pauli operators that constitute the target operator \( A \). In particular, we decompose \( A \) as

\[
A = \sum_{i=1}^{N} |\alpha_i|^2 P_i, \tag{2}
\]

where we set \( \sum_{i=1}^{N} |\alpha_i|^2 = 1 \). In the context of Hamiltonian simulation, the number of coefficients required to implement a \( k \)-local Hamiltonian on a system consisting of \( n_{\text{site}} \) qubits is \( N = O(n_{\text{site}}^k) \), while for geometrically local Hamiltonians where the number of atoms within an interaction range is \( N_I \), we have \( N = O(N_{I}^{2}n_{\text{site}}) \). It is important to note that in this decomposition we assume that the coefficients \( \{|\alpha_i|\} \) are given and cannot be further compressed into a smaller set.

In the following, we first review the LCU method formally, and then discuss how its structure results in a circuit that loads the classical data describing \( A \) into a quantum processor in constant time. In Section V, we demonstrate how the constraints associated with the physical setup of Rydberg atom experiments affects this efficiency, and provide the relevant gate and ancilla counts.

A. Algorithm:

The LCU decomposition of the block-encoding unitary in Eq. (3) consists of three unitaries [26].

\[
U = V^\dagger \bar{U} V. \tag{3}
\]

The block-encoding unitary acts on \( n_a \) ancilla qubits and \( n_{\text{site}} \) system qubits. The unitary \( V \) rotates the \( n_a \)-qubit initial ancilla state \( |0\rangle^\otimes n_a \) to a linear combination of the computational basis states \( \{|x_i\} \) which encode the precomputed classical coefficients \( \alpha_i \).

\[
|\Psi\rangle_a \equiv V |0\rangle^\otimes n_a = \sum_{i=1}^{N} \alpha_i |x_i\rangle. \tag{4}
\]

The operator \( V \) can be understood as an amplitude-encoding state-preparation unitary [37]. We note that the number of ancilla qubits \( n_a > \lceil \log N \rceil \) depends on the choice of the basis \( \{|x_i\}\}

Then, we apply the following conditional unitary operation

\[
\bar{U} \equiv \sum_{i=1}^{N} |x_i\rangle \langle x_i| \otimes P_i. \tag{5}
\]

The action of \( \bar{U} \) entangles each Pauli operator with an orthogonal address state of the ancilla register.

\[
\sum_{i=1}^{N} \alpha_i |x_i\rangle \otimes |\Psi_{\text{sys}}\rangle \bar{U} = \sum_{i=1}^{N} |x_i\rangle \otimes (\alpha_i P_i |\Psi_{\text{sys}}\rangle). \tag{6}
\]

Finally, a block-encoding of a superposition of multi-qubit Paulis \( \{P_i\} \) is obtained by rotating the address space by an application of \( V^\dagger \)

\[
V^{-1} \bar{U} |\Psi_{a}\rangle \otimes |\Psi_{\text{sys}}\rangle = \sum_{i=1}^{N} |\alpha_i|^2 (|0\rangle \otimes |\Psi_{\text{sys}}\rangle) + |\Psi_{-}\rangle
\]

\[
= |0\rangle \otimes |A |\Psi_{\text{sys}}\rangle + |\Psi_{-}\rangle \tag{7}
\]

where the unnormalized wavevector \( |\Psi_{-}\rangle \) satisfies \( \left( |0\rangle \otimes |\Psi_{a}\rangle \langle 0| \otimes 1 \right) |\Psi_{-}\rangle = 0 \), in accordance with the form of the block-encoding unitary in Eq. (1). We remind the reader that the unitarity of \( U \) implies that the Hermitian operator block-encoded in this way satisfies \( ||A|| \leq 1 \).

As we noted before, the ancillary Hilbert space is not constrained in the above discussion. While the original discussion of block-encoding unitary sets \( n_a = \lfloor \log (N) \rfloor \) [13], we refrain from this choice for two reasons. First, preparing the state \( |\Psi\rangle_a \) using only \( \lfloor \log (N) \rfloor \) ancillas is practically difficult and additional ancillae are needed to realize the state preparation in a geometrically local system. Secondly, increasing \( n_a \) allows us to impose a desired structure onto the basis states \( \{|x_i\} \). For instance, in a system evolving under a local Hamiltonian it is desirable that the states \( \{|x_i\} \) reflect the geometric locality of the Pauli operators \( \{P_i\} \). In Section VII A, we demonstrate the advantages of choosing \( \{x_i\} \) as One-Hot encoded bitstrings.

It should be emphasized that there are infinitely many ancillary states which result in a block-encoding of the same signal operator. To see this, divide the ancilla register into two parts \( a_1 \) and \( a_2 \) consisting of \( n_{a_1} \) and \( n_{a_2} \) ancillary qubits, respectively. Then we can construct two state-preparation unitaries \( V_{a_1} \) and \( V_{a_2} \) that are equivalent.
from the perspective of LCU-based block-encoding, if $\hat{U}$ acts on only the system and second ancilla register $a_2$:

$$
|0\rangle^\otimes n_{a_1} \otimes |0\rangle^\otimes n_{a_2} \rightarrow |0\rangle^\otimes n_{a_1} \otimes \sum_{i=1}^{N} \alpha_i |x_i\rangle
$$

$$
|0\rangle^\otimes n_{a_1} \otimes |0\rangle^\otimes n_{a_2} \overset{V_a}{\rightarrow} \sum_{i=1}^{N} \alpha_i |\Psi_i\rangle |x_i\rangle,
$$

where the states $\{|\Psi_i\rangle\}$ can be any state of the Hilbert space of $n_{a_1} \geq n_{a_2}$ qubits. This freedom becomes useful in our constructions. In Section VLA3, we show that the state preparation protocols of the form $V_a$ allow us to construct corresponding circuits which are exponentially more robust to gate errors.

B. Relation to the control challenge

Overall, the LCU algorithm can be thought of as a way of encoding the classical information describing the Hermitian operator $A$ into a quantum processor. First of all, any Hermitian operator $A$ is expressed by $\hat{2}$ and any Pauli operator $P_i$ can be factorized as

$$
P_i = \prod_{j=1}^{n_{\text{site}}} \left((-i)^{r_{ij}} \sigma_{x,j}^{\alpha} \sigma_{y,j}^{\beta} \sigma_{z,j}^{\gamma} \right)^{S_{ij}(i)} (\sigma_{x,j}^{\alpha} \sigma_{y,j}^{\beta} \sigma_{z,j}^{\gamma})^{S_{ij}^\dagger(i)}
$$

where $\sigma_j^{\alpha}$ is a single-qubit Pauli operator acting on $j$. The classical data which describes $A$ has the form of a tree structure depicted in Fig. 1 with three levels and two types of nodes. Each node stores some classical data which can be encoded in the amplitude of laser pulses incident on the quantum processor. The rectangular nodes store the information of the amplitudes $\{|\alpha_i\rangle\}$, while the circular modes store the binary data which describes the multi-qubit Pauli operator $P_i$. In the case of the data tree in Fig. 1, the data stored in level 1 is an $N$ dimensional vector $\alpha$ of the set of real coefficients $\{|\alpha_i\rangle\}$. In level 2 each data is binary, i.e., $S_{ij}(i) \in \{0, 1\}$, where $S_{ij}$ is a $n_{\text{site}}$-dimensional vector. Finally, in level 3, the data for each $i$ is stored in $n_{\text{site}} \times 2$ matrix, $r_{ij}(i) \in \{0, 1\}$

1. For each circular node which corresponds to an input operator $O$ and has a binary data $a \in \{0, 1\}$, output $O^a$. In the third level, $O \in \{i1, \sigma_x^\alpha, \sigma_y^\beta, \sigma_z^\gamma\}$ at site $j$.

2. For each group of circular nodes sharing the same parent node, multiply the output operator of each circular node set the result as the input of the parent. I.e., prepare $O = (-i)^{r_{ij}^1} \sigma_{x,j}^{\alpha} \sigma_{y,j}^{\beta} \sigma_{z,j}^{\gamma}^{r_{ij}^2}$.

3. Run 1 in the second level. Here $O = (-i)^{r_{ij}^1} \sigma_{x,j}^{\alpha} \sigma_{y,j}^{\beta} \sigma_{z,j}^{\gamma}^{r_{ij}^2}$.

4. For each rectangular node storing $\alpha_i$ and the input operator $O_i$, output $\alpha_i O_i$.

5. For each group of rectangular nodes sharing the same parent node, add the output operators of each rectangular node and set the result as the input of the parent: $\sum_{i=1}^{n_{\text{site}}} \alpha_i O_i$.

Each of the 2 distinct units that make up LCU-based block-encoding traverse the levels of the data tree in Fig. 1. The amplitude encoding unitary $V$ connects the root of the tree to the first layer, and $\hat{U}$ connects the first layer to the leaves. Ideally, we would like to implement each step of the LCU in constant time (i.e., independent of $n_{\text{site}}$ and $N$), which would entail that we can succeed in loading classical data of size $O(2Nn_{\text{site}})$ into the quantum degrees of freedom in constant time. However, physical constraints associated with the Rydberg atom platform limits parallelized data loading into a quantum processor realized in experiments. As we discuss in Section VLA3, our concrete implementation of LCU-based block-encoding unitaries highlights the important consequences of physical constraints, such as the rotational symmetry of blockade interactions and that each atom is manipulated by a few laser pulses at a time, that limit the number of operations that can be realized simultaneously.

The LCU-based block-encoding scheme described in this enjoys a nice property that the resulting unitary is also Hermitian (i.e., $U^\dagger = U$). In other words, the LCU-based block-encoding is “qubitized” [32]. In the next section, we discuss how the qubitized block-encoded unitaries allow for a simple implementation of algorithms in the framework of Quantum Signal Processing (QSP), which uses only one additional ancillary qubit [8].

IV. PROCESSING OF BLOCK ENCODED MATRICES BY QSP

In this section, we review QSP framework introduced in Refs. [8, 32]. From the perspective of quantum control, QSP can be understood as a efficient way of manipulating a block-encoded operator $A$ to realize the block-encoding of a polynomial function $P(A)$. As we discuss in the following, the polynomial $P(A)$ is defined through an ordered list of angles $\{|\phi_i\rangle\}$, whose size determines the order of the polynomial as well as the query complexity of QSP.

We begin our discussion with the quantum control theory of a single qubit. The pioneering work of Ref. [8] asked the following question. Given two unitary operations on a single-qubit $G(\phi) = e^i\phi\sigma_z$, and $R(\theta) = -ie^{i\theta/4\sigma_x}e^{i\theta/4\sigma_y}e^{-i\theta/4\sigma_x}$, with Pauli operators $\sigma_x, \sigma_y, \sigma_z$, what single-qubit transformations can we design by the
the constraint \( |P(x)|^2 + (1 - x^2)|Q(x)|^2 = 1 \) for \( x \in [0, 1] \).
eigenvalues $\lambda_i$ of $A$ (e.g., $A |\lambda\rangle = \lambda |\lambda\rangle$) to rotation angles $\theta_i$ for the exit ancilla.

Formally, given that the block-encoding unitary is Hermitian $U^2 = 1$, we can introduce an iterate $W \equiv (2 (|0\rangle \langle 0|)^{\otimes n_a} - 1)U$, which can be written as a direct sum over $SU(2)$ invariant subspaces associated with each eigenvalue of $H$

$$W = \bigoplus_\lambda \left( \frac{\lambda}{\sqrt{1-\lambda^2}} - \frac{\sqrt{1-\lambda^2}}{\lambda} \right) \lambda,$$

where the subscript $\lambda$ means that the matrix representation is written in the following basis

$$|G_\lambda\rangle = |0\rangle^{\otimes n_a} |\lambda\rangle \quad |G_\lambda^\perp\rangle = \frac{\lambda |G_\lambda\rangle - U |G_\lambda\rangle}{\sqrt{1-\lambda^2}}.$$

Hence, the eigenvectors of $W$ are given by

$$|G_{\lambda^\perp}\rangle = \frac{1}{\sqrt{2}} (|G_\lambda\rangle \pm |G_\lambda^\perp\rangle),$$

with associated eigenvalues $e^{\pm i\theta_\lambda}$, where $\theta_\lambda \equiv \arccos(\lambda)$.

Ref. [8] showed that using a controlled version of $W$ with the exit-ancilla as the control, it is possible to implement the unitary,

$$U_\phi = \sum_{\lambda,\eta=\pm} R_\phi(\theta_\lambda) \otimes |G_{\lambda\eta}\rangle \langle G_{\lambda\eta}|,$$

which rotates the exit-ancilla along a fixed axis on the $x$-$y$ plane as determined by $\phi$ and by an angle determined by the phased of the eigenvalue $e^{i\theta_\lambda}$. The decomposition of $U_\phi$ in terms of a controlled version of $W$ and single-qubit rotations of the exit-ancilla is the following:

$$U_\phi = (e^{-i\phi \sigma_z^{(\text{ex})}}/2 \otimes 1) U_0 (e^{-i\phi \sigma_z^{(\text{ex})}}/2 \otimes 1)$$

$$U_0 \equiv |+\rangle_{\text{ex}} \langle +| \otimes 1 + |-\rangle_{\text{ex}} \langle -| \otimes W$$

$$= \sum_{\lambda,\eta=\pm} e^{i\eta \theta_\lambda}/2 R_{\text{ex}}(\eta \theta_\lambda) \otimes |G_{\lambda\eta}\rangle \langle G_{\lambda\eta}|,$$

and $\sigma_z^{(\text{ex})}$ is a Pauli operator acting on the exit ancilla.

Noting that the eigenvectors of $W$ satisfy

$$\langle 0^{\otimes n_a} \otimes 1_{\text{sys}} |G_{\lambda^\perp}\rangle = \sqrt{\frac{1}{2}} |\lambda\rangle,$$

an arbitrary-parity polynomial of a Hermitian signal operator can be block-encoded. That is,

$$|+\rangle_{\text{ex}} 0^{\otimes n_a} \prod_{j=1}^{k/2} U_{\phi_{2j}} U_{\phi_{2j+1}+\pi} |0\rangle^{\otimes n_a} |+\rangle_{\text{ex}}$$

$$= \sum_\lambda \tilde{P}(\lambda) \langle \lambda | \langle \lambda |,$$

where $\tilde{P}(x)$ is an arbitrary parity polynomial of degree $k$. As a result, QSP provides an indispensable tool for processing block-encoded signal operators. Most importantly, for the following discussion on Hamiltonian simulation we would like to implement $\tilde{P}(\lambda) \approx e^{i\lambda t}$. In Fig. 2 we provide the circuit diagrams for implementing QSP-based algorithms.

V. RYDBERG ATOM GATES

In this section, we introduce the building blocks for implementing algorithms in the QSP framework on the Rydberg atom platform. We start the section with a general discussion of what constitutes an error-bounded gate, and how to calculate the error-bounded gate count (EBGC) of a particular protocol implemented on the Rydberg atom platform. Crucially, EBGC does not correspond to the size of the circuit decomposition of our protocols, as it takes into account the information on the input states. We briefly review the relevant level diagrams and single-qubit gates in subsection V.B. In Sections V.C and V.D we introduce two multi-qubit gates utilizing the Rydberg blockade mechanism. Each multi-qubit gate serves a different function in the efficient implementation of the LCU-based block-encoding unitary. The first multi-qubit gate, which we name “One-Hot Encoding” (OHE) gate, (see Section V.C) is used to load the classically-stored coefficient data $\{\alpha_i\}$ to orthogonal ancillary states in constant time and EBGC. In Section V.D we introduce a multi-qubit controlled Pauli operation, which can be expressed formally as

$$C_xP_1 \equiv |x_i\rangle \langle x_i| \otimes P_i + [1 - |x_i\rangle \langle x_i|] \otimes 1,$$

where the bitstring $x_i$ will be referred to as the control condition. Intuitively, the unitaries $\{C_x, P_i\}$ are the building blocks of $U$ in Eq. 5 and they “load” the classical data encoded in vectors $S^{(i)}$ and $v^{(i)}$ (see Section IV.I) into quantum mechanical address states $|x_i\rangle$.

Our multi-qubit gates combine the techniques introduced in Refs. [16, 30]. However, the EBGC we introduce highlights an property of these techniques which, to our knowledge, is both very advantageous and under-appreciated from the perspective of quantum algorithm design; namely, that they allow the realization of gates which introduce errors in a highly biased manner. In Section V.D and V.E we describe Electromagnetically Induced Transparency (EIT) as a physical mechanism for realizing single-qubit controlled gates that introduce substantial errors only when the control condition is satisfied. Moreover, the ratio between the errors introduced when the control condition is satisfied or not can be tuned using the single-atom drive strengths an detunings, by the virtue of the underlying EIT mechanism [30]. We demonstrate the advantage of such biased errors for achieving low EBGC implementations of block-encoding unitaries in Section V.F.

In Section V.F we turn our attention the advantages of using the blockade mechanism (as in Ref. [16]) for implementing $C_xP_i$ and $V_{\text{OHE}}$ in comparison to the more
conventional strategy of utilizing tunable two-qubit interactions. We refer to the latter as the two-qubit control strategy and assume that interactions between any two qubits can be turned on and off, to emphasize the advantage of using blockade gates. Lastly, in Section V G, we discuss the conditions necessary to parallelize of multiple \( C_x, P_i \) gates. We show that under the assumption that each qubit can be driven with a constant number of laser pulses, we can implement at most \( O(n_{\text{site}}) \) \( C_x, P_i \) gates at a time.

The results of this section sets the stage for a concrete blueprint of an efficient and scalable implementation of the QSP-based optimal Hamiltonian simulation of Refs. [31, 32], including the geometric arrangement Rydberg atoms and pulse sequences.

### A. Error-bounded gate counts (EBGCs) and the subadditivity of errors

In the following, we will utilize the number of gates involved in a given protocol to quantify the way that the errors grow as a function of scaling variables \( n_{\text{site}} \) and \( N \). Conventionally, the gate counts are equated to the size of a quantum circuit. The relationship between the circuit size and the errors is then established by via the subadditivity property of errors [38], which gives an upper bound for the spread of the errors introduced with each additional gate. However, the subadditivity bound may be extremely loose for a given protocol as it completely disregards both the structure of input states as well as the structure of the errors specific to an experimental implementation, which may be biased to introduce different errors for different input states.

On the bright side, given the error model for a specific experimental platform such as one consisting of Rydberg atoms, we can design control protocols which take advantage of the structure of the error model to reduce the required number of error-bounded gates. Our gate counting method, which we will call “error-bounded gate count” (EBGC) take as its unit the largest error that can be introduced by a single CNOT gate implemented on the Rydberg platform. Hence, when the information regarding the input state of the CNOT gate is taken into account, the resulting EBGC of the CNOT gate is smaller than or equal to 1. In this work, we demonstrate that we can take advantage of the structure of the input state and the errors not only at the level of a single CNOT gate but also at the level of many gates that implement block-encoding unitaries. Most importantly, we find that the asymptotic scaling of the EBGC for LCU-based block-encoding can be made independent of \( n_{\text{site}} \) and \( N \), in stark contrast to the scaling of the circuit size. Intuitively, this result follows from the fact that the EIT-based Rydberg atom implementations of single-qubit controlled unitary operations introduce errors that are heavily biased with respect to the input state of the control register.

The gate counting method discussed below considers only the fundamental sources of error, given by nonadiabatic contributions and radiative decay processes, and assume that the error rates of each source is the same. Our method hence assumes that the errors due to laser phase and amplitude fluctuations, leakage to unwanted Rydberg states due to blackbody radiation, as well as those due to the finite temperature atomic motion and the associated Doppler shift can all be eliminated [39, 40]. We emphasize that although our error-bounded
gate count is specific to the Rydberg atom platform, the strategy to design control protocols that take advantage of the biases in the relevant error model can be applied to any experimental platform.

1. Subadditivity of errors

To put the discussion on firm footing, we sketch the proof of subadditivity of errors, and underline its shortcomings. Consider a circuit $C$ that can be described by an ordered product of $T$ unitaries $C = \prod_i^T W_i$, and an imperfect implementation $\tilde{C}$ of $C$, where each $W_i$ is replaced by $\tilde{W}_i$. We assume $\tilde{W}_i$ to be unitary for simplicity. Now, given the same input state $|\phi_0\rangle$, we are interested in the difference between the outputs $|\phi_T\rangle$ and $|\tilde{\phi}_T\rangle$ of $C$ and $\tilde{C}$, respectively. Define

$$|\phi_1\rangle \equiv W_1 |\phi_0\rangle$$
$$\tilde{\phi}_1 = \tilde{W}_1 |\phi_0\rangle = \frac{1}{N} (|\phi_1\rangle + |E_1\rangle),$$

where we define the error vector $|E_1\rangle$ and the normalization $N$. The size of the error vector satisfies the following inequality

$$\frac{||E_1||}{N} = \left|\left|\left(\tilde{W}_1 - W_1\right)|\phi_0\rangle\right|\right| \leq \left|\left|\left(\tilde{W}_1 - W_1\right)|\phi_0\rangle\right|\right|_{\text{sup}} = \epsilon_1,$$

where the error $\epsilon_1$ associated with $\tilde{W}_1$ is defined via the spectral norm, which is completely oblivious to the input vector $|\phi_0\rangle$. The worst case scenario is that all errors from each $W_i$ constructively interfere. Since $\{W_i\}$ are all unitary, the errors introduced by the $i^{\text{th}}$ step is not amplified for any later step, and we obtain the inequality

$$\left|\left|\left(C - \tilde{C}\right) |\phi_0\rangle\right|\right| \leq \sum_i \left|\left|\tilde{W}_i - W_1\right|\right|_{\text{sup}} = \sum_i \epsilon_i.$$

As a result, decomposing each $W_i$ using a universal gate-set with known error rates, we can relate the size of the circuit to the total error of the circuit. However, we emphasize again that in the above discussion the definition of errors $\epsilon_i$ in Eq. (25) is independent of the structure of the input state. To understand the shortcomings of this definition, notice that the omission of the particularities of $|\phi_0\rangle$ forgets about the fact that the ancillary registers are initiated in the $|0\rangle^\otimes n$ state and that we know how this initial state transforms at each step of our circuit. In principle, we could use our knowledge of the trajectory of ancilla qubits to design error robust protocols.

Here, we use the error-bounded gate count to take into account our knowledge of the error model and the input state at each step. We show that for most of our protocols, we obtain a better scaling of the number of gates than as indicated by Eq. (25). In the following, we introduce the rules for calculating the gate count for single-qubit rotations and controlled unitaries in the form $C_x U_1 \cdots U_k$. We support the assumptions that go into the gate count with the physical error mechanisms relevant to the Rydberg atom system in Sections $V B$, $V C$, $V D$, and $V E$.

2. Error-bounded gate count (EBGC)

We distinguish three factors which determine EBGC. These factors are (i) the rotation angle of single-qubit rotations (ii) the dimensionality of the local Hilbert space of each Rydberg atom, and (iii) the dependence of the errors introduced during controlled unitary operations on the state of the control register. We introduce EBGC in order to keep track of the errors introduced by all three of these factors. We note that the EBGC is normalized such that the Rydberg atom implementation of a CNOT gate requires at most 1 error-bounded gate. Although in this section, we provide the rules for calculating the EBGC in an ad-hoc fashion for simplicity, we support the associated error model in more detail in Section $V$ where we discuss specific experimental proposals for implementing the relevant gates on the Rydberg platform. The scaling of EBGC compared to the conventional gate count is a main focus of our paper.

As for the first factor, we note that our protocols often use a continuous family of gates, such as single-qubit rotations by an arbitrary angle. We introduce an error model in which the error rate increases monotonically with the rotation angle. For example, given the single qubit rotation

$$R_{\theta} |0\rangle \equiv \cos (\theta) |0\rangle + \sin (\theta) |1\rangle,$$

the error associated with implementation of $R_{\theta}$ on the Rydberg atom platform is proportional to $|\theta/\pi|$. We thus assign an EBGC of $|\theta/\pi|$ to $R_{\theta}$. Notice that this rule associates 1/3 error-bounded gates for each single qubit Pauli operator.

Second, the protocols discussed in the rest of the paper take advantage of the fact that each Rydberg atom has more than two-states. A local Hilbert space of more than two-dimensions entails that the experimentalist can choose laser pulses which only acts on a two-dimensional subspace of the local Hilbert space. As a result, the errors are introduced only when the Rydberg atom is in a state with a non-zero overlap with the subspace acted on by the laser pulse. Consider as an example a laser pulse sequence implementing the unitary $X^{(h)}$ that transfers an atom from the logical state $|1\rangle$ to the Rydberg state $|R\rangle$ via the intermediate state $|r\rangle$ [the level diagram associated with each atom is discussed in more detail in Section $V B$, $V D$]. Given the initial state $\sqrt{1 - \alpha^2} |0\rangle + \alpha |1\rangle$, the transfer has an EBGC of $2|\alpha^2|$ error-bounded gates.

We also use a generalization of this rule to count the number of gates associated with our multi-qubit One-Hot amplitude-encoding gate $V_{\text{OHE}}$ in Section $V C$ and its controlled counterpart in Section $V E$. Surprisingly, the
EBGC of $V_{\text{OE}}$ is independent of the number of qubits involved, and it is equivalent to that of a single CNOT gate.

Lastly, our gate count makes sure that the cost of controlled unitaries $C_{x_i}P_i$ are assessed in accordance with the physical error model, and sums up the errors in the target and control registers separately. While the errors in the control register occur while checking whether a control condition $x_i$ is satisfied, the errors in the target register are assumed to be introduced only when the state of the control register satisfies the control condition. In Section $V D$ we discuss the concrete experimental protocol which can be modelled by such a biased error model. More specifically, we show that even when the errors are not perfectly biased, it is in principle possible to increase the bias arbitrarily by changing only the amplitudes and detunings of the drive lasers. We also discuss a scenario where the information on the input state in the target register can be used to obtain a smaller EBGC.

As a concrete example, consider a single CNOT gate, where the control register is initially in $|\psi_c\rangle = \sqrt{1 - |\alpha|^2} |0_c\rangle + \alpha |1_c\rangle$. We assume that the errors introduced into the control register during the CNOT gate operation scales with

$$\langle \psi_c | \hat{n}_1^{(c)} | \psi_c \rangle \equiv \langle \psi_c | 1_c \rangle |1_c | \psi_c \rangle = |\langle 1_c | \psi_c \rangle|^2.$$  

(27)

On the other hand, if the input state of the target register is not known, the errors introduced to the target register is proportional to the probability that the control condition is satisfied (i.e., $\langle \psi_c | n_1^{(c)} | \psi_c \rangle$). Hence, given no information on the state of the target qubit, the protocol for implementing the CNOT gate on the Rydberg platform (see Section $V D$) has an EBGC of $|\alpha|^2$. The knowledge of the state of the target register can be used to further reduce the EBGC (see Section $V I$). In particular implementing the controlled unitary $C_X^{(R)}$ given the target input state $|\psi_t\rangle = \sqrt{1 - |\beta|^2} |0 + \beta |1\rangle$, results in an EBGC of $|\alpha|^2(2 + 2|\beta|^2)$. Notice that this gate count is not identical to that of the CNOT gate when $|\beta|^2 = 1$ because the transfer from $|1\rangle$ to $|R\rangle$ of the target qubit requires the use of the intermediate hyperfine state $|r\rangle$.

We can extend the above discussion of EBGCs for controlled unitaries of the form $C_{x_i}U_1 \cdots U_k \equiv C_{x_i}P_i$. The errors introduced in the target and control registers depend on the size $m$ of the control register (see Section $V D$ for the different physical mechanism relevant to the two cases). When $m = 1$, and there is no knowledge of the state of the target register, the errors introduced into the target register during a $CU_1 \cdots U_k$ gate is proportional to the $k$-times the probability that the control condition is satisfied. On the other hand, when $m > 1$, the errors introduced into the target register are assumed to be independent of the input state of the control register, drastically increasing the EBGC (see Section $V D 2$ for the physical mechanism).

Similarly to the case of the CNOT gate, when $m = 1$, the error introduced in the control register with the input state $|\psi_c\rangle = \sqrt{1 - |\alpha|^2} |0\rangle + \alpha |1\rangle$ results in an EBGC of $|\alpha|^2/3$. In contrast, for $m > 1$, the errors introduced in the control register is proportional to the expectation value of the Hamming distance between the control condition and the input state in the control register. Given the control condition $|x_0\rangle = \Sigma \{0\}^m$ encoded in the control register of size $m$, the Hamming distance to any input state $|\psi_c\rangle$ can be calculated as

$$\text{Ham}(|\psi_c\rangle, |x_0\rangle) \equiv \langle \psi_c | \Sigma \{ \sum_{i=1}^m \hat{n}^{(c,i)}_1 \} \Sigma_0 |\psi_c\rangle,$$  

(28)

where $n^{(c,i)}_1$ is similar to the operator defined in Eq. (27) but only for the $i$th qubit in the control register. The resulting EBGC for the $C_{x_0}U_1 \cdots U_k$ gate with $m > 1$ is obtained by summing up the errors in the control and target registers. In the absence of any information of the state of the target register, the resulting EBGCs are $|\alpha|^2(2 + k)/3$ for $m > 1$, and $|\alpha|^2(2 + k)/3$ for $m = 1$. We emphasize that the difference between $m = 1$ and $m > 1$ cases can be drastically different when $\alpha \ll 1$. In particular for the implementation of the LCU-based block-encoding unitaries, we may have the situation where $\alpha = O(1/N)$, entailing a different scaling behaviour of the number of error-bounded gates compared to that resulting from a conventional gate count. As before, the EBGCs are subject to modification when the state of the target register is known. The gate counts are summarized in Table $7$ for a given input state $|\psi_c\rangle$ of the control register and the control condition $|x_0\rangle$. We evaluate the depth to implement each gate using unit time bins of size $t_{\text{step}}$, given by the time it takes to achieve a complete transfer of the $|0\rangle$ state to $|1\rangle$ state.

Our gate count does not only to assess an experimental scenario accurately, but also allows for designing algorithms with lower gate counts by taking full advantage of the structure of the errors relevant in an experimental scenario. In the following sections, we show that the structure of the errors on the Rydberg atom platform can be used to construct quantum algorithms whose EBGC scales differently than the circuit size. For instance, in Section $L X$ we demonstrate that utilizing our knowledge of the error model and the ancillary input states allows for an implementation of the QSP-based simulation algorithm of Ref. [32] with optimal gate complexity that grows with the simulated space-time volume, when the simulated Hamiltonian is $k$-local.

**B. Rydberg Interactions, Level Diagrams and Single Qubit Rotations**

1. Dipolar interactions:

Although all the gates that we will be discussing rely on the same Rydberg blockade mechanism as discussed in Ref. [27] [28] [41], we require both short- and long-range dipolar interactions in order to implement the full variety of multi-qubit gates that we utilize in this work.
The two main factors which effect the range of dipolar interaction between Rydberg atoms are (i) whether the dipolar interactions are of long-ranged resonant dipole-dipole type or of short-ranged Van der Waals type and (ii) the dipole moments associated with different Rydberg states [15, 40]. While the long-ranged dipolar interactions between the Rydberg states are useful for the One-Hot amplitude encoding gate we discuss in Section \[V\ C\] the possibility of controlling the range of short-ranged interactions will play an important role in implementing a parallelized version of our scheme in Section \[\ V\ G\]. Fortunately, the required characteristics can be in principle realized with the current experimental setups [15, 40, 42].

2. Level Diagrams and Single Qubit Rotations:

The four level diagrams that are relevant to our implementation are shown in Fig. 2. The diagrams consist of three types of states. Although these diagrams greatly simplify the experimental reality, the three types of states provide sufficient correspondence between our work and the experimental setup. First, we have long-lived hyperfine states \(|0\), \(|1\), and \(|r\)\), which make up the two logical states and an auxiliary state for each Rydberg atom. Second, we have an intermediate state \(|P\) which is useful to implement rotations within the hyperfine manifold, but which have a much shorter lifetime than the hyperfine states due to a larger radiative decay rate. The intermediate state is also crucial for the realization of the EIT scheme that we will discuss in the Section \[\ V\ D\ 1\]. Lastly, the high-energy Rydberg states \(|R\) which not only have a shorter lifetime than the hyperfine states, due to radiative decay, but also evolve under an interacting Hamiltonian, which can be written as

\[
H_R = \sum_{i,j} J_{ij} |R_i R_j \rangle \langle R_i R_j|,
\]

where \(|R_i R_j \rangle \equiv |R_i \rangle \otimes |R_j \rangle\) is the two-particle state where the \(i\)th and \(j\)th atoms located at positions \(r_i\) and \(r_j\) are in the Rydberg state. Although in reality the interaction strength has the form \(J_{ij} \propto \frac{1}{|r_i - r_j|^3}\) with \(\nu \in \mathbb{N}\), it is reasonable to model its spatial dependence as a step function which takes the value \(J\) when \(|r_i - r_j| < R_0\) and vanishes otherwise. We refer to the distance \(R_0\) as the “blockade radius”. The interaction strength \(J\) is finite. As a consequence, even when the radiative decay rate is not taken into account, the two-qubit blockade gate cannot be implemented perfectly. The errors due to the imperfect blockade will be referred to as non-adiabatic errors which grow as \(O(J^2/\Omega^2)\), where \(\Omega\) is the characteristic Rabi frequency of the transition between the low energy states to the Rydberg state. We remind the reader that in the following, we assume that these non-adiabatic errors are as large as the errors introduced by the radiative decay rate.

For the implementation of single-qubits, we choose to use \(|P\) as the intermediate state (see Fig. 2 a). Specifically, we can drive transitions between the logical states \(|0\) and \(|1\) using a Raman scheme which virtually excites the short-lived intermediate state \(|P\). The errors associated with the virtual occupation of \(|P\) motivate our rule for counting single-qubit gates in Section \[\ V\ A\ 2\]. Specifically, given \(|0\) as our initial state, the errors scale with the time that the short-lived state is virtually occupied during the rotation to the superposition state \(\sqrt{1 - |\alpha|^2} |0\rangle + \alpha |1\rangle\), resulting in the EBGC of \(\arcsin(\alpha)/\pi\) as in Table 4.

Next, we discuss a multi-qubit generalization of the single-qubit rotation, which we call the One-Hot amplitude encoding gate \(V_{OHE}\). The reason that \(V_{OHE}\) is a generalization of the single qubit gate is that the long-range Rydberg interactions constrain the many-body Hilbert space relevant for the evolution to a two-dimensional subspace.

C. One-Hot amplitude encoding gate

In the following, we introduce a new gate which can be thought of as a multi-qubit generalization of the single-qubit gate. Both the single-qubit gate and the One-Hot encoding gate are used to store classical information encoded in the duration \(t_0\) and the amplitude \(\Omega\) of the laser drive in quantum mechanical degrees of freedom. More specifically, the single-qubit rotation loads a single amplitude \(\alpha_i \equiv \arcsin(\Omega t)\) on a single qubit. Similarly, the One-Hot amplitude encoding gate \(V_{OHE}\) is a way of loading \(M\) amplitudes \(\{\alpha_i\}\) where \(\alpha_i \propto \Omega_i\) into \(M\) qubits in constant time. Because our scheme implements \(M\) amplitudes in the computational basis states with only one-Rydberg-excitation, we refer to it as the “One-Hot” amplitude-encoding gate. From a physical point of view \(V_{OHE}\) gate achieves to load all of the information encoded in the relative local intensity of the laser field into orthog-
onal computational basis states of a quantum register. Thus, our proposal for \( V_{OHE} \) provides an interesting case study for one of the most pressing challenges for applications of quantum information processing [12].

The sequence of unitaries that implement \( V_{OHE} \) builds on a similar gate discussed in the context of preparing the \( W \) state on the Rydberg platform [13]. Starting from the state \( |0\rangle^{\otimes M} \), we coherently drive the ancilla with \( M \) amplitudes \( \{\Omega_0\alpha_i\} \) where \( \Omega_0/J^2 \ll 1 \). Starting from the \( |0\rangle^{\otimes M} \) state, and assuming that each Rydberg level causes an energy shift of \( J \) on the Rydberg states of all other qubits, the dynamics is constrained to a two-dimensional Hilbert space spanned by

\[
|0\rangle^{\otimes M} \quad \text{and} \quad \sum_i \alpha_i |\text{OHE}, i\rangle,
\]

where we define the One-Hot encoding basis states \( |\text{OHE}, i\rangle \equiv |0\cdots 0_{R_1} 0\cdots 0\rangle \), each of which has only one Rydberg excitation. Projecting the drive Hamiltonian \( H_d = \sum_{i=1}^{M} \Omega_0 \alpha_i \sigma^+ + \text{h.c.} \) onto this subspace yields the effective Hamiltonian

\[
\hat{H} \equiv P_t H_d P_t = \Omega_0 \left( \sum_{i=1}^{M} \alpha_i |\text{OHE}, i\rangle \langle 0^{\otimes M}| + \text{h.c.} \right),
\]

(30)

which is analogous to a Pauli operator in the constrained Hilbert space (notice \( \hat{H}^2 = 1 \)). A schematic for the implementing \( V_{OHE} \) is given in Fig. [1].

Starting from the state \( |0\rangle^{\otimes M} \) and evolving the system under \( \hat{H} \) for time \( t^* = \frac{\pi}{\Omega_0} \), results in

\[
U_{0r} |0\rangle^{\otimes M} = e^{-i t^* \hat{H}} |0\rangle^{\otimes M} = \sum_{i=1}^{M} \alpha_i |\text{OHE}, i\rangle \equiv |\text{OHE}\rangle.
\]

(31)

While the time to implement \( U_{0r} \) scales as \( O(1/\sqrt{M}) \), the runtime of One-Hot encoding gate is increased drastically (i.e., \( \sqrt{M} \)-fold) due to the requirement that our final state needs to be within the long-lived logical subspace of each atom. In other words, we are required to transfer each ancilla atom excited to their Rydberg state to the long-lived hyperfine \( |1\rangle \) state using the following evolution operator

\[
U_{1r} = \exp \left( it_1^* \sum_{i=1}^{M} (\Omega_1 |r_i\rangle \langle 1| + \text{h.c.}) \right),
\]

(32)

where \( t_1^* = \left| \frac{\pi}{\Omega_1} \right| \). Assuming that the Rabi frequencies of local drives are the same, \( \Omega_0/\Omega_1 = O(\sqrt{M}) \), as the second part of the evolution does not take advantage of the collective enhancement of the effective Rabi frequency in the presence of blockade interactions. Given this bottleneck, we chose the single-qubit drive strengths in the implementation of \( U_{0r} \) as \( \Omega_0 = O(1) \) such that the runtime of the \( V_{OHE} \) gate is \( 2t_1^* = 2t_{\text{step}} \). Thus, \( V_{OHE} \) has an implementation depth of 2. We emphasize that this result holds only in the limit of infinite blockade radius. We discuss the case of finite maximum blockade radius in Section [11].

To arrive at the error bounded gate counts, we consider two sources of errors: (i) those that result from the radiative decay rate of the atoms in their Rydberg states and (ii) the non-adiabatic errors that result from the imperfect blockade interactions. Because we have at most one atom in the Rydberg state during the implementation of \( V_{OHE} \), the errors due to the radiative decay mechanism is the same as those associated with a single-qubit gate where the initial \( |0\rangle \) state is completely transferred to the \( |1\rangle \) state. On the other hand, the non-adiabatic errors resulting from the finite value of the strength \( J \) of dipolar interactions grow as \( O \left( (\Omega_0/J)^2 \right) = O(1) \), since the bottleneck discussed in the previous paragraph entails that
we set $O(\Omega_0) = O(1)$. Including the errors introduced by the radiative decay of the Rydberg states during $U_{1r}$, the number of gates involved in implementing $V_{\text{OHE}}$ is $3/3=1$. It is important to emphasize that the above error cost of $V_{\text{OHE}}$ is calculated assuming that the coupling between the Rydberg and hyperfine manifolds is induced by a single photon transition, as the introduction of intermediate states which do not experience an energy shift due to dipolar interactions result in radiative errors that scale as $O(M)$.

Before we continue our discussion of the Rydberg atom gates with the implementation of controlled unitary gates, we note that the controlled version of $V_{\text{OHE}}$ can be easily implemented by introducing a single additional control ancilla which can block the excitation of the rest of the qubits to the Rydberg state. The depth of the implementation of controlled $V_{\text{OHE}}$ is therefore at most 4. However, without the use of EIT-based blockade gate that we introduce in the next subsection, the errors introduced are independent of the state of the control register, which is undesirable if we want to reduce EBGC. We give the protocol for the biased error implementation of $CV_{\text{OHE}}$ in Subsection V.A.

**D. Multi-qubit controlled unitary gates $C_{x_0}U_1 \cdots U_k$**

While the unitary $V_{\text{OHE}}$ offers a way of loading the classical data stored in $\vec{\alpha}$ into quantum degrees of freedom, the controlled unitary gates load the classical data stored in $S$ and $r$ (see Section II.B) into the quantum processor. In particular, implementation of each single-qubit rotation $U_j \equiv e^{i\theta_j \hat{n}_j \cdot \vec{\sigma}}$ on the target atoms load the information regarding the position of that single qubit as well as the axis $\hat{n}_j \in \{\hat{x}, \hat{y}, \hat{z}\}$, and the angle $\theta_j$ associated with the rotation. By conditioning products of single-qubit rotations on the ancillary states $|x_0\rangle$, we can make sure the relevant data is associated with orthogonal address states.

We claimed in Section V.A that the error model which reflects the physical implementation of a controlled unitary operation is completely biased in the sense that it introduces errors only if the control condition is satisfied. Here, we describe the concrete protocol for which such an error model is valid. In particular, we discuss the multi-target controlled unitary proposed in Ref. [30], which utilizes an interference phenomenon called Electro-dynamically Induced Transparency (EIT) to ensure that the evolution of the target atoms can be made near trivial and error-free when the control condition is not satisfied. This protocol thus motivates the way we count the gates for each single-control conditional unitary in Section V.A.2. The EIT-based Rydberg-blockade gate is valid for unitaries conditioned on a single control atom. Although this may at first seem like a disadvantage, in Section VI, we will demonstrate that it is possible to implement multi-target-based block-encoding unitaries using only the single-qubit-controlled version of $V_{\text{OHE}}$ and single-qubit-controlled unitaries in Table II.

We also consider a protocol for implementing multiple control multi-qubit unitaries. However, as reflected in our gate counting method, this protocol has a crucial shortcoming, that the controlled gate suffers from non-adiabatic and radiative errors due to imperfect blockade interactions independent of whether they control condition is satisfied. However, the multiple control unitaries are still useful for implementing QSP-based algorithms, most notably through the implementation of the unitary $2(|0\rangle\langle 0|)^{\otimes n_a} - 1$.

1. **EIT-based single-control multi-target unitary on the Rydberg platform**

The EIT-based controlled unitary operations utilize interference to ensure that if the control condition is not satisfied, the evolution of the target atoms initiated in the logical subspace stays in a non-radiative "dark" subspace, thereby drastically reducing the errors due to radiative decay from both the intermediate state $|P\rangle$ and the Rydberg state $|R\rangle$. Moreover, the Rydberg interactions become relevant to the evolution only when the control condition is satisfied such that the non-adiabatic correlations are only relevant to this case. As a result, the EIT-based blockade gates can in principle introduce errors in a completely biased way, and the errors due to imperfect blockade and radiative decay rate should be weighted by the probability that the control condition is satisfied. This is the core justification of our error model for single-qubit controlled unitaries discussed in Section

![Diagram](image_url)
We start the discussion of the EIT-based blockade-gates with the implementation of a CNOT gate \[30\]. The scheme uses the level scheme in Fig. 3 \((a)\) and \((b)\) for the control and target qubits, respectively. The target qubit is continuously driven by a control field \(\Omega_c\) during the three-step protocol. In the first step, the control atom is excited to a Rydberg state if it satisfies the control condition. Secondly, lasers inducing the two probe Rabi frequencies \(|\Omega_{p1}|e^{i\theta_1}\) and \(|\Omega_{p2}|e^{i\theta_2}\) are shone on for the target atom. The frequencies of the control and probe frequencies are such that the excitation to the Rydberg state is two-photon resonant. Denoting the detuning between the hyperfine and intermediate state as \(\Delta\), and defining the radiative decay rates of the states \(|R_j\rangle\) and \(|P\rangle\), we consider an experiment satisfying, \(\{\Delta, J\} \gg \{\Omega_1, \Omega_2, \Omega_c\} \gg \{\gamma_R, \gamma_P\}\). If the control atom is not excited to the Rydberg state during this step, then both logical states of the target atom evolve adiabatically in a dark (non-radiative) subspace spanned by

\[
\begin{align*}
|\psi_I\rangle &= \frac{1}{\sqrt{1-x^2}} (|\psi_I\rangle + x |R\rangle) \\
|\tilde{\psi}_I\rangle &= \frac{1}{\sqrt{2}} \left( |0\rangle - e^{i(\theta_1-\theta_2)} |1\rangle \right),
\end{align*}
\]

and eventually return back to the initial state \[30\]. In the above expression we define \(x = |\sqrt{2}\Omega_p/\Omega_c|\) as a time-dependent dimensionless quantity, and \(|\psi_I\rangle = \frac{1}{\sqrt{2}} (|0\rangle + e^{i(\theta_1-\theta_2)} |1\rangle)\) as a state in the logical subspace and is determined by the relative phase of the two probe lasers.

Most importantly, because the dark-state has no contribution from the short-lived intermediate state \(|P\rangle\). Hence, the errors come solely due to decay of the small occupation in the Rydberg state \(|R_k\rangle\), which scales as \(x^2\). Given no limit to the achievable Rabi frequencies \(\Omega_p\) and \(\Omega_c\), \(x\) can be reduced arbitrarily without changing the runtime of the CNOT gate or violating conditions for adiabaticity. In particular, to reduce the errors by a factor of \(w\), we need \(\Omega_c \rightarrow \Omega_c/w\), \(\Delta \rightarrow w\Delta\), and \(\Omega_p \rightarrow \sqrt{w}\Omega_p\). As a result, we can neglect the errors introduced during the single-qubit controlled unitaries when the control condition is not satisfied.

If, on the other hand, the control condition is satisfied and the control atom is excited to its Rydberg state, then the EIT condition that ensure an evolution only within the dark-state manifold is no longer satisfied, and the transitions between the two logical states \(|0_l\rangle\) and \(|1_l\rangle\) are mediated by the virtual excitation of the short-lived state \(|P\rangle\), introducing errors due to \(\gamma_P\). In the last step, the control atom is transferred back to its logical subspace.

Whether the Pauli gate applied on the target qubit is \(\sigma_x^{(t)}\) or \(\sigma_y^{(t)}\) is determined by the phase difference between the control pulses \(\Omega_{p1}\) and \(\Omega_{p2}\). The multi-target generalization of the EIT-based controlled unitary is obtained by simply increasing the number of target qubits within the blockade radius of the control qubit. As a result, given the control condition \(|x_0\rangle\) and the input state of the control register \(|\psi_c\rangle\), the single-control k-target unitaries \(CU_1 \cdot \cdots \cdot U_k\) can be implemented using \((2+k)/3\) \(|x_0\rangle\langle \psi_c|\) error-bounded gates and a depth of 3.

We will also use controlled unitaries where the evolution of the target qubit is not within the logical subspace. In particular it is desired to transfer one of the logical states of the target atom to the Rydberg state conditionally on the state of the control atom. To this end, we need to use an additional hyperfine state \(|r_i\rangle\) (see Fig. 5\((c)\)), and apply the EIT-based blockade gate where the probe lasers on the target atom induce transitions between \(|1_l\rangle\) and \(|r_i\rangle\). Then, the population in \(|r_i\rangle\) can be transferred to the Rydberg state \(|R_t\rangle\) using an additional \(\pi\) pulse. As a result, the depth of the implementation is 4. On the other hand, the EBGC depends on the input state of both the control and the target registers. Given the input state \(|\psi_c\rangle = \sqrt{1-|\alpha|^2} |0_c\rangle + \alpha |1_c\rangle\) and
\[ |\psi_i\rangle = \sqrt{1 - |\beta|^2} |0_i\rangle + \beta |1_i\rangle, \]
the EBGC of the controlled unitary is \[2|\alpha|^2(1 + |\beta|^2)\] for a single target qubit. The calculation of the EBGC for larger number of target atoms is straightforward.

Unfortunately a multi-control generalization of the EIT-based blockade gate is not possible as the target goes through a non-trivial evolution whenever the control condition is not satisfied. In the next section we discuss a different multi-qubit Rydberg blockade gate, which does not use EIT and allows for both multiple control and target qubits. However, the multiple control unitary does not enjoy the completely biased error mechanisms of its EIT counterpart.

2. Multi-control multi-target unitaries on the Rydberg platform

The original conception of the blockade gate did not utilize the EIT mechanism discussed above \[27,28\]. As a result, the errors associated with this conventional blockade gate is not biased unlike the blockade gate discussed in the previous subsection. However, the conventional blockade gate is useful because it allows for a straightforward implementation of multi-control multi-target gates. Since such gates are used in our implementation of block-encoding unitaries, we briefly review the protocol and compare it to the EIT-based blockade gate.

As discussed above the EIT-based blockade gate uses an interference phenomena to result in a trivial evolution of the target qubits whenever the control qubit is in the logical subspace. On the other hand, the conventional blockade gate freezes the dynamics of the target qubits whenever the control qubit is in the Rydberg state \[|\psi_c\rangle\] due to an energy shift of \[J\] resulting from dipolar interactions. Hence, while the trivial evolution in the EIT-based blockade gate is error-free due to deconstructive interference, the trivial evolution in the conventional blockade gate acquire errors due to the radiative decay rate of the intermediate state \[|P\rangle\] and the non-adiabatic processes virtually exciting the target qubit to the Rydberg state. As a result, the errors associated with the conventional blockade gate is unbiased (i.e., independent of the state of the control register), invalidating the gate counting scheme we discussed in Section \[V.A\].

To clarify this conclusion, let us consider the implementation of the CNOT gate using the conventional blockade gate. When the control qubit is in the \[|1_c\rangle\] state, then the target qubit goes through a NOT operation which introduces errors due to the radiative decay rate of the intermediate high-energy state (see Section \[V.B.2\]). On the other hand, when the control qubit is in the \[|0_c\rangle\] state, the evolution of the target qubit is frozen, up to the non-adiabatic errors due to the imperfect blockade. Assuming the rate of non-adiabatic errors are comparable to the errors due to radiative decay, the implementation of CNOT introduces unbiased errors into the target register in accordance with Table \[7\].

On the bright side, the conventional blockade gate allows for a straightforward implementation of multi-control multi-target gates. In particular, the controlled unitary is realized by first transferring all atoms in the control register to a state with at least one Rydberg excitation whenever they do not satisfy the control condition \[|x_0\rangle \equiv \bigotimes_{j=1}^{m} |(x_0)_{j}\rangle\] (where \[m\] is the number of qubits in the control register). The transfer can be easily implemented by simply transferring each control qubit to the Rydberg state if they do not occupy the logical state given by \[|(x_0)_{j}\rangle\]. As a result, the radiative decay rate in the control register grows as the expectation value of the Hamming distance between \[|x_0\rangle\] and the input control state \[\psi_c\], in accordance with our gate count in Section \[V.A\] (see Table \[7\]).

E. Implementation of the CVoHE gate

As we will discuss in Section \[V.A.3\] the implementation of the state-preparation unitary \[V\] in Eq. \[\text{I}\] for \(k\)-local signal operators require an EBGC of only \(O(k)\) if they can be implemented using single-qubit controlled \(V_{\text{OHE}}\) gates that employ the EIT-based blockade gate (see Section \[V.D\]). However, unlike the tensor products of Pauli operators, the \(V_{\text{OHE}}\) gate utilizes interactions between the Rydberg states of many atoms. Thus, we need to introduce a new mechanism to implement \(CV_{\text{OHE}}\). Overall, the strategy is to use the One-Hot encoding state \[|\text{ohe}\rangle\] (see Eq. \[31\]) to realize a multi-qubit version of the \(CX^{(k)}\) operation where the relevant level diagram consists of only 4 non-degenerate states.

Besides \(|\text{ohe}\rangle\), the other states in our scheme are (i) the initial state \(|0\rangle^n\) where \(n\) is the number of qubits, (ii) \(|\text{ohe}^{(r)}\rangle\) which has at most one excitation in the state \(|r\rangle\) and (iii) \(|\text{ohe}^{(p)}\rangle\) with at most one excitation in the Rydberg state \(|R_p\rangle\). The Rydberg state \(|R_p\rangle\) has two important properties. First, it is only accessible from \(|R\rangle\) via a microwave transition (see Fig. \[3\] \[44,45\]). Second, the angular momentum quantum numbers of \(|R\rangle\) and \(|R_p\rangle\) are different, such that the two states experience different energy shifts due to dipolar interactions. Thus, we can assume that it is possible to have an energy shift on \(|R_p\rangle\) while the energy of \(|R\rangle\) stays constant. We emphasize that it is possible to describe the evolution of \(n\) atoms with only 4 states as a result of the blockade interactions between the Rydberg states. In particular, starting from the state \(|\text{ohe}\rangle\) and transferring each qubit to a non-interacting state results again in a One-Hot encoding state with only the characteristics of the One-Hot entries changed.

Using the Rydberg state \(|R_p\rangle\), we can implement a single-qubit controlled version of \(V_{\text{OHE}}\). In particular, we obtain an EIT scheme where the two states in the logical subspace are \(|0\rangle^n\) and \(|\text{ohe}^{(r)}\rangle\), and the intermediate state of the EIT scheme is the One-Hot encoding Rydberg
state $|\text{OHE}\rangle$. Finally, the state which controls whether the EIT condition is satisfied is $|\text{OHE}^{(R_p)}\rangle$.

The first step of $CV_{\text{OHE}}$ is to implement a transition between $|0\rangle^\otimes n_a$ and $|\text{OHE}^{(r)}\rangle$ controlled by the energy shift of $|\text{OHE}^{(R_p)}\rangle$. In the second step, $|\text{OHE}^{(r)}\rangle$ is transferred to the $|\text{OHE}\rangle$ state by a tensor product of single-qubit rotations. The depth of the implementation is 4 and the EBGC given the state of the control register $|\psi_c\rangle = \sqrt{1 - |\beta|^2} |0\rangle + |\beta\rangle |1\rangle$ is $5|\beta|^2$. The prefactor 5 is a result of taking into account both the radiative and non-adiabatic errors into account. We emphasize that the One-Hot encoding keeps EBGC small during the transfer between $|\text{OHE}^x\rangle$ and $|\text{OHE}\rangle$.

F. Advantages of the blockade based gates for implementing OHE and $C_{x_0}U_{1} \cdots U_{k}$ gates

In this subsection, we clarify the advantage of implementing the multi-qubit gates $V_{\text{OHE}}$ and $C_{x_0}U_1 \cdots U_k$ using the blockade mechanism native to the Rydberg atom platform. In particular, we answer the question of whether the implementation of these gates via the blockade mechanism that utilizes always-on two-atom interactions is advantageous compared to an implementation which controls arbitrary two-atom interactions in a time-dependent way. In the following, we call the latter implementation method Two-Qubit Control (TQC).

In our discussion of the runtime and the errors associated with the TQC and blockade strategies, we do not consider the specific error model relevant for either implementation, and assume that each gate operation introduces a constant error independent of the input states. Without loss of generality [46], we consider systems where the two-qubit interactions are diagonal in the computational basis, namely those generated by a mutually commuting subset of Pauli operators, $\sigma_i^x \sigma_j^z$, describing the interaction between the $i$th and $j$th qubits.

In the case of the TQC method, we allow the control of two-qubit interactions between any two qubits at a time. Moreover, we assume that multiple interaction terms can be turned on simultaneously. To compare the runtime of the two implementation strategies, we consider a unit of time given by the maximum strength of two-qubit interactions $J_{\text{max}}$. Regarding errors, we assume that the time-dependent control of effective interaction strength $J$ leads to non-adiabatic errors which scale as $O((J/J_{\text{max}})^2)$ for each pair of qubits involved. This adiabaticity assumption puts the errors introduced by the blockade and TQC strategies on equal footing by demanding that the error scaling of the gates implemented using the two strategies to be identical. Therefore, in the following, we will be only considering scaling of the time complexity and errors of the implementations with respect to the number of qubits involved. In making these assumption on adiabaticity, we do not consider shortcuts to adiabaticity and their applications to implementing two-qubit gates [17, 20]. Yet, to the best of our knowledge, although the shortcuts to adiabaticity can in principle reduce the error, the protocols which aim to counteract the non-adiabatic errors by designing custom engineered drive pulses typically require expensive numerical optimization routines that become unfeasible in the scaling limit relevant to our discussion. Moreover, it is important to remind the reader that our definition of the TQC method is motivated mainly by the purpose of accentuating the advantages of the blockade gates and that the current experimental platforms do not have the resources required to control two-qubit interactions between any two qubits in a time-dependent way.

We start our comparison by considering the implementation of the OHE gate in a system of $M$ qubits. As discussed in Section V C, the OHE gate can be implemented using the blockade mechanism in time $1/\Omega_o$ as in Section V C. The errors due to non-adiabatic process scale independently of $M$. We can compare this performance to that of the TQC implementation of the OHE gate. To the best of our knowledge, the lowest depth implementation of the OHE gate has circuit depth $O(\log(M))$ and requires $O(M)$ two-qubit gates [50]. Therefore, the total runtime of the implementation based on tunable two-qubit interactions scales as $O(\log(M))$ while the total error scales as $O(M)$. As a result, the optimal scaling both the runtime and the errors of the blockade implementation drastically improves that of a TQC implementation of the same gate.

Next, we consider the implementation of the $C_{x_0}U_1 \cdots U_k$ gates. The blockade-based implementation of $C_{x_0}U_1 \cdots U_k$, has a constant runtime, and the total error due to non-adiabatic and radiative processes scale as $O(m^2 + k)$, where $m$ is the number of qubits in the control register. In the case that of TQC, on the other hand, the depth of implementing $C_{x_0}U_1 \cdots U_k$ gate is the same as that of an $m$-bit Toffoli gate. Specifically, an $m$-bit Toffoli gate can be used to entangle the state of the control register to the $|1\rangle_{\text{anc}}$ or $|0\rangle_{\text{anc}}$ of the ancilla qubit depending on whether or not it satisfies the control condition. Then the unitary $U_1 \cdots U_k$ can be implemented conditionally on the single additional ancilla in constant time with the errors scaling as $O(k)$. To our knowledge, the lowest depth implementation of the $m$-bit Toffoli gate without ancillas require a circuit of depth $O(m)$ and the errors scale as $O(m^2)$ [51, 52]. Hence, the runtime of $C_{x_0}U_1 \cdots U_k$ gate implementation is $O(m)$, and the total error scales as $O(m^2 + k)$. Again, the blockade based implementation drastically improves the scaling of the runtime and errors for the TQC implementation of the $C_{x_0}U_1 \cdots U_k$ gates.

As a result, the blockade-based implementation of set of multi-qubit gates that we propose as building blocks for QSP-based algorithms provide drastic advantages in terms of runtime and errors with respect implementations based on TQC. We emphasize again that this advantage is present despite the fact that the TQC method requires...
a level of control over two-qubit interactions that has not been realized in current experimental platforms.

G. Parallelization of multiple \( C_{x_i} P_i \)

Having discussed the scaling of computational resources associated with a single \( C_{x_i} P_i \) gate, we now turn our attention to the conditions under which multiple conditional gates can be parallelized while still respecting the two physical constraints relevant for implementing blockade gates: (i) the spherical symmetry of the Rydberg blockade radius, and (ii) that each Rydberg atom can be driven by a fixed and small set of control lasers at each step of the implementation.

To better understand the effects of these physical constraints on efficient parallelization, consider \( M \) controlled unitaries \( \{C_{x_i} P_i\} \). In the context of quantum circuit models, whether any two controlled unitaries can be implemented in parallel is solely determined by the commutation relations between \( \{P_i\} \). That is, in principle, the implementation of controlled unitaries requires a circuit depth \( O(L) \), given that we can decompose the set \( \{P_i\} \) of Pauli operators as a union of \( L \) subsets each consisting of mutually commuting elements. However, such a criterion for parallelization is only valid if it is possible to implement a system evolution via the generator of each \( C_{x_i} P_i \) gate (i.e., \(-i \log(C_{x_i} P_i)\)), and yet, the generators that one can experimentally control are typically restricted to those that involve two or fewer Pauli operators at a time. In the case of a blockade-based implementation of \( C_{x_i} P_i \), we need to introduce two additional criteria for determining whether two gates \( C_{x_i} P_i \) and \( C_{x_j} P_j \) can be implemented in parallel. The first criterion captures the fact that in our implementation of the blockade gate (see Section 5), we realize only a single polarity in the control register at each step. The second criterion captures the assumption that each target qubit can be controlled by a fixed and small set of laser drives at a time, and that the Rydberg blockade volume is spherically symmetric.

To state these two criteria more formally, we define the following quantities. First, we define the multi-qubit Pauli operator \( \Sigma_i \), which prepares the address state \( |x_i\rangle \) on the control register according to

\[
\Sigma_i |0^n_a\rangle = |x_i\rangle.
\]  

Second, we define the support \( S_i \equiv \text{supp}(P_i) \) of the Pauli operator \( P_i \) as the set of qubits that \( P_i \) acts non-trivially. Then we define \( \text{ball}(S_i) \) as the set of qubits that are enclosed by the smallest sphere that also encloses \( S_i \). Two Paulis \( P_i \) and \( P_j \) commute with respect to the blockade radius if \( \text{ball}(S_i) \bigcap \text{ball}(S_j) = 0 \). This criterion captures the spherical symmetry of the Rydberg blockade radius and ensures that the Pauli operators are blocked only by the control atoms that are designated to control them. Then, given two Pauli operators \( P_i \) and \( P_j \), and the associated control conditions \( x_i \) and \( x_j \), the criteria for parallelizing blockade-based implementations of \( C_{x_i} P_i \) and \( C_{x_j} P_j \) are

\[
\text{supp}(\Sigma_i) \bigcap \text{supp}(\Sigma_j) = 0
\]

\[
\text{ball}(S_i) \bigcap \text{ball}(S_j) = 0.
\]

We depict these parallelization conditions in Fig. 5. Given that the \( \text{ball}(S_i) \bigcap \text{ball}(S_j) = 0 \), we can satisfy the condition \( \text{supp}(\Sigma_i) \bigcap \text{supp}(\Sigma_j) = 0 \) by assigning an ancilla for each \( \text{ball}(S_i) \).

These physical criteria for parallelization of multiple \( C_{x_i} P_i \) gates have two important consequences. First, the parallelization of a set of \( M \) \( \{C_{x_i} P_i\} \) gates requires \( M \) control qubits, each designated to one element in the set. Second, the maximum rate of parallelization, defined as the number of \( C_{x_i} P_i \) gates that can be applied simultaneously on the system, is \( O(n_{\text{site}}) \). Notice that the second consequence entails that we need at least \( O(n_{\text{site}}) \) ancilla qubits to achieve the maximum parallelization rate. In the next section, we use the above parallelization criteria to design a shallow depth circuit to implement LCU-based block-encoding unitaries.

VI. IMPLEMENTATION OF STATE PREPARATION AND LCU-BASED BLOCK-ENCODING

Having established the characteristics of the native Rydberg gates and their advantages compared to the gates implemented via time-dependent two-qubit interactions, we now present the protocols for the efficient LCU-based implementation of block-encoding unitaries in the Rydberg atom platform. The main goal of this section is to demonstrate that the biased error model associated with the Rydberg blockade gates discussed Section 5 can be utilized to implement LCU-based block-encoding unitaries whose EBGC scales exponentially slower than the size of the corresponding circuit.

In the following, we discuss two concrete circuit decompositions of LCU-based block-encoding unitaries on the Rydberg atom platform. The first circuit decomposition demonstrates that the existence of a single \( \bar{U} \) unitary (see Eq. 5) which allows for the block-encoding of any signal operator \( A \). In the following, we denote this unitary as \( \bar{U}_{\text{lib}} \). Surprisingly, \( \bar{U}_{\text{lib}} \) can be implemented in constant depth for any signal operator. As a result, the lower bound for the circuit depth implementing the block-encoding of a signal operator \( A \) acting on \( n_{\text{site}} \) system qubits is given by that of a circuit which prepares an appropriate state on \( O(n_{\text{site}}) \) qubits. Moreover, we show that the associated state preparation unitary \( V \) acting on \( 2n_{\text{site}} \) qubits can be implemented on the Rydberg atom platform with optimal circuit depth; assuming that at each time step, each atom can be manipulated using only a constant number of laser drives. Unfortunately, the circuit decomposition based on \( \bar{U}_{\text{lib}} \) relies heavily on the multi-qubit controlled unitary operations, which does
not allow us to take advantage of the error biases present for single-qubit controlled unitaries (see Section V D). As a result, for a signal operator composed of $M$ Pauli strings, $U_{\text{lib}}$-based block-encoding scheme achieves the same EBGC scaling of $O(M)$ as the earlier work in Ref. 33.

To improve the EBGC scaling, we propose a new circuit decomposition for block-encoding unitaries, which only uses single-qubit controlled unitaries, and show that the scaling of EBGC can be reduced to $O(k)$ for any $k$-local signal operator, composed of $O(n_{\text{site}}^k)$ Paulis. Hence, our construct realizes an exponential improvement in EBGC for any signal operator. However, we note that both the size and the depth of this circuit may scale as $O(n_{\text{site}}^k)$.

For both circuit decompositions, the depth and the number of ancillae required are determined by the parallelization rate of the relevant set of $\{C_x, P_1\}$ gates (see Section V G). Hence, the maximum parallelization rate is $O(n_{\text{site}})$ given that we use $\Omega(n_{\text{site}})$ ancillae.

In Sections V TA1 and V TA2 we discuss two state preparation protocols which rely on multi-qubit controlled unitaries. These protocols can be used to prepare any state. In Section V TA3 we present the state preparation unitary based on CV QHE gates (see Section V E), which efficiently prepares ancillary states that are customized for an efficient implementation of LCU-based block-encoding. In Sections V TB1 and V TB2 we discuss the implementation of $U$ for the two state preparation protocols in Sections V TA2 and V TA3 respectively.

A. Implementation of state preparation unitary $V$

In this section, we discuss protocols to implement the following state-preparation map

$$|0\rangle^{\otimes n_{a_1}} |0\rangle^{\otimes n_{a_2}} \rightarrow |\Psi\rangle \equiv \sum_{i=1}^{M} \alpha_i |y_i\rangle |x_i\rangle , \quad (35)$$

where $|x_i\rangle$ and $|y_i\rangle$ are computational basis states, and $n_{a_1}$ and $n_{a_2}$ are the number of ancillae in the first and second ancillary registers $a_1$ and $a_2$, respectively. The two registers have different purposes. The controlled unitary operations which make up the $U$ act on the system degrees of freedom conditionally on the state of $a_2$, while $a_1$ can be thought of as a workspace register, which primarily allows us to load the coefficients $\{\alpha_i\}$ into quantum mechanical degrees of freedom efficiently.

In Sections V TA1 and V TA2 we provide the serial and parallelized versions of a generic state-preparation protocol. Both protocols have an iterative structure, and the ancillae in $a_1$ are uncomputed at the end of each step (i.e., $|y_i\rangle = |0\rangle^{\otimes n_{a_2}}$) in order for them to be reused. These schemes allow us to prepare arbitrary states in the second ancillary register. However, the uncomputation step requires many multiple-control Pauli operators to be implemented, which is unwanted because of their high EBGC cost (see Section V D). Then, in Section V TA3 we introduce a customized state-preparation protocol the implementation of LCU-based block-encoding unitaries. For our customized state-preparation protocol, the ancillae in $a_1$ are not uncomputed. Although the scheme does not allow the preparation of an arbitrary state, the states prepared in this way are suitable for the implementation of the LCU-based block-encoding unitaries (see Section III). Most importantly, the customized state-preparation protocol uses only single-qubit controlled unitaries, allowing us to exponentially improve the EBGC.

1. Serial state-preparation

Consider the set of $M$ unitaries $\{\Sigma_{i,a_2}\}$ [see Eq. 33], where each $\Sigma_{i,a_2}$ prepares a computational basis state $|x_i\rangle$ in the second ancillary register $a_2$ (i.e., $\Sigma_{i,a_2}|0\rangle^{\otimes n_{a_2}} = |x_i\rangle$) that has a finite overlap with the target state $|\Psi\rangle$. Formally,

$$(I \otimes |x_i\rangle\langle x_i|) |\Psi\rangle \neq 0 \quad \forall i \in \mathbb{N}.$$
In the worst case scenario where no two elements of \( \{ \Sigma_i \} \) can be applied simultaneously, we can use a single ancilla in the first register \( a_1 \) (i.e., \( n_1 = 1 \)) to prepare \( |\Psi\rangle \) in \( O(M) \) steps. The protocol consists of the following sequence of transformations for each coefficient \( \alpha_i \):

1. Apply \( C_{g,\alpha_2} R_{\beta_2,\alpha_1} \): rotate \( a_1 \) to
   \[
   |\psi\rangle_{a_1} \equiv \sqrt{1 - |\alpha_{11}|^2} |0\rangle_{a_1} + \alpha_{11} |1\rangle_{a_1}, \]
   conditionally on the ancilla qubit in \( a_2 \) being in state \(|0\rangle_{\otimes n_2}\).

2. Apply \( C_{1,\alpha_1} \Sigma_{i,\alpha_2} \).

3. Apply \( C_{x_i,\alpha_2} X_{a_1} \): a multi-qubit controlled NOT gate on \( a_1 \) register conditioned on the \( n_{a_2} \) ancilla qubits being in state \( \Sigma_{i,0}\rangle_{\otimes n_2} \).

The circuit implementing the above state-preparation protocol is depicted in Fig. 7(a). From an intuitive point of view, each step of the above protocol prepares one of \( M \) components \( |x_i\rangle \) of the ancillary state \( |\Psi\rangle \), thus completing in depth \( 9M \) (see Fig. 7). The number of error-bounded gates required for this protocol depends on the state being prepared. For instance, given a target state \( |\Psi\rangle = \sum_i \alpha_i |0\rangle_{a_1} |x_i\rangle_{a_2} \) the EBGC \( g_{\Sigma_i}^l \) of the \( i \)th iteration of the above protocol is [see Eq. (28) for the definition of \( \text{Ham}(x, y) \)]

\[
1 < g_{s,i} = g_R + g_{\Sigma} + g_{\text{NOT}}
\]

\[
= \frac{1}{3} \left\{ \left[ \frac{\arcsin (\alpha_{i1})}{\pi} + 2 \text{Ham} \left( |0\rangle_{\otimes n_{a_2}} , |\tilde{\psi}^{(i-1)}\rangle \right) \right]
+ \left[ \frac{\arcsin (\alpha_{i1})}{\pi} \text{Ham} \left( |0\rangle_{\otimes n_{a_2}} , |x_i\rangle \right) \right]
+ \left[ 1 + 2 \text{Ham} \left( |x_i\rangle , |\tilde{\psi}^{(i)}\rangle \right) \right] \right\},
\]

where we defined \( g_R \), \( g_{\Sigma} \), and \( g_{\text{NOT}} \) as the EBGCs for implementing the first, second, and third steps of the above protocol, respectively. Furthermore, \( |\tilde{\psi}^{(i)}\rangle \equiv \sum_j g_{s,i} |0\rangle_{\otimes n_{a_2}} \otimes 1 |\tilde{\psi}^{(i-1)}\rangle \), where the state \( |\tilde{\psi}^{(i)}\rangle = \sum_j \alpha_j |0\rangle_{\otimes n_{a_1+n_{a_2}}} |x_j\rangle_{\otimes n_{a_2}} \) is that of the total ancillary register at the end of \( i \)th iteration, with boundary conditions \( |\tilde{\psi}^{(0)}\rangle = |0\rangle_{\otimes (n_{a_1}+n_{a_2})} \) and \( |\tilde{\psi}^{(M)}\rangle = |\Psi\rangle \). Hence, the total gate count depends on the path that the initial ancilla state takes. However, we can lower bound the total number of gates \( g_{s,\text{tot}} = \sum_i g_{s,i} \) as \( O(N) \) because \( g_{\text{NOT}} \) for uncomputing \( a_1 \) at the end of the \( i \)th step has an EBGC larger than or equal to \( 1 - 2/3|\alpha_{i1}|^2 \), and \( \sum_i |\alpha_i|^2 = 1 \).

2. Parallelized state-preparation

In the case that the components of \(|\Psi\rangle\) can be prepared simultaneously, we can utilize our discussion on parallelization \( C_{g,\alpha_1} \) (see Section V C) gates to improve the runtime of the serial protocol in the previous subsection by introducing \( O(n_{\text{state}}) \) ancillae in register \( a_1 \).

Consider the case that \( \{ \Sigma_i \} \) associated with preparing each component \( |x_i\rangle \) can be grouped in \( L \) layers, where the \( n_i \) operators \( \{ \Sigma_i \} \) in the \( l \)th layer mutually commute with respect to the Rydberg blockade (see Section V C). Each element in the \( l \)th layer has an associated coefficient \( \hat{\alpha}_l \), such that \( \sum_{l=1}^{M} |\hat{\alpha}_l|^2 = 1 \). Moreover, each layer has an overall coefficient \( \beta_l \) which satisfies \( \sum_{l=1}^{L} |\beta_l|^2 = 1 \) and can be decomposed the following product form

\[
\beta_l = \sin (\theta_l) \prod_{j=1}^{l-1} \cos (\theta_j), \tag{36}
\]

where \( \{ \theta_j \} \) will be defined below. For the \( l \)th step of the state-preparation, we pair each \( \Sigma_i^{(l)} \) with an ancilla qubit \( a_{1,i} \) in the same local neighborhood (see Section VI B). This pairing requires a number of ancillae \( n_{a_1} = O(\max x_i n_l) \).

In this setting, we can implement the state preparation unitary layer by layer. More specifically, we apply the following protocol for each layer \( l \):

1. Apply \( e^{-i\theta_0 \hat{H}} \) [see Eq. (31)] for duration \( t_0 = \theta_1/|\Omega_{a_1}| \) on the ancilla in register \( a_1 \) conditionally on the second register being in state \( |0\rangle_{\otimes n_{a_2}} \). We denote the partial rotation generated by \( \hat{H} \) as \( V_{\text{OHE}}^{(l)} (\theta_1) \).

2. Apply a parallelized \( C_{a_1,i} \hat{\Sigma}_i^{(l)} \) for each component \(|x_i^{(l)}\rangle\) in layer \( l \). Here, \( a_{1,i} \) is the \( i \)th atom of the first ancillary register.

3. Apply \( X \) operation on each \( a_{1,i} \) controlled by all the qubits in \( \text{supp} \left( \hat{\Sigma}_i^{(l)} \right) \) being in the logical state \(|1\rangle_{a_2,i} \).

The circuit implementing the above state-preparation protocol is depicted in Fig. 7(b). As in the serialised state-preparation protocol, the first step loads the amplitudes \( \{ \hat{\alpha}_l \} \) into the ancilla \( \{ a_{1,i} \} \) in the first ancillary register, with probability \( |\beta_l|^2 \). The second step simultaneously prepares the states \( \{ \Sigma_i^{(l)} |0\rangle_{\otimes n_{a_2}} \} \), associated with the \( l \)th layer, each conditioned on \( a_{1,i} \). In the last step, we uncompute the state of the ancillary register \( a_1 \). The depth of this parallelized state-preparation is \( 9L \) and is limited by the maximum parallelization rate [i.e., \( L = \Omega(M/n_{a_2}) \)]. The number of gates associated with the protocol can again be lower bounded as \( \Omega(M) \) due to the cost of uncomputing the ancillary register \( a_1 \).
3. State preparation for LCU-based block-encoding on the Rydberg atom platform

The two state-preparation protocols discussed above demonstrate that while it is possible to parallelize the a state-preparation unitary, it is not possible to obtain a substantial advantage compared to other state preparation protocols [53] with respect to the EBGC, because of the error cost associated with uncomputing the first ancillary register $a_1$. In this subsection, we demonstrate that it is possible to exponentially reduce the EBGC of state preparation at the expense of a modest increase in $n_{a_1}$ and relaxing the requirement to uncompute $a_1$. To this end, we construct a protocol that only consists of $V_{OHE}$ and its single-qubit controlled counterpart $CV_{OHE}$. The depth of the state-preparation protocol is $O(L)$ given $L$ layers of Pauli strings that commute with respect to Rydberg blockade. On the other hand, the EBGC of implementing a state preparation unitary is only $O(k)$, for signal operators that can be decomposed in terms of $k$-local Pauli operators. This result is at the core of the optimal EBGC we obtain for the implementation of QSP-based $k$-local Hamiltonian simulation algorithm in Section [7X].

As a first step, we describe a state preparation protocol which uses 2 ancilla registers $a_1$ and $a_2$, and prepares the following two-hot encoded state

$$|\Psi_{2HE}\rangle = \sum_{i=1}^L \beta_i^{(1)} |\text{ohe}, l\rangle \otimes \left( \sum_{i=1}^n \beta_i^{(2,l)} |\text{ohe}, i\rangle \right), \tag{37}$$

where $\beta_i^{(n,l)}$ is the coefficient of the state $|\text{ohe}, i\rangle$ of the $n^{th}$ ancillary register $a_n$, conditioned on the $(n-1)^{st}$ ancilla register being in the state $|\text{ohe}, l\rangle$. As we will discuss in Section [7X], this ancillary state allows us to simultaneously apply $n_l$ Pauli operators $\{P_i^{(l)}\}$ conditioned on the ancillary state $|\text{ohe}, l\rangle \otimes |\text{ohe}, i\rangle$, given that the Pauli operators $\{P_i^{(l)}\}$ commute with respect to the blockade radius (see Section [7X]). To prepare $|\Psi_{2HE}\rangle$, we use two ancilla registers $a_1$ and $a_2$ consisting of $n_{a_1} = L$ and $n_{a_2} = \max(n_l)$ qubits, respectively. The state preparation unitary can then be implemented by first applying a $V_{OHE,a_1}$ on the first ancilla register, followed by an application of $V_{OHE,a_2}^{(l)}$ on the second ancilla register conditional on the $l^{th}$ qubit in $a_1$ (we denote the...
this operation by $C_{1a1i}V^{(i)}_{\text{OHE}a}$. The state-preparation protocol requires $n_2 + L$ ancillary qubits and takes only
\[
\frac{1}{3} \left( 3 + 5 \sum_{l=1}^{L} |\beta_l^{(1)}|^2 \right) = 8/3 \text{ error-bounded gates and a depth of } 2 + 4L \text{ (see Section V E for EBGC calculation for single-qubit controlled } \text{VHE}). \] 
When the signal operator $A$ can be decomposed into $O(n_{\text{site}})$ local Pauli operators, the state preparation protocol takes $O(1)$ time, $O(1)$ EBGC, and $O(n_{\text{site}})$ ancillae. The circuit preparing $|\Psi_{\text{OHE}}\rangle$ is depicted in Fig. 3(c).

However, for signal operators that are $k$-local, then $L = n_{a1} = O(n_{\text{site}}^k)$, and the ancilla requirement of the state preparation protocol described above becomes unfavorable. To address this problem, it is desirable to design a protocol, which reduces the number of required ancillae, while not affecting the scaling of EBGC substantially. Such a state preparation protocol can be obtained by concatenating the circuit described above for 2 ancillary registers over $k$ ancillary registers [see Fig. 7(d)]. The concatenated protocol requires $O(kn_{\text{site}})$ ancillae and the number EBGC becomes
\[
\frac{1}{3} \left( 3 + \sum_{l=1}^{n_1} |\beta_{l1}^{(1)}|^2 \left( 5 + \sum_{l_2=1}^{n_2} |\beta_{l_2}^{(2)}|^2 \left( 5 + \cdots \right) \right) \right) = \frac{3 + 5(k - 1)}{3} = O(k). \quad (38)
\]

The prepared state is a product of $k$ One-Hot encoded states, each associated with a different ancillary register
\[
|\Psi_{\text{KHE}}\rangle = \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \cdots \sum_{l_k=1}^{n_k} \beta_{l_1}^{(1)} \beta_{l_2}^{(2)} \cdots \beta_{l_k}^{(k)} \mid x_{l_1}, l_2, \ldots, l_k \rangle \otimes \mid \text{OHE}, l_1 \rangle \rangle. \quad (39)
\]

We emphasize that the number of computational basis components of $|\Psi_k\rangle$ is $O(n_{k\text{site}}^k)$, and our protocol allows one to adjust the amplitude associated with each $k$-Hot computational basis state, for instance by using a regression tree decomposition of the sorted list of coefficients \{\alpha_i\} \cite{34}.

B. Implementation of $\tilde{U}_{\text{lib}}$

In the following, we first discuss the implementation of $\tilde{U}$ for the state-preparation protocols discussed in Section V TA1 and V TA2. Although the EBGCs of these state-preparation protocols makes them not favorable for the implementation of LUC-based block-encoding unitaries, the constant depth construction of $\tilde{U}$ (denoted $\tilde{U}_{\text{lib}}$) presented here provides the insight that from the point of view of circuit depth, the implementation of LUC-based block encoding is equivalent to that of the associated state preparation protocol.

Next, we discuss the implementation of $\tilde{U}_{k\text{HE}}$ for the ancillary state $|\Psi_{\text{KHE}}\rangle$. Our protocol in this case is straightforward. In particular, the unitary $\tilde{U}_{k\text{HE}}$ consists of controlled Pauli operations conditioned on the states of one qubit from each of $k$ ancilla registers. In order implement block-encoding with low EBGC, we construct the multi-qubit controlled Paulis using only their single-qubit counterparts, which allow us to utilize EIT (see Section V D 1) and result in an exponentially more error-robust implementation of the the block-encoding unitary. We emphasize again that the reduction in EBGC would not be possible if we used the multi-qubit controlled unitaries discussed in Sec. V D 2.

1. $U_{\text{lib}}$

The constant depth implementation of $\tilde{U}_{\text{lib}}$ requires that the address states (prepared by the unitary $V$) of each Pauli operator mirror the classical description of the signal operator $A$ stored in $(S_j^{(i)}) \in \{0,1\}^{n_{\text{site}}}$ and $(r_j^{(i)}) \in \{0,1\}^{n_{\text{site}} \times 2}$ defined in Section III. As a result, the implementation of $\tilde{U}_{\text{lib}}$ uses $n_2 = 2^{n_{\text{site}}}$ "library" ancillae. We remind the reader that here $i$ indexes the multi-qubit Paulis $\{P_i\}$, and $j$ and $k$ index the system sites in the support of $\{P_i\}$ and the type of single-qubit Pauli operator that acts on the $j$th site.

Intuitively, $\tilde{U}_{\text{lib}}$ simply creates a library of all Pauli operators acting on $n_{\text{site}}$ system qubits. We consider $2^{n_{\text{site}}}$ library ancilla which can be partitioned into $n_{\text{site}}$ pairs $\{a_{x,y,j}, a_{y,j}\}$, where the subscript $x$ and $y$ denotes that these ancillae control the application of $\sigma_x^j$ and $\sigma_y^j$ operators acting on the $j$th site of the system. Hence, each pair stores the entries of the classical data stored in $r_j^{(i)}$. In the following, we denote the system qubits as $q_i$, and denote the phase gate as
\[
S = \begin{pmatrix}
1 & 0 \\
0 & i
\end{pmatrix}
\]

$\tilde{U}_{\text{lib}}$ can be implemented in only three steps as follows

1. Apply $C_{a_{x,j}, S_{a_{y,j}}}$, for each pair of ancilla qubits $\{a_{x,i}, a_{y,i}\}$
2. Apply a $C_{a_{y,j}, X_{q_j}}$ gate for each pair $\{a_{y,i}, q_i\}$
3. Apply $C_{a_{x,j}, X_{q_j}}$ for each pair $\{a_{x,i}, q_i\}$.

The above protocol results in the following unitary
\[
\tilde{U}_{\text{lib}} = \sum_{i=1}^{2^{n_{\text{site}}}} |x_i\rangle \langle x_i| \otimes P_{x_i}, \quad (40)
\]

where $|x_i\rangle$ is a computation basis address state which encodes the classical description of $P_{x_i}$ stored in $S^{(i)}$ and $r^{(i)}$. We emphasize that $\tilde{U}_{\text{lib}}$ can be implemented in depth $3 \times 3 = 9$. Given an arbitrary ancillary state $|\Psi\rangle$ output by the state-preparation unitary $V$ in Sections V TA1 and V TA2, the EBGC for the implementation of $\tilde{U}_{\text{lib}}$ is
\[
\sum_j \langle \Psi | (2n_j^{(x,j)} + n_j^{(y,j)}) | \Psi \rangle = O \left( \max_i \text{Ham} \left( |x_i\rangle , |0\rangle^{2n_{\text{site}}} \right) \right),
\]
where \( \hat{a}^{(s,j)} \equiv \frac{1}{2} (1 + Z_{a,s,j}) \).

As mentioned earlier, the unitary \( \tilde{U}_{lh} \) allows us to bound the maximum depth of a circuit which implements the LCU-based block-encoding using \( \Omega(n_{\text{site}}) \) ancillae, using the depth of the state preparation circuit acting on \( O(n_{a_1} + n_{a_2}) \) ancilla qubits.

2. Implementation of \( \tilde{U}_{k\text{HKE}} \)

Next, we discuss the implementation of \( \tilde{U}_{k\text{HKE}} \), which, together with the protocol for preparing the state in Eq. (37), allows us to implement the LCU-based block-encoding unitary with an EBGC that is exponentially smaller than the associated circuit size. To ensure that the EBGC of realizing \( \tilde{U}_{k\text{HKE}} \) is small, we only use the EFT-based one-qubit controlled unitary operations discussed in Section V.D.

The unitary \( \tilde{U}_{k\text{HKE}} \) that can be used to block-encode a \( k \)-local Hamiltonian can be decomposed as a sum of \( k \)-qubit controlled unitaries. In the following, we first describe a case where we use only \( 2 \) ancillary registers \( a_1 \) and \( a_2 \), consisting of \( n_{a_1} = L \) and \( n_{a_2} = O(n_{\text{site}}) \) ancillae. Then we extend the scheme to \( k \) ancillary registers, compatible with the ancillary state \( |\Psi_{k\text{HKE}}\rangle \).

3. Controlled-PHASE and Controlled-W gates

We also comment on the gates that accompany the block-encoding unitary in the realization of QSP-based algorithms discussed in Section IV. In particular, we are required to implement the reflection operator \( 2 \langle 0 | 0 \rangle^{\otimes n_a} - 1 \), where \( n_a \) is the total number of ancillae in the first and the second ancilla layers. To this end, we introduce an additional "phase" ancilla \( a_{\text{ph}} \) which is the target qubit of a multi-qubit controlled phase gate. To calculate the EBGC of the multi-qubit controlled NOT gate, we need to calculate the Hamming distance between \( |\Psi_{k\text{HKE}}\rangle \) of \( k \) ancilla registers and the all zeros state

\[
\frac{1}{3} \left( 2 \text{Ham}(|\Psi_{k\text{HKE}}\rangle, |0\rangle^{\otimes n_a} + 2 \right) = \frac{1}{3} \left( 2 \langle \Psi_{k\text{HKE}} | \sum \sum n_{i,l} a_{i,l} | \Psi_{k\text{HKE}} \rangle + 2 \right)
\]
TABLE II. The EBGCs and the depth associated with the protocols introduced in Section VI. $M$ is the number of multi-qubit Pauli operators that make up the signal operator, and $L$ is the number of Pauli operators that commute with respect to Rydberg blockade. Notice that the error-bounded gate cost of $\bar{U}_{k\text{HE}}$ only depends on the maximum support size of the Pauli operators, and not on $M$ or the system size. In the case that the ancilla state One-Hot encoding, the error-bounded gate cost of CPHASE is $5/3$. The controlled versions of the LCU protocol and the CPHASE can be implemented with an additional $2/3$ error-bounded gates and 2 steps.

![Circuit Diagrams](image)

FIG. 8. The circuit diagrams for implementing the unitaries: (a) $\bar{U}_{2\text{HE}}$ and (b) $\bar{U}_{k\text{HE}}$. The ancillary registers are denoted as $a_i$. In a), the conditional Pauli operations with the curly brackets indicate a parallelized application. The same building block is repeated $n_{a1}$ times to complete the application of $\bar{U}_{2\text{HE}}$. In b), we depict the concatenation of unitaries which result in the application of $\bar{U}_{k\text{HE}}$. The building block is to be repeated $n_{a1}(k-1)$ times.

encoded states (see Section VI A 3) the expectation value of the Hamming distance is only $2k$, resulting in a total EBGC of $\frac{2}{3}(k+1)$.

On the other hand, we also need to implement a controlled version of the iterate $W$ in Eq. (13). To this end, we require the single-qubit controlled versions of both the LCU-based block-encoding unitary as well as that of the multi-qubit controlled-PHASE gate discussed in the previous paragraph. For the latter, we can simply block the evolution of the phase ancilla $a_{ph}$, conditionally on an additional ancilla that we call the “exit” ancilla in the following. The version of the controlled-PHASE gate conditioned on the state of the exit ancilla introduces an additional EBGC of $2/3$.

The exit ancilla can also be used to implement the controlled-$W$ operation efficiently when we use the One-Hot encoding state-preparation protocol in Section VI A 3. In this case, the LCU-based block-encoding unitary can be blocked by only blocking the first ancillary register $a_1$ using the EIT-based blockade mechanism discussed in Section V E. The EBGC for implementing controlled-LCU vanishes given that the exit ancilla does not satisfy the control condition; otherwise the EBGC is that of the LCU. Hence, in the following, we again assume that the EBGC of the LCU and its controlled counterpart differ only by $2/3$, since the conditional operations require the exit ancilla to be excited to the Rydberg manifold and back. The depth of the controlled versions of LCU and CPHASE protocols are both increased by 2.

We provide the EBGCs of the LCU-based block-encoding, controlled-PHASE gate, and their exit ancilla controlled versions in Table VII.

VII. SCALABLE IMPLEMENTATION OF LCU ON THE RYDBERG ATOM PLATFORM

So far we have considered the situation where the largest blockade radius attainable is infinite. In this section, we consider the more realistic situation where the maximum range $R_{b,\text{max}}$ of blockade interactions is finite. In a typical experiment the range of the resonant dipole interactions that result in Förster processes do not exceed $30\mu m$, while the separation of the Rydberg atoms...
FIG. 9. The problem of scalability and its resolution through FANIN and FANOUT protocols. a) Due to the finite length scale of the Rydberg blockade, the conditional unitary gates cannot be applied when the size of the target or control register is larger than the maximum blockade volume $V_{b,\text{max}}$. b) The communication between registers whose size exceeds $V_{b,\text{max}}$ can be facilitated by an additional network of ancillae, and c) the implementation of FANIN (↑) and FANOUT (↓) protocols.

trapped by holographic optical tweezers is around $2\mu m$ \[55\]. Hence, the scalability of the protocols introduced in the last two sections is restricted ultimately by $R_{b,\text{max}}$.

To engineer scalable protocols, we divide the system and ancilla qubits into a total of $n_{\text{sub}}$ modules whose sizes are determined by $R_{b,\text{max}}$.

Remarkably, the scalable protocols for the state preparation step of LCU-based block-encoding [i.e., $|\Psi_{\text{KHE}}\rangle$] in Eq. (37) and $C_{x_0}U_1 \cdots U_k$ gates distributed on $n_{\text{sub}}$ modules requires only a single qubit of information to be communicated between the modules. The information can be communicated either by what we call “connector” ancillae which serve as wires connecting different modules, or by physically transporting the ancillae appropriately using the optical tweezers. The processing of the incoming information as well as the calculation of the output message are performed by a gadget we refer to as the telecommunication port, which introduces only three ancilla qubits per subsystem.

Here, we describe explicit protocols to realize a modular and distributed implementation of the block-encoding unitary constructed out of multi-qubit gates $V_{\text{KHE}}$ and $C_{x_0}U_1 \cdots U_k$. The most important contribution of this section is a demonstration of the $n$-qubit Toffoli gate whose circuit depth matches the Lieb-Robinson bounds resulting from the locality constraint $[35, 36]$. As such, our work demonstrates the power of considering concrete implementations to achieve efficient decompositions of target unitaries. On the other hand, the circuit depth for implementing $V_{\text{KHE}}$ in a scalable fashion is optimal only in one dimensions [i.e., $O(n_{\text{sub}})$].

As discussed in the previous sections, the single-qubit controlled unitaries are especially suited to realize a protocols that introduce highly biased errors. While we do not explicitly calculate the EBGC for each protocol in this section, we note that (i) the FANOUT protocol described below allows one to control $n_{\text{sub}}$ subsystems using a single control qubit in state $|\psi_c\rangle$ with an EBGC of $|\langle 1_c|\psi_c\rangle|O(n_{\text{sub}})$, and (ii) that the scalable version of $V_{\text{KHE}}$ results in constant EBGC.

In the absence of additional ancillae, blockade interactions cannot be used to entangle registers larger than the blockade volume $V_{b,\text{max}}$. We depict the geometric constraints resulting from a finite $R_{b,\text{max}}$ in Fig. 9. Similarly, the finite blockade radius does not allow the implementation of the $V_{\text{OHE}}$ gate when the qubits in the relevant register occupy a volume larger than the blockade volume.

The solution to this problem requires the ability (i) to broadcast the information regarding a single subsystem to many others (1-to-many communication), and (ii) to bring the relevant information of many subsystems to one particular subsystem (many-to-1 communication). We satisfy these requirements by introducing the FANOUT and FANIN protocols build out of the multi-qubit gates $V_{\text{OHE}}$ and $C_{x_0}U_1 \cdots U_k$. For each subsystem $s_i$, we also introduce a telecommunication port (see Fig. 10) consisting of 3 ancilla qubits referred as: antenna ($A_i$), receiver ($R_i$), and processor ($Q_i$). For simplicity, we also assume that a set of connector ancillae $\{T_{ij}\}$ connecting $A_i$ to $R_j$. The role of $A_i$ and $R_j$ is to facilitate the communication of whether a control condition is satisfied or violated between $s_i$ and $s_{j \neq i}$. The processor ancilla $Q_i$, is multi-purpose. In the following, we use $Q_i$ as an indicator of whether $s_i$ satisfies a control condition (see Section VIIIB1.), as well as to load amplitude information into the subsystem (see Section VIIIB2.).

1. Many-to-one communication (FANIN)

In the following, we denote a controlled unitary $U$ acting on register $e$ conditioned on the state $|a\rangle_c |b\rangle_d$ as $C_{abcd} U_{e}$. Then, the FANIN protocol is simply implemented by $C_{x_0,s_i} \text{NOT}_{Q_i}$, a unitary which transfers the state $|0\rangle_{Q_i}$ to $|1\rangle_{Q_i}$ conditioned on $s_i$ being in the computational basis state $|x_0\rangle_{s_i}$. Thus, the FANIN gate de-
scribed by the map
\[ |\psi\rangle_{s_i} \otimes |0\rangle_{Q_i} \rightarrow \langle x_0 |\psi\rangle_{s_i} |x_0\rangle_{s_i} \otimes |1\rangle_{Q_i} + \sum_{j \neq 0} \langle x_j |\psi\rangle_{s_i} |x_j\rangle_{s_i} \otimes |0\rangle_{Q_i}. \]  

(45)

Intuitively, the FANIN gate compresses the information whether the subsystem \(s_i\) satisfies the control condition to the single processor ancilla \(Q_i\). The above FANIN gate is equivalent to the \(m\)-qubit Toffoli gate (where \(m\) is the number qubits in the subsystem). We prefer to refer it as FANIN to emphasize its operational characteristic.

The implementation of the FANIN gate is scalable with a circuit depth of \(O(n_{sub})\) and ancilla count of \(O(n_{sub})\). To this end, consider the arrangement of subsystems in a regular 2D structure as depicted in Fig. 11. In order to transfer a single processor ancilla \(Q_i\) from its initial state \(|0\rangle_{Q_i}\) to \(|1\rangle_{Q_i}\) conditionally on all subsystems \(s_i\) being in the state \(|x_0^{(i)}\rangle_{s_i}\), we propose the following protocol, which also makes use of the antenna and receiver ancillae.

1. Apply \(C_{x_0^{(i)}} \text{ NOT}_{Q_i}\) for each subsystem

2. For the outermost subsystems, transfer the state of \(Q_i\) to the \(R_j\) of neighboring subsystems according to the arrows depicted in Fig. 11. For the corner cases, use FANIN protocol on the three corner subsystems. Uncompute the ancillary qubits \(\{T_{i,j}\}\)

3. Do the following until you reach the central subsystem (shown in green in Fig. 11)

(a) Apply \(C_{11;R_i} \text{ NOT}_{A_j}\)

(b) Transfer the state of \(A_j\) to \(R_i\) of the neighboring states according to the arrows in Fig. 11

Treat the corner cases as in the outermost layer.

(c) Uncompute \(\{T_j\}, A_j, R_j,\) and \(Q_j\).

4. Apply \(C_{x_0^{(c)}} A_c\), where \(x_0^{(c)}\) denotes the state where all antenna ancillae surrounding \(A_c\) are in state \(|1\rangle\).

As a result, the antenna ancilla of the central subsystem \(A_c\) is in the \(|1\rangle\) state only if the state of the composite system is

\[ \bigotimes_i |x_0^{(i)}\rangle_{s_i}. \]

The number of gates for the scaled implementation of the FANIN protocol requires \(O(n_{sub})\) gates and an implementation depth of \(O(\sqrt{n_{sub}})\) matching the lower bound imposed by the Lieb-Robinson bounds [45, 36]. In three-dimensions, a similar scheme can be implemented to complete the FANIN protocol in depth \(O(n_{sub}^{1/3})\). To our knowledge our protocol is the first which implements the FANIN (i.e., \(n_{sub}\)-qubit Toffoli gate) with a depth which matches this lower bound. We emphasize the role of considering the concrete experimental implementation in this result. Lastly, we note that due to the use of multi-qubit controlled unitaries, the EBGC of the above protocol is not optimal.
2. 1-to-many communication (FANOUT)

The FANOUT protocol broadcasts the state of a single qubit to the receiver ancilla of many subsystems \([57]\). It can be implemented using single control multi-target CNOT gates. Considering the 2D layout depicted in Fig. 11, the state of \(A_c\) can be broadcasted using a parallelized implementation of CNOT and CNOT\(_3\) gates in accordance with the arrows connecting the different antenna ancillae in Fig. 11. This scheme implements the FANOUT gate using \(O(n_{sub})\) gates and in \(O(d\sqrt{n_{sub}})\) steps, which is optimal for local systems as in the case of the FANIN protocol. Because at each step, we are using single-qubit controlled unitaries, this protocol is naturally robust to errors. In particular, the EBGC for broadcasting a state \(|\psi_c\rangle\) to \(n_{sub}\) subsystems scale as \(|(1_c|\psi_c\rangle)^2O(n_{sub})\).

B. Scalable implementation of \(C_{x0}U_1 \cdots U_k\) and \(V_{OHE}\)

In this section, we utilize the FANIN and FANOUT protocols discussed above to construct the modular and distributed versions of the \(C_{x0}U_1 \cdots U_k\) and \(V_{OHE}\) gates. Given these constructions, the implementation of modular and distributed versions of LCU-based block-encoding unitaries, including the use of exit and phase ancillae, are straightforward.

1. \(C_{x0}U_1 \cdots U_k\)

A modular and distributed version of \(C_{x0}U_1 \cdots U_k\) can be implemented by first applying the FANIN protocol to transfer the central antenna \(A_c\) to the state \(|1_0\rangle\) only if each of \(n_{sub}^{(c)}\) subsystems making up the control register satisfy the associated control condition. Then the state of \(A_c\) can be broadcasted to the \(n_{sub}^{(t)}\) receiver ancillae \(R^{(t)}\) of the subsystems that make up the target register using a FANOUT protocol. Lastly, the controlled unitary \(C_{q_i}U_1 \cdots U_k\) can be applied \(i\)th target subsystem, after which all of the ancillae can be uncomputed. Thus, \(C_{x0}U_1 \cdots U_k\) can be implemented in \(O(d^2\sqrt{n_{sub}})\) steps, and using \(O(n_{sub}^{(c)} + n_{sub}^{(t)})\) gates (see Fig. 10).

2. \(V_{OHE}\)

The unitary \(V_{OHE}\) acting on \(n_{sub}\) subsystems of size \(n_j\) achieves the following transformation

\[
V_{OHE} \prod_j |0\rangle^{\otimes n_j} = \left( \prod_j \beta_j V_{OHE}^{(j)} \right) \prod_j |0\rangle^{\otimes n_j}
\]

\[
= \sum_{j=1}^{n_{sub}} \beta_j \sum_{i=1}^{n_j} V_{OHE}^{(j)} |0\rangle^{\otimes n_0} \cdots |0\rangle^{\otimes n_{j-1}} \otimes |\text{ohe, i}\rangle_j \otimes |0\rangle^{\otimes n_{j+1}} \cdots |0\rangle^{\otimes n_{sub}},
\]

(46)

where \(|\text{ohe, i}\rangle_j\) is a One-Hot encoding computational basis state of the \(j\)th subsystem, and \(V_{OHE}^{(j)}\) is the One-Hot encoding unitary which acts non-trivially only on \(s_j\). Unitarity entails \(\sum_j |\beta_j|^2 = \sum_i |\alpha_i^{(j)}|^2 = 1\).

In order to implement a linear superposition of \(n_{sub}\) \(V_{OHE}^{(j)}\), it is required to condition the application of \(V_{OHE}^{(j)}\) on it being the only non-trivial unitary to be applied on the system. However, because \(V_{OHE}^{(j)} \prod_i |n_i\rangle\) is a linear superposition of \(n_j\) One-Hot encoding bit-strings, it is challenging to verify whether \(V_{OHE}^{(j)}\) is applied on \(s_j\). More generally, in order to verify that \(V_{OHE}^{(j)}\) is applied, we need to use \(O(n_{j})\) controlled unitaries.

Thus, the scalable implementation of \(V_{OHE}\) poses two important challenges. First, we need to be able to communicate to each subsystem \(s_j\) that for all subsystems \(s_j\) such that \(j \neq i\), the unitary \(V_{OHE}^{(j)}\) was not applied. Second, we need to make sure to implement \(V_{OHE}^{(j)}\) only with probability \(|\beta_j|^2\), such that the resulting unitary is Eq. \([46]\). To overcome these challenges, we use two ingredients. First, we implement controlled versions of \(V_{OHE}^{(j)}\) conditioned on the state of the processor ancilla \(Q_j\). Thanks to the controlled nature of the operation, the probability that \(Q_j\) is in state \(|1\rangle\) \(Q_j\) is identical to the probability that \(V_{OHE}^{(j)}\) was applied. Second, to ensure that each subsystem \(s_i\) learns about all previous subsystems \(s_j < i\), we order the subsystems in a linear manner [see Fig. 11(b)].

The protocol that we propose for implementing the distributed version of \(V_{OHE}\) is the following. For each subsystem \(s_i\) with \(i \geq 1\),

1. (Query whether \(V_{OHE}^{(j)}\) applied for \(1 < j < i\).) Transfer the state of \(A_{i-1}\) to \(R_i\). Uncompute the ancillary qubits \(\{T_{i-1,1}\}\).
2. (Input coefficient \(\beta_j\).) Apply a single-qubit rotation to \(Q_i\) along the \(-\)-axis by an angle \(\theta_i = \arcsin (\beta_i)\).
3. Apply \(C_{01:R_iQ_iV_i^{(i)}}\).
4. (If any \(V_{OHE}^{(j)}\) for \(i \leq j\) is applied, turn on antenna.) Apply \(C_{\neq 00:Q_iR_iX_{A_i}}\).
5. Repeat for the \(i + 1\)-st subsystem, until \(i = n_{\text{sub}}\).

The protocol completes in \(O(n_{\text{sub}})\) steps, which is optimal in 1D, but suboptimal in higher dimensions. However, the condition that each subsystem should receive the information from every other subsystem is a constraint that likely makes an implementation with circuit depth \(O(d \sqrt{n_{\text{sub}}})\) impossible. Because for each subsystem, we apply controlled unitaries conditioned on few-qubit ancillary states that are unlikely to be occupied, the EBGC of the above protocol is \(O(1)\), when we use the biased error model introduced in Section \(\S\ A.2\) On the other hand, the size of the circuit required to implement the above protocol is \(O(n_{\text{sub}})\), with a small overhead associated with the gates applied on the telecommunication port. Lastly, we note that the ancillae of the telecommunication port are not uncomputed. Yet, as discussed in Section \(\S\ A\) this is not a problem for the implementation of the LCU-based block encoding.

VIII. IMPLEMENTATION OF OPTIMAL HAMILTONIAN SIMULATION PROTOCOLS ON THE RYDBERG ATOM PLATFORM

In previous sections, we demonstrated that the implementation of the algorithmic primitives of the QSP framework in Rydberg atoms is efficient, robust to errors and scalable. In the remaining of the sections, we focus on Hamiltonian simulation as a particular application of QSP. In this introductory section, we overview three different approaches to the Hamiltonian simulation problem: (i) Hamiltonian simulation algorithms based on product formulas, (ii) QSP-based optimal Hamiltonian simulation of generic Hamiltonians, and (iii) optimal simulation algorithm of Ref. \(\text{[31]}\) for local Hamiltonians. In Section \(\S\ A\) we provide the explicit error-bounded gate and depth counts to implement the algorithms of these three approaches utilizing the native \(V_{\text{OHE}}\) and \(C_{z^0}U_1 \cdots U_k\) gates of the Rydberg platform.

A. Hamiltonian simulation

The use of physical quantum systems to simulate quantum dynamics has a rich tradition. The task of quantum Hamiltonian simulation is simply stated: given any initial state \(|\psi_0\rangle\) of \(n\) qubits, a Hamiltonian \(H\) and evolution time \(t\), construct a sequence of quantum gates, which approximates the final state \(|\psi_f\rangle = e^{-iHt} |\psi_0\rangle\). In Ref. \(\text{[4]}\), Lloyd provided the first demonstration that this task is feasible. The strategy of what is now known as product formulas (PF) \(\text{[5]}\) is to make use of the algebraic structure of the local terms in the expansion \(H = \sum_{l=1}^{L} H_l\) through the Baker-Campbell-Hausdorff identity

\[
\exp(-iHt) = \left(e^{-iH_1t/r}e^{-iH_2t/r} \cdots e^{-iH_Nt/r}\right)^r + O\left(\frac{(\Lambda t)^2}{r}\right),
\]

where \(\Lambda = \max_i |H_i|\), and \(r\) is the number of time slices used in the approximation. For a fixed error tolerance \(\epsilon\), and a geometrically local Hamiltonian for which \(O(L) = O(n)\) the number of time slices required is quadratic in the simulated space-time volume \(r = O(n^2t^2/\epsilon)\). Since each time slice has \(O(n)\) operations, the total gate complexity of the PF algorithm is \(O(n^3t^2/\epsilon)\). Higher order PF exist \(\text{[27]}\), and at order \(2k\) the dependence of the gate complexity on the system size improves to \(O(5^{2k}n^2t^2/\epsilon^{1/2k})\), although in the limit of large \(k\) the prefactor becomes prohibitive. Recently Ref. \(\text{[34]}\) showed that in the case of a one-dimensional system with nearest-neighbor interactions, the gate complexity can be reduced by a factor of \(n\) using an integral representation of the Trotterization error, and the resulting algorithm has a gate complexity of \(O((nt)^{1+1/2k}/\epsilon^{1/2k})\) which scales almost linearly in the simulated space-time volume.

The PF algorithm of Ref. \(\text{[34]}\) for the 1D system with nearest-neighbor interactions analytically demonstrates the validity of arguments put forward by Jordan, Lee, and Preskill \(\text{[59]}\) which claimed that the simulation of quantum dynamics generated by geometrically local Hamiltonians requires a gate complexity at least linear in the simulated space-time volume. The more general question: “Can the same gate complexity be obtained for any time-dependent local Hamiltonian?” was answered affirmatively by Haah, Hastings, Kothari, and Low \(\text{[31]}\). Perhaps more importantly, the gate complexity of the algorithm, \(O(nt\text{polylog}(nt/\epsilon))\) was proved to be optimal even for simulating only local observables associated with the time evolution. In comparison, we note that the gate complexity \(O((nt)^{1+1/2k}/\epsilon^{1/2k})\) in Ref. \(\text{[34]}\) is not optimal but nearly-optimal due to the extra factor \((nt)^{1/2k}\). Optimal Hamiltonian simulation algorithm of Ref. \(\text{[31]}\) makes use of two facts. The first is known as Lieb-Robinson bounds \(\text{[35, 36]}\), which constrain how information spreads in local Hamiltonian systems. The second is that the novel algorithmic frameworks of LCU and QSP enables optimal Hamiltonian simulation in a small subsystem with respect to all parameters (i.e, with polynomial cost in the system size while achieving a polylogarithmic dependence on the error threshold) \(\text{[22]}\). Next, we detail QSP-based optimal Hamiltonian simulation algorithms first for arbitrary Hamiltonians and then for local systems.

B. QSP-based optimal Hamiltonian simulation

The Hamiltonian simulation algorithm based on the QSP framework uses the block-encoding of the Hamiltonian \(H\) to construct a polynomial approximation of
FIG. 12. Overview of the algorithm in Ref. [31]. (a) the first approximation depicts the principle relation between the initial and decimated circuits for Hamiltonian simulation. On the other hand, the equivalence relation depicts the spread and reversal of errors originating from the left (yellow) and right (blue) edges of each block. At the first time step, the error due to the left edge of a block simulating $e^{-iH_{BC}t}$ spreads within the light-cone depicted in yellow. At the second step, the evolution operator $e^{iH_Bt}$ fixes this error but it also introduces new errors due to the right boundary of the block. The final evolution by $e^{-iH_{AB}t}$ reverses the error introduced by $e^{iH_Bt}$, and the approximation error is bounded by the exponentially small error outside of the light cone given the intermediate block is large enough to contain the light-cone. (b) Demonstration of how the block decimation operator can be repeated in space and the pattern of errors induced.

$$P(H) \approx e^{-iHt}.$$ Ref. [32] proved that this method results in an optimal query complexity, which is

$$k^* = O \left( \alpha t + \frac{\log(1/\epsilon)}{\log \log (1/\epsilon)} \right),$$ (47)

where we define $\alpha \equiv \|H\| = O(N)$, for a Hamiltonian composed of $N$ Pauli strings. The linear scaling of $k^*$ with respect to the spectral norm of the Hamiltonian is due to the unitarity block-encoding utilized by QSP. We also emphasize that the scaling of the query complexity with respect to error tolerance $\epsilon$ is exponentially improved compared to the simulation algorithms based on Trotterization [33]. Moreover, it is possible show that the number of queries $k$ can be bound by the following inequality [9]

$$k^* \leq e^{q \alpha t} + \frac{\ln (1/\epsilon)}{q} \quad \forall q \in \mathbb{R}.$$

(48)

The query complexity of the QSP-based Hamiltonian simulation can be related to the time and error-bounded gate complexities, given a specific implementation of the query. We emphasize that even if the time required to implement the query is constant, the time complexity of Hamiltonian simulation of a system of size $n_{\text{site}}$ for a time $t = n_{\text{site}}$ results in an implementation time of $O(n^2_{\text{site}})$, which is suboptimal. On the other hand, implementing the query with constant EBGC implies an implementation optimal with respect to errors. In Section [IX] we demonstrate that this optimal error complexity is achieved for our proposal for $k$-local Hamiltonians.

C. Optimal Quantum Hamiltonian simulation of local Hamiltonians

Since Ref. [32] argued that the optimal gate complexity of an algorithm simulating local Hamiltonian simulations should scale roughly as the simulated space-time volume, there has been a renewed interest in finding rigorous tight bounds for optimal simulation of local Hamiltonians [31, 34]. In 2018, an algorithm by Haah et al. [31], constructed an algorithm with a gate complexity linear in the simulated $d + 1$ dimensional space-time volume $O(n \text{polylog}(n^{1/d}/\epsilon))$, where $\epsilon$ is the total error tolerance of Hamiltonian simulation, and proved the optimality of this bound. In this section, we give a brief review of this work to motivate the constructions in Section [IX].

From the bird’s eye view, the algorithm in Ref. [31] uses a block-decimation of the time-evolution unitary $e^{-iHt}$ which guarantees that the error due to the decomposition is bound by the Lieb-Robinson (LR) theorems [35, 36]. LR theorems formalize the intuition that for local Hamiltonians, the maximum speed that information can travel is a well-defined constant, called the Lieb-Robinson velocity $v$. The overall structure of the algorithm in Ref. [31] is depicted in Fig. [12] for the case of a one-dimensional system, which we consider for sake of simplicity. Any block-decimation of the evolution operator introduces local errors at each boundary. These errors can be described as the emission of Hamiltonian terms $H_{bd}$ at the boundaries of neighboring blocks. Upon time evolution by a local Hamiltonian, it is possible to ensure that these errors propagate only within the associated light-cones (depicted as yellow and blue triangles...
in Fig. 12] up to exponentially small corrections. The knowledge of the local spread of such errors allows on to design a spin-echo-like algorithm which reverses the spread of the errors, leaving behind only the exponentially small corrections. Formally, one can show that for a given a block decimation of the sites $X = A \cup B \cup C$, a constant $\mu = O(1)$, Lieb-Robinson velocity $v$, and $vt \ll l = \text{dist}(B)$

$$|U_t - e^{-iH_A \cup B \cup C} e^{iH_B} e^{-iH_A \cup C}| < O(|H_{bd}| \exp(-\mu l)), \quad (49)$$

where $H_X$ denotes the sum of Hamiltonian terms which have a support on a region $X$. Longer times can be simulated by applying the same decimated evolution operator repeatedly $t$ times and obtain an error linearly increasing with $t$.

Given the above discussion, the choice of the smallest dimensions of each block, denoted $l$ in the space axis and $t_\square$ in the time axis, is determined by (i) the tolerance $\epsilon_{LR}$ for errors resulting from the block-decimation, (ii) the tolerance $\epsilon_{\square}$ for errors due to the QSP-based approximate Hamiltonian simulation of each block, and (iii) the number of blocks $m = O(nt/t_\square)$. In particular, given a total error tolerance $\epsilon$, we would like to associate with each block to be $\epsilon_\square + \epsilon_{LR} = O(\epsilon/m)$ \[35, 38\]. In the following, we assume $\epsilon_\square = \epsilon_{LR}$. Given Eq. (49), we have $\epsilon_{LR} = O(e^{-\mu l})$, and the spatial dimension of each block is $l = O(\log (nt/t_\square \epsilon))$. We are then left with the challenge of realizing the Hamiltonian simulation of a system of size $O(\log (nt/t_\square \epsilon))$, with error tolerance $\epsilon_\square = O(e^{-\mu/l})$ using only $O(\text{polylog}(nt/t_\square \epsilon))$ gates. Fortunately, the QSP-based Hamiltonian simulation algorithm discussed in Section [IV] has a query complexity that scales logarithmically with $e^{-1}$, resulting in a the gate complexity that scales almost linearly with the simulated space-time volume

$$O(nt \text{polylog}(nt/t_\square \epsilon)). \quad (50)$$

In summary, the optimal quantum Hamiltonian simulation algorithm by Haah et al. uses the QSP-based query optimal Hamiltonian simulation algorithm in combination with a clever block-decimation of the space-time evolution to obtain a near optimal gate count for Hamiltonian simulation algorithm for local Hamiltonians.

We emphasize that the aforementioned optimal gate count is not error-bounded in the sense discussed in Section [IV], as it does not take into account the specific error model of the implementation. Moreover, the near optimal gate count of the algorithm in Ref. [31] comes at a price of a high implementation overhead [33], which makes it less efficient than the product-formula algorithms for Hamiltonian simulation on near-term intermediate scale quantum (NISQ) processors. In the next section, we calculate the overhead associated with the Rydberg platform implementations of both the product formula and the QSP-based Hamiltonian simulation protocols. While the overhead of the algorithm in Ref. [31] makes it undesirable for NISQ devices, we find that the EBGC of QSP-based Hamiltonian simulation of Ref. [32] has orders of magnitude of smaller overhead for gate errors compared to its competitors.

IX. CONCRETE CIRCUITS AND GATE COUNTS

Here, we use the results of the Section [VII] to analyze the resource requirements for the implementation of QSP-based and Hamiltonian simulation algorithms on the Rydberg platform, and compare them to those needed to implement Hamiltonian simulation based on product formula. To this end, we briefly discuss the implementation of product-formula-based Hamiltonian simulation algorithms on the Rydberg platform and calculate the associated EBGCs.

A. Hamiltonian simulation

Here, we explicitly calculate the resources needed for implementing algorithms in the framework of QSP, using LCU-based block-encoding. In particular, for a system of $n_{\text{site}}$ qubits, we consider the number of error-bounded gates, the runtime, and the number of ancillas required to implement a (i) conditional version of LCU block-encoding, (ii) QSP-based Hamiltonian simulation, (iii) Optimal simulation of local Hamiltonians in Ref. [31], and finally (iv) Hamiltonian simulation using 4th order product formula. We note that all of calculations in this section, we assume that the blockade radius can be taken large enough such that the scalable protocols discussed in Section [VII] are not necessary.

1. Implementing Haah’s Optimal Hamiltonian Simulation (Ref. [31])

To facilitate the calculation of resources needed for Hamiltonian simulation, it is necessary to chose an explicit Hamiltonian to be simulated. Here, we chose a the one-dimensional disordered Heisenberg Hamiltonian $H_{\text{DH}}$ as our target system

$$H_{\text{DH}} = \sum_{i \prec j} \sigma_i \cdot \sigma_j + \sum_i h_i \sigma_i^{(z)}. \quad (51)$$

The choice of the disordered Heisenberg Hamiltonian as our target allows us to directly compare the cost of our implementation of QSP-based Hamiltonian simulation to that implied by the previous empirical studies that use product formulas [33, 34].

Our first task is to find the dimensions of each block in the decimation given the parameters $n_{\text{site}}$, $l$, and the error tolerance $\epsilon$. 

In one-dimensions, the number of blocks is

\[ m = 4 \left( \frac{2t_{\text{site}}}{t_{\square}} \right), \]  

where \( l \) and \( t_{\square} \) are the shortest dimensions of each block along space and time coordinates (see Fig. 13), respectively. Hence, we have \( m/2 \) blocks of length \( t \) and \( m/2 \) blocks of length \( 2t \), and the overall factor of 4 in Eq. (52) is due to the normalization of the Hamiltonian for each site (i.e., \( H_{i,t+1} \leq 1 \)). For a given spatial extent \( l \) of each block, the parameter \( t_{\square} \) can be determined by studying how the errors due to the decimation scale as a function of time for a single block. This was done in Ref. [31] and the following relation was found

\[ 0.175 \left( \frac{7.9 t_{\square}}{t+0.95} \right)^{1+0.95} = \frac{\epsilon}{3m} \equiv \epsilon_{\text{LR}}. \]  

For the data presented here, we set \( t = 4n_{\text{site}} \) and \( m_{\square} = m_{\text{LR}} = 10^{-3}/2 \).

Once the parameters \( l \) and \( t_{\square} \) are determined, we can also calculate the order \( k_{\square} \) of the polynomial approximation to the Hamiltonian evolution associated with each block using Eq. (48). For the smaller blocks of spatial size \( l \), we get

\[ k_{\square}(l) = \min_q \left[ e^q t_{\square} l + \ln \left( \frac{l_{\square} + 0.95}{3m} \right) / q \right], \]  

while for blocks of spatial size \( 2t \), we replace \( l \rightarrow 2l \).

The simultaneous implementation of Hamiltonian simulation in each block may introduce unwanted crosstalk errors due to the algebraic decay of the dipolar interactions, even if the blockade radii of adjacent blocks do not overlap (see Fig. 13 a). We circumvent this problem by doubling the implementation time required to simulate evolution for a time \( t_{\square} \). The scheme is depicted in Fig. 13 for a one-dimensional system. For each time step, we require that the blockade radii associated with different spatial blocks have negligible overlap. We also note that, in principle, the block-encoding and the QSP-based Hamiltonian simulation algorithms have the flexibility to implement the simulation of a Hamiltonians with different boundary terms are omitted at each step (see Fig. 12), as well as changing the overall sign of the Hamiltonian \( H \rightarrow -H \).

Having determined \( l \), \( t_{\square} \), and \( k_{\square} \), we next calculate the depth and EBGCs for the CPHASE gate and LCU-based block-encoding using the techniques discussed in Section VI (see also Table II), and assuming that we are using the customized One-Hot encoding state-preparation protocol

\[ d_{\text{LCU}} = 2d_{\text{OHE}} + d_{\text{CZ}} + 1 = 4(1+2L) + 9L + 1 = 124, \]  

\[ n_{\text{LCU}} = 2n_{\text{OHE}} + n_{\text{CZ}} = 16 + 10 = 26, \]  

\[ a_{\text{LCU}} = n_{\text{site}} + L; \quad d_{\text{CPHASE}} = 3; \quad n_{\text{CPHASE}} = \frac{4}{3}, \]

where \( d_{\star} \), \( n_{\star} \), and \( a_{\star} \) denote the circuit depth, EBGC, and the number of ancillae needed for implementing “\( * \)”. In order to calculate the EBGC for \( U_{\text{OHE}} \), we used Eq. (42), and set \( L = 2 \times 3 + 1 = 7 \) to implement Pauli terms associated with the 3 axes of interactions and the disordered field in Eq. (51). Hence, implementing the conditional version of the iterate \( W \) in Eq. (13) costs depth \( d_{\text{CW}} = 2 + d_{\text{CPHASE}} + d_{\text{LCU}} \), where we added depth 2 for each controlled version of the unitaries. On the other hand, the EBGC of \( CW \) is \( n_{\text{CW}} = 2 + 2 + d_{\text{CPHASE}} + n_{\text{LCU}} \). As a result, the QSP-based Hamiltonian simulation algo-

![FIG. 13. The overall schematic for the paralleled implementation of the optimal Hamiltonian simulation of [31] on the Rydberg platform. a) A depiction of the sources of crosstalk errors due to the algebraic decay of the interaction potential, which makes the definition of Rydberg blockade radius \( R_b \) fuzzy. The two system atoms at the boundary between the two blockade volumes introduce errors due to unwanted blockade interactions. b) The experimental configuration for the parallelized application of the algorithm in Ref [31]. We use twice as many overlapping simulation regions (i.e., \( A, B, C \), and \( D \)) as depicted in Fig. 12 to reduce the errors due to the algebraic decay of the dipolar interactions. c) The schedule for applying the SP-based Hamiltonian simulation algorithm on 4 different simulation regions to reduce the crosstalk errors. The crosstalk errors can be reduced arbitrarily at the expense of an increase in the circuit depth.](image-url)
algorithm of Ref. [32] can be implemented using
\[
d_{QSP} = k^s (d_{CW} + 1) \\
a_{QSP} = a_{LCU} + 1 \\
n_{QSP} = k^s (n_{CW} + 1),
\] (58)
where \(k^s\) is calculated using Eq. (48). Lastly, the computation resources to implement the local Hamiltonian simulation algorithm of Ref. [31] are
\[
d_{HID} = 2 \frac{t}{\delta t} (k^s (l + 2) (2l + 2))(d_{CW} + 1) \\
a_{HID} = \frac{n_{site}}{2l} (4 + 2l + 2) \\
n_{HID} = \frac{m}{2} k^s (2l + 2) [n_{CW} + 1].
\]
Notice that the overall factor of 2 in the depth of implementation comes from our method of reducing the crosstalk between the blocks (see Fig. [13]). The number of ancillae is increased by an additional \(O (\frac{n_{site}}{2l})\) ancillae compared to the requirements for QSP-based Hamiltonian simulation in order to facilitate the parallelization by the block-decomposition.

2. Comparison to Hamiltonian simulation with product formulas

Here, we compare the resource requirements for our implementation of the QSP-based Hamiltonian simulation algorithm on the Rydberg platform to those of Hamiltonian simulation using product formulas [33] [34].

In order to have a fair comparison for the disordered Heisenberg model, we consider the ordering structure of the product formula proposed by Ref. [34]. In particular, we focus on the types of Hamiltonians that can be written in the following form
\[
H = \sum_{i}^{n_{site}-1} H_{i,i+1},
\] (59)
where the terms \(H_{i,i+1}\) can be decomposed into Pauli operators which act non-trivially only on sites \(i\) and \(i+1\). Then the first order product formula has the following form
\[
e^{-iH\delta t} \approx \rho_1 (\delta t) = \prod_{k=1}^{n/2-1} e^{-i\delta t H_{2k,2k+1}} \prod_{l=1}^{n/2} e^{-i\delta t H_{2k-1,2k}}
= e^{-i\delta t H_{\text{even}}} e^{-i\delta t H_{\text{odd}}}.
\] (60)
Moreover, the higher order product formulas can be constructed as the following [34]
\[
\rho_2 (\delta t) = e^{-i \frac{\delta t}{2}} H_{\text{even}} e^{-i \delta t H_{\text{odd}}} e^{-i \frac{\delta t}{2}} H_{\text{even}}
\] (61)
\[
\rho_{2k} (\delta t) = \rho_{2k-2} (p_k \delta t)^2 \rho_{2k-2} ((1-4p_k) \delta t) \rho_{2k-2} (p_k \delta t)^2,
\] (62)
where \(p_k = 1/4 - 4^{\frac{k}{2}}\). In the following, we will only focus on the 4th order product formula algorithm as it results in the best conventional gate counts in Ref. [33]. The errors induced by the 4th order product formula approximation to the evolution operator \(e^{-iH_{\text{even}}(\delta t)}\) scale as \(O(n (\delta t)^5)\) for small \(\delta t\) [34]. In order to simulate larger times, one conducts the simulation in \(r = t/\delta t\) segments. Ref. [34] utilized a numerical optimization algorithm which determined that the number of segments \(r\) for an error threshold \(\epsilon = 10^{-3}\) and 4th-order product formula
\[
r_4 \approx 4n_{\text{site}}^{1.555}.
\] (63)

In order to calculate the resource costs of implementing product formulas on the Rydberg atom platform, we consider the specific protocols proposed in Ref. [41]. This proposal is based on an implementation of the exponential of a Pauli term in the Hamiltonian (i.e., \(e^{-i\alpha_j P_j}\)) using a single ancilla \(a\). In particular, the scheme uses the following gate sequence
\[
e^{-i\alpha_j P_j} = G e^{i\alpha_j \sigma_z^a} G^*,
\] (64)
where
\[
G = e^{-i\pi/4 \alpha_j^a} U_j e^{i\pi/4 \alpha_j^a},
\] (65)
and \(U_j = |1\rangle_a \langle 1| \otimes P_j + (I - |0\rangle_a \langle 0|) \otimes I\). Intuitively, the transformation \(G\) maps the eigenstates of the Pauli operator \(P_j\) with eigenvalues \(\pm 1\) onto the \(|\pm 1\rangle_{\text{anc}}\) states of the ancilla qubit. The two eigenstates acquire phases with opposite signs using the single-qubit phase rotation \(e^{i\alpha_j^a \sigma_z^a}\). The exponentiation requires 1 ancilla per two-qubit Pauli operator and can be implemented in depth \(8 + \alpha_j / \pi\). The EBGC of the exponentiation step is
\[
\frac{1}{3} \left[ 2 \left( 2 + \frac{\text{supp}(P_j)}{2} \right) + \frac{\alpha_j}{\pi} \right].
\] (66)
Thus, the exponentiation of each two-qubit Pauli takes \(2 + \alpha_j / (3 \pi)\) error-bounded gates.

The resources needed for the simulation of the 1D disordered Heisenberg model using 1st order product formula for a single segment [33] that implements a time evolution for \(\delta t\) are the following
\[
d_{P_F} = 2 \times 3 \times (8 + \delta t / \pi) + \delta t / \pi = 48 + 7 \delta t / \pi \\
an_{P_F} = n_{\text{site}} / 2
\]
\[
n_{P_F} = n_{\text{site}} \left[ 3 \left( 2 + \frac{\delta t}{3 \pi} \right) + \frac{\delta t}{3 \pi} \right] = n_{\text{site}} \left( 6 + \frac{4 \delta t}{3 \pi} \right),
\]
where we note the factors of 2 in the calculation of \(d_{P_F}\) arise from the serial application of evolution by \(H_{\text{even}}\) and \(H_{\text{odd}}\), and we assume that the local random field can be implemented using 1 single-qubit rotation without any need for ancillas. Because the ancillae are uncomputed after each step, \(a_{P_F}\) does not change with increasing \(k\). Here, we compare the QSP-based Hamiltonian simulation to the 4th order product formula according to Eq. [62], as it results in the lowest gate counts in Ref. [33]. To calculate the resources for longer time evolution, the above expressions should be multiplied by \(r_4\) in Eq. [63].
FIG. 14. a) The EBGCs and b) the circuit depth of Hamiltonian simulation algorithms (i) based on the 4th-order product formula (blue, Ref. [33]), (ii) QSP-based local Hamiltonian simulation (orange, Ref. [31]) which uses block-decimation, and (iii) QSP-based Hamiltonian simulation (green, Ref. [32]). We assume that the QSP-based Hamiltonian simulation of Ref. [32] can be implemented without the scalable protocols discussed in Section VII. While from the point of view of depth complexity, the 4th-order product formula is superior in all system sizes considered, the QSP-based Hamiltonian simulation of Ref. [32] exhibits the lowest implementation overhead in terms of EBGCs, achieving more than an order of magnitude reduction compared to the 4th order product formula, while simultaneously exhibiting better asymptotic scaling.

3. Results

The results for both the product-formula- and QSP-based Hamiltonian simulation algorithms are displayed in Fig. 14.

Most strikingly, the QSP-based Hamiltonian simulation of Ref. [32] has an EBGC (green in Fig. 14 a) which is more than an order of magnitude smaller than that of the simulation based on the 4th order product formula (in blue), for \( n_{\text{site}} = 50 \). The optimal local Hamiltonian simulation algorithm of Ref. [31] has an increased overhead (in orange) due to the block-decimation, which results in an overhead in the query complexity in comparison to the algorithm of Ref. [32]. Moreover, since the scaling of EBGC for both QSP-based Hamiltonian simulations are optimal, there is no system size for which the 4th order product formula is more robust to errors than its QSP-based counterparts.

From the point of view of the circuit depth, the 4th order product formula results in the shortest circuit depth Hamiltonian simulation for NISQ devices, although the optimal local Hamiltonian simulation of Ref. [31] has a better scaling. Note that the time complexity of the QSP-based Hamiltonian simulation of Ref. [32] shows suboptimal scaling as it does not take advantage of the geometric locality to parallelize the simulation.

As a result, the choice of using product formula vs. QSP-based Hamiltonian simulation on the Rydberg atom platform depends on the errors relevant for the implementation. If the lifetime of the logical states are the main contributor to the decoherence, then using the product-formula-based Hamiltonian simulation is the most advantageous. On the other hand, if the logical states are long-lived and the majority of errors are introduced during gate operations, and it is possible to implement controlled unitaries with biased errors, then QSP-based Hamiltonian simulation has a clear advantage.

X. CONCLUSIONS AND OUTLOOK

In this work, we proposed control protocols that implement QSP-based algorithms on the Rydberg atom platform. Most importantly, the proposed multi-qubit Rydberg blockade gates utilize Electromagnetically Induced Transparency to introduce highly state-dependent errors. Our analysis demonstrates how such biased error models allow the design of control sequences whose an error-bounded gate counts (EBGCs) exhibit exponentially improved scaling compared to the associated circuit size. Our work highlights the importance of including the physical constraints relevant for concrete experimental platforms when analysing (i) the resources needed to implement a particular quantum algorithm and (ii) the rate at which classical data can be encoded in quantum degrees of freedom. Moreover, we demonstrated that QSP-based algorithms can be implemented in a scalable manner on a modular and distributed network of Rydberg atoms. Lastly, we demonstrated the efficiency of our proposed implementation of Hamiltonian simulation in terms of the error-bounded gate complexity, by comparing it to a state-of-the-art implementation of product-formula-based Hamiltonian simulation algorithm.

Given the generality of QSP framework, and the recent successes of the Rydberg atom platform [24, 56], we foresee many promising avenues of research that can originate from our work. Our techniques can be extended to implementing algorithms based on Quantum Singular Value Transformation (QSVT) in a straightforward manner. A crucial question in this direction is whether the
properties of Rydberg atoms provide other substantial advantages in realizing specific QSVT-based algorithms. For instance, the result that local signal operators are much easier to implement than those that are non-local can serve as a basis to explore the power of QSP-based quantum algorithms instantiated by local signal operators. Furthermore, the efficiency of the proposed V_OHE gate in loading classical data into quantum mechanical degrees of freedom should be further investigated in the context of a more general theory of communication between quantum and classical information processors.

ACKNOWLEDGMENTS

S.Z. and S.S. thank I. Chuang and M.D. Lukin, D. Bluvestein, I. Cong, S. Ebadi, and A. Keesling for insightful comments and helpful discussions. S.Z. acknowledges the financial support from the Swiss National Science Foundation through the Early Postdoc.Mobility grant (P2EZP2_184320) and from the Army Research Office, ARO MURI (grant no W911NF1910517). S.S. acknowledges NTU Research for their financial and technical support.

[1] D. Deutsch and R. Jozsa, Rapid solution of problems by quantum computation, Proceedings: Mathematical and Physical Sciences 439, 553 (1992).
[2] L. K. Grover, Quantum mechanics helps in searching for a needle in a haystack, Physical review letters 79, 325 (1997).
[3] P. W. Shor, Algorithms for quantum computation: discrete logarithms and factoring, in Proceedings 35th annual symposium on foundations of computer science (IEEE, 1994) pp. 124–134.
[4] S. Lloyd, Universal quantum simulators, Science, 1073 (1996).
[5] M. A. Nielsen and I. Chuang, Quantum computation and quantum information (American Association of Physics Teachers, 2002).
[6] A. Y. Kitaev, Quantum measurements and the abelian stabilizer problem, arXiv preprint quant-ph/9511026 (1995).
[7] G. H. Low, T. J. Yoder, and I. L. Chuang, Methodology of resonant equiangular composite quantum gates, Physical Review X 6, 041067 (2016).
[8] G. H. Low, Quantum signal processing by single-qubit dynamics, Ph.D. thesis, Massachusetts Institute of Technology (2017).
[9] 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.
[10] J. M. Martyn, Z. M. Rossi, A. K. Tan, and I. L. Chuang, Grand unification of quantum algorithms, PRX Quantum 2, 040203 (2021).
[11] A. W. Harrow and A. Montanaro, Quantum computational supremacy, Nature 549, 203 (2017).
[12] S. Aaronson, Read the fine print, Nature Physics 11, 291 (2015).
[13] S. Chakraborty, A. Gilyén, and S. Jeffery, The Power of Block-Encoded Matrix Powers: Improved Regression Techniques via Faster Hamiltonian Simulation, in 46th International Colloquium on Automata, Languages, and Programming (ICALP 2019) Leibniz International Proceedings in Informatics (LIPIcs), Vol. 132, edited by C. Baier, I. Chatzigiannakis, P. Flodchin, and S. Leonardi (Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2019) pp. 33:1–33:14.
[14] M. Saffman, Quantum computing with atomic qubits and rydberg interactions: progress and challenges, Journal of Physics B: Atomic, Molecular and Optical Physics 49, 202001 (2016).
[15] C. S. Adams, J. D. Pritchard, and J. P. Shaffer, Rydberg atom quantum technologies, Journal of Physics B: Atomic, Molecular and Optical Physics 53, 012002 (2019).
[16] K. Molmer, L. Isenhower, and M. Saffman, Efficient grover search with rydberg blockade, Journal of Physics B: Atomic, Molecular and Optical Physics 44, 184016 (2011).
[17] H.-J. Briegel, T. Calarco, D. Jaksch, J. I. Cirac, and P. Zoller, Quantum computing with neutral atoms, Journal of modern optics 47, 415 (2000).
[18] L. Henriet, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Reymond, and C. Jurczak, Quantum computing with neutral atoms, Quantum 4, 327 (2020).
[19] H. Bernien, S. Schwartz, A. Keesling, H. Levine, A. Omran, H. Pichler, S. Choi, A. S. Zibrov, M. Endres, M. Greiner, et al., Probing many-body dynamics on a 51-atom quantum simulator, Nature 551, 579 (2017).
[20] A. Browaeys and T. Lahaye, Many-body physics with individually controlled rydberg atoms, Nature Physics 16, 132 (2020).
[21] A. Omran, H. Levine, A. Keesling, G. Semeghini, T. T. Wang, S. Ebadi, H. Bernien, A. S. Zibrov, H. Pichler, S. Choi, et al., Generation and manipulation of schrödinger cat states in rydberg atom arrays, Science 365, 570 (2019).
[22] R. Samajdar, W. W. Ho, H. Pichler, M. D. Lukin, and S. Sachdev, Quantum phases of rydberg atoms on a kagome lattice, Proceedings of the National Academy of Sciences 118, 10.1073/pnas.2015785118 (2021). https://www.pnas.org/content/118/4/e2015785118.full.pdf.
[23] R. Verresen, M. D. Lukin, and A. Vishwanath, Prediction of toric code topological order from rydberg blockade, arXiv preprint arXiv:2011.12310 (2020).
[24] G. Semeghini, H. Levine, A. Keesling, S. Ebadi, T. T. Wang, D. Bluvestein, R. Verresen, H. Pichler, M. Kaliowski, R. Samajdar, et al., Probing topological spin liquids on a programmable quantum simulator, arXiv preprint arXiv:2104.04119 (2021).
[25] M. Endres, H. Bernien, A. Keesling, H. Levine, E. R. Anschuetz, A. Krajenbrink, C. Senko, V. Vuletic, M. Greiner, and M. D. Lukin, Atom-by-atom assembly of defect-free one-dimensional cold atom arrays, Science 354, 1024 (2016).
[26] A. M. Childs and N. Wiebe, Hamiltonian simulation
