Strategies for simulating time evolution of Hamiltonian lattice field theories

Siddharth Hariprakash[*,1,2,3] Neel S. Modi[*,1,2] Michael Kreshchuk,[2] Christopher F. Kane,[4] and Christian W Bauer[2]

[1]*Center for Theoretical Physics and Department of Physics, University of California, Berkeley, California 94720, USA*
[2]*Physics Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA*
[3]*National Energy Research Scientific Computing Center (NERSC), Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA,*
[4]*Department of Physics, University of Arizona, Tucson, Arizona 85719, USA*

Simulating the time evolution of quantum field theories given some Hamiltonian H requires developing algorithms for implementing the unitary operator e^{-iHt}. A variety of techniques exist that accomplish this task, with the most common technique used so far being Trotterization, which is a special case of the application of a product formula. However, other techniques exist that promise better asymptotic scaling in certain parameters of the theory being simulated, the most efficient of which are based on the concept of block encoding.

In this work we study the performance of such algorithms in simulating lattice field theories. We derive and compare the asymptotic gate complexities of several commonly used simulation techniques in application to Hamiltonian Lattice Field Theories. Using the scalar $\hat{\varphi}^4$ theory as a test, we also perform numerical studies and compare the gate costs required by Product Formulas and Signal Processing based techniques to simulate time evolution. For the latter, we use the the Linear Combination of Unitaries construction augmented with the Quantum Fourier Transform circuit to switch between the field and momentum eigenbases, which leads to immediate order-of-magnitude improvement in the cost of preparing the block encoding.

The paper also includes a pedagogical review of utilized techniques, in particular Product Formulas, LCU, Qubitization, QSP, as well as a technique we call HHKL based on its inventors' names.

CONTENTS

---

[*] These authors contributed equally to this work.

I. INTRODUCTION

Performing precise numerical simulations is an essential ingredient in the studies of Quantum Field Theories (QFTs), which describe the most fundamental interactions within the Standard Model of particle physics. Some of the difficulties arising in such studies include: a large, non-conserving number of degrees of freedom; different types of particles and their interactions; and non-perturbativity of the strong interaction at large distances. A common approach to numerical simulation of field theories is provided by Euclidean Lattice Field Theory, where space-time is discretized and observables are calculated using the imaginary-time path integral formalism. For a recent overview of the field, see [1]. Such a conventional approach to lattice field theory relies on Monte-Carlo integration to perform the integral over the vast Hilbert space [2], which requires the weight of each field configuration to be real and positive. While this has been very successful, there are many observables, including those requiring dynamical information, for which Monte-Carlo integration is not possible due to the so-called sign problem [3, 4]. The issue of the sign problem, however, can be avoided entirely by computing such observables within the framework of Hamiltonian Lattice Field Theories (HLFTs), in which space is discretized, while time is kept continuous [5–8].[2] While resources required for the simulation of HLFTs on classical computers scale exponentially with the number of lattice sites [15], Jordan, Lee, and Preskill showed that quantum computers provide an exponential advantage over classical computers for this problem [16]. In fact, simulating QFT with the aid of quantum computers served as one of the motivations for the development of quantum computation [17], and is still considered as one of its most promising applications. For a recent review of the applications of quantum computing to high-energy physics, see [18].

A major task in such a scenario is the ability to efficiently simulate on a quantum computer the time evolution of a given Hamiltonian. In Ref. [19] Jordan, Lee, and Preskill made a detailed proposal to simulate a scalar field theory on quantum computers. Similarly to the earlier research [20–22], their work relies on the Suzuki-Trotter approximation [23–28] to compute the exponential of the Hamiltonian, and hence time evolve the scalar field theory.

This paper compares the asymptotic resource requirements of various techniques to implement the time evolution operator of quantum field theories, including Suzuki-Trotter expansions and nearly-optimal simulation strategies, such as Quantum Signal Processing (QSP) [29–33] and the HHKL method named after the authors of Ref. [34]. For each method, we compute the asymptotic dependence with respect to various parameters characterizing the problem at hand, such as the error allowed in the approximation of the evolution operator, measures of Hamiltonian size / complexity, as well as the time for which the system is evolved. This comparison is carried out for two very general classes of lattice Hamiltonians, but we also provide more detailed results for a scalar quantum field theory as an explicit example.

Using nearly-optimal methods requires querying matrix elements of the Hamiltonian, typically in the form of the *Block Encoding* (BE) subroutine. In this work, we consider the BE for a bosonic degree of freedom, based on the Linear Combination of Unitaries (LCU) algorithm [35], and achieve a significant reduction in the number of gates and ancillary qubits by taking advantage of the quantum Fourier transform for switching between the eigenbases of the field and conjugate momentum operators. We do not consider BEs based on the sparse oracle access input model, for two reasons. First, their construction is highly problem-specific [36–40]. Second, for local lattice field theories, which we consider as one of the major applications of the present work, our studies [40] indicate that the LCU-based approach is more efficient than that based on the sparse oracle access.

A. Paper Structure

We start with setting up our notation in Sec. I B before summarizing the main results of this paper in Sec. I C, with most of the information contained in Tables II to IV. In Sec. II we present a review of the various simulation algorithms. No new information is presented in this section, and the presentation is generic, not yet adapted for lattice field theory simulations. In Sec. III we compute the asymptotic gate complexity of using these techniques to simulate the time evolution of 2 generic classes of lattice field theories. This section will provide derivations for the results presented in Sec. I C. In Sec. IV we present concrete asymptotic gate complexities for a quartic scalar field theory. We provide a discussion

---

[2] Note that while HLFTs sometimes refer solely to quantum field theories in the equal-time formulation, discretized on a spatial lattice [5, 6], in this work we assume (unless explicitly stated otherwise) a more general scenario which applies to any type of spatial discretization (e.g., spatial/momentum lattice [9] or any other single-particle basis set [10]), any type of field discretization (first or second quantization [9, 11]), as well as to both equal-time and light-front quantization [7, 8, 12–14].

of our results in Sec. V and conclude in Sec. VI.

B. Definitions and Notations used in this work

We begin by presenting the standard mathematical notation used to describe the asymptotic behavior of functions

Definition 1 (Asymptotic Notations). *Let $f, g : \mathbb{R} \to \mathbb{R}$ be functions of real variables. Then we say that $f = \mathcal{O}(g)$ if there exists a positive constant c such that $|f(x)| \leq c|g(x)|$ for all sufficiently large choices of x, i.e. $\forall x \geq x_0 \in \mathbb{R}$. We say that $f = \Omega(g)$ if $g = \mathcal{O}(f)$ and that $f = \Theta(g)$ if $f = \mathcal{O}(g)$ and $g = \mathcal{O}(f)$.*

We see that $\mathcal{O}(\cdot)$ and $\Omega(\cdot)$ denote asymptotic upper and lower bounds respectively, while $\Theta(\cdot)$ denotes an asymptotically *tight* bound. We will make use of these notations throughout this work to denote the asymptotic scaling / behavior of the variables we present in this section.

In HLFTs the degrees of freedom are typically the discretized quantum fields (in first quantization) or the eigenstates of the number operator (in second quantization) which live on the sites, links, or plaquettes of a lattice Λ.[3] For many Hamiltonians of interest, each term in the Hamiltonian couples fields belonging to a small subset of sites [5]. Depending on which fields get coupled together, one finds different types of Hamiltonians.

We first consider k-*site local* Hamiltonians of the form

$$H = \sum_{J \in S} H_J \, ,$$
$$J = \{j_1, \ldots, j_k\} \in S \, , \quad j_i \in \Lambda \, , \tag{1}$$

where each H_J acts nontrivially only on at most k sites. The multi-index variable J, used to label each summand appearing in Eq. (1), consists of a subset (containing k elements) of the N_Λ total lattice sites. Each lattice site is comprised of n_q qubits, each term H_J acts on kn_q qubits, and the total system con-

tains $n = N_\Lambda n_q$ qubits.[4] The cardinality of the set S, which contains sets of indices used to label individual terms H_J, is also the total number of individual terms (denoted by N_H) that form the full Hamiltonian H. We assume that in general, for a k-site-local Hamiltonian, the following scaling relation holds:

$$N_H \equiv |S| = \mathcal{O}(N_\Lambda{}^m) \, , \tag{2}$$

where $m \left(\in \mathbb{R}_{\geq 0} \right) \leq k$ is a parameter fixed by the choice of a specific theory.

We note that each summand H_J can be expressed in the Pauli basis as

$$H_J = \sum_{i \in [N_P{}^{(J)}]} c_i^{(J)} P_i^{(J)} \, , \tag{3}$$

where each $P_i^{(J)} \in \mathbb{C}^{2^n \times 2^n}$ represents an n-qubit Pauli operator, each $c_i^{(J)} \in \mathbb{C} \setminus \{0\}$ is a complex coefficient, and $N_P{}^{(J)}$ enumerates the number of terms with non-zero coefficient in the Pauli decomposition of H_J. We introduce the quantity N_P as the maximum number of Pauli strings appearing in the decomposition of any single term H_J. We define it as follows:

$$N_P \equiv \max_J \left(N_P{}^{(J)} \right) . \tag{4}$$

In this way, the total number of Pauli strings appearing in a Hamiltonian with N_H summands can be upper bounded by $N_H N_P$. Note that for a k-site-local Hamiltonian $N_P = \mathcal{O}(4^{kn_q})$ always holds as an upper bound.

An important quantity in asymptotic gate complexity estimates is the 1-norm of a Hamiltonian H, denoted by $\|H\|_1$, which we define as

$$\|H\|_1 \equiv \sum_{J \in S} \|H_J\| \, , \tag{5}$$

where $\| \cdot \|$ represents the spectral / operator norm. The *induced* 1-norm of the Hamiltonian, denoted by $|||H|||_1$, can then be defined by maximizing the following sum over the qubit-index (i.e. the location within the multi-index variable $J \in S$) $\ell \in \{1, \ldots, k\}$ and the site-index $j_\ell \in \Lambda$:

$$|||H|||_1 \equiv \max_\ell \max_{j_\ell} \sum_{J_\ell^{(j_\ell)} \in S_\ell^{(j_\ell)}} \|H_J\| \, ,$$
$$J_\ell^{(j_\ell)} = \{j_1, \ldots, j_{\ell-1}, j_{\ell+1}, \ldots, j_k\} \in S_\ell^{(j_\ell)} \, , \tag{6}$$
$$J = \{j_1, \ldots, j_{\ell-1}, j_\ell, j_{\ell+1}, \ldots, j_k\} \in S \, . \quad j_i \in \Lambda \, .$$

---

[3] In some cases, the lattice is naturally embedded into some metric space, e.g., when a lattice in real or momentum space is used. More generally, it can be considered as the space which the index enumerating the degrees of freedom belongs to. Furthermore, we use the term "lattice site" for any local ingredient of the lattice that can support a Hilbert space. While for fermions and scalar fields these are the actual lattice sites, for gauge fields this terminology also covers the links and plaquettes of the lattice.

---

[4] We note that our definition of k-site-local Hamiltonian reduces to the usual definition of a k-local Hamiltonian on n-qubits [27] if we set $n_q = 1$ and $\Lambda = \{1, 2, \ldots, n\}$, in which case $N_H \sim n^k$ and $N_P \sim 4^k$.

In other words, the induced 1-norm performs the sum over all norms $\|H_J\|$ which have the lattice site j_ℓ at the ℓ'th position, both of which are chosen to maximize the sum appearing in Eq. (6). This induced norm provides an upper bound on the *strength* of the Hamiltonian at any single site due to interactions with the remaining (allowed by locality restrictions and the specific theory under consideration) sites in the lattice, such that the qubit-index of that site within the multi-index variable $J \in S$ is fixed. We denote by $\mathsf{N}_{\mathrm{ind}}$ the number of terms in the sum appearing in Eq. (6):

$$\mathsf{N}_{\mathrm{ind}} \equiv |S_{\ell^*}^{(j_{\ell^*})}|\,,$$
$$\ell^*, j_{\ell^*} = \operatorname*{argmax}_{\ell} \operatorname*{argmax}_{j_\ell} \sum_{J_\ell^{(j_\ell)} \in S_\ell^{(j_\ell)}} \|H_J\|\,. \quad (7)$$

For a general k-site-local Hamiltonians, we write

$$\mathsf{N}_{\mathrm{ind}} = \mathcal{O}\!\left(\mathsf{N}_\Lambda{}^{\mathsf{m}_{\mathrm{ind}}}\right)\,, \quad (8)$$

where $\mathsf{m}_{\mathrm{ind}}$ $(\in \mathbb{R}_{\geq 0}) \leq \mathsf{k} - 1$ is again a parameter fixed by the choice of a specific theory. We also define a quantity $\mathsf{c}_\mathsf{P} \in \mathbb{R}$ to be the maximum absolute value of the individual coefficients in the expansion of each H_J in terms of Pauli operators. Writing each summand H_J as in Eq. (3), one finds

$$\mathsf{c}_\mathsf{P} \equiv \max_{J,i}\{|c_i^{(J)}|\}\,, \quad (9)$$

so that $\|H\|_1 \leq \mathsf{c}_\mathsf{P}\mathsf{N}_\mathsf{P}\mathsf{N}_\mathsf{H}$ and $\||H|\|_1 \leq \mathsf{c}_\mathsf{P}\mathsf{N}_\mathsf{P}\mathsf{N}_{\mathrm{ind}}$.

We define the *exponentiation error* ε as the difference between the exact time evolution operator e^{-iHt} and an approximate implementation $U(t)$

$$\varepsilon \equiv \|U(t) - \mathrm{e}^{-iHt}\|\,. \quad (10)$$

We summarize these results in a few definitions, that will be used throughout this paper

Definition 2 (Site-local lattice Hamiltonian). *A k-site-local Hamiltonian on a lattice Λ of size $\mathsf{N}_\Lambda = |\Lambda|$ is one of the form:*

$$H = \sum_{J \in S} H_J\,,$$
$$J = \{j_1, \ldots, j_\mathsf{k}\} \in S\,, \quad j_i \in \Lambda\,,$$

where the number of terms in the set S are given by N_H, each each H_J acts nontrivially only on at most k sites j_i, labeled by $J = \{j_1, \ldots, j_\mathsf{k}\}$, and each lattice site is comprised of n_q qubits. Let each term H_J take the form (3) and, as in (4), let N_P be an upper bound on the number of distinct Pauli strings appearing in the decomposition of any single term H_J. Let c_P be the largest magnitude of any Pauli coefficient over all terms as in (9), and $\mathsf{N}_{\mathrm{ind}}$ be the number of terms in the induced 1-norm of the Hamiltonian as in Eq. (7).

Many HLFT Hamiltonians have the property that they only couple fields at neighboring lattice locations together, giving rise to a *geometric* locality. Defining a distance metric to the different lattice sites, allows to define a k-geometrically-site-local Hamiltonian, which is a sum of operators, each involving only sites belonging to a connected set.

Definition 3 (Geometrically-site-local lattice Hamiltonian). *A k-geometrically-site-local lattice Hamiltonian is a k-site-local lattice of Hamiltonian in which each individual term H_J acts on a connected set of sites J.*[5]

The restriction of geometric locality implies that each term H_J acts upon belong to a ball of diameter k, which imposes a tighter bound on $\mathsf{N}_{\mathrm{ind}}$:[6]

$$\mathsf{N}_{\mathrm{ind}} \leq \binom{\mathsf{k}^\mathsf{d}}{\mathsf{k}} = \mathcal{O}(1)\,, \quad (11)$$

i.e., $\mathsf{N}_{\mathrm{ind}}$ no longer scales with N_Λ (note that here we have assumed a constant value for the spatial dimension d). Furthermore, since each set J can be labeled by the center of the corresponding ball, the number of sets and hence terms in the Hamiltonian is given by

$$\mathsf{N}_\mathsf{H} = \mathcal{O}(\mathsf{N}_\Lambda)\,. \quad (12)$$

Hamiltonians of relativistic quantum field theories typically depend on the values of fields and their derivatives at a given point and thus one can expect that lattice formulations of these theories exist such that they are geometrically local. This is the case for the scalar field theory considered later in this paper for example. For gauge theories, gauge invariance needs to be included into the considerations, and the locality of the Hamiltonian now depends on what basis is chosen for the fields. In so-called electric bases, Gauss's law constraints are still local, such that the final Hamiltonian is geometrically local in this basis. On the contrary, magnetic bases, behaving better at small bare coupling (which is required in the continuum limit), require non-geometrically local interactions [41–46].

As we will also consider Hamiltonians that act only on $\mathcal{O}(1)$ number of lattice sites, we will also provide the definition for them:

---

[5] One typically assumes that the lattice is embedded into some metric space. In our work, we assume a hypercubic lattice with spacing 1. In principle, one could define locality without introducing a metric structure, e.g., by using a more general topology on S.

[6] The precise upper bound in Eq. (11) is given for a hypercubic lattice, but statement on independence of $\mathsf{N}_{\mathrm{ind}}$ on N_Λ is general.

Λ	Lattice
$N_\Lambda = |\Lambda|$	Lattice size
J	Multi-index enumerating Hamiltonian terms
$H = \sum_{J\in S} H_J$	Hamiltonian
k	Hamiltonian site-locality
H_J	Term acting on k sites
S	Set containing sets of indices J
$N_H = |S|$	Number of summands forming H
n_q	Number of qubits per lattice site
$n = N_\Lambda n_q$	Total number of qubits
$N_P^{(J)}$	Number of Pauli terms representing a single term H_J
N_P	$\max_J(N_P^{(J)})$
c_P	Maximum Pauli coefficient in H
D	Hamiltonian geometric locality
d	Spatial dimension
d	Degree of Chebyshev polynomial
ε	Exponentiation error
m	Parameter describing the scaling of N_H with respect to N_Λ, see Eq. (2)
m_{ind}	Parameter describing the scaling of N_{ind} with respect to N_Λ, see Eq. (8)
χ	Total number of elementary gates

Table I. Notations used throughout the paper.

Definition 4. [$\mathcal{O}(1)$ lattice Hamiltonian] *Let H be a ($k = N_\Lambda = \mathcal{O}(1)$)-site-local Hamiltonian, with N_P and c_P defined as in Eqs. (4) and (9). Note that $N_\Lambda = \mathcal{O}(1)$ also implies $N_H = \mathcal{O}(1)$, and that there is no difference between a site-local and geometrically site-local Hamiltonian in this case.*

C. Main results

The main results of this paper are summarized in Tables II to IV. In this section we summarize only the results, while the detailed derivations of these expressions will be given in Sec. III.

We first consider the results for a lattice containing $\mathcal{O}(1)$ sites, and compare the implementation of the time evolution operator after a *Product Formula* (PF) based splitting to the implementation using Quantum Signal processing (QSP). Both of those techniques will be introduced and defined in Sec. II. The important parameters that the time evolution operator depends on are the time t the system is evolved for, the exponentiation error ε, as well as the maximum number of Pauli terms N_P and the maximum coefficient c_P appearing in the Pauli decomposition of any single term in the Hamiltonian. The scaling with respect to these parameters is shown in Table II, where p denotes the order of the PF

used. While the scaling of the PF implementation on all parameters improves as p is increased, the number of gates needed is typically exponential in p, so the order of the PF is typically chosen as $p = \mathcal{O}(1)$. One can therefore see that QSP has a better scaling than PFs in all parameters, with the most dramatic improvement being in the scaling in ε, which is exponentially better.

Next, we include the asymptotic scaling with the size of the lattice N_Λ for a k-site local Hamiltonian. The Hamiltonian is a sum of terms H_J, each of which acts on $k = \mathcal{O}(1)$ lattice sites and in general do not commute with one another. One approach is to assemble the terms H_J using a PF, with the implementation of each H_J performed either using a PF or QSP, as just discussed. These two choices are compared in the first column of Table III. One can see that the scaling is more favorable in all parameters when the individual terms H_J are themselves implemented using a PF.

A second approach is to use QSP for the entire Hamiltonian, with the corresponding scaling shown in the second column of Table III. One can see that this leads to worse scaling (in general) with the number of lattice sites than the PF, while giving a much better scaling with ε. Therefore, the usage of PF may be preferential in situations when $N_H \gg 1/\varepsilon$, which is often the case in field theory.

Local implementation	Complexity for $\mathcal{O}(1)$ sites
Product formula	$\mathcal{O}\left[(N_P)^{2+\frac{1}{p}}(c_P t)^{1+\frac{1}{p}}\varepsilon^{-\frac{1}{p}}\right]$
QSP	$\mathcal{O}\left[N_P \log N_P \left(N_P c_P t + \log(1/\varepsilon)\right)\right]$

Table II. Scaling of a Hamiltonian defined on $\mathcal{O}(1)$ lattice sites. The scaling is given in terms of the number of Pauli strings N_P of a single term in the Hamiltonian Eq. (4), the largest coefficient of each of these Pauli strings c_P Eq. (9), and the exponentiation error ε, defined in Eq. (10).

Finally, we consider the case of geometrically-local Hamiltonians. In this case one still has the same two possibilities for the implementation, shown in the first two columns of Table IV. Both have an improved scaling in the number of lattice sites N_Λ due to the geometric locality, and as before the QSP approach wins in terms of dependence on ε while the PF-based wins in terms of scaling with N_Λ. The geometric locality of the Hamiltonian, however, allows for the use of the HHKL algorithm, introduced in Sec. II C, to assemble the H_J terms, with each local term again implemented via either PF or QSP. The scaling of this approach is shown in the third column of Table IV. We can see that this leads to a better scaling in N_Λ, with a combination of HHKL with QSP

leading to the best overall scaling. The scaling of course does not give any information on the prefactor, that could be considerably larger for HHKL than for product based approaches [26]. Furthermore, the scaling given here is not tight, so it could be that for realistic scenarios product based formulas still outperform HHKL based methods. Note that the scaling of PFs is identical to that of HHKL combined with PFs.

As a case study, we consider a scalar lattice field theory and express the obtained asymptotic gate complexities in terms of model parameters. This requires specifying the BE used, and we consider two methods for constructing the BE subroutine U_H in the scalar lattice field theory. The first method uses the standard Linear Combination of Unitaries (LCU) algorithm [35] to prepare a BE of the Hamiltonian H of the entire system; this approach is efficient for arbitrary local Hamiltonians. The second method improves upon the first by using the Fourier Transform to switch between the eigenbases of position and momentum operators, which exponentially improves the asymptotic cost of constructing each U_{H_J} in terms of the number of qubits n_q per site. The BE of the full Hamiltonian is then constructed using LCU to combine each U_{H_J}. The comparisons of costs for these approaches is shown in Fig. 9. Finally, we use the best available construction of BE in order to compare the gate counts between the PF and QSP approaches in application to a single site of the φ^4 theory, see Fig. 10.

II. REVIEW OF SIMULATION ALGORITHMS

In this section, we review some algorithms commonly used in the literature for simulating the time evolution of generic quantum systems. We will focus on algorithms based on Product Formulas (PFs) [26, 27] and Quantum Signal Processing (QSP) [29, 30], both of which can be applied to small and spatially extended systems. We also review the HHKL algorithm [34], which may serve as a "nearly-optimal wrapper" of local evolution operators in geometrically-local system. These algorithms will serve as the building blocks from which we express algorithms for Hamiltonian lattice field theories in later chapters.

The discussion in this section aims to be a standalone discussion that gives the reader a basic understanding of the different tools used to implement a time evolution operator. While we aim to make this section accessible to readers without much background in the topic, we will still provide a precise set of statements and theorems that will be used later.

Note that this section is not detailed enough to re-derive all of these results, and for this we refer the reader to the papers cited in this section.

Note also that this discussion is kept very general, without restricting ourselves to the Hamiltonians considered in this work (except when stated explicitly otherwise).

A. Time evolution by Product Formulas

We begin by considering a time-independent Hamiltonian

$$H = \sum_{\gamma=1}^{\Gamma} H_\gamma \qquad (13)$$

comprised of Γ summands, such that one can efficiently construct quantum circuits to implement the exponential $\mathrm{e}^{-iH_\gamma t}$ of each summand appearing on the RHS of Eq. (13). The unitary operator (called the time evolution operator for H) given by the exponential e^{-iHt} describes evolution under the full Hamiltonian H for some arbitrary time t. PFs allow us to construct an approximation to the full time evolution operator by decomposing it into a product of exponentials, each involving a single summand. The general form of a PF (see Ref. [27]) is given by

$$\mathrm{e}^{-iHt} \approx S(t) = \prod_{v=1}^{\Upsilon} \mathcal{P}_v \left(\prod_{\gamma=1}^{\Gamma} \mathrm{e}^{-ita_v^{(\gamma)} H_\gamma} \right), \qquad (14)$$

such that each $a_v^{(\gamma)} \in \mathbb{R}$ and $\sum_{v=1}^{\Upsilon} a_v^{(\gamma)} = 1 \ \forall \gamma \in [\Gamma] \equiv \{1, 2, \ldots, \Gamma\}$. The latter condition ensures that each individual term H_γ is evolved for the total time t. The parameter Υ determines the total number of stages in the PF, and for each stage $v \in [\Upsilon]$ there exists a corresponding permutation operator $\mathcal{P}_v$ that determines the ordering of the individual exponentials within that stage (the specific actions of the operators $\mathcal{P}_v$ depend on the chosen PF construction). PFs of the form Eq. (14) provide good approximations to the full time evolution operator when the evolution time t is small, and in particular one defines a pth order PF $S_p(t)$ as one that is correct up to p^{th} order in t

$$S_p(t) = \mathrm{e}^{-iHt} + \mathcal{O}(t^{p+1}). \qquad (15)$$

The simplest example of a PF arises from the use of the Baker-Campbell-Hausdorff formula

$$\mathrm{e}^{A+B} = \mathrm{e}^A \mathrm{e}^B + \mathcal{O}([A, B]), \qquad (16)$$

and thus we can write

$$\mathrm{e}^{-iHt} \approx S_1(t) = \prod_{\gamma=1}^{\Gamma} \mathrm{e}^{-iH_\gamma t}. \qquad (17)$$

Local implementation	Assembly using PF	QSP for entire system
Product formula	a) $\mathcal{O}\left(N_{ind} N_P \left(N_H N_P c_P t\right)^{1+\frac{1}{p}} \varepsilon^{-\frac{1}{p}}\right)$	
QSP	b) $\mathcal{O}\left[N_{ind} \left(N_H N_P c_P t\right)^{1+\frac{1}{p}} \varepsilon^{-\frac{1}{p}} N_P \times \log N_P \log(N_{ind} N_H N_P c_P t/\varepsilon)\right]$	c) $\mathcal{O}\left[N_H N_P \log(N_H N_P)\left(N_H N_P c_P t + \log(1/\varepsilon)\right)\right]$

Table III. Resource scaling for site-local Hamiltonians as in Definition 2. The two rows label the local implementation for each term $\mathrm{e}^{-iH_J t}$ either using a PF (top row) or using QSP (bottom row). The two columns show how the various terms H_J are combined together, either using a PF (left column) or doing the whole evolution using QSP (bottom right). Cases a), b), and c) are studied in Sec. III A 1, Sec. III C, and Sec. III B 1, correspondingly.

Local implementation	Assembly using PF	QSP for entire system	Assembly using HHKL
Product formula	d) $\mathcal{O}\left(N_P \left(N_\Lambda N_P c_P t\right)^{1+\frac{1}{p}} \varepsilon^{-\frac{1}{p}}\right)$		e) $\mathcal{O}\left(N_P \left(N_\Lambda N_P c_P t\right)^{1+\frac{1}{p}} \varepsilon^{-\frac{1}{p}}\right)$
QSP	f) $\mathcal{O}\left[\left(N_\Lambda N_P c_P t\right)^{1+\frac{1}{p}} \varepsilon^{-\frac{1}{p}} N_P \times \log N_P \log(N_\Lambda N_P c_P t/\varepsilon)\right]$	g) $\mathcal{O}\left[N_\Lambda N_P \log(N_\Lambda N_P) \times \left(N_\Lambda N_P c_P t + \log(1/\varepsilon)\right)\right]$	h) $\mathcal{O}\left[N_\Lambda N_P^2 c_P t \left[\log(N_\Lambda t/\varepsilon)\right]^d \times \log(N_P \log(N_\Lambda t/\varepsilon))\right]$

Table IV. Resource scaling for geometrically site-local Hamiltonians as in Definition 2. The two rows label the local implementation for each term $\mathrm{e}^{-iH_J t}$ either using a PF (top row) or using QSP (bottom row). The three columns show how the various terms H_J are combined together, either using a PF (left column), using the HHKL algorithm (right column), or doing the whole evolution using QSP (bottom middle). Cases d), f), and g) are obtained from cases a), b), and c), respectively, in Table III upon setting $N_H = \mathcal{O}(N_\Lambda)$ and $N_{ind} = \mathcal{O}(1)$. Cases e) and h) are studied in Sec. III D.

This approximation is known as the first order Lie-Trotter formula. The higher (even) order Suzuki-Trotter formulas are defined recursively as

$$S_2(t) := \prod_{i=1}^{\Upsilon} \mathrm{e}^{\frac{-iH_i t}{2}} \prod_{i=\Upsilon}^{1} \mathrm{e}^{\frac{-iH_i t}{2}},$$
$$S_{2k}(t) := S_{2k-2}(u_k t)^2 S_{2k-2}((1-4u_k)t) \times S_{2k-2}(u_k t)^2, \tag{18}$$

where $u_k \equiv 1/\left(4 - 4^{1/(2k-1)}\right)$, and the product given by $\prod_{i=\Upsilon}^{1}(\cdot)$ corresponds to reversing the order in which we choose the variable i with respect to the first product given by $\prod_{i=1}^{\Upsilon}(\cdot)$. In theory, one could use such a construction to build arbitrarily large order PFs. However the number of stages Υ, and hence the number of elementary gate operations required to construct a circuit implementing such a PF, grows exponentially with the order p (see [27]). For this reason, we use relatively low order formulas for practical applications and our results assume that $p, \Upsilon(p) = \mathcal{O}(1)$. Note that Eq. (18) is only one possible form of Eq. (14), and investigating alternatives to it is an active area of research [47–53].

Since PFs have corrections that depend polynomially on the evolution time t, they give good approximations for small t. Given a PF valid for small t, however, we can obtain one for longer times. The general procedure is to divide the full evolution time t into r (known as the *Trotter number*) smaller time steps, and apply the chosen p^{th} order PF to each step, describing the evolution for a smaller time t/r, before multiplying together the results from the individual steps

$$S_p(t) = (S_p(t/r))^r, \tag{19}$$

and the associated exponentation error ε is given by

$$\varepsilon = \|(S_p(t/r))^r - \mathrm{e}^{-iHt}\|. \tag{20}$$

To work out the resource requirements to satisfy such an error bound, we ask for the required Trotter number r for a pth order formula to satisfy Eq. (20) given a fixed value of ε. In general the answer depends on the asymptotic scaling on the evolution time t, as well as the norms of the individual summands H_γ and the commutators amongst them.

While Eq. (15) provides the correct asymptotic dependence of the error on t, [27] and [54] present bounds that show explicit dependence on the 1-norm of the full Hamiltonian (given by $\sum_{\gamma=1}^{\Gamma} \|H_\gamma\|$). Note however, that these bounds are not tight. For example, when all the summands H_γ commute, the exponentiation error ε becomes zero while these error bounds can become arbitrarily large. This was further addressed in [27], where the authors present another (tighter) bound that results from exploiting the commutativity properties of the summands H_γ. This led to the following asymptotic scaling of the

Trotter number

$$r = \mathcal{O}\left(\frac{\tilde{\alpha}^{\frac{1}{p}} t^{1+\frac{1}{p}}}{\varepsilon^{\frac{1}{p}}}\right), \qquad (21)$$

where $\tilde{\alpha} \equiv \sum_{\gamma_1, \gamma_2, ..., \gamma_{p+1}=1}^{\Gamma} ||[H_{\gamma_{p+1}}, ...[H_{\gamma_2}, H_{\gamma_1}]]||$.

Note that not every Trotter number r satisfying the asymptotic upper bound (21) will yield the desired error bound. Rather, the above result merely provides an asymptotic upper bound on the *minimum* Trotter number r needed.

B. Time evolution by Qubitization

While most applications to date simulating relativistic quantum field theories use PFs as discussed in Sec. II A, there is an alternate approach for simulating the time evolution of quantum systems that is based on a very general technique called *Quantum Signal Processing* (QSP) [29]. As described in more detail in Sec. Sec. II B 4, QSP can be used to apply broad classes of matrix functions of the Hamiltonian to a quantum state.

We begin by clarifying the terminology as confusion often arises in the literature. QSP is an algorithm which, given access to a circuit implementing a unitary operator W with eigenvalues $\{e^{i\theta_\lambda}\}$, constructs a circuit implementing a unitary V with eigenvalues $\{e^{ih(\theta_\lambda)}\}$, for a wide class of polynomials h. In the context of quantum simulation the unitary W is taken to be the so-called *Szegedy quantum walk operator* W_H associated with the Hamiltonian H. The quantum walk operator W_H is a particular type of Hamiltonian *block encoding* operator, which can be constructed from more general block encodings U_H by means of the *Qubitization* [30] procedure.

Confusingly, the term *Qubitization* is also often used for the entire process of simulating time evolution based on the usage of QSP and calls to W_H. Moreover, sometimes *Qubitization* serves as an umbrella term for all the simulation techniques which are not based on product formulæand/or rely on some form of block encoding. On top of that, *Qubitization* has also been used in a completely unrelated context, for denoting the process of mapping physical degrees of freedom onto qubits [55, 56]. In this work, *Qubitization* will be used in the first, restricted sense, referring to the process of constructing W_H.

There are two main motivations for studying QSP-based quantum simulation algorithms. First, they have a better asymptotic complexity than PFs [29, 30]. Second, they are compatible with a wider class of Hamiltonians than those which can be efficiently mapped onto Pauli Hamiltonians, and for which the number of qubits is sublinear in the number of degrees of freedom in the system [38, 39, 57].

As with PFs, the aim of this section is a pedagogical overview of techniques based on QSP, however we do provide precise statements of the theorems needed in the rest of the paper. We will rely on results from the literature to provide proofs of these theorems.

Simulating the time evolution operator of a given Hamiltonian using QSP relies on a a few key insights, which we will now review in turn.
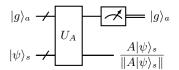
1. Idea of block encoding

Figure 1. Circuit implementing the block encoding U_A of an operator A, as defined in Eq. (22). If the ancillary register is measured in the state $|g\rangle_a$, the state of the system is $A|\psi\rangle_s/\|A|\psi\rangle_s\|$.

Block encoding (BE) is a technique that allows one to implement the action of a given operator A on an arbitrary quantum state directly in a quantum circuit, despite this operator not being unitary. This can be accomplished by rescaling A and padding it with extra entries to fit it into a unitary matrix that can act simultaneously on the given quantum state together with an appropriate number of ancilla qubits. Schematically, this means that the operator A, which acts on a system Hilbert space $\mathcal{H}_s$, is embedded into a unitary matrix acting on a larger Hilbert space $\mathcal{H}_a \otimes \mathcal{H}_s$, where $\mathcal{H}_a$ is the Hilbert space of the ancillary qubits. Schematically, U_A is given by

$$U_A = \begin{pmatrix} A/\alpha & * \\ * & * \end{pmatrix}. \qquad (22)$$

The rescaling by the parameter $\alpha > 0$, known as *scale factor*, is required, since the norm of any sub-block of a unitary matrix has to be less than unity. Equation (22) therefore implies that U_A is, in fact, a BE for the entire equivalence class of matrices A differing by a constant factor.

If we define the state $|g\rangle_a = G|0\rangle_a$ as the state that projects out the block of U_A containing the operator A/α, we can write

$$A/\alpha = (\langle g|_a \otimes \mathbb{1}_s) U_A (|g\rangle_a \otimes \mathbb{1}_s), \qquad (23)$$
$$U_A |g\rangle_a |\psi\rangle_s = |g\rangle_a (A/\alpha |\psi\rangle_s) + |\perp\rangle_{as}, \qquad (24)$$

where $|\perp\rangle_{as}$ is a vector perpendicular to $|g\rangle_a$. This implies that A is only applied to $|\psi\rangle_s$ iff the ancillary register is measured in $|g\rangle_a$. We shall term the probability of this to happen the "success probability of

the block encoding", which is given by $||A|\psi\rangle_s||/\alpha$. In the commonly considered case $|g\rangle_a = |0\rangle_a$ one can schematically write

$$|0\rangle_a|\psi\rangle_s = \begin{pmatrix} |\psi\rangle_s \\ 0 \end{pmatrix}, \tag{25}$$

$$U_A|0\rangle_a|\psi\rangle_s = \begin{pmatrix} (A/\alpha)|\psi\rangle_s \\ * \end{pmatrix}. \tag{26}$$

The success probability depends on how exactly U_A is constructed. In order for the algorithm to be considered efficient, the success probability has to scale inverse polynomially with problem parameters. The scale factor plays its role in defining the complexity of the simulation algorithm, as will be discussed later.

In much of this work we will use this technique to block encode a Hamiltonian H, and we call the corresponding BE U_H. The scale factor can then be absorbed by evolving a rescaled Hamiltonian

$$\tilde{H} = H/\alpha, \tag{27}$$

for the time

$$\tilde{t} = \alpha \, t \,, \tag{28}$$

so that

$$Ht = \tilde{H}\tilde{t} \,. \tag{29}$$

In the following discussion, we will choose a diagonal basis for the (rescaled) system Hamiltonian, such that basis vectors of the system Hilbert space are eigenstates of the Hamiltonian

$$\tilde{H}|\lambda\rangle_s = \lambda|\lambda\rangle_s \,, \tag{30}$$

with $|\lambda| < 1$. This basis is used only to simplify the discussion of the technique. Being able to diagonalize the Hamiltonian is not required for its use.

As already discussed, the vector $|g\rangle_a$ of the ancillary Hilbert space $\mathcal{H}_a$ defines the block in which the Hamiltonian is encoded in the unitary matrix U_H. In other words, the BE acts on the tensor product of a state $|\lambda\rangle_s$ and $|g\rangle_a$ as

$$U_H|g\rangle_a|\lambda\rangle_s = \lambda\,|g\rangle_a|\lambda\rangle_s + \sqrt{1-\lambda^2}|\perp^\lambda\rangle_{as} \,, \tag{31}$$

where we have defined a normalized transverse state that is orthogonal to the vector $|g\rangle_a$, such that $((\langle g|_a \otimes \mathbb{1}_s)|\perp\rangle_{as} = 0$. Thus, the action of $\tilde{H}$ can be extracted by acting with U_H on an enlarged Hilbert space and post-selecting on the ancilla qubit being in the $|g\rangle_a$ state, i.e.

$$\langle g|_a U_H|g\rangle_a|\lambda\rangle_s = \lambda\,|\lambda\rangle_s \,. \tag{32}$$

Note that if the ancillary Hilbert space $\mathcal{H}_a$ is 2-dimensional, one could choose $|g\rangle_a = |0\rangle_a$ and $|\perp\rangle_a = |1\rangle_a$.

Many different techniques for BE have been discussed in the literature. Among many others, one can use general algorithms such as the FABLE algorithm [58], approaches that rely on a technique called linear combination of unitaries (LCU) [35, 59], a technique called QETU [60], and techniques using access to the matrix elements of the sparse matrix and compact mappings H [30, 36, 38, 39, 59, 61]. The choice of BE is typically dictated by the number of ancillary qubits required, the value of the scale factor α, the gate complexity required for its implementation, whether an approximate implementation of the Hamiltonian is sufficient, and whether it is applicable to general sparse Hamiltonians.

One issue with BEs discussed so far is that the action of the BE does not stay in the Hilbert space spanned by span$\{|g\rangle_a|\lambda\rangle_s, |\perp^\lambda\rangle_{as}\}$. In other words, the action of U_H on $|\perp^\lambda\rangle_{as}$ can produce a state with different value of $\lambda' \neq \lambda$, such that in general

$$\langle g|_a\langle\lambda'|_s U_H|g\rangle_a|\perp^\lambda\rangle_{as} \neq 0$$
$$\langle\perp^{\lambda'}|_{as} U_H|\perp^\lambda\rangle_{as} \neq 0 \,. \tag{33}$$

Therefore, repeated actions of the unitary U_H are not easily known. This can be dealt with by constructing a more constrained BE W_H, for which one can ensure that multiple actions of this BE on a given eigenstate will never produce a state with a different eigenvalue. This constrained BE is called a *qubitized block encoding* and is discussed in Sec. II B 3.

2. Block encoding by LCU

A generic subroutine, called Linear Combination of Unitaries (LCU), can be used as a general-purpose technique for constructing BEs of qubit Hamiltonians. After introducing the general technique, we briefly discuss several modifications of LCU as well as alternative approaches to constructing BEs.

Consider the situation were one has a combination of M unitary operators U_0, ..., U_{M-1}, each acting on a system Hilbert space $\mathcal{H}_s$ and the goal is to construct the operator

$$T = \sum_{i=0}^{M-1} \beta_i \, U_i \,, \tag{34}$$

acting on the same system Hilbert space.[7][8]

---

[7] We assume $\beta_i > 0$ since the complex phases of β_i can be absorbed by U_i.
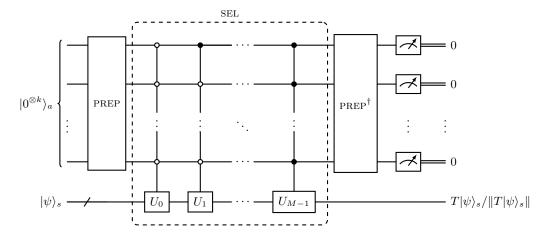[8] This technique can also be used to add non-unitary op-

Figure 2. Linear Combination of Unitaries (LCU) implements the BE U_T of an operator T expressed as a sum of elementary operators with known circuit implementations, reproduced from [35]. Each operator U_i is controlled on the binary string representing its index i. The size of the ancillary register is $k = \lceil \log_2 M \rceil$. For illustration purposes, $\log_2(M)$ is assumed to be an integer k such that the binary string of $M - 1$ contains all 1's.

This can be achieved in the following way. First, consider an ancilla register with Hilbert space $\mathcal{H}_a$ containing $k = \lceil \log M \rceil$ qubits. Labeling the computational basis of these states through the integers corresponding to the binary string encoded in the qubits, one defines a so-called *prepare oracle* (PREP) to create a state $|\beta\rangle$:

$$|\beta\rangle_a = \mathrm{PREP}_a|0\rangle_a = \frac{1}{\sqrt{\beta}} \sum_{i=0}^{M-1} \sqrt{\beta_i}|i\rangle_a, \qquad (35)$$

where $|0\rangle_a$ corresponds to the state with all qubits in $\mathcal{H}_a$ in the $|0\rangle_a$ state and we have defined $\beta \equiv \|\vec{\beta}\|_1 = \sum_i |\beta_i|$.

Next, using the so-called *select oracle* (SEL), one performs operations U_i on states in $\mathcal{H}_s$, controlled on the values of qubits in $\mathcal{H}_a$

$$\mathrm{SEL} = \sum_{i=0}^{M-1} |i\rangle_a\langle i|_a \otimes U_i. \qquad (36)$$

At the end one reverses the prepare oracle. The complete circuit on the Hilbert space $\mathcal{H}_{as} = \mathcal{H}_a \otimes \mathcal{H}_s$ can therefore be written as

$$U_T = \left(\mathrm{PREP}_a^\dagger \otimes \mathbb{1}_s\right) \mathrm{SEL} \left(\mathrm{PREP}_a \otimes \mathbb{1}_s\right). \qquad (37)$$

Letting this operator act on a state $|0\rangle_a|\psi\rangle_s$ one finds

$$U_T|0\rangle_a|\psi\rangle_s = \frac{1}{\beta}|0\rangle_a T|\psi\rangle_s + |\perp\rangle_{as}, \qquad (38)$$

———————

erators by first block-encoding them and then adding the unitaries using LCU.

where $|\perp\rangle$ is an unnormalized state transverse to $(|0\rangle_a \otimes \mathbb{1}_s)$. This implies that U_T is a BE of T with a scale factor β.

From the above discussion it is clear that the number of ancilla qubits to combine M unitaries is given by $\dim(\mathcal{H}_a) \sim \log M$, and the number of gates is given by M gates that are controlled on $\log M$ qubits, which can be reduced to $\mathcal{O}(M \log M)$ elementary gates.

While the presented approach to LCU relies on the binary encoding of terms on the RHS of Eq. (34) in the ancillary register, other encodings can be used as well. In particular, a variation of the LCU algorithm exists, which takes advantage of the Hamiltonian locality [62]. For an k-local $\mathsf{n_q}$-qubit Hamiltonian, it requires $2\mathsf{n_q}$ ancillary qubits yet only uses $2k$-controlled gates.

The LCU algorithm can be applied to a Hamiltonian provided in a form of a $2^{\mathsf{n_q}} \times 2^{\mathsf{n_q}}$ matrix, upon decomposing it into Pauli strings as $H = \sum_{i=0}^{4^{\mathsf{n_q}}-1} \beta_i P_i$, where P_i are the Pauli strings and the coefficients c_i are found as $c_i = \frac{1}{2^{\mathsf{n_q}}} \mathrm{Tr}(H P_i)$. An alternative approach to constructing BEs for Hamiltonians given in the matrix form is considered in [58].

The LCU approach is often combined with other methods. In Sec. IV C 2 we shall consider its modification which takes into account the structure of particular lattice Hamiltonians.

Lastly, we briefly mention that simulation techniques based on BE, considered in the next section, are compatible with a wide class of *sparse* Hamiltonians, including those which cannot be efficiently mapped onto Pauli strings. A general approach to constructing BEs for such systems is based on the usage of *sparse oracle* subroutines. As was mentioned in Sec. I, their construction is highly problem-

specific [36–40] and does not seem to provide advantage over LCU for local HLFTs [40], which are considered the main application of this work.

3. Qubitization

In approaches to quantum simulation based on Quantum Signal Processing (QSP), one approximates the time evolution operator e^{-iHt} by constructing polynomials of the Hamiltonian operator. In what follows we shall base our algorithms upon the Qubitization technique [30], which utilizes a special *qubitized* form of block encoding W_H, also known as *Szegedy quantum walk operator* or the *iterate*).[9]

The qubitized block encoding W_H is designed so that its successive applications to states of the form $|g\rangle_a|\lambda\rangle_s$ produce a state belonging to the subspace $\mathrm{span}\{|g\rangle_a|\lambda\rangle_s, |\perp^\lambda\rangle_{as}\}$, for all eigenstates $|\lambda\rangle$. This fact implies that repeated applications of W_H produce block encodings of powers of H, which would not be true for U_H. Such a scenario is realized iff each state $|\perp^\lambda\rangle_{as}$ factorizes as $|\perp^\lambda\rangle_{as} \equiv |\perp^\lambda\rangle_a|\lambda\rangle_s$:

$$W_H|g\rangle_a|\lambda\rangle_s = \lambda|g\rangle_a|\lambda\rangle_s - \sqrt{1-\lambda^2}|\perp^\lambda\rangle_{as}$$
$$= \left(\lambda|g\rangle_a - \sqrt{1-\lambda^2}|\perp^\lambda\rangle_a\right)|\lambda\rangle_s. \quad (39)$$

Note that the minus sign between the two terms is conventional.

For any dimension of $\mathcal{H}_a$, the orthogonal state $|\perp^\lambda\rangle_a$ can now be defined as

$$|\perp^\lambda\rangle_a|\lambda\rangle_s = \frac{\lambda\mathbb{1}_{as} - W_H}{\sqrt{1-\lambda^2}}|g\rangle_a|\lambda\rangle_s, \quad (40)$$

which implies that for each eigenvalue λ of the Hamiltonian there is one specific transverse vector $|\perp^\lambda\rangle_a$ in $\mathcal{H}_a$.

From Eq. (39) it follows that the operator W_H is block diagonal (in the sense that it doesn't mix the Hamiltonian eigenstates), and can be written as

$$\left(\mathbb{1}_a \otimes \langle\lambda'|_s\right) W_H \left(\mathbb{1}_a \otimes |\lambda\rangle_s\right) = \widetilde{W}_H^{(\lambda)}\delta_{\lambda\lambda'}, \quad (41)$$

---

[9] A different QSP-based approach to approximating e^{-iHt} amounts to using Quantum Singular Value Transformation [59, 63]. This method is advantageous in the sense that it avoids controlled calls to the block encoding, and uses directly U_H instead of W_H. However, as such an approach does not allow one to simultaneously implement both even and odd parts of e^{-iHt}, it requires adding them using LCU at the final stage. This, in turn, leads to success probabilities which (even upon potential amplitude amplification) cannot be made exponentially close to 1, as required by the HHKL algorithm.

where $\widetilde{W}_H^{(\lambda)}$ is some operator acting on the register a. Information relevant to block encoding of H is contained in two-dimensional blocks of $\widetilde{W}_H^{(\lambda)}$ corresponding to subspaces

$$\hbar_a^{(\lambda)} = \mathrm{span}\{|g\rangle_a, |\perp^\lambda\rangle_a\} \in \mathcal{H}_a, \quad (42)$$

which we denote by $W_H^{(\lambda)}$:

$$W_H^{(\lambda)} = \begin{pmatrix} \langle g|_a\widetilde{W}_H^{(\lambda)}|g\rangle_a & \langle g|_a\widetilde{W}_H^{(\lambda)}|\perp^\lambda\rangle_a \\ \langle\perp^\lambda|_a\widetilde{W}_H^{(\lambda)}|g\rangle_a & \langle\perp^\lambda|_a\widetilde{W}_H^{(\lambda)}|\perp^\lambda\rangle_a \end{pmatrix}_{\hbar_a^{(\lambda)}}$$
$$= \begin{pmatrix} \lambda & -\sqrt{1-\lambda^2} \\ \sqrt{1-\lambda^2} & \lambda \end{pmatrix}_{\hbar_a^{(\lambda)}}$$
$$= \exp(-iY\theta_\lambda), \quad (43)$$

where

$$\theta_\lambda = \arccos(\lambda). \quad (44)$$

While the qubitized BE W_H gives the same result as the original BE U_H when acted on $|g\rangle_a|\lambda\rangle_s$, it has a much simpler structure when applied multiple times:

$$\left(\mathbb{1}_a \otimes \langle\lambda'|_s\right)(W_H)^k\left(\mathbb{1}_a \otimes |\lambda\rangle_s\right) = \left(\widetilde{W}_H^{(\lambda)}\right)^k\delta_{\lambda'\lambda}. \quad (45)$$

Using Eq. (43), one can show that

$$\left(W_H^{(\lambda)}\right)^k$$
$$= \begin{pmatrix} \langle g|_a\left(\widetilde{W}_H^{(\lambda)}\right)^k|g\rangle_a & \langle g|_a\left(\widetilde{W}_H^{(\lambda)}\right)^k|\perp^\lambda\rangle_a \\ \langle\perp^\lambda|_a\left(\widetilde{W}_H^{(\lambda)}\right)^k|g\rangle_a & \langle\perp^\lambda|_a\left(\widetilde{W}_H^{(\lambda)}\right)^k|\perp^\lambda\rangle_a \end{pmatrix}_{\hbar_a^{(\lambda)}}$$
$$= \begin{pmatrix} T_k(\lambda) & -\sqrt{1-\lambda^2}\,U_{k-1}(\lambda) \\ \sqrt{1-\lambda^2}\,U_{k-1}(\lambda) & T_k(\lambda) \end{pmatrix}_{\hbar_a^{(\lambda)}}$$
$$= \exp\left(-iYk\theta_\lambda\right), \quad (46)$$

where $T_k(\lambda)$ and $U_k(\lambda)$ are Chebyshev polynomials of the first and second kind, respectively. Given that Chebyshev polynomials are an excellent polynomial basis for approximating L^∞ functions on a finite interval, one can hope to construct a general function of the eigenvalues (and therefore of the Hamiltonian matrix itself) by combining powers of the W_H operator. Note that while the matrix of $\widetilde{W}_H^{(\lambda)}$ may contain blocks of size larger than 2×2 when the ancillary register contains more than a single qubit, only its 2×2 blocks $W_H^{(\lambda)}$ are relevant for the block encodings of H and its powers. In the next section we discuss

how this technique can be used for approximating the time evolution operator.

While so far we have only defined the unitary W_H through its action on a given state, a very important result is that it can be obtained from a BE of the Hamiltonian U_H using a process known as *qubitization* [30], which we now describe. In many cases of interest, it is possible to construct the iterate by assuming a form $W_H = S'U_H$, and searching for some S' that acts only on the ancillary qubits. The conditions S' must satisfy can be stated more succinctly by first letting S' be a product of a reflector R_g about the state $|g\rangle_a$ and another operator S that acts only on the ancillary qubits, i.e.

$$W_H = \left(R_g \otimes \mathbb{1}_s\right)\left(S \otimes \mathbb{1}_s\right)U_H\,, \qquad (47)$$

where

$$R_g = \left(2|g\rangle_a\langle g|_a - \mathbb{1}_a\right)\,. \qquad (48)$$

The circuits for W_H and R_g are shown in Figs. 3 and 4.
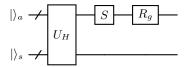
Figure 3. Circuit for the qubitized block encoding W_H defined in Eq. (47).
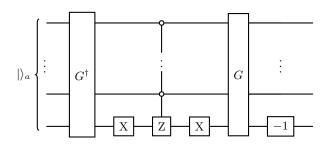

Figure 4. Circuit for the reflection operator R_a defined in Eq. (48) and used in Eq. (47), see [59] for alternative implementations. G is the oracle for preparing the state $|g\rangle_a$, i.e. $G|0\rangle_a = |g\rangle_a$. The $\boxed{-1}$ gate adds an overall phase to the circuit (negative sign), and can be implemented, e.g., as XZXZ.

Written in this way, it was shown [30] that one can construct W_H from Eq. (47) if S satisfies

$$\left(\langle g|_a \otimes \mathbb{1}_s\right)\left(S \otimes \mathbb{1}_s\right)U_H\left(|g\rangle_a \otimes \mathbb{1}_s\right) = H/\alpha, \quad (49\text{a})$$

$$\left(\langle g|_a \otimes \mathbb{1}_s\right)\left[\left(S \otimes \mathbb{1}_s\right)U_H\right]^2\left(|g\rangle_a \otimes \mathbb{1}_s\right) = \mathbb{1}_a\,. \qquad (49\text{b})$$

One particularly simple case is when U_H is its own inverse, *i.e.* $U_H^2 = \mathbb{1}_{as}$. In this case, one can simply set $S = \mathbb{1}_a$. This construction can be applied to the case when the block encoding U_H is constructed via the LCU procedure [37]. To see this, we must modify our perspective slightly. In the original construction, we had $U_H = (\text{PREP}^\dagger)\text{SEL}(\text{PREP})$ and $|g\rangle_a = |0\rangle_a$. Alternatively, we can regroup terms and instead take $U_H \to U_H^{\text{LCU}} \equiv \text{SEL}$ and $|g\rangle_a \to |g^{\text{LCU}}\rangle_a \equiv \text{PREP}|0\rangle_a = |\beta\rangle_a$. Because $(U_H^{\text{LCU}})^2 = \text{SEL}^2 = \mathbb{1}_{as}$, we can simply set $S = \mathbb{1}_a$ to construct W_H:

$$\left(\langle 0|_a \otimes \mathbb{1}_s\right)U_H\left(|0\rangle_a \otimes \mathbb{1}_s\right)$$
$$= \left(\langle 0|_a \otimes \mathbb{1}_s\right)(\text{PREP}^\dagger)\text{SEL}(\text{PREP})\left(|0\rangle_a \otimes \mathbb{1}_s\right) \quad (50)$$
$$= \left(\langle \beta|_a \otimes \mathbb{1}_s\right)U_H^{\text{LCU}}\left(|\beta\rangle_a \otimes \mathbb{1}_s\right)\,.$$

It is important to note that the BE U_H^{LCU} is defined with $|g\rangle_a = \text{PREP}|0\rangle_a$.

The construction described above also applies to cases where the block encoding is based on Hamiltonian sparsity, since one can use sparse oracles to construct block encodings that are Hermitian and thus self-inverse [59]. In more general scenarios where this is not necessarily the case (e.g., as in Ref. [64]), the operator S' can always be constructed at the expense of introducing a single ancillary qubit and performing two controlled calls to U_H [30]. Note that while this procedure does not increase the asymptotic cost of preparing W_H, it does result in a significant increase of the overall prefactor. Importantly, for all methods used in this work, the BE's are also Hermitian, and the general procedure requiring an extra qubit was avoided. For completeness, the procedure for constructing W_H from an arbitrary U_H is presented in App. A.

4. QSP using Qubitization

Having just demonstrated that repeated applications of W_H construct block encodings of Chebyshev polynomials of the Hamiltonian, one natural strategy for constructing a polynomial approximation to e^{-itH} is as follows. One would first construct circuits with different powers of W_H and then add these circuits with the appropriate coefficients using an LCU procedure. However, this would result in a block encoding approximating e^{-iHt} with a scale factor that grows with the degree of the polynomial approximation and cannot be made arbitrarily close to one. This is highly undesirable for the reasons previously discussed in Footnote 9, and below we discuss how this limitation can be overcome with the aid of QSP.

In order to use the operator W_H of the previous section to calculate the exponential of the
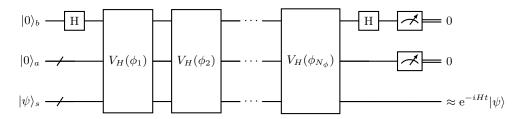
Figure 5. Quantum Signal Processing (QSP) circuit used for simulating time evolution, reproduced from [64]. The circuit prepares a block encoding for the approximation of the time evolution operator $e^{-iHt}|\psi\rangle$. Circuit for $V_H(\phi)$ is shown in Fig. 6. When all the ancillary qubits are measured in the zero states, the desired operator is applied to the system register $|\psi\rangle_s$. Without loss of generality, $|g\rangle_a$ is set to $|g\rangle_a = |0\rangle_a$.
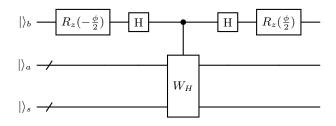

Figure 6. Circuit for $V_H(\phi)$, as defined is Eq. (54).

Hamiltonian, we first note that within the subspaces $\text{span}\{|g\rangle_a|\lambda\rangle_s, |\perp^\lambda\rangle_{as}\}$ the qubitized operator W_H can be diagonalized by defining the states

$$|\theta^\lambda_\pm\rangle_{as} \equiv \frac{1}{\sqrt{2}}\left(|g\rangle_a \pm i|\perp^\lambda\rangle_a\right)|\lambda\rangle_s . \qquad (51)$$

Using Eq. (39) one can easily show that

$$W_H|\theta^\lambda_\pm\rangle_{as} = e^{i\theta^\lambda_\pm}|\theta^\lambda_\pm\rangle_{as} , \qquad (52)$$

with

$$\theta^\lambda_\pm = \mp\theta_\lambda , \qquad (53)$$

where θ_λ is defined in Eq. (44).[10]

Given this form, one can now define a new operator $V_H(\phi)$, parameterised by ϕ, which can be constructed from the operator W_H. It requires one more ancillary qubit $|b\rangle \in \mathcal{H}_b$, such that the operator $V_H(\phi)$ acts on the Hilbert space $\mathcal{H}_{bas} \equiv \mathcal{H}_b \otimes \mathcal{H}_a \otimes \mathcal{H}_s$. It is defined by (see Fig. 6)

$$V_H(\phi) = (e^{-i\phi Z_b/2} \otimes \mathbb{1}_{as})W'_H(e^{i\phi Z_b/2} \otimes \mathbb{1}_{as}), \qquad (54)$$

_____

[10] The fact that W_H encodes information about the spectrum of the system via $\text{spec}\, W_H^{(\lambda)} = \{e^{\pm i\arccos(\lambda)}\}$ can be used for implementing a highly efficient version of Quantum Phase Estimation algorithm based using controlled calls to W_H [37].

where Z_b is the usual Pauli Z operator acting on the qubit $|b\rangle$, and W'_H is the W_H operator controlled on the qubit $|b\rangle$, which now acts on the Hilbert space $\mathcal{H}_{bas}$:

$$W'_H = |+\rangle_b\langle+|_b \otimes \mathbb{1}_{as} + |-\rangle_b\langle-|_b \otimes W_H . \qquad (55)$$

One can show through explicit computation that

$$\left(\mathbb{1}_b \otimes \langle\theta^{\lambda'}_\pm|_{as}\right)V_H(\phi)\left(\mathbb{1}_b \otimes |\theta^\lambda_\pm\rangle_{as}\right) \\ = V_H^{(\theta^\lambda_\pm)}(\phi)\,\delta_{\lambda\lambda'} , \qquad (56)$$

with

$$V_H^{(\theta)}(\phi) = e^{i\frac{\theta}{2}}\begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2}e^{-i\phi} \\ -i\sin\frac{\theta}{2}e^{i\phi} & \cos\frac{\theta}{2} \end{pmatrix}_b \\ = e^{i\frac{\theta}{2}}e^{-i\frac{\theta}{2}(X_b\cos\phi+Y_b\sin\phi)} , \qquad (57)$$

where X_b and Y_b denote the Pauli-X and Y matrices in $\mathcal{H}_b$.

What this construction has accomplished is to create a unitary operator $V_H(\phi)$ that is block diagonal in the original Hilbert space $\mathcal{H}_s$, with the entries of the 2×2 matrix in $\mathcal{H}_b$ given by functions of the eigenvalues of the Hamiltonian (through $\cos\theta^\lambda_\pm$), parameterized by some ϕ. By multiplying several of these operators in sequence (each with a different value of ϕ), one obtains an operator

$$V_H^{(\theta)}(\vec{\phi}) \equiv V_H^{(\theta)}(\phi_{N_\phi})\ldots V_H^{(\theta)}(\phi_1), \qquad (58)$$

that is still block diagonal in $\mathcal{H}_s$. The resulting operator is a 2×2 matrix in $\mathcal{H}_b$ which acquires the form of

$$V_H^{(\theta)}(\vec{\phi}) = \mathcal{A}(\theta)\mathbb{1}_b + i\mathcal{B}(\theta)X_b + i\mathcal{C}(\theta)Y_b + i\mathcal{D}(\theta)Z_b , \qquad (59)$$

where the functions $\mathcal{A}(\theta)\ldots\mathcal{D}(\theta)$ are all parameterized by the angles $\vec{\phi} = (\phi_1,\ldots,\phi_{N_\phi})$ and, similarly to Eq. (46), can be expressed in terms of Chebyshev polynomials of λ upon substituting $\theta = \arccos(\lambda)$.

The classes of functions $\mathcal{A}(\theta)\ldots\mathcal{D}(\theta)$ that can be realized by the above procedure are described by the theory of Quantum Signal Processing. In particular, it was shown in [65] that one can construct the vector of angles $\vec{\phi}$ to obtain functions $\mathcal{A}(\theta)$ and $\mathcal{C}(\theta)$ of the form

$$|\mathcal{A}^2(\theta)| + |\mathcal{C}^2(\theta)| \leq 1, \qquad \mathcal{A}(0) = 1,$$

$$\mathcal{A}(\theta) = \sum_{k=0}^{N_\phi/2} a_k \cos(k\theta),$$

$$\mathcal{C}(\theta) = \sum_{k=1}^{N_\phi/2} c_k \sin(k\theta). \tag{60}$$

The two other functions, $\mathcal{B}(\theta)$ and $\mathcal{D}(\theta)$ are not relevant to the calculation since projecting onto $|+\rangle_b$ leaves only $\mathcal{A}(\theta)$, $\mathcal{C}(\theta)$. Reversing this logic, one can start from some desired functions $\mathcal{A}(\theta)$, $\mathcal{C}(\theta)$, and determine phases $\vec{\phi}$ such that $V_H^{(\theta)}(\vec{\phi})$ reproduces these functions (approximately). This can be accomplished using an efficient classical procedure given in [65].[11]

We omit now a few final details, which can be found in [30], and simply state the relevant results. As can be confirmed from Eq. (59), letting the operator $V_H^{(\theta)}(\vec{\phi})$ act on the state $|+\rangle_b$ and post-selecting on $\langle+|_b$ picks out the function

$$\langle+|_b\langle g|_a\langle\lambda'|_s V_H(\vec{\phi})|+\rangle_b|g\rangle_a|\lambda\rangle_s$$
$$= \delta_{\lambda\lambda'}\left[A(\lambda) + iC(\lambda)\right]. \tag{61}$$

The functions $A(\lambda)$ and $C(\lambda)$ are directly related to the functions $\mathcal{A}(\theta)$, $\mathcal{C}(\theta)$, with the detailed relation given in Ref. [30]. The success probability for measuring the system in the state $|+\rangle_b|g\rangle_a$ for a given value λ depends on the functions and is given by

$$p \geq |A^2(\lambda)| + |C^2(\lambda)|. \tag{62}$$

The final task is to determine which functions $A(\lambda)$ and $C(\lambda)$ (and therefore which phases $\vec{\phi}$) approximate the time evolution operator with uncertainty bounded by ε,

$$\left|\mathrm{e}^{i\lambda\tilde{t}} - [A(\lambda) + iC(\lambda)]\right| < \varepsilon. \tag{63}$$

Since $A(\lambda)$ and $C(\lambda)$ can be expressed in terms of Chebyshev polynomials, they can be used for approximating the real and imaginary part of $\mathrm{e}^{i\lambda\tilde{t}}$

through the rapidly converging Jacobi-Anger expansion [29]:

$$A(\lambda) = J_0(t) + 2\sum_{k \text{ even}>0}^{N_\phi/2} (-1)^{k/2} J_k(\tilde{t}) T_k(\lambda), \tag{64a}$$

$$C(\lambda) = 2\sum_{k \text{ odd}>0}^{N_\phi/2} (-1)^{(k-1)/2} J_k(\tilde{t}) T_k(\lambda), \tag{64b}$$

(note that N_ϕ is always chosen to be even). In this case, the circuit in Fig. 5 would implement the approximate BE of the evolution operator e^{-iHt}. The number of phases needed for a given value of ε is bounded by [29]

$$N_\phi = \mathcal{O}\left(\tilde{t} + \log(1/\varepsilon)\right) = \mathcal{O}\left(\alpha t + \log(1/\varepsilon)\right). \tag{65}$$

Note also that Eqs. (62) and (63) imply that

$$p \geq 1 - \mathcal{O}(\varepsilon), \tag{66}$$

since the time evolution operator has unit magnitude. Note that due to the $\log(1/\epsilon)$ scaling, one can require the success probability to be exponentially close to one for only a polynomial cost (see also Footnote 9).

It is worth mentioning that the success rate of QSP is always $\mathcal{O}(1)$, independent of the success rate $\mathcal{O}(1/\alpha)$ of the BE U_H used. This is due to the fact that we have used QSP to implement a unitary function (e^{-iHt}) which has unit norm. This implies that $A(\lambda)^2 + C(\lambda)^2 \approx 1$, which together with Eq. (62) gives a success probability that is close to one. While this is a desirable feature, the cost of a poor success probability of the BE is now reflected in the number of phases N_ϕ required to approximate the time evolution operator to an error ϵ, as seen by Eq. (65). For this reason, significant savings can be achieved by performing dedicated studies to minimize α.

A number of algorithms, based on the concept of QSP, can be used for quantum simulation. Algorithms such as Quantum Singular Value Transformation (QSVT) [59, 63] and (QETU) [60, 66, 67] can be utilized for near-optimal state preparation [60, 67, 68] and construction of block encodings [69], while utilizing the recently developed Generalized QSP (GQSP) algorithm [32] can readily improve by a constant factor the gate counts obtained with the QSP procedure [69].

C. HHKL

The results presented so far (PFs and QSP) can be applied to general site-local Hamiltonians (see Definition 2). For geometrically-site-local Hamiltonians

---

[11] A recent version of QSP, *generalized QSP*, significantly improves the cost of calculating phases $\vec{\phi}$ [32].

(see Definition 3) one can use the fact that information from one area of the lattice can only spread to areas not directly connected to it through repeated actions of the time evolution operators, which implies that the speed with which this information spreads is bounded. This is formalized by the Lieb-Robinson bound [70]. This idea was used in a paper by Haah, Hastings, Kothari and Low (HHKL) [34] to devise a scheme for simulating the time evolution of geometrically-site-local Hamiltonians.

As before, this section is aimed at giving the reader a basic understanding of the technique, without going into too many of the details. In order to understand these details the reader is encouraged to read the excellent description of the algorithm in the original work, where all statements are also carefully proved.

The main ingredient to the HHKL algorithm is that the time evolution of a large quantum system with geometrically local Hamiltonian can be obtained from the time evolution of two smaller systems that share some overlap, which has to be of at least the size given by the ball of size $\mathcal{O}(1)$ used in Definition 3. Thus, the system Hilbert space $\mathcal{H}_s$ is divided into three parts, $\mathcal{H}_{a,b,c}$ such that $\mathcal{H}_s = \mathcal{H}_a \otimes \mathcal{H}_b \otimes \mathcal{H}_c$. In the discussion of [34] is assumed that the local Hamiltonian H_J such that $||H_J|| \leq 1$. Thus, in the remainder we work with

$$\tilde{H}_J = \frac{H_J}{||H_J||}\,, \qquad \tilde{t} = ||H_J||t\,. \qquad (67)$$

First, one uses that the evolution of a system for some long time $\tilde{t}$ can be split into the product of unitaries over shorter times $\Delta t = \mathcal{O}(1)$:

$$U(\tilde{t}, 0) = U(\tilde{t}, \tilde{t} - \Delta t) \cdots U(\Delta t, 0)\,. \qquad (68)$$

The total number of such unitaries is $\mathcal{O}(\tilde{t})$.

Lemma 6 of [34] states that the time evolution $U_s(\Delta t)$ of a Hamiltonian on the whole system over time Δt can be obtained by multiplying together the forward time evolution in the system $\mathcal{H}_{a\cup b} = \mathcal{H}_a \otimes \mathcal{H}_b$ and $\mathcal{H}_{b\cup c} = \mathcal{H}_b \otimes \mathcal{H}_c$ and the backwards evolution in the system $\mathcal{H}_b$ as

$$U_{a\cup b\cup c}(\Delta t) = U_{a\cup b}(\Delta t)U_b^\dagger(\Delta t)U_{b\cup c}(\Delta t) + \mathcal{O}(\delta)\,, \qquad (69)$$

where the error due to this approximation is given by

$$\delta \equiv \delta(a, b, c) = \mathcal{O}\left(\mathrm{e}^{-\mu\,\mathrm{d}(a,c)} \sum_{X\in\mathrm{BD}(ab,c)} ||\tilde{H}_X||\right)\,. \qquad (70)$$

For now we assume that each of the $U(\Delta t)$ on the RHS of Eq. (69) is implemented exactly. In Eq. (70)

$\mu > 0$ is a constant, $\mathrm{d}(a, c)$ is the smallest distance between any point in a and any point in c, and the norm $||\tilde{H}_J||$ is summed over all terms that are on the boundary of $a\cup b$ and c, indicated by $\mathrm{BD}(ab, c)$. The uncertainty is therefore exponentially suppressed in the distance between the regions a and c. This implies that this approximation works well for large enough distance $\ell \equiv \mathrm{d}(a, c)$.

This result can be applied recursively, splitting the evolution into smaller and smaller pieces, each of which just have to be larger than the ball of size $\sim \mathcal{O}(1)$. If the size of the original system is given by L^{d} (where d represents the dimension of the Euclidean space we consider with size L in each dimension) and the final pieces are each of size ℓ^{d}, one ends up with

$$m = \mathcal{O}\left(\frac{L^{\mathsf{d}}}{\ell^{\mathsf{d}}}\right) \qquad (71)$$

individual pieces. Taking into account the $\mathcal{O}(\tilde{t})$ evolution operators we started from, the final error $\mathcal{O}(\varepsilon)$ can be reached by requiring

$$\delta = \mathcal{O}\left(\frac{\varepsilon}{\tilde{t}\,m}\right)\,. \qquad (72)$$

We see that the sum of the norms of the boundary terms appearing in Eq. (70) is $\mathcal{O}(L^{\mathsf{d}-1})$ and thus from Eqs. (70) and (72) one finds it suffices to choose

$$\ell = \Theta\left(\log\left(\frac{L^{\mathsf{d}}\tilde{t}}{\varepsilon}\right)\right)\,. \qquad (73)$$

One can then estimate the required resources using the fact that one needs to compute

$$\mathcal{O}(\tilde{t}m) = \mathcal{O}\left(L^{\mathsf{d}}\tilde{t}\log^{-\mathsf{d}}\left(\frac{L^{\mathsf{d}}\tilde{t}}{\varepsilon}\right)\right) \qquad (74)$$

time evolution operators on a system of size $\mathcal{O}(\ell^{\mathsf{d}})$.[12] The cost and the associated additional error for implementing each of these "local" pieces depends on the algorithm used for their implementation. In particular, if an algorithm with cost polynomial in ℓ and polylogarithmic in $1/\varepsilon$ is used (such as Qubitization), then the cost of the HHKL algorithm remains $\mathcal{O}(L^{\mathsf{d}}\tilde{t}\,\mathrm{polylog}(L^{\mathsf{d}}\tilde{t}/\varepsilon))$. Implementing the time evolution of local blocks using PF would lead to an algorithm whose asymptotic cost dependence on ε is

————

[12] Note that decreasing ε is logarithmically decreasing the number of evolution operators. This number clearly has to be larger than unity, meaning ε cannot be exponentially small with system size.

given by that of the employed PF method, i.e., no longer polylogarithmic in ε. This can make sense if, for a given value of ε, the PF-based method leads to lower gate counts as compared to a nearly-optimal implementation – for a system of size ℓ but not of size L [26].

D. Summary and precise statement of techniques used

1. General Product Formulas

Definition 5. A pth order PF $S_p(t)$ is a PF of the form (14), such that it approximates the true exponential e^{-iHt} to p^{th} order in t, i.e.,

$$\mathrm{e}^{-iHt} = S_p(t) + \mathcal{O}(t^{p+1}). \tag{75}$$

Theorem 6 (Trotter number with commutator scaling). *Let $H = \sum_{\gamma=1}^{\Gamma} H_\gamma$ be an operator comprised of Γ Hermitian summands. Let $S_p(t)$ be a pth order PF as defined by (75), with $t \geq 0$. Then, if we require an exponentiation error $\mathcal{O}(\varepsilon)$, as defined by (20), it suffices to choose*

$$r = \mathcal{O}\left(\frac{\tilde{\alpha}^{\frac{1}{p}} t^{1+\frac{1}{p}}}{\varepsilon^{\frac{1}{p}}}\right), \tag{76}$$

where $\tilde{\alpha} \equiv \sum_{\gamma_1,\gamma_2,\ldots,\gamma_{p+1}=1}^{\Gamma} ||[H_{\gamma_{p+1}}, \ldots [H_{\gamma_2}, H_{\gamma_1}]]||$

Proof. This theorem follows directly from Theorem 11 and Corollary 12 of [27]. $\square$

2. QSP

In this section we summarize the results of the section on QSP in a concise and precise form which will be referred to later in the text.

Definition 7 (Block encoding). Let H be a Hamiltonian acting on the n-qubit Hilbert space $\mathcal{H}_s \cong \mathbb{C}^{2^n}$. Let $k \geq 0$ be an integer and let U_H be a unitary operator on the joint Hilbert space $\mathcal{H}_a \otimes \mathcal{H}_s$, where $\mathcal{H}_a \cong \mathbb{C}^{2^k}$ denotes the Hilbert space of k ancillas. Then, for $\alpha \in \mathbb{R}^+$, the unitary operator U_A is called an (α, k)-*block-encoding* of A if for every state $|\psi\rangle \in \mathcal{H}_s$:

$$\left(\mathbb{1}_s \otimes \langle g|_g\right) U_A \left(\mathbb{1}_s \otimes |g\rangle_a\right) = A/\alpha, \tag{77}$$

where $|g\rangle_a$ is some state in $\mathcal{H}_a$, $\mathbb{1}_s$ is the identity operator on $\mathcal{H}_s$. The value α chosen such that $\|A\|/\alpha \leq 1$ is called the *scale factor* of the BE.

BEs can be constructed using LCU as follows

Lemma 8 (LCU block encoding). *Let $U_0, \ldots, U_{M-1}$ be M unitary operators acting on a system Hilbert space $\mathcal{H}_s$ and define the operator $T = \sum_{i=0}^{M-1} \beta_i U_i$. The LCU method provides a $(\beta, \log M)$ BE of the operator T, where $\beta \equiv \sum_{i=0}^{M-1} \|\beta_i\|$. The gate complexity is given by $\mathcal{O}(M \log M)$ elementary gates.*

Proof. The proof of this Lemma can be found in many textbooks, see for example [59]. The asymptotic cost stems from the complexity of implementing the SEL subroutine, since the cost of preparing an arbitrary state on $\log_2 M$ qubits in PREP is $\mathcal{O}(M)$ [71, 72], which is cheaper than the cost of SEL, as we explain next. The gate complexity of SEL is $\mathcal{O}(M \log M)$, which comes from sequentially implementing M unitaries, each being controlled on $k = \log M$ ancilla qubits. Decomposing k-controlled gates into elementary gates can be achieved at a cost linear in k, with the constant prefactor being smallest if k additional ancilla qubits are allowed to be used [37, 73]. For applications considered in this work, this gives a minor overhead in the qubit cost since k is logarithmic is the parameters of the system. $\square$

Given a BE of the Hamiltonian, one can obtain an approximation to the time evolution operator using QSP.

Theorem 9 (QSP with Qubitization). *Let U_H be a an (α, k)-block-encoding of a Hamiltonian H acting on a Hilbert space $\mathcal{H}_s$. Furthermore, let t and ε be positive real values. Then there is a quantum circuit that produces a $\left(1 \pm \mathcal{O}(\varepsilon), k+2\right)$-block-encoding of the time evolution operator e^{-iHt} up to exponentiation error ε, and possesses a complexity of*

$$\mathcal{O}\left(\alpha t + \log(1/\varepsilon)\right), \tag{78}$$

relative to the oracle U_H. The probability for this BE to succeed is

$$p > 1 - \mathcal{O}(\varepsilon). \tag{79}$$

Proof. We will not prove this result rigorously, but point the reader to the pedagogical sections of this paper, with all statements made there having been rigorously proved in the literature [30]. The fact that the BE is $(\tilde{\alpha}, \tilde{k}) = (1 \pm \mathcal{O}(\varepsilon), k+2)$ can be easily understood. First, the BE of the time evolution operator requires two additional qubits, one ($|q\rangle$) for the qubitization and one ($|b\rangle$) to add the functional dependence on ϕ, giving $\tilde{k} = k+2$. The fact that $\tilde{\alpha} = 1 \pm \mathcal{O}(\varepsilon)$ follows from the fact that the operator we are block-encoding is a unitary operator, and our approximation is good up to error ε. Since the norm of a unitary operator is 1, this gives that both $\tilde{\alpha}$ and the success probability are unity (up to $\mathcal{O}(\varepsilon)$). $\square$

3. HHKL

Theorem 10 (HHKL Algorithm). *Let Λ be a lattice of qubits embedded in d-dimensional Euclidean space of size L in each dimension, and let $\hat{H}$ be a geometrically-local Hamiltonian, normalized such that each local term has a norm of order unity. Furthermore, let $\tilde{t}$ and ε be positive real values. Then there exists an algorithm that approximates $\mathrm{e}^{-i\tilde{H}\tilde{t}}$ to exponentiation error ε, which requires[13]*

$$\mathcal{O}\left(\frac{L^{\mathsf{d}}\tilde{t}}{\log^{\mathsf{d}}\left(L^{\mathsf{d}}\tilde{t}/\varepsilon\right)}\right) \tag{80}$$

calls to a helper function that implements the time evolution for a system of size

$$\ell = \Theta\left(\log\left(L^{\mathsf{d}}\tilde{t}/\varepsilon\right)\right), \tag{81}$$

with error

$$\delta = \mathcal{O}\left(\frac{\varepsilon \log^{\mathsf{d}}(L^{\mathsf{d}}\tilde{t}/\varepsilon)}{L^{\mathsf{d}}\tilde{t}}\right) \tag{82}$$

Proof. The results follow directly from our discussion in Sec. II C. For formal proofs of the results stated there we refer the reader to the original paper [34]. □

III. SIMULATING TIME EVOLUTION ON THE LATTICE

In this section, we apply the algorithms and methods of Section II to the case of Hamiltonians defined in Sec. I B. Similar to Sec. II, this section is organized by the techniques used (PFs, QSP, and HHKL). For each of these sections, we begin by considering the case of $\mathcal{O}(1)$-site-local RHLFTs defined on $\mathcal{O}(1)$-many sites, which behave very similarly to generic quantum systems, and will serve as a review of the relationship between our notation and other common notations. We then proceed to the case of k-site-local and k-geometrically-site-local Hamiltonians, whose simulation can roughly be categorized as being composed of simulation methods for $\mathcal{O}(1)$-site-local Hamiltonians that are "glued" via PFs, LCU, or HHKL, depending on the flavor of locality whose preservation is desired.

———————

[13] We assume that the error ε large enough that the number of calls to the helper function is larger than one.

A. Product Formulas

Before we investigate the scaling of PFs for different local Hamiltonians with different numbers of lattice sites, we give two results that apply to all Hamiltonians considered in this work.

The general results on PFs depended on a parameter $\tilde{\alpha}$, which was related to the commutator of terms in the Hamiltonian. Our first Lemma relates this to the norm of the Hamiltonian

Lemma 11. *Let H be a k-site-local Hamiltonian, as defined in Definition 2. Then the nested commutator $\tilde{\alpha} \equiv \sum_{J_1,J_2,\ldots,J_{p+1}\in S}||[H_{J_{p+1}},\ldots[H_{J_2},J_{J_1}]]||$ can be asymptotically bounded as*

$$\tilde{\alpha} = \mathcal{O}\left(|||H|||_1^p ||H||_1\right). \tag{83}$$

Proof. This result has been proven in Appendix G of [27]. □

The next Lemma will give an expression that will allow to combine a PF with another approximation to the exponentials that make up the PF. The PF introduced in Eq. (14) is used as an approximation to e^{-iHt}, and another approximation $U_v^{(J)}(t/r)$ is used to approximate each term of the form $\mathrm{e}^{-ita_v^{(J)}H_J/r}$.

Lemma 12. *Let H be a site-local Hamiltonian as defined by 2. Let $S_p(t)$ be a pth order PF for e^{-iHt} as given in Eq. (14), and let the unitary $U_v^{(J)}(t/r)$ be an approximation to $\mathrm{e}^{-ita_v^{(J)}H_J/r}$ for all choices of allowed pairs (v,J), with $v \in \{1,\ldots,\Upsilon(p)\}$ and $J \in S$. Define the PF $\tilde{S}_p(t)$ as the PF for e^{-iHt} obtained by replacing $\mathrm{e}^{-ita_v^{(J)}H_J/r}$ in Eq. (14) with $U_v^{(J)}(t/r)$*

$$\tilde{S}_p(t/r) \equiv \prod_{v=1}^{\Upsilon(p)} \mathcal{P}_v\left(\prod_{J\in S} U_v^{(J)}(t/r)\right). \tag{84}$$

If

$$\left\|U_v^{(J)}(t/r) - \mathrm{e}^{-ita_v^{(J)}H_J/r}\right\| = \mathcal{O}\left(\frac{\varepsilon}{\mathsf{N_H}r\Upsilon(p)}\right), \tag{85}$$

it is true that

$$\left\|\tilde{S}_p(t) - \mathrm{e}^{-iHt}\right\| = \mathcal{O}(\varepsilon). \tag{86}$$

Proof. Demanding that the new PF still reproduces the time evolution operator with the same exponen-

tiation error we can write

$$\left\|\mathrm{e}^{-iHt} - \tilde{S}_p(t)\right\|$$

$$= \left\|\mathrm{e}^{-iHt} - S_p(t) + \left[S_p(t/r)\right]^r - \left[\tilde{S}_p(t/r)\right]^r\right\|$$

$$\leq \mathcal{O}(\varepsilon) + \left\|\left[\tilde{S}_p(t/r)\right]^r - \left[S_p(t/r)\right]^r\right\|$$

$$= \mathcal{O}(\varepsilon) + r\,\mathsf{N_H}\Upsilon(p)\left\|U_v^{(J)}(t/r) - \mathrm{e}^{-it/ra_{vJ}H_J}\right\|$$

$$= \mathcal{O}(\varepsilon)\,.$$

$$(87)$$

□

1. General site-local Hamiltonian

We can now give results that apply to both general site-local Hamiltonians (see Definition 2).

Theorem 13 (Product Formulas for site-local Hamiltonians)**.** *Let H be a site-local Hamiltonian as defined in Definition 2 or Definition 3. To simulate the time evolution operator e^{-iHt} up to exponentiation error $\mathcal{O}(\varepsilon)$ it suffices to choose a Trotter number r such that:*

$$r = \mathcal{O}\left(\mathsf{N_{ind}}\mathsf{N_H}^{1/p}\frac{(\mathsf{N_P}\mathsf{c_P}t)^{1+\frac{1}{p}}}{\varepsilon^{\frac{1}{p}}}\right).\qquad(88)$$

Furthermore, if the Trotter number r is chosen with tight the asymptotic scaling

$$r = \Theta\left(\mathsf{N_{ind}}\mathsf{N_H}^{1/p}\frac{(\mathsf{N_P}\mathsf{c_P}t)^{1+\frac{1}{p}}}{\varepsilon^{\frac{1}{p}}}\right),\qquad(89)$$

then the associated elementary gate count χ can be asymptotically bounded as

$$\chi = \mathcal{O}\left(\mathsf{N_{ind}}\mathsf{N_P}\frac{(\mathsf{N_H}\mathsf{N_P}\mathsf{c_P}t)^{1+\frac{1}{p}}}{\varepsilon^{\frac{1}{p}}}\right).\qquad(90)$$

Proof. Using Theorem 6 and Lemma 11 one finds that the Trotter number r is given by

$$r = \mathcal{O}\left(\frac{|||H|||_1||H||_1^{\frac{1}{p}}t^{1+\frac{1}{p}}}{\varepsilon^{\frac{1}{p}}}\right).\qquad(91)$$

Using previously established notation, we have

$$||H||_1 = \mathcal{O}\left(\mathsf{N_H}\sum_J ||H_J||\right) = \mathcal{O}(\mathsf{N_H}\mathsf{N_P}\mathsf{c_P})\,,$$
$$|||H|||_1 = \mathcal{O}(\mathsf{N_{ind}}\mathsf{N_P}\mathsf{c_P})\,,\qquad(92)$$

which immediately gives

$$r = \mathcal{O}\left(\mathsf{N_{ind}}\mathsf{N_H}^{1/p}\frac{(\mathsf{N_P}\mathsf{c_P}t)^{1+\frac{1}{p}}}{\varepsilon^{\frac{1}{p}}}\right).\qquad(93)$$

Assume now that we wish to compute the total gate count χ for a quantum circuit implementing an approximation of e^{-iHt} to total exponentiation error $\mathcal{O}(\varepsilon)$ based on the splitting $\left(S_p(t/r)\right)^r$, where the Trotter number r in fact satisfies (89) (i.e., the asymptotic scaling is tight) so that the spectral norm $\|\left(S_p(t/r)\right)^r - \mathrm{e}^{-iHt}\| = \mathcal{O}(\varepsilon)$. This splitting takes the following form (see Eq. (14))

$$\left[S_p(t/r)\right]^r \equiv \left(\prod_{v=1}^{\Upsilon(p)}\mathcal{P}_v\left(\prod_{J\in S}\exp\left(\frac{-ita_v^{(J)}H_J}{r}\right)\right)\right)^r.$$
$$(94)$$

Each $S_p(t/r)$ thus consists of $\mathcal{O}\left(\mathsf{N_H}\Upsilon(p)\right)$-many exponentials of the form $\exp\left(-ita_v^{(J)}H_J/r\right)$. Each of these exponentials can be approximated using a different PF $\tilde{S}_{\tilde{p}}^{(v,J)}(t/r\tilde{r})$ based on the Pauli decomposition of each term H_J (see Eq. (3)). This splitting takes the form

$$\left[\tilde{S}_{\tilde{p}}^{(v,J)}(t/(r\tilde{r}))\right]^{\tilde{r}}$$

$$\equiv \left(\prod_{v=1}^{\Upsilon(\tilde{p})}\tilde{\mathcal{P}}_v\left(\prod_{i=1}^{\mathsf{N_P}^{(J)}}\exp\left(\frac{-it\tilde{a}_v^{(i)}c_i^{(J)}P_i^{(J)}}{r\tilde{r}}\right)\right)\right)^{\tilde{r}}.$$
$$(95)$$

From Lemma 12 we know that to ensure that the total error remains $\mathcal{O}(\varepsilon)$, the accuracy of this PF has to be

$$\left\|\left(\tilde{S}_{\tilde{p}}^{(v,J)}(t/r\tilde{r})\right)^{\tilde{r}} - \exp\left(\frac{-ita_v^{(J)}H_J}{r}\right)\right\|$$
$$= \mathcal{O}\left(\frac{\varepsilon}{\mathsf{N_H}r\Upsilon(p)}\right).$$
$$(96)$$

Substituting this into the result of Theorem 6 we therefore find

$$\tilde{r} = \Omega\left(\left(\frac{\mathsf{N_P}\mathsf{c_P}t}{r}\right)^{1+\frac{1}{p}}\left(\frac{\varepsilon}{\mathsf{N_H}\Upsilon(p)r}\right)^{-\frac{1}{p}}\right)$$

$$= \Omega\left(\frac{\Upsilon(p)}{\mathsf{N_{ind}}}\left(\frac{\mathsf{N_P}\mathsf{c_P}t}{\mathsf{N_H}\varepsilon}\right)^{\frac{p}{p}}\right).$$
$$(97)$$

We choose $\tilde{p} = p$ and use that $\Upsilon(p) = \mathcal{O}(1)$. We also have that $\tilde{r}, \mathsf{N_{ind}} \in \mathbb{N}$ have to both be at least 1, and hence one finds $\tilde{r} = \Omega(1)$.

Using that $\Upsilon(\tilde{p}) = \mathcal{O}(1)$, we have that the resource scaling is thus given by

$$\chi = \mathcal{O}(N_H N_P r)$$
$$= \mathcal{O}\left(N_{ind} N_P \frac{(N_H N_P c_P t)^{1+\frac{1}{p}}}{\varepsilon^{\frac{1}{p}}}\right). \qquad (98)$$

$\square$

2. $\mathcal{O}(1)$ site-local Hamiltonian

Having obtained the results for a general site-local Hamiltonian, one can obtain the results for an $\mathcal{O}(1)$ site-local Hamiltonian

Theorem 14 (Product Formulas for $N_\Lambda = \mathcal{O}(1)$ sites). *Let H be an $\mathcal{O}(1)$ Hamiltonian as defined in Definition 4 Let $S_p(t)$ be a pth order PF as given by Eq. (14) and Eq. (75). Then to simulate the time evolution operator e^{-iHt} up to exponentiation error $\mathcal{O}(\varepsilon)$ it suffices to choose a Trotter number r such that:*

$$r = \mathcal{O}\left(\frac{(N_P c_P t)^{1+\frac{1}{p}}}{\varepsilon^{\frac{1}{p}}}\right). \qquad (99)$$

Furthermore, if the Trotter number r is chosen with the tight asymptotic scaling

$$r = \Theta\left(\frac{(N_P c_P t)^{1+\frac{1}{p}}}{\varepsilon^{\frac{1}{p}}}\right), \qquad (100)$$

then the associated elementary gate count χ can be asymptotically bounded as

$$\chi = \mathcal{O}\left(\frac{N_P (N_P c_P t)^{1+\frac{1}{p}}}{\varepsilon^{\frac{1}{p}}}\right). \qquad (101)$$

Proof. This result is obtained immediately by setting $N_H \sim N_{ind} = \mathcal{O}(1)$ in Theorem 13. $\square$

3. Geometrically-site-local Hamiltonian

Theorem 15 (Product Formulas for geometrically-site-local Hamiltonians). *Let H be a geometrically site-local Hamiltonian as defined in Definition 3. To simulate the time evolution operator e^{-iHt} up to exponentiation error $\mathcal{O}(\varepsilon)$ it suffices to choose a Trotter number r such that:*

$$r = \mathcal{O}\left(N_\Lambda^{1/p} \frac{(N_P c_P t)^{1+\frac{1}{p}}}{\varepsilon^{\frac{1}{p}}}\right). \qquad (102)$$

Furthermore, if the Trotter number r is chosen with tight the asymptotic scaling

$$r = \Theta\left(N_\Lambda^{1/p} \frac{(N_P c_P t)^{1+\frac{1}{p}}}{\varepsilon^{\frac{1}{p}}}\right), \qquad (103)$$

then the associated elementary gate count χ can be asymptotically bounded as

$$\chi = \mathcal{O}\left(N_P \frac{(N_\Lambda N_P c_P t)^{1+\frac{1}{p}}}{\varepsilon^{\frac{1}{p}}}\right). \qquad (104)$$

Proof. This result follows directly from Theorem 13 making the replacements $N_{ind} = \mathcal{O}(1)$ and $N_H = \mathcal{O}(N_\Lambda)$ as given in Eqs. (11) and (12). $\square$

B. Quantum Signal Processing

As discussed in detail in Sec. II D 2, QSP provides a technique to take the BE of a given Hamiltonian and obtain the approximation to the BE of its time evolution operator. We will begin by deriving the general result of using QSP to implement the time evolution operator of the full system Hamiltonian.

We will also derive results that combine the techniques of PFs with those of QSP. In this approach we will use a PF to split the full time evolution operator into products of exponentials of the form $\mathrm{e}^{-ita_v^{(J)} H_J / r}$, and then use QSP to to obtain expression for these exponentials requiring only $\mathcal{O}(1)$ lattice sites.

In both of these approaches we assume that the original BE is achieved using LCU, as explained in Sec. II B 2, but we note that other techniques are possible as well.

1. General site-local Hamiltonian

We first establish the scaling of the LCU that will be used for the BE

Lemma 16 (LCU for site-local Hamiltonians). *Let H be a site-local Hamiltonian as as in Definition 2. There exists a quantum circuit acting on $n \equiv N_\Lambda n_q$-many qubits together with $k = \mathcal{O}\left(\log(N_H N_P)\right)$-many ancillas that produces an $\left(\mathcal{O}\left(N_H N_P c_P\right), k\right)$-block-encoding of H on the n qubits, and requires a total of $\mathcal{O}\left(N_H N_P \log(N_H N_P)\right)$-many elementary quantum gates.*

Proof. This follows directly from Lemma 8, if we observe that the total number of Pauli strings is given by $N_H N_P$ and the sum of the coefficients of all Pauli strings is bounded by $\beta < N_H N_P$. $\square$

Given this result now allows us to state the central QSP result we use in this work:

Theorem 17 (QSP for site-local Hamiltonians). *Let H be a site-local Hamiltonian as defined by 2, and let t and ε be positive values. Then there exists a quantum circuit that serves as a $\left(1 \pm \mathcal{O}(\varepsilon), \mathcal{O}(\log(N_H N_P))\right)$-block-encoding for e^{-iHt} up to exponentiation error ε, and requires an elementary gate count of*

$$\chi = \mathcal{O}\left[N_H N_P \log(N_H N_P)\left(N_H N_P c_P t + \log(1/\varepsilon)\right)\right].$$
(105)

Proof. This result follows from Lemma 16 and Theorem 9. From Lemma 16 one obtains that LCU give a $\left(\mathcal{O}\left(N_H N_P c_P\right), \mathcal{O}\left(\log(N_H N_P)\right)\right)$ BE of the Hamiltonian. Using this result in Theorem 9 one finds that QSP gives a $\left(1 \pm \mathcal{O}(\varepsilon), \mathcal{O}(\log(N_H N_P))\right)$-block-encoding for e^{-iHt}. Using the resource scaling from Theorem 9, with the scale factor and resource scaling of LCU from Lemma 16, gives the resource requirements stated in the theorem. $\square$

Remark 18. Note that we have assumed that the block-encoding is formed from a sum over all Pauli strings appearing in the Hamiltonian. When more structure is known about a given field theory, the factor in the complexity due to the block-encoding may be reduced.

2. $\mathcal{O}(1)$ site-local Hamiltonian

As for the PF, we can obtain the QSP requirements of an $\mathcal{O}(1)$ site-local Hamiltonian:

Theorem 19 (QSP for $\mathcal{O}(1)$ sites). *Let H be an $\mathcal{O}(1)$ Hamiltonian as defined in Definition 4, and let t and ε be positive values. Then there exists a quantum circuit that serves as a $\left(1 \pm \mathcal{O}(\varepsilon), \mathcal{O}(\log(N_P))\right)$-block-encoding for e^{-iHt} up to exponentiation error ε, and requires an elementary gate count of*

$$\chi = \mathcal{O}\left[N_P \log N_P\left(N_P c_P t + \log(1/\varepsilon)\right)\right].$$
(106)

Proof. This follows directly from Theorem 17 by taking the limit $N_H = \mathcal{O}(1)$. $\square$

3. Geometrically-site-local Hamiltonian

Theorem 20 (QSP for geometrically-site-local Hamiltonians). *Let H be a geometrically site-local Hamiltonian as defined by 2, and let t and ε be positive values. Then there exists a quantum circuit that*

serves as a $\left(1 \pm \mathcal{O}(\varepsilon), \mathcal{O}(\log(N_H N_P))\right)$-block-encoding for e^{-iHt} up to exponentiation error ε, and requires an elementary gate count of

$$\chi = \mathcal{O}\left[N_\Lambda N_P \log(N_\Lambda N_P)\left(N_\Lambda N_P c_P t + \log(1/\varepsilon)\right)\right].$$
(107)

Proof. This result follows directly from Theorem 17 making the replacements $N_H = \mathcal{O}(N_\Lambda)$ as given in Eq. (12). $\square$

C. Combining Product Formulas with QSP

In this section we consider combining a PF with QSP. We will use Lemma 12 to write a PF for e^{-iHt}, but then use QSP for $U_v^{(J)}(t/r)$, which serves as an approximation to each term of the form $\exp\left(-it a_v^{(J)} H_J/r\right)$ appearing in the PF.

Theorem 21 (QSP applied to Product Formulas). *Let H be a site-local Hamiltonian as defined by 2, and let t and ε be positive values. By using QSP (see Theorem 19) to simulate each exponential term $e^{-i a_v^{(J)} H_J t/r}$ appearing in the PF $S_p(t/r)$ to error $\varepsilon/(r N_H)$, and taking*

$$r = \Theta\left(N_{ind} N_H^{1/p} \frac{(N_P c_P t)^{1+\frac{1}{p}}}{\varepsilon^{\frac{1}{p}}}\right),$$
(108)

we obtain a quantum circuit that serves as a $(1 \pm \mathcal{O}(\varepsilon), \log(N_H N_P))$-block-encoding for the full time evolution operator e^{-iHt} to exponentiation error $\mathcal{O}(\varepsilon)$. The gate count for this circuit is given by

$$\chi = \mathcal{O}\Big[N_{ind}\left(N_H N_P c_P t\right)^{1+\frac{1}{p}} \varepsilon^{-\frac{1}{p}} N_P \log N_P$$
$$\times \log(N_{ind} N_H N_P c_P t/\varepsilon)\Big].$$
(109)

Proof. From Lemma 12 we know that the simulation of a single exponential of the form $e^{-i a_v^{(J)} H_J t/r}$ acting on $\mathcal{O}(1)$ sites is needed to accuracy $\varepsilon/(N_H r)$. Performing this approximation using QSP (see Theorem 19) requires a gate count

$$\chi_J = \mathcal{O}\left(N_P \log N_P\left(N_P c_P \frac{t}{r} + \log(N_H r/\varepsilon)\right)\right)$$
$$= \mathcal{O}\left(N_P \log N_P \log(N_H r/\varepsilon)\right),$$
(110)

where the first term can be discarded because the asymptotic assumption on r given by (108) indicates that $N_P c_P t/r = \mathcal{O}(\varepsilon/(N_P c_P t)^{1/p})$, which is asymptotically bounded from above by a constant.

The total gate count from the $(N_H r)$-many applications of this QSP subroutine is then given by

$$
\begin{aligned}
\chi &= N_H \, r \, \chi_J \\
&= \mathcal{O}\Big[N_{ind} \, (N_H N_P c_P t)^{1+\frac{1}{p}} \, \varepsilon^{-\frac{1}{p}} \\
&\quad \times N_P \log N_P \log(N_{ind} N_H N_P c_P t/\varepsilon) \Big] .
\end{aligned}
\tag{111}
$$

Since the block-encoding produced by QSP has a scale factor $1 \pm \mathcal{O}(\varepsilon/(r N_H))$, the final block-encoding produced by multiplying $(\Upsilon(p) N_H r)$-many such block-encodings will again be a block-encoding, this time with scale-factor $1 \pm \mathcal{O}(\varepsilon)$. This completes the proof. $\qquad\square$

D. HHKL

As discussed in Sec. II C the HHKL technique is a method of "gluing" together implementations of a time evolution operator on a "local" lattice with size ℓ^d. In this section we consider HHKL with two approaches for the implementation of the local evolution operator, first using a PF and then QSP.

Theorem 23 (HHKL with PF as helper). *Let H be a geometrically site-local Hamiltonian as defined by 3, and let t and ε be positive values. Using the HHKL algorithm with a PF for the helper function gives rise to an algorithm that approximates the time evolution operator e^{-iHt} with exponentiation error requiring a gate complexity of*

$$
\chi = N_P \, (N_\Lambda N_P c_P t)^{1+\frac{1}{p}} \, \varepsilon^{-\frac{1}{p}} .
\tag{112}
$$

Proof. From Theorem 10 we need the helper function to implement a time evolution operator for time $\tilde{t} = \|H_J\| t = \mathcal{O}(N_P c_P t)$ on a system of size $N_\ell = \mathcal{O}(\ell^d)$ with exponentiation error δ, where ℓ and δ are given in Eqs. (81) and (82).

From Theorem 13 we find that a PF to implement the time evolution on a system with N_ℓ lattice sites and exponentiation error δ requires resources (taking

$t = \mathcal{O}(1)$ in Eq. (105))

$$
\begin{aligned}
\chi_{\mathrm{PF}} &= \mathcal{O}\Big(N_P \, (N_\ell N_P c_P)^{1+\frac{1}{p}} \, \delta^{-\frac{1}{p}} \Big) \\
&= \mathcal{O}\bigg(N_P \, (N_P c_P)^{1+\frac{1}{p}} \Big[\log \big(N_\Lambda t/\varepsilon \big) \Big]^{d\left(1+\frac{1}{p}\right)} \\
&\quad \times \bigg(\frac{N_\Lambda t}{\varepsilon \log^d \big(N_\Lambda t/\varepsilon \big)} \bigg)^{\frac{1}{p}} \bigg) \\
&= \mathcal{O}\bigg(N_P \, (N_P c_P)^{1+\frac{1}{p}} \log^d \Big(\frac{N_\Lambda t}{\varepsilon} \Big) \Big(\frac{N_\Lambda t}{\varepsilon} \Big)^{\frac{1}{p}} \bigg) .
\end{aligned}
\tag{113}
$$

Furthermore, we know from Theorem 10 that the number of calls to this helper function is given by Eq. (80), such that the final gate complexity is

$$
\begin{aligned}
\chi &= \mathcal{O}\bigg(\frac{N_\Lambda t}{\log^d \big(N_\Lambda t/\varepsilon \big)} \chi_{\mathrm{PF}} \bigg) \\
&= \mathcal{O}\Big(N_P \, (N_\Lambda N_P c_P t)^{1+\frac{1}{p}} \, \varepsilon^{-\frac{1}{p}} \Big) .
\end{aligned}
\tag{114}
$$

as desired. $\qquad\square$

Theorem 24 (HHKL with QSP as helper). *Let H be a geometrically site-local Hamiltonian as given by Definition 3, and let t and ε be positive values. Using the HHKL algorithm with QSP for the helper function gives rise to an algorithm that approximates the time evolution operator e^{-iHt} with exponentiation error requiring a gate complexity of*

$$
\chi = \mathcal{O}\bigg(N_P{}^2 N_\Lambda c_P t \log^d \big(N_\Lambda t/\varepsilon \big) \log \Big(N_P \log \big(N_\Lambda t/\varepsilon \big) \Big) \bigg) .
\tag{115}
$$

Proof. Theorem 10 states that we need the helper function to implement a time evolution operator for time $\tilde{t} = \|H_J\| t = \mathcal{O}(N_P c_P t)$ on a system of size $N_\ell = \mathcal{O}(\ell^d)$ with exponentiation error δ, where ℓ and δ are given in Eqs. (81) and (82). From Theorem 17 we find that implementing the time evolution on a system with N_ℓ lattice sites and exponentiation error δ with QSP requires resources (taking $t = \mathcal{O}(1)$ in Eq. (105))

$$
\begin{aligned}
\chi_{\mathrm{QSP}} &= \mathcal{O}\Big(N_\ell N_P \log \big(N_\ell N_P \big) \big(N_\ell N_P c_P + \log(1/\delta) \big) \Big] \\
&= \mathcal{O}\bigg(\log^d \big(N_\Lambda t/\varepsilon \big) N_P \log \Big[N_P \log \big(N_\Lambda t/\varepsilon \big) \Big] \\
&\quad \times \Big[N_P c_P \log^d \big(N_\Lambda t/\varepsilon \big) \\
&\quad\quad + \log \bigg(\frac{N_\Lambda t}{\varepsilon \log^d \big(N_\Lambda t/\varepsilon \big)} \bigg) \Big] \bigg) .
\end{aligned}
\tag{116}
$$

Remark 22. The results above were for a general site-local Hamiltonian. To obtain the results for a geometrically site-local Hamiltonian one makes the replacement $N_{ind} = \mathcal{O}(1)$ and $N_H = \mathcal{O}(N_\Lambda)$ from Eqs. (11) and (12).

In the limit of large system size, the second term in the last square bracket (the one originating from the $\log(1/\delta)$ term) is subdominant to the first, which allows us to write

$$\chi_{\mathrm{QSP}} = \mathcal{O}\left(\mathsf{N_P} \mathsf{c_P} \log^{2\mathsf{d}}\left(\mathsf{N_\Lambda} t/\varepsilon \right) \log\left[\mathsf{N_P} \log\left(\mathsf{N_\Lambda} t/\varepsilon \right) \right] \right). \tag{117}$$

Using the number of calls to this helper function from Eq. (80), the final gate complexity is

$$\begin{aligned}
\chi &= \mathcal{O}\left(\frac{\mathsf{N_\Lambda} t}{\log^{\mathsf{d}}\left(\mathsf{N_\Lambda} t/\varepsilon \right)} \chi_{\mathrm{QSP}} \right) \\
&= \mathcal{O}\left(\mathsf{N_P}^2 \mathsf{N_\Lambda} \mathsf{c_P} t \log^{\mathsf{d}}\left(\mathsf{N_\Lambda} t/\varepsilon \right) \log\left[\mathsf{N_P} \log\left(\mathsf{N_\Lambda} t/\varepsilon \right) \right] \right),
\end{aligned} \tag{118}$$

as desired. $\square$

IV. LATTICE QUARTIC THEORY

The results we have discussed so far are theoretical worst-case asymptotic complexities, applicable to any lattice field theory with prescribed locality properties. In this section, we specialize to a particular model of interacting quantum field theories: the lattice scalar field theory. From the computational perspective, this theory is similar to lattice gauge theories and so intuition can be drawn from its consideration. Furthermore, it is often used as a proverbial example in the studies of quantum simulation algorithms [9, 15, 19], and so detailed comparisons with past work can be performed.

In this section, we review the Hamiltonian of the theory and our procedure for digitizing the bosonic field. Throughout the discussion, we will keep track of the parameters of the theory required for quoting the asymptotic gate complexities derived in Sec. III, e.g., $\mathsf{N_P}$, $\mathsf{c_P}$, $\mathsf{N_H}$, etc. We then determine the asymptotic gate complexities for implementing time evolution using PFs, QSP with two different BE methods, and the HHKL algorithm. To determine the coefficients in these asymptotic gate complexities, we explicitly construct circuits that implement time evolution for a single site. We first compare the gate cost of constructing controlled calls to the walk operator W_H obtained using the naïve and improved LCU procedures. Then we compare the gate costs for time evolution simulated via 4th order PF and QSP.

A. Introduction

The model we consider is defined on a hypercubic lattice Λ of spatial dimension d with lattice spacing

a. Assuming the simplest finite-difference approximation to the spatial derivative $(\nabla_\mu \hat{\varphi})^2$, the Hamiltonian is given by[14]

$$\hat{H}_{\mathrm{quartic}} = \hat{H}_\varphi + \hat{H}_\pi, \tag{119a}$$

$$\hat{H}_\varphi = a^{\mathsf{d}} \left[\sum_{i=1}^{\mathsf{N_\Lambda}} \frac{m}{2} \hat{\varphi}_i^2 + \frac{\lambda}{4!} \hat{\varphi}_i^4 + \sum_{\langle ij \rangle} \frac{(\hat{\varphi}_j - \hat{\varphi}_i)^2}{2a^2} \right], \tag{119b}$$

$$\hat{H}_\pi = a^{\mathsf{d}} \sum_{i=1}^{\mathsf{N_\Lambda}} \frac{1}{2} \hat{\pi}_i^2, \tag{119c}$$

where the indices i and j label lattice sites, and the notation $\langle ij \rangle$ runs over all distinct links joining nearest neighbors in the lattice. In the remainder of this work, we will denote operators with a hat to differentiate between their digitized eigenvalues.

The Hamiltonian in Eq. (119) is 2-site-geometrically-local, which implies that $\mathsf{N_{ind}} = \mathcal{O}(1)$. We see that the Hamiltonian is a sum of $\mathsf{N_H} = \mathcal{O}(\mathsf{N_\Lambda})$ terms.

The operators $\hat{\pi}_i$ and $\hat{\varphi}_i$ are conjugate operators satisfying $[\hat{\varphi}_j, \hat{\pi}_i] = ia^{-\mathsf{d}} \delta_{ij}$. This relation implies that the so-called field basis, in which the φ_i operators are diagonal, is related to the so-called momentum basis, where the π_i operators are diagonal, by the usual Fourier transform. The parameters m and λ are bare quantities, and are in general complicated non-perturbative functions of the lattice spacing, *i.e.* $m = m(a)$ and $\lambda = \lambda(a)$. In principle, this functional dependence could be included to determine the asymptotic complexity of a realistic physical calculation in terms of the lattice spacing. In order to keep our expressions as general as possible, however, we choose to leave the bare parameters m and λ as free variables in our quoted asymptotic complexities; this allows our results to be adapted in a straightforward way to any specific physical use case. For the remainder of this work, we use units where the lattice spacing $a = 1$.

The above Hamiltonian, although discretized on the lattice, is still an unbounded operator acting on an infinite dimensional Hilbert space. In order to obtain concrete time-evolution operators whose action can actually be simulated on a system of qubits, we must first truncate this Hamiltonian to a finite-dimensional operator. This is achieved by a procedure known as *digitization*, in which the Hilbert

---

[14] The following discussion can be generalized to the case of higher-order finite-difference approximations to the derivative operator. One common approach is to first perform integration by parts to get the term $\hat{\varphi} \nabla^2 \hat{\varphi}$ and then use a symmetric approximation for the second derivative.

space at each lattice site is represented using n_q qubits; the details of our digitization procedure are given in the next section.

1. Digitization

The basic idea is that we wish to construct a sequence of matrix operators H_{n_q} that can act on n_q qubits at a time, and whose spectrum converges to any prescribed single-site Hamiltonian H as $n_q \to \infty$. There are numerous ways to proceed with such a construction [74–76], but the one we follow is outlined in [11, 77].

The strategy is to *digitize* the field on a given lattice site into $N = 2^{n_q}$ allowed field values

$$\varphi \in \{-\varphi_{\max}, -\varphi_{\max} + \delta\varphi, \dots, \varphi_{\max} - \delta\varphi, \varphi_{\max}\}, \tag{120}$$

where the spacing $\delta\varphi$ is given by

$$\delta\varphi = \frac{2\varphi_{\max}}{N - 1}. \tag{121}$$

This restriction of the field values is to be understood as a restriction of the Hilbert space at the given lattice site, identifying the eigenvalues of the restricted field operator precisely with this set of allowed field values. In other words, the digitized field operator in the field basis at site i becomes the diagonal matrix

$$\hat{\varphi}_i = \mathrm{diag}\left(\varphi_{\min}, \varphi_{\min} + \delta\varphi, \dots, \varphi_{\max}\right)_i. \tag{122}$$

This single-site field operator acts nontrivially only on the n_q-qubit Hilbert space attributed to the site labeled by i. Note that all lattice sites use the same digitized φ values.

Because $\hat{\pi}_i$ and $\hat{\varphi}_i$ are conjugate operators, the momentum operator in the field basis can be written as

$$\hat{\pi}_i = \mathrm{FT} \cdot \hat{\pi}_i^{(p)} \cdot \mathrm{FT}^{-1},$$
$$\hat{\pi}_i^{(p)} = \mathrm{diag}\left(-\pi_{\max}, -\pi_{\max} + \delta\pi, \dots, \pi_{\max}\right)_i, \tag{123}$$

where the superscript (p) indicates the operator is represented in the momentum basis, FT denotes the *symmetric Fourier transform* [11] and

$$\pi_{\max} = \frac{\pi}{\delta\varphi}, \quad \delta\pi = \frac{2\pi_{\max}}{N - 1}. \tag{124}$$

Note that this choice implies anti-periodic boundary conditions in field space. Using this definition, the momentum part of the Hamiltonian can be written as

$$\hat{H}_\pi = \sum_{i=1}^{N_\Lambda} \frac{1}{2} \mathrm{FT}_i \cdot \left(\hat{\pi}_i^{(p)}\right)^2 \cdot \mathrm{FT}_i^{-1}. \tag{125}$$

The final component of the digitization strategy is the choice of $\varphi_{\max}$. It was shown in [11] that, for a given value of n_q and λ, there is an optimal choice of $\varphi_{\max}$ that minimizes the digitization error of the low lying spectrum of the digitized Hamiltonian. The procedure for determining $\varphi_{\max}$ involved scanning over many values to minimize the digitization error. While this procedure is in general possible, an explicit formula for the optimal value of $\varphi_{\max}$ for the $\lambda = 0$ case was given in [74, 76, 78]. As argued in [11], because $\lambda \neq 0$ results in a smooth deformation of the wavefunction, this choice of $\varphi_{\max}$ is expected to work well even in the $\lambda \neq 0$ case. Therefore, we use the value

$$\varphi_{\max} = 2^{n_q} \sqrt{\frac{2\pi}{2^{n_q+1} + 1}}, \tag{126}$$

as in [43].

Altogether, these definitions allow for the construction of $2^{n_q} \times 2^{n_q}$ Hermitian matrices that digitize each of the single-site terms in Hamiltonian (119) into operators that can act on n_q qubits at a time. An appropriate full digitization of Hamiltonian (119) is then obtained by directly substituting the definitions of the digitized single-site operators $\hat{\varphi}_i$ and $\hat{\pi}_i$ into (119), together with the link terms $\hat{\varphi}_i \hat{\varphi}_j = \hat{\varphi}_i \otimes \hat{\varphi}_j$.

We are now in a position to discuss the parameters of this theory necessary for determining the cost of the various methods described in Sec. II D. Recall that we found $N_H = \mathcal{O}(N_\Lambda)$ and $N_{ind} = \mathcal{O}(1)$. The final parameters needed are N_P and c_P, which we now discuss.

To determine the maximum number of Pauli strings N_P appearing in any term in $\hat{H}$, we start by writing the various operators in terms of Pauli operators. Note that the n_q qubit operators $\hat{\varphi}_i$ and $\hat{\pi}_i^{(p)}$ are diagonal operators with evenly spaced eigenvalues. It was shown in [11] that operators of this form can be decomposed into n_q single Pauli-Z gates. For example, one has $\hat{\varphi} = \frac{\varphi_{\max}}{2^{n_q}-1} \sum_{j=0}^{n_q-1} 2^j Z_j$, where Z_j is a single Pauli-Z gate acting on qubit j. Using this relation, we see that N_P for the various terms in H_φ are

$$N_P^{(\varphi_i^2)} = \mathcal{O}(n_q^2), \tag{127a}$$

$$N_P^{(\varphi_i^4)} = \mathcal{O}(n_q^4), \tag{127b}$$

$$N_P^{(\varphi_i \varphi_j)} = \mathcal{O}(n_q^2). \tag{127c}$$

Turning now to $\hat{H}_\pi$, while the operator $(\hat{\pi}_i^{(p)})^2$ is a sum of $\mathcal{O}(n_q^2)$ gates, the same is not true for $\hat{\pi}_i^2 = \mathrm{FT} \cdot (\hat{\pi}_i^{(p)}) \cdot \mathrm{FT}^\dagger$; after using the Fourier transform to change to the field basis, the operator $\hat{\pi}_i^2$ is dense and requires $\mathcal{O}(4^{n_q})$ Pauli strings. Since the quantum

Fourier transform requires $\mathcal{O}(n_q^2)$ gates, it is more cost effective to first decompose $(\hat{\pi}_i^{(p)})^2$ into $\mathcal{O}(n_q^2)$ Pauli strings, and then change to the field basis using the Fourier transform. Using this method results in

$$N_P^{(\pi^2)} = \mathcal{O}(n_q^2). \tag{128}$$

From this we see that N_P for the entire Hamiltonian is dominated by the $\hat{\varphi}^4$ term, and is given by

$$N_P = \mathcal{O}(n_q^4). \tag{129}$$

This result for N_P is significantly more favorable than the worst case scaling for an arbitrary 2-site-local Hamiltonian of $\mathcal{O}(4^{2n_q})$.

The final piece is to determine the maximum coefficient c_P of any Pauli string in $\hat{H}$. Using the fact that our choice of digitization leads to $\pi_{\max} = \mathcal{O}(2^{n_q/2})$ and $\varphi_{\max} = \mathcal{O}(2^{n_q/2})$, we see

$$c_P^{(\pi^2)} = \mathcal{O}(\pi_{\max}^2) = \mathcal{O}(2^{n_q}), \tag{130}$$

$$c_P^{(\varphi^2)} = \mathcal{O}(m^2\varphi_{\max}^2) = \mathcal{O}(m^2 2^{n_q}), \tag{131}$$

$$c_P^{(\varphi^4)} = \mathcal{O}(\lambda\varphi_{\max}^4) = \mathcal{O}(\lambda 4^{n_q}), \tag{132}$$

$$c_P^{(\varphi_i\varphi_j)} = \mathcal{O}(\varphi_{\max}^2) = \mathcal{O}(2^{n_q}). \tag{133}$$

As was the case with N_P, the overall value of c_P is dominated by the $\hat{\varphi}^4$ term, and is given by

$$c_P = \mathcal{O}(\lambda 4^{n_q}). \tag{134}$$

These values for N_P, c_P and N_H will be used in the following sections to determine the asymptotic gate complexity of the various time-evolution algorithms discussed in Sec. II D for geometrically site-local Hamiltonians.

B. Product Formulas

In this section, we consider using a pth order PF (introduced in Sec. II A) to time evolve the scalar field theory Hamiltonian (given by Eq. (119)) that has been digitized using the formalism presented in Section IV A 1.

Specifically, we construct concrete error bounds and asymptotic gate complexity estimates as a special case of the results obtained in Sec. III A for general geometrically site-local Hamiltonians. We use the result of Theorem 15 for the asymptotic bound on the elementary gate count χ (see Eq. (104)) and make the following substitutions based on the discussion provided in Section IV A 1:

$$N_H = \mathcal{O}(N_\Lambda), \tag{135}$$

$$N_P = \mathcal{O}(n_q^4), \tag{136}$$

$$c_P = \mathcal{O}(\lambda 4^{n_q}). \tag{137}$$

These follow from Eq. (12), Eq. (129), and (134) respectively. Putting this all together, the asymptotic gate complexity is thus given by

$$\chi = \mathcal{O}\left(n_q^{4\left(2+\frac{1}{p}\right)} \frac{(4^{n_q} N_\Lambda \lambda t)^{1+\frac{1}{p}}}{\varepsilon^{\frac{1}{p}}} \right). \tag{138}$$

C. Quantum Signal Processing

Obtaining gate counts for the case of pure QSP requires two logical steps. The first is to determine the number of calls to the local block encoding, and the second is to estimate the complexity of implementing this block encoding. By Theorem 9, the query complexity for simulating with evolution time t to exponentiation error ε is given by

$$\mathcal{O}(\alpha t + \log(1/\varepsilon)), \tag{139}$$

where α is the scale factor of the BE used.

We now consider two approaches to implement the local block encoding. The first approach is based on the fully general LCU procedure, and does not take into account any features of the model. The second approach is a modification of the former which takes advantage from splitting the Hamiltonian into the field and momentum parts, in analogy with how it was done in the product-formula-based approach. In this section, we consider only asymptotic gate complexities; detailed gate count comparisons of these techniques will be given in Sec. IV E.

1. Block encoding via LCU

In this section, we determine the cost of simulating time-evolution via QSP when the BE is implemented using a fully general LCU procedure. This involves first digitizing the Hamiltonian $\hat{H}$ and then decomposing it into Pauli operators. In such a scenario, the dominant cost arises from the $\hat{\pi}_i^2$ operator, which is represented by $\mathcal{O}(3^{n_q})$ Pauli strings (excluding Z gates in all combinations). The value of c_P in this case is the same as in Eq. (134). Lastly, we have $N_H = \mathcal{O}(N_\Lambda)$. Plugging these results into Theorem 17 leads to gate complexity

$$\chi = \mathcal{O}\left[N_\Lambda 3^{n_q} \log\left(N_\Lambda 3^{n_q}\right) \left(N_\Lambda 3^{n_q}(\lambda 4^{n_q})t + \log(1/\varepsilon)\right) \right]. \tag{140}$$

We see that the naive decomposition of the $\hat{\pi}_i^2$ operators into Pauli gates results in a poor scaling with n_q. By exploiting the fact that $\text{SEL}^2 = \mathbb{1}$, the walk oper-
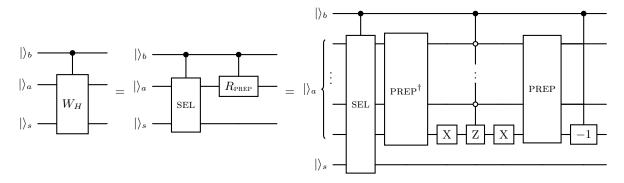
Figure 7. Circuit for controlled call to W_H when the BE is prepared using LCU. Note that $|g\rangle_a = \text{PREP}|0\rangle_a$ and so the reflector R_{PREP} (see Fig. 4) uses $G = \text{PREP}$.

ator W_H can then be constructed without ancillary qubits using Eq. (50), as was discussed in Sec. II B 3.

The final step is to construct a controlled call to W_H. Rather than naively controlling every gate in W_H, one can achieve this by only controlling the SEL oracle and the multi-controlled Z gate in the reflector R; the PREP oracle does not need to be controlled [37]. This can be understood by noting that if one wanted to implement a controlled call to some operator $U = A^\dagger B A$ where $A^\dagger A = \mathbb{1}$, one only has to control B. The circuit for the controlled implementation of W_H is shown in Fig. 7.

2. Block encoding via LCU and FT

In order to take advantage of the decomposition (125) within the LCU construction, one needs to slightly modify the PREP and SEL oracles. To write $\hat{H}$ in the form required for LCU, we first write the momentum component of $\hat{H}$ in the momentum basis as a sum of M_π unitary operators

$$\hat{H}_\pi^{(p)} = \sum_{j=0}^{M_\pi} \beta_j^{(\pi^{(p)})} \hat{U}_j^{(\pi^{(p)})} \,. \tag{141}$$

Next, we write the full Hamiltonian as

$$\hat{H} = \sum_{i=0}^{M_\varphi - 1} \beta_i^{(\varphi)} \hat{U}_j^{(\varphi)} + \text{FT} \left(\sum_{j=0}^{M_\pi - 1} \beta_j^{(\pi^{(p)})} \hat{U}_j^{(\pi^{(p)})} \right) \text{FT}^\dagger , \tag{142}$$

where it is understood that the Fourier transform is performed locally at each site. The Hamiltonian is now in a form suitable for the modified LCU procedure.

First, the PREP oracle is used to encode the real positive coefficients $\beta_i^{(\varphi)}$ and $\beta_j^{(\pi^{(p)})}$. The new SEL

oracle is shown in Fig. 8 and is given by

$$\begin{aligned} \text{SEL} = &(\mathbb{1}_a \otimes \text{FT}) \\ &\otimes \left(\sum_{j=0}^{M_\pi - 1} |j\rangle_a \langle j|_a \otimes \hat{U}_j^{(\pi^{(p)})} \right) (\mathbb{1}_a \otimes \text{FT}^\dagger) \\ &+ \sum_{i=0}^{M_\varphi - 1} |i + M_\pi\rangle_a \langle i + M_\pi|_a \otimes \hat{U}_i^{(\varphi)} , \end{aligned} \tag{143}$$

where, again, it is understood that the Fourier transform is performed locally at each site. It is important to note that the modified SEL operator still satisfies $\text{SEL}^2 = \mathbb{1}$, which enables the use of the efficient construction of W_H.

Note that the SEL oracle prepared using this method encodes the same operators as the SEL oracle prepared using the naive LCU procedure in Sec. IV C 1; the only difference is how it is implemented. This different implementation, however, results in a more favorable scaling for $\mathsf{N_P}$. This can be understood by noting that the $\mathsf{N_P}$ associated with the momentum Hamiltonian is reduced to $\mathsf{N_P}^{(\pi^{(p)})} = \mathcal{O}(\mathsf{n_q}^2)$, implying that the value of $\mathsf{N_P}$ is dominated by the $\hat{\varphi}^4$ term. Substituting $\mathsf{N_P} = \mathcal{O}(\mathsf{n_q}^4)$, $\mathsf{c_P} = \mathcal{O}(\lambda 4^{\mathsf{n_q}})$ and $\mathsf{N_H} = \mathcal{O}(\mathsf{N_\Lambda})$ into the result in Theorem 17 gives the gate complexity

$$\chi = \mathcal{O}\left[\mathsf{N_\Lambda} \mathsf{n_q}^4 \log(\mathsf{N_\Lambda} \mathsf{n_q}^4) \left(\mathsf{N_\Lambda} \mathsf{n_q}^4 (\lambda 4^{\mathsf{n_q}}) t + \log(1/\varepsilon) \right) \right], \tag{144}$$

which has a more favorable scaling with $\mathsf{n_q}$ than the naive LCU implementation in Sec. IV C 1. Since $\text{SEL}^2 = \mathbb{1}$, the controlled call to W_H can be constructed in the same way as the naive LCU case in Sec. IV C 1.
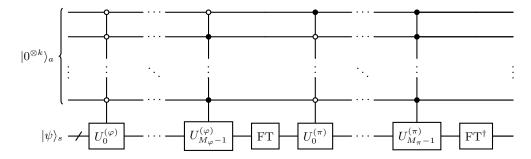
Figure 8. Circuit for SEL oracle in Eq. (143). As compared to $\mathcal{O}(3^{n_q})$ and $\mathcal{O}(n_q)$ in the usual LCU implementation from Sec. IV C 1, the usage of Fourier Transform described in Sec. IV C 2 allows one to reduce the circuit depth and the number of ancillary qubits to $\mathcal{O}(n_q^4)$ and $\mathcal{O}(\log n_q)$, correspondingly. Note that while in the usual LCU procedure the gate complexity $\mathcal{O}(3^{n_q})$ stems from the number of terms in $\hat{H}_\pi$, upon adding the Fourier Transform to the circuit, the $\mathcal{O}(n_q^4)$ complexity stems from the $\hat{\varphi}^4$ term in $\hat{H}_\varphi$.

D. HHKL

In order to apply the HHKL results formulated in our conventions in Theorem 23 and Theorem 24, note that we must scale down the Hamiltonian (119) by the appropriate factor that brings the local terms to at most unit spectral norm, and simultaneously rescale evolution time up by the same factor. This factor is precisely the maximum spectral norm of any local term $\tilde{H}_J$,

$$
\max_{J \in S} \|\tilde{H}_J\|
$$
$$
= \left\| \frac{1}{2}\hat{\pi}_i^2 + \frac{1}{2}m^2\hat{\varphi}_i^2 + \frac{\lambda}{4!}\hat{\varphi}_i^4 + \sum_{j \in \langle ij \rangle} \frac{(\hat{\varphi}_j - \hat{\varphi}_i)^2}{2} \right\|
$$
$$
= \mathcal{O}(N_P c_P) = \mathcal{O}(n_q^4(\lambda 4^{n_q})),
$$
$$
\tag{145}
$$

where we have used $N_P = \mathcal{O}(n_q^4)$ and $c_P = \mathcal{O}(\lambda 4^{n_q})$, and where any lattice site with index i in the interior of the lattice (or on the boundary if there are no sites in the interior) can be chosen to compute the spectral norm, and the sum runs over all lattice sites j neighboring i. We therefore make the rescalings

$$
\tilde{H} = \frac{H}{\max_{J \in S} \|H_J\|} \tag{146}
$$
$$
\tilde{t} = t \max_{J \in S} \|H_J\| = \mathcal{O}(4^{n_q} n_q^4 \lambda t). \tag{147}
$$

For HHKL built out of local Trotter operations, Theorem 23 implies a gate complexity

$$
\chi = \mathcal{O}\left[n_q^4 (N_\Lambda n_q^4 (\lambda 4^{n_q}) t)^{1 + \frac{1}{p}} \varepsilon^{-\frac{1}{p}} \right]. \tag{148}
$$

where we substituted $c_P = \mathcal{O}(\lambda 4^{n_q})$ and $N_P = \mathcal{O}(n_q^4)$.

Similarly, for HHKL built out of local QSP oper-

ations, Theorem 24 then implies a gate complexity

$$
\chi = \mathcal{O}\Big[(n_q^4)^2 N_\Lambda (\lambda 4^{n_q}) t \log^d \left(N_\Lambda t / \varepsilon \right)
$$
$$
\times \log \left(n_q^4 \log \left(N_\Lambda t / \varepsilon \right) \right) \Big]. \tag{149}
$$

To conclude our discussion on HHKL scalings, we again stress an important point. As we have previously remarked, while the purely asymptotic scaling of the Trotter-based HHKL gate count is worse than that for QSP-based HHKL, the difference is not striking when the Trotter order p is taken to be large. This allows for the possibility of lower constant prefactors due to the smaller overhead involved in implementing Trotter operations compared to QSP operations, and may even be a prudent choice for a more practical implementation on near-term devices, if the exact gate counts follow suit before asymptotics take over.

E. Numerical results.

In the previous section we worked out the asymptotic gate count scaling for several time-evolution algorithms. While these expressions are helpful for identifying which methods have the best scaling in a particular parameter, the practical question of the size of the prefactors requires a dedicated numerical study, which we perform in this section. In particular, we first compare the cost of the two BE methods described in the previous section. For the comparison, we choose to compare the cost of constructing a single controlled call to the W_H operator. This choice allows a more direct comparison of the cost of each method as in some cases significant circuit simplifications occur compared to naively controlling every gate in the W_H operator. From there, we turn to time evolution and compare the cost of using QSP to

that of a 4th order PF, focusing on the scaling with time t and error ε.

We choose to focus on a single site Hamiltonian given by

$$\hat{H} = \frac{1}{2}\hat{\pi}^2 + \frac{1}{2}\hat{\varphi}^2 + \frac{\lambda}{4!}\hat{\varphi}^4 \,, \qquad (150)$$

where we set $\lambda = 32$ (similarly to [11]) as an instance of the strongly coupled regime where perturbation theory breaks down. While this will not provide an explicit scaling with the number of lattice sites, analyzing the single site case in detail will provide intuition for what values of t and ε QSP outperforms PF methods for $N_\Lambda > 1$.

Before comparing gate counts of the various methods, we describe the details of constructing the circuits. All gate counts were found using QISKIT's transpile function with an optimization level of 1; higher optimization levels result in the dropping of gates with small arguments and were therefore not used. We use the universal gate-set containing R_x, R_z, and CNOT gates. When implementing multi-controlled gates, we compared the cost using two different methods. The first uses QISKIT's built-in functionality to decompose the multi-controlled gates into $\mathcal{O}(M^2)$ CNOT gates, where M is the number of qubits being controlled on. The second method uses $\mathcal{O}(M)$ additional work qubits to decompose the multi-controlled gates into $\mathcal{O}(M)$ gates [73]. For each value of n_q, we compare the two and choose the method that results in the smallest number of gates. Lastly, it is important to note that, when implementing multi-controlled gates, global phases must also be controlled. Because using LCU requires the coefficients in the PREP oracle to be real and positive, any overall phase must necessarily be absorbed into the individual operators in the SEL oracle. Controlling these phases requires an additional controlled single-qubit rotation. Alternatively, one could use the generalized LCU method to allow the coefficients in the PREP oracle to be complex and set the phase of the operators in the SEL oracle to have zero overall phase (see e.g. [59]). While this avoids having to control the phases, it leads to a more complicated PREP oracle. Additionally, because the generalized LCU method uses circuits of the form $\widehat{\text{PREP}} \times \text{SEL} \times \text{PREP}$ with $\widehat{\text{PREP}} \times \text{PREP} \neq \mathbb{1}$, one will have to also control both PREP and $\widehat{\text{PREP}}$ in the controlled W_H circuits in Fig. 7. For these reasons, we use the former method to control the overall phases.

We now compare the cost of implementing the controlled W_H operator using the two different BE methods. The top and middle plots in Fig. 9 show the CNOT gate count and rotation gate count, respectively, as a function of n_q. The blue points show the gate counts using the LCU method where one naively decomposes the full digitized Hamiltonian into Pauli strings as described in Sec. IV C 1. As expected, this method scales least favorably with n_q. As previously discussed in Sec. IV C 2, this poor gate scaling with n_q can be improved by utilizing the Fourier Transform to implement the momentum component of the Hamiltonian. This improved scaling can be seen by looking at the orange squares in the top and middle plots in Fig. 9.

The bottom plot in Fig. 9 shows the number of ancillary qubits required to implement a controlled call to W_H for each method. As expected from the $\mathcal{O}(n_q)$ scaling, using the naive LCU method requires the most ancillary qubits. Using LCU combined with the FT reduces the scaling to $\mathcal{O}(\log n_q)$. Note that, for both methods, the overall coefficient in the number of ancillary qubits is large due to needing additional work qubits to efficiently implement multi-controlled gates.

Taken together, the gate count and ancillary qubit count comparisons show that using LCU combined with the FT is superior to the naive LCU method for all values of n_q.

Now that we have identified the optimal method for preparing BEs, we turn to the cost of performing time evolution. The first method we implement is a 4th order PF. The second uses QSP, where the BE is prepared using the BE method that minimizes the number of rotation gates, i.e., using LCU combined with the FT. The top (bottom) plot in Fig. 10 show the rotation (CNOT) gate count for both methods as a function of evolution time t and error ε for $n_q = 3$. The value of the scale factor α for the QSP data is $\alpha = 134.5$. Due to QSP's large overall prefactor from implementing the BE, using the 4th order PF requires less rotation and CNOT gates for errors $\varepsilon \gtrsim 10^{-9}$. For errors $\varepsilon \lesssim 10^{-9}$, QSP wins due to its exponentially better scaling with respect to ε.

It is interesting to note that the QSP gate count scaling is almost independent of the desired error ε. This is due to the fact that the αt term in the asymptotic cost is significantly larger than the $\log(1/\varepsilon)$ term (see e.g. Eq. (139)). These findings therefore show that the additional cost of implementing time evolution to floating point precision using QSP is negligible compared to implementing time evolution approximately (not accounting for the increased cost from requiring higher precision rotation gates). In Sec. V we discuss the implications of these results for multi-site systems.

V. DISCUSSION

In this section we discuss how results of Sec. III provide guidance for choosing a simulation algorithm
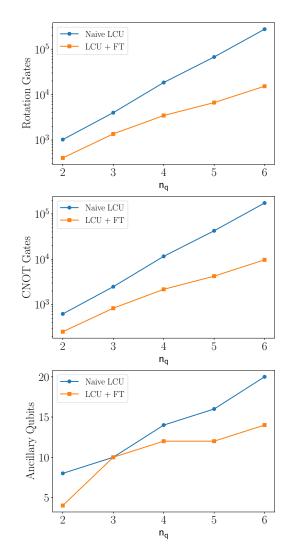
Figure 9. The top, middle, and bottom plots show the CNOT gate count, rotation gate count, and number of ancillary qubits needed to prepare a controlled call to the W_H operator corresponding to the single site Hamiltonian in Eq. 150 as a function of n_q. Blue circles correspond to using the naive LCU method described in Sec. IV C 1. Orange squares correspond to using LCU combined with the Fourier Transform as described in Sec. IV C 2. Using LCU combined with the FT results in an exponential improvement with n_q in both the gate count and number of ancillary qubits required.

		ε	t	N_H
		Non-geometrically-local		
a)	PF	$\varepsilon^{-\frac{1}{p}}$	$t^{1+\frac{1}{p}}$	$\mathbf{N_H}^{1+\frac{1}{p}}\mathbf{N_{ind}}$
b)	QSP+PF	$\varepsilon^{-\frac{1}{p}}\log(1/\varepsilon)$	$t^{1+\frac{1}{p}}$	$N_H^{1+\frac{1}{p}}N_{ind}$ $\times \log(N_H N_{ind})$
c)	QSP	$\mathbf{\log(1/\varepsilon)}$	$\mathbf{t}$	$\mathbf{N_H}^2 \mathbf{\log(N_H)}$

		ε	t	N_Λ
		Geometrically-local		
d)	PF	$\varepsilon^{-\frac{1}{p}}$	$t^{1+\frac{1}{p}}$	$N_\Lambda^{1+\frac{1}{p}}$
e)	PF+HHKL	$\varepsilon^{-\frac{1}{p}}$	$t^{1+\frac{1}{p}}$	$N_\Lambda^{1+\frac{1}{p}}$
f)	QSP+PF	$\varepsilon^{-\frac{1}{p}}\log(1/\varepsilon)$	$t^{1+\frac{1}{p}}$	$N_\Lambda^{1+\frac{1}{p}}\log(N_\Lambda)$
g)	QSP	$\mathbf{\log(1/\varepsilon)}$	$\mathbf{t}$	N_Λ^2
h)	QSP+HHKL	$\mathbf{\log(1/\varepsilon)}$	$\mathbf{t}$	$\mathbf{N_\Lambda[\log(N_\Lambda)]^d}$ $\times \mathbf{\log\log(N_\Lambda)}$

Table V. Asymptotic dependence ($\mathcal{O}$ omitted) on major problem parameters for algorithms in Tables III and IV. Boldface indicates best asymptotic scaling among applicable methods. While N_{ind} is constant for geometrically-local theories, it typically scales polynomially with N_Λ otherwise. For most non-geometrically-local theories, the order p can be chosen large enough so that PFs demonstrate better scaling with N_Λ than QSP.

for the particular physical model and then examine the results of Sec. IV.

The major parameters of interest defining the asymptotic complexities of time evolution algorithms are the evolution time t, the error of the time-evolved state ε, as well as some measure of Hamiltonian size/complexity. For lattice systems, this can be the number of terms in the Hamiltonian N_H. For a particular theory, N_H can also be expressed in terms of

lattice size N_Λ. The dependence of N_H on N_Λ is linear for geometrically-site-local Hamiltonians and is, more generally, polynomial. For algorithms based on PFs, one additionally needs to consider a parameter N_{ind} related to the induced norm of the Hamiltonian. The asymptotic dependence on individual parameters for algorithms discussed in Secs. II and III is summarized in Table V.

The parameter for which the difference in asymptotic dependence is most striking for PF- and QSP-based algorithms is the error ε. Nevertheless, using algorithms with $\log(1/\varepsilon)$ dependence may not necessarily be an optimal choice for several reasons. First, their asymptotic cost comes with large constant prefactors stemming from the implementation of the BE subroutine. While progress is being made in this direction, the value of ε at which using such techniques clearly becomes advantageous is $\varepsilon \sim 10^{-9}$, which is much smaller than uncertainties coming from other sources (e.g., from digitizing a theory with bosonic degrees of freedom, finite lattice spacing and volume effects, to name just a few). Second, as discussed below, in some scenarios PF-based algorithms have better dependence on problem size than those based on QSP. Third, and least importantly (in the fault-
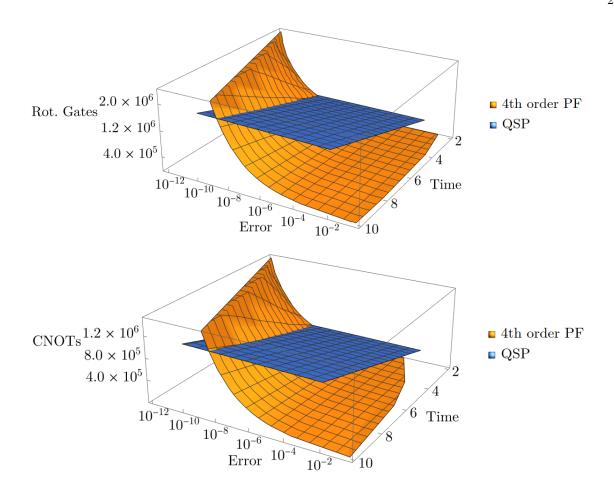
Figure 10. The top (bottom) plot shows the rotation (CNOT) gate count for implementing the time evolution operator for the single site Hamiltonian in Eq. (119) with $n_q = 3$. Results are shown as a function of time t and error ε. The orange and blue colored surfaces show results using a 4th order PF and QSP, respectively. The BE used in the QSP calculation was implemented using the method in Sec. IV C 2. The PF requires less rotation and CNOT gates for errors $\varepsilon \gtrsim 10^{-9}$. For error tolerances $\varepsilon \lesssim 10^{-9}$, QSP wins due to its exponentially better scaling with ε. Furthermore, the QSP gate count is almost independent of the error due to the optimal scaling with ε.

tolerant regime), PF based algorithms do not require the usage of ancillary qubits, while QSP-based algorithms typically require a logarithmic (in problem parameters) number of those.

Recent research [40] indicates that the error tolerance at which near-optimal algorithms start outperforming the PF-based ones is highly problemdependent. In particular, when simulating lattice gauge theories, the complexity of PF-based simulation is likely to scale exponentially with the number of colors [79, 80], while the near-optimal algorithms can have polynomial scaling with the number of colors, which is achieved by leveraging the sparsity of gauge field operators. As a result, the usage of nearoptimal techniques may lead to tremendous savings starting at error rates as low as $\varepsilon = 10^{-1}$ [40].

In terms of the asymptotic dependence on problem size, a clear winner is the HHKL algorithm, assum-

ing that for implementing the time evolution of local blocks, an algorithm with $\log(1/\varepsilon)$ scaling is chosen, such as QSP. Note, however, that for higher-order PFs [26] the improvement reached by the HHKL algorithm in terms of problem size may not be particularly pronounced, especially, unless a very high precision is required, as discussed above.

For non-geometrically-local Hamiltonians, the situation is largely determined by the Hamiltonian structure defining the dependence of N_{ind} on N_Λ. In particular, in the second-quantized formulation, this dependence will be defined by the choice of single-particle basis states (plane waves [9, 38], wavelets [81], harmonic oscillator eigenstates [10], etc.). Importantly, whether the Hamiltonian is geometrically-local or not, one can imagine scenarios in which, at a fixed value of ε, switching from PF to QSP would make sense as one increases the problem

size. However, given the sharp difference in the dependence on ε for these approaches, such scenarios are not highly probable.

Lastly, we note that the dependence on t is likely to not be a crucial parameter in choosing the simulation algorithm. Nevertheless, a situation in which considering longer times can motivate one to switch from PF to QSP is possible as well.

In addition to factors considered in this work, other considerations should also be taken into account. Most notably, the compatibility of methods based on BE with general sparse Hamiltonians and modern simulation techniques [82–84] and the potential for applying the PF-based methods on early-fault-tolerant quantum computers [60, 85].

Construction of efficient block encodings (BEs) is critical for improving the efficiency of QSP-based algorithms. In [11, 19] it was shown how digitizing a bosonic degree of freedom in the eigenbases of position/momentum operator is advantageous from the perspective of using PF algorithms. There, the key factors are a) an economical qubit representation of arbitrary powers of position/momentum operators in their respective eigenbases and b) the efficiency of switching between the position and momentum bases with the aid of Fourier Transform implemented on the quantum computer. In Sec. IV C 2 we showed how this approach can be equally well applied when constructing a BE with LCU.

VI. CONCLUSION

In this work we studied various approaches to simulating time evolution in application to Hamiltonian Lattice Field Theories (HLFTs). We started off by providing a review of existing techniques based on Product Formula (PF) splitting [27], Quantum Signal Processing (QSP) [29] and Qubitization [30], Linear Combination of Unitaries (LCU) [35], as well as the HHKL algorithm [34]. Following that, we derived asymptotic costs of these methods in terms of parameters describing site-local and geometrically-site-local Hamiltonians arising in HLFT, see Tables III and IV. While in this work we chose to contrast the most accessible and widely used algorithms based on PF with the methods having the best asymptotic scaling (QSP, HHKL), we note that there exist intermediate approaches. Most notably, this includes the direct usage of Taylor series and LCU without QSP [86].

Next, we focused on a particular type of HLFT, the quartic scalar field theory defined on a hypercubic lattice. Our specific choice was guided by the fact that all the fundamental interactions are mediated by bosons, and from the computational per-

spective the scalar field theory is largely reminiscent of lattice gauge theories [41–46, 87]. Besides this, an extensive volume of literature exists focusing on simulations of fermionic systems [36, 37, 57, 88–95]. Developing simulation algorithms for systems combining particles of both statistics is an exciting direction which was pursued in [74, 75, 96] where authors assumed the usage of PF-based methods, and is currently investigated in the context of nearly-optimal techniques [40].

For the scalar field theory, we worked out the query complexities of various simulation methods as well provided explicit gate counts (not the upper bounds) for the case of simulating the system on a single lattice site using PF and QSP. Next, we focused on block encoding a single site in scalar field theory and left to further work additional improvements, such as reducing the cost of SEL operator from $\mathcal{O}(N_\Lambda \log N_\Lambda)$ to $\mathcal{O}(N_\Lambda)$ [37] or taking into account the structure of coefficients in the Hamiltonian operator while/in constructing PREP operator. We developed an improved version of the LCU block encoding for digitized bosons [11, 19] which utilizes the Quantum Fourier Transform circuit for switching between the field and momentum eigenbases, see Fig. 8. We then provided explicit gate counts for simulating time evolution which showed that PF-based simulation works exceptionally well for simple systems, such as anharmonic oscillator, and outperforms the QSP-based algorithm for $\varepsilon \gtrsim 10^{-9}$. We note, however, that considering more complex models (such as non-Abelian lattice gauge theories) and larger lattice system sizes may result in near-term algorithms outperforming PF for error tolerances as little as $\varepsilon \lesssim 10^{-1}$ [40].

Another important distinction between PF and QSP methods is the ease of quoting accurate error bounds for the final result. Starting with QSP methods, all that is needed to place an upper bound on the error is the scale factor α, the time t, and the degree of the polynomial approximation. While the error determined in this way is only an upper bound, because QSP methods have optimal scaling $\log(1/\varepsilon)$ in the error, the cost of overestimating the error is low compared to the contribution stemming from the αt term in Eq. (65).

For PF methods on the other hand, while rigorous error bounds also exist, they are known to generally significantly overestimate the actual error [27]. Choosing the value of the Trotter number r based on these loose error bounds would therefore result in significant extra computational costs due to the suboptimal $(1/\varepsilon)^{1/p}$ scaling of PF methods. Tightening error bounds in this case implies either performing laborious problem-specific commutator calculations [27, 96] or performing numerical simulations at

several values of the Trotter number in order to extrapolate the results. As shown in Ref. [97], since using PFs to approximate the time evolution operator is equivalent to introducing a temporal lattice with spacing a_t, such an extrapolation would necessitate a renormalization procedure with a costly scale setting procedure at each value of a_t.[15]

ACKNOWLEDGEMENTS

---

[15] As was also shown in Ref. [97], to avoid a double extrapolation $a \to 0$ and $a_t \to 0$, one should work at fixed anisotropy $\xi \equiv a/a_t$ and take the limit $a \to 0$.

[1] Z. Davoudi *et al.*, in *Snowmass 2021* (2022) arXiv:2209.10758 [hep-lat].

[2] C. Morningstar, in *21st Annual Hampton University Graduate Studies Program (HUGS 2006)* (2007) arXiv:hep-lat/0702020.

[3] E. Y. Loh, J. E. Gubernatis, R. T. Scalettar, S. R. White, D. J. Scalapino, and R. L. Sugar, Phys. Rev. B **41**, 9301 (1990).

[4] G. Pan and Z. Y. Meng, (2022), 10.1016/B978-0-323-90800-9.00095-0, arXiv:2204.08777 [cond-mat.str-el].

[5] J. B. Kogut and L. Susskind, Phys. Rev. D **11**, 395 (1975).

[6] S. D. Drell, H. R. Quinn, B. Svetitsky, and M. Weinstein, Physical Review D **19**, 619 (1979).

[7] W. A. Bardeen and R. B. Pearson, Physical Review D **14**, 547 (1976).

[8] W. A. Bardeen, R. B. Pearson, and E. Rabinovici, Phys. Rev. D **21**, 1037 (1980).

[9] J. Liu, Z. Li, H. Zheng, X. Yuan, and J. Sun, Mach. Learn. Sci. Tech. **3**, 045030 (2022), arXiv:2109.05547 [quant-ph].

[10] J. P. Vary, H. Honkanen, J. Li, P. Maris, S. J. Brodsky, A. Harindranath, G. F. de Teramond, P. Sternberg, E. G. Ng, and C. Yang, Phys. Rev. C **81**, 035205 (2010), arXiv:0905.1411 [nucl-th].

[11] N. Klco and M. J. Savage, Phys. Rev. A **99**, 052335 (2019), arXiv:1808.10378 [quant-ph].

[12] H. C. Pauli and S. J. Brodsky, Phys. Rev. D **32**, 1993 (1985).

[13] H.-C. Pauli and S. J. Brodsky, Physical Review D **32**, 2001 (1985).

[14] S. J. Brodsky, H.-C. Pauli, and S. S. Pinsky, Phys. Rept. **301**, 299 (1998), arXiv:hep-ph/9705477.

[15] S. P. Jordan, H. Krovi, K. S. M. Lee, and J. Preskill, Quantum **2**, 44 (2018), arXiv:1703.00454 [quant-ph].

[16] S. P. Jordan, K. S. M. Lee, and J. Preskill, Science **336**, 1130 (2012), arXiv:1111.3633 [quant-ph].

[17] R. P. Feynman, Int. J. Theor. Phys. **21**, 467 (1982).

[18] C. W. Bauer *et al.*, PRX Quantum **4**, 027001 (2023), arXiv:2204.03381 [quant-ph].

[19] S. P. Jordan, K. S. M. Lee, and J. Preskill, Quant. Inf. Comput. **14**, 1014 (2014), arXiv:1112.4833 [hep-th].

[20] S. Lloyd, Science **273**, 1073 (1996).

[21] C. Zalka, Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences **454**, 313 (1998).

[22] S. Wiesner, (1996), arXiv:quant-ph/9603028.

[23] J. E. Cohen, S. Friedland, T. Kato, and F. P. Kelly, Linear Algebra and its Applications **45**, 55 (1982).

[24] M. Suzuki, Communications in Mathematical Physics **51**, 183 (1976).

[25] M. Suzuki, J. Math. Phys. **32**, 400 (1991).

[26] A. M. Childs and Y. Su, Physical review letters **123**, 050503 (2019).

[27] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu, Phys. Rev. X **11**, 011020 (2021), arXiv:1912.08854 [quant-ph].

[28] N. Hatano and M. Suzuki, Lect. Notes Phys. **679**, 37 (2005), arXiv:math-ph/0506007.

[29] G. H. Low and I. L. Chuang, Phys. Rev. Lett. **118**, 010501 (2017), arXiv:1606.02685 [quant-ph].

[30] G. H. Low and I. L. Chuang, Quantum **3**, 163 (2019), arXiv:1610.06546 [quant-ph].

[31] S. Zeytinoğlu and S. Sugiura, Phys. Rev. Res. **6**, 013003 (2024), arXiv:2201.04665 [quant-ph].

[32] D. Motlagh and N. Wiebe, (2023), arXiv:2308.01501 [quant-ph].

[33] Y. Kikuchi, C. Mc Keever, L. Coopmans, M. Lubasch, and M. Benedetti, npj Quantum Inf. **9**, 93 (2023), arXiv:2303.05533 [quant-ph].

[34] J. Haah, M. B. Hastings, R. Kothari, and G. H. Low, (2018), 10.1137/18M1231511, arXiv:1801.03922 [quant-ph].

[35] A. M. Childs and N. Wiebe, Quant. Inf. Comput. **12**, 0901 (2012), arXiv:1202.5822 [quant-ph].

[36] B. Toloui and P. J. Love, (2013), arXiv:1312.2579 [quant-ph].

[37] R. Babbush, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, A. Paler, A. Fowler, and H. Neven, Phys. Rev. X **8**, 041015 (2018).

[38] M. Kreshchuk, W. M. Kirby, G. Goldstein, H. Beauchemin, and P. J. Love, Phys. Rev. A **105**, 032418 (2022), arXiv:2002.04016 [quant-ph].

[39] W. M. Kirby, S. Hadi, M. Kreshchuk, and P. J. Love, Phys. Rev. A **104**, 042607 (2021), arXiv:2105.10941 [quant-ph].

[40] M. Rhodes, M. Kreshchuk, and S. Pathak, arXiv preprint arXiv:2405.10416 (2024).

[41] J. F. Haase, L. Dellantonio, A. Celi, D. Paulson, A. Kan, K. Jansen, and C. A. Muschik, Quantum **5**, 393 (2021), arXiv:2006.14160 [quant-ph].

[42] Y. Ji, H. Lamm, and S. Zhu (NuQS), Phys. Rev. D **102**, 114513 (2020), arXiv:2005.14221 [hep-lat].

[43] C. W. Bauer and D. M. Grabowska, Phys. Rev. D **107**, L031503 (2023), arXiv:2111.08015 [hep-ph].

[44] C. W. Bauer, I. D'Andrea, M. Freytsis, and D. M. Grabowska, (2023), arXiv:2307.11829 [hep-ph].

[45] D. M. Grabowska, C. Kane, B. Nachman, and C. W. Bauer, (2022), arXiv:2208.03333 [quant-ph].

[46] C. Kane, D. M. Grabowska, B. Nachman, and C. W. Bauer, (2022), arXiv:2211.10497 [quant-ph].

[47] M. B. Hastings, D. Wecker, B. Bauer, and M. Troyer, (2014), arXiv:1403.1539 [quant-ph].

[48] D. Poulin, M. B. Hastings, D. Wecker, N. Wiebe, A. C. Doberty, and M. Troyer, Quant. Inf. Comput. **15**, 0361 (2015), arXiv:1406.4920 [quant-ph].

[49] A. Tranter, P. J. Love, F. Mintert, and P. V. Coveney, Journal of chemical theory and computation **14**, 5617 (2018).

[50] Y. Hu, F. Meng, X. Wang, T. Luan, Y. Fu, Z. Zhang, X. Zhang, and X. Yu, Quantum Sci. Technol. **7**, 045001 (2022).

[51] A. Tranter, P. J. Love, F. Mintert, N. Wiebe, and P. V. Coveney, Entropy **21**, 1218 (2019).

[52] A. T. Schmitz, N. P. D. Sawaya, S. Johri, and A. Y. Matsuura, (2021), arXiv:2103.08602 [quant-ph].

[53] J. Ostmeyer, J. Phys. A **56**, 285303 (2023), arXiv:2211.02691 [quant-ph].

[54] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Commun. Math. Phys. **270**, 359 (2007), arXiv:quant-ph/0508139.

[55] X.-Y. Huang, L. Yu, X. Lu, Y. Yang, D.-S. Li, C.-W. Wu, W. Wu, and P.-X. Chen, "Qubitization of Bosons," (2021), arXiv:2105.12563 [quant-ph].

[56] A. Alexandru, P. F. Bedaque, A. Carosso, M. J. Cervia, and A. Sheng, Phys. Rev. D **107**, 034503 (2023), arXiv:2209.00098 [hep-lat].

[57] R. Babbush, D. W. Berry, J. R. McClean, and H. Neven, npj Quantum Inf. **5**, 92 (2019).

[58] D. Camps and R. Van Beeumen, in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, 2022) pp. 104–113.

[59] L. Lin, (2022), arXiv:2201.08309 [quant-ph].

[60] Y. Dong, L. Lin, and Y. Tong, PRX Quantum **3**, 040305 (2022), arXiv:2204.05955 [quant-ph].

[61] R. Babbush, D. W. Berry, Y. R. Sanders, I. D. Kivlichan, A. Scherer, A. Y. Wei, P. J. Love, and A. Aspuru-Guzik, Quantum Sci. Technol. **3**, 015006 (2017).

[62] W. Kirby, M. Motta, and A. Mezzacapo, Quantum **7**, 1018 (2023), arXiv:2208.00567 [quant-ph].

[63] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (2019) pp. 193–204.

[64] I. Novikau, E. A. Startsev, and I. Y. Dodin, Phys. Rev. A **105**, 062444 (2022), arXiv:2112.06086 [physics.plasm-ph].

[65] G. H. Low, T. J. Yoder, and I. L. Chuang, Phys. Rev. X **6**, 041067 (2016).

[66] H. H. S. Chan, D. M. n. Ramo, and N. Fitzpatrick, (2023), arXiv:2303.06161 [quant-ph].

[67] C. F. Kane, N. Gomes, and M. Kreshchuk, (2023), arXiv:2310.13757 [quant-ph].

[68] L. Lin and Y. Tong, Quantum **4**, 372 (2020), arXiv:2002.12508 [quant-ph].

[69] C. F. Kane, S. Hariprakash, N. S. Modi, M. Kreshchuk, and C. W. Bauer, ""Block encoding by signal processing (manuscript in preparation)"," (2024).

[70] E. H. Lieb and D. W. Robinson, Commun. Math. Phys. **28**, 251 (1972).

[71] V. V. Shende, S. S. Bullock, and I. L. Markov, IEEE Trans. Comput. Aided Design Integr. Circuits Syst. **25**, 1000 (2006).

[72] M. Plesch and v. Brukner, Phys. Rev. A **83**, 032302 (2011).

[73] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2012).

[74] A. Macridin, P. Spentzouris, J. Amundson, and R. Harnik, Phys. Rev. Lett. **121**, 110504 (2018), arXiv:1802.07347 [quant-ph].

[75] A. Macridin, P. Spentzouris, J. Amundson, and R. Harnik, Phys. Rev. A **98**, 042312 (2018), arXiv:1805.09928 [quant-ph].

[76] A. Macridin, A. C. Y. Li, S. Mrenna, and P. Spentzouris, Phys. Rev. A **105**, 052405 (2022), arXiv:2108.10793 [quant-ph].

[77] T. Farrelly and J. Streich, (2020), arXiv:2002.02643 [quant-ph].

[78] C. W. Bauer, M. Freytsis, and B. Nachman, Phys. Rev. Lett. **127**, 212001 (2021), arXiv:2102.05044 [hep-ph].

[79] A. Kan and Y. Nam, (2021), arXiv:2107.12769 [quant-ph].

[80] Z. Davoudi, A. F. Shaw, and J. R. Stryker, Quantum **7**, 1213 (2023), arXiv:2212.14030 [hep-lat].

[81] G. K. Brennen, P. Rohde, B. C. Sanders, and S. Singh, Phys. Rev. A **92**, 032315 (2015), arXiv:1412.0750 [quant-ph].

[82] G. H. Low and N. Wiebe, (2018), arXiv:1805.00675 [quant-ph].

[83] M. Kieferová, A. Scherer, and D. W. Berry, Phys. Rev. A **99**, 042314 (2019).

[84] Y. Tong, V. V. Albert, J. R. McClean, J. Preskill, and Y. Su, Quantum **6**, 816 (2022), arXiv:2110.06942 [quant-ph].

[85] L. Lin and Y. Tong, PRX Quantum **3**, 010318 (2022), arXiv:2102.11340 [quant-ph].

[86] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Phys. Rev. Lett. **114**, 090502 (2015), arXiv:1412.4687 [quant-ph].

[87] A. N. Ciavarella and C. W. Bauer, (2024), arXiv:2402.10265 [hep-ph].

[88] S. B. Bravyi and A. Y. Kitaev, Annals Phys. **298**, 210 (2002), arXiv:quant-ph/0003137.

[89] F. Verstraete and J. I. Cirac, J. Stat. Mech. **0509**, P09012 (2005), arXiv:cond-mat/0508353.

[90] A. Tranter, S. Sofia, J. Seeley, M. Kaicher, J. Mc-Clean, R. Babbush, P. V. Coveney, F. Mintert, F. Wilhelm, and P. J. Love, International Journal of Quantum Chemistry **115**, 1431 (2015).

[91] R. Babbush, J. McClean, D. Wecker, A. Aspuru-Guzik, and N. Wiebe, Phys. Rev. A **91**, 022311 (2015).

[92] A. Aspuru-Guzik, P. J. Love, A. Y. Wei, I. D. Kivlichan, D. W. Berry, and R. Babbush, New J. Phys. **18**, 033032 (2016).

[93] K. Setia and J. D. Whitfield, J. Chem. Phys. **148**, 164104 (2018).

[94] W. Kirby, B. Fuller, C. Hadfield, and A. Mezzacapo, PRX Quantum **3**, 020351 (2022), arXiv:2109.14465 [quant-ph].

[95] K. Sakamoto, H. Morisaki, J. Haruna, E. Itou, K. Fujii, and K. Mitarai, arXiv preprint arXiv:2311.17388 (2023).

[96] J. D. Watson, J. Bringewatt, A. F. Shaw, A. M. Childs, A. V. Gorshkov, and Z. Davoudi, (2023), arXiv:2312.05344 [quant-ph].

[97] M. Carena, H. Lamm, Y.-Y. Li, and W. Liu, Phys. Rev. D **104**, 094519 (2021), arXiv:2107.01166 [hep-lat].

Appendix A: General qubitization procedure

In all the cases considered in this work the quantum walk operator W_H can be constructed from the block encoding U_H using the recipe given in Eq. (47). For completeness, below we provide a general approach from [30] which allows one to assemble the circuit for the W_H operator using an ancillary qubit and two controlled calls to U_H. This is necessary in cases when one is unable to find an operator S satisfying Eq. (49).

The general construction requires adding one qubit $|q\rangle$ to the ancillary Hilbert space, such that $\mathcal{H}_a \to \mathcal{H}_{qa} = \mathcal{H}_q \otimes \mathcal{H}_a$. In this new ancillary Hilbert space we define

$$|g\rangle_{qa} = |+\rangle_q |g\rangle_a \,, \tag{A1}$$

and the operator W_H is defined on the Hilbert space $\mathcal{H}_{qas} = \mathcal{H}_q \otimes \mathcal{H}_a \otimes \mathcal{H}_s$. As was shown in Section 4
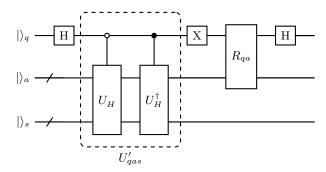

Figure 11. Circuit for the qubitized block encoding W_H, given an arbitrary block encoding U_H. The circuit for R_{qa} is shown in Fig. 12.
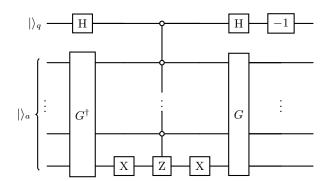

Figure 12. Circuit for the reflection operator R_{qa} defined in Eq. (A3) and used in Eq. (A2), see [59] for alternative implementations. G is the oracle for preparing the state $|g\rangle_a$, i.e. $G|0\rangle_a = |g\rangle_a$. The $\boxed{-1}$ gate adds an overall phase to the circuit (negative sign), and can be implemented, e.g., as XZXZ.

of [30], it is given by

$$\begin{aligned} W_H =&(\mathrm{H}_q \otimes \mathbb{1}_{as})(R_{qa} \otimes \mathbb{1}_s)(\mathrm{X}_q \otimes \mathbb{1}_{as}) \\ &\times U'_{qas}(\mathrm{H}_q \otimes \mathbb{1}_{as}) \,, \end{aligned} \tag{A2}$$

where H_q and X_q are the Hadamard and X gate on the qubit $|q\rangle$, R_{qa} is a standard reflection operator

$$R_{qa} = 2|g\rangle_a|+\rangle_q\langle+|_q\langle g|_a - \mathbb{1}_{qa} \,, \tag{A3}$$

and

$$U'_{qas} = |0\rangle_q\langle 0|_q U_H + |1\rangle_q\langle 1|_q U_H^\dagger \,, \tag{A4}$$

is an operator that applies the U_H or $U_H^\dagger$ operator, depending on the state of $|q\rangle$.

One can show through explicit computation that

$$W_H|g\rangle_{qa}|\lambda\rangle_s = \left(\lambda\,|g\rangle_{qa} - \sqrt{1-\lambda^2}|\perp^\lambda\rangle_{qa}\right)|\lambda\rangle_s\,, \tag{A5}$$

with

$$|\perp^\lambda\rangle_{qa} = |-\rangle_q|\perp^\lambda\rangle_a\,, \tag{A6}$$

verifying that Eq. (39) gives indeed a qubitized BE of H. The circuits for operators W_H, U'_{qas}, and R_{qa} are shown in Figs. 11 and 12.