Block encoding bosons by signal processing

Christopher F. Kane[1], Siddharth Hariprakash[2,3,4], Neel S. Modi[2,3], Michael Kreshchuk[3,5], and Christian W Bauer[3]

[1]Department of Physics, University of Arizona, Tucson, Arizona 85719, USA
[2]Center for Theoretical Physics and Department of Physics, University of California, Berkeley, California 94720, USA
[3]Physics Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA
[4]National Energy Research Scientific Computing Center (NERSC), Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA,
[5]Phasecraft Inc., Washington, DC 20001, USA

Block Encoding (BE) is a crucial subroutine in many modern quantum algorithms, including those with near-optimal scaling for simulating quantum many-body systems, which often rely on Quantum Signal Processing (QSP). Currently, the primary methods for constructing BEs are the Linear Combination of Unitaries (LCU) and the sparse oracle approach. In this work, we demonstrate that QSP-based techniques, such as Quantum Singular Value Transformation (QSVT) and Quantum Eigenvalue Transformation for Unitary Matrices (QETU), can themselves be efficiently utilized for BE implementation. Specifically, we present several examples of using QSVT and QETU algorithms, along with their combinations, to block encode Hamiltonians for lattice bosons, an essential ingredient in simulations of high-energy physics. We also introduce a straightforward approach to BE based on the exact implementation of Linear Operators Via Exponentiation and LCU (LOVE-LCU). We find that, while using QSVT for BE results in the best asymptotic gate count scaling with the number of qubits per site, LOVE-LCU outperforms all other methods for operators acting on up to $\lesssim 11$ qubits, highlighting the importance of concrete circuit constructions over mere comparisons of asymptotic scalings. Using LOVE-LCU to implement the BE, we simulate the time evolution of single-site and two-site systems in the lattice φ^4 theory using the Generalized QSP algorithm and compare the gate counts to those required for Trotter simulation.

Contents

1 Introduction **2**

2 Algorithms **3**
2.1 Block encoding and walk operator 3
2.2 Linear Combination of Unitaries . 4
2.3 Simulating time evolution using Generalized QSP 5
2.4 QSVT 5
2.5 QETU 6

3 Block encodings for a bosonic lattice field theory **7**
3.1 General Form of Hamiltonians of Interest 8
3.2 Implementation of field operators and their functions 9
3.3 LCU to Block Encode Individual Terms 9
3.4 QSVT to Block Encode Individual Terms 10
3.5 QETU to Block Encode Individual Terms 10
3.5.1 Technical details 11
3.5.2 Cost to prepare block-encodings of bosonic operators 11
3.6 LOVE-LCU 12
3.6.1 Main idea 12
3.6.2 Technical details 13
3.7 Constructing full Hamiltonian Block Encoding and Walk Operator 14

4 Numerical demonstration **15**

5 Discussion and Conclusion **19**

Acknowledgements 22

References 22

A Constructing the walk operator 28

B CNOT gate counts for scalar field theory simulation 30

C Scale factor for Block Encodings constructed using LCU 33

1 Introduction

Quantum simulation of quantum many-body physics was the original motivation for the development of quantum computing [14, 40, 68] and continues to be considered one of its most compelling applications. For many areas of interest, such as quantum chemistry [42], nuclear [81], or high-energy physics [12], fault-tolerant devices will be necessary to perform calculations of practical value.

Most quantum simulation proposals involve the stages of state preparation and/or time evolution. Both of these tasks are commonly accomplished with the aid of algorithms based on Product Formulas (PF) [64, 89], an approach often referred to as *Trotterization*. Recent years, however, have witnessed a surge in the development of algorithms based on the so-called *post-Trotter* methods such as Linear Combination of Unitaries (LCU) [27] or Quantum Signal Processing (QSP) [65, 66]. Although the asymptotic cost of these algorithms can achieve optimal or near-optimal dependence on problem parameters, they often come with large constant prefactors compared to asymptotically sub-optimal methods based on Product Formulas (PF) [17, 26, 48, 69, 85, 86]. Consequently, their usage is sometimes limited to highly specific parameter ranges, such as long evolution times or small errors of the final state [47].

An important ingredient of post-Trotter methods is to encode the Hamiltonian directly into a quantum circuit. Given that quantum circuits can only implement unitary, not Hermitian, operators, the Hamiltonian is typically encoded as a sub-block of a larger unitary matrix. This process is called Block Encoding (BE), and is the source for most of the prefactors in the algorithmic cost of post-Trotter algorithms. The commonly used approaches to constructing block encodings include the LCU algorithm and the sparse oracle approach [4, 34, 56, 60, 63]. Highly optimized constructions of BEs based on both approaches have been developed for various types of physical systems [5, 6, 78].

In this work, we present several novel approaches to constructing BEs for a certain class of Hamiltonians. We demonstrate that, while BEs are commonly seen as building blocks for QSP-based algorithms, *QSP-based algorithms themselves can be utilized for the highly efficient construction of BEs*. Our approaches apply to Hamiltonians which can be written as

$$\hat{H} = \sum_{\ell=0}^{N-1} \hat{H}_\ell \,, \tag{1}$$

where each $\hat{H}_\ell$ acts on at most n qubits. Many Hamiltonians have this structure, and we shall term them as *n-site-local* Hamiltonians. We will focus on the situation where each term in the Hamiltonian can be diagonalized via a known and efficiently implementable transformation

$$\hat{H}_\ell = U_\ell^\dagger \hat{H}_\ell^{(D)} U_\ell \,. \tag{2}$$

For some of our methods, we will make a further assumption about the structure of the Hamiltonian, which is satisfied for certain bosonic lattice field theories, which have recently garnered attention for their crucial role in high-energy and low-energy nuclear physics [7–9, 11, 12, 25, 29–31, 34, 36–39, 43, 44, 47, 50, 57, 59, 67, 72, 75–78, 87]. Specifically, we consider theories for which the Hamiltonians are formulated in terms of conjugate operators $[\hat{\varphi}_i, \hat{\pi}_j] = i\delta_{ij}$ at lattice sites i, j. Using $\mathsf{n_q}$ qubits, the operators $\hat{\varphi}_i$ are digitized by sampling at equally spaced values. Since the operator $\hat{\pi}_i$ is conjugate to $\hat{\varphi}_i$, it can be diagonalized using a Fourier transform $\hat{\pi}_i = \widehat{\mathcal{F}}^\dagger \hat{\pi}_i^{(D)} \widehat{\mathcal{F}}$. The Hamiltonian can be written as functions of simple operators $\hat{\varphi}_i$ and $\hat{\pi}_i$, and possibly combinations like $\hat{\varphi}_i - \hat{\varphi}_j$ needed for gradient terms. We will use the general notation $\hat{\xi}_k$ ($k \in \{0, 1, 2\}$) to denote any of the operators

$$\hat{\xi}_0 \in \{\hat{\varphi}_i\} \,, \quad \hat{\xi}_1 \in \{\hat{\pi}_i^{(D)}\} \,, \quad \hat{\xi}_2 \in \{\hat{\varphi}_i - \hat{\varphi}_j\} \,, \tag{3}$$

such that each $\hat{H}_\ell^{(D)}$ is then given by a function of one such operator

$$\hat{H}_\ell^{(D)} = f_{k_\ell}(\hat{\xi}_{k_\ell}) \,. \tag{4}$$

A key property of the operators $\hat{\xi}_{k_\ell}$ we exploit in this work is that each of them can be written as a linear combination of $\mathcal{O}(\mathsf{n_q})$ many Pauli Z gates.

Based on this set up and these assumptions, we present 3 different methods for the BE of the individual Hamiltonian terms $\hat{H}_\ell$, which are then combined via LCU, see Fig. 1. The first method first obtains a block encoding for the operator $\hat{\xi}_{k_\ell}$ and then uses QSVT to construct the BE for $\hat{H}_\ell^{(D)}$. The second method constructs the unitary operator $\mathrm{e}^{-i\tau\hat{\xi}_{k_\ell}}$ (or $e^{-2i\arccos(\hat{\xi}_k/\alpha)}$) and uses it as a building block to construct the BE for $\hat{H}_\ell^{(D)}$ using QETU. The final method, which we call Linear Operators Via Exponentiation and Linear Combination of Unitaries (LOVE-LCU) constructs the unitary operators $\mathrm{e}^{\pm i\arccos(f_{k_\ell}(\hat{\xi}_{k_\ell})/\alpha)}$, from which $\hat{H}_\ell^{(D)}$ can easily be constructed by adding the two terms via LCU. Following the construction of the BE for $\hat{H}_\ell^{(D)}$ we conjugate it by U_ℓ resulting in a BE for $\hat{H}_\ell$.

In Sec. 2, we briefly review the notion of BE as well as the GQSP, QSVT, and QETU algorithms. In Sec. 3, we first review the general formulation of bosonic field theories considered in this work and then discuss the 3 methods in more detail. We also demonstrate that, due to the highly symmetric structure of QSVT, QETU, and LOVE-LCU circuits, the qubitized walk operator can be constructed without ancillary qubits in a straightforward way. In Sec. 4 we use the proposed methods to construct BEs for several models of interest, construct explicit circuits, and provide metrics such as gate count and the number of ancillary qubits. For a single site system, we find that using our methods allows to implement time evolution with less gates than PFs for errors as small as $\epsilon \sim 10^{-2}$. Our conclusions are presented in Sec. 5.

2 Algorithms

In this section, we provide brief reviews of the simulation algorithms and techniques we consider in this work, along with associated circuit diagrams. Readers familiar with these concepts are encouraged to skip directly to Sec. 3.

2.1 Block encoding and walk operator

The key ingredient of most near-optimal simulation algorithms is the construction of the so-called

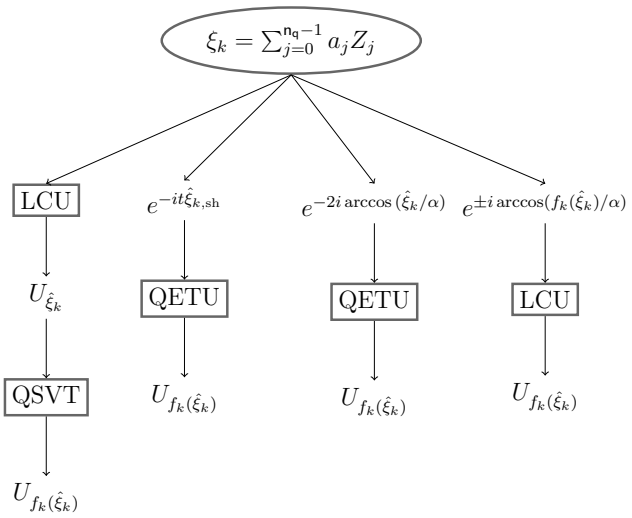

Figure 1: Various methods for block encoding operators in Hamiltonians of scalar field theories (see Sec. 3). In the first two approaches, the cost of constructing the building block is polynomial in $\mathsf{n_q}$, while in the latter two, the cost is exponential in $\mathsf{n_q}$. In the first three approaches, the query depth is determined by the convergence of the polynomial approximation to the desired function. In the last approach, the query depth is constant (two).

Block Encoding (BE) of a given operator. BE typically refers to the embedding of an n-qubit operator $A \in \mathcal{C}^{2^n\times 2^n}$, acting on an n-qubit Hilbert space $\mathcal{H}_s$, in the (without loss of generality) principle block of a larger $(m+n)$-qubit unitary operator $U_A \in \mathcal{C}^{2^{n+m}\times 2^{n+m}}$, acting on a larger Hilbert space $\mathcal{H}_a \otimes \mathcal{H}_s$, where $\mathcal{H}_a$ is the m-qubit ancilla space. The BE U_A has the following form:

$$U_A = \begin{pmatrix} A/\alpha & * \\ * & * \end{pmatrix}, \tag{5}$$

where each $*$ represents some block to be chosen such that U_A is unitary and the quantity α, known as the scale factor, is chosen such that $||A/\alpha||_2 \leq 1$ since any sub-block of a unitary matrix must have its singular values upper bounded by 1. In this case, we call U_A the (α, m)-block-encoding of A. Schematically, we see that for $|0\rangle_a \equiv |0^{\otimes m}\rangle_a \in \mathcal{H}_a$ and $|\psi\rangle_s \in \mathcal{H}_s$ the action of U_A can be written as

$$U_A\,|0\rangle_a\,|\psi\rangle_s = \frac{1}{\alpha}\begin{pmatrix} A\,|\psi\rangle_s \\ * \end{pmatrix}, \tag{6}$$

and, thus, we recover the action of the operator A on $|\psi\rangle_s$ only if the ancillary register is $|0\rangle_a$, see Fig. 2.[1] The probability that this oc-

[1]Note that while this is a common choice, in general one could use a different state to project out the sub-block containing the rescaled matrix of interest [66].

curs, known as the *success probability*, is given by $(||A|\psi\rangle_s||/\alpha)^2$. This can also be written as

$$A/\alpha = (\langle 0|_a \otimes \mathbb{1}_s) U_A (|0\rangle_a \otimes \mathbb{1}_s)\,, \tag{7}$$

which implies that

$$U_A|0\rangle_a|\psi\rangle_s = |0\rangle_a(A/\alpha|\psi\rangle_s) + |\perp\rangle_{as}\,, \tag{8}$$

where $|\perp\rangle_{as}$ is a vector perpendicular to $|0\rangle_a$ in the sense that $(\langle 0|_a \otimes \mathbb{1}_s)|\perp\rangle_{as} = 0$. For a more formal treatment of block encodings, see Refs. [21, 22, 41, 47, 63].

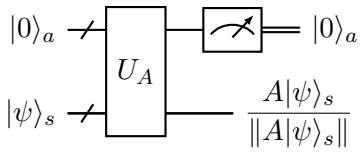

Figure 2: Circuit for implementing the block encoding U_A of an operator A, as defined in Eq. (7). Upon measuring the ancillary register in the state $|0\rangle_a$, the state of the system $|\psi\rangle_s$ is mapped to $A|\psi\rangle_s/\|A|\psi\rangle_s\|$.

The most common QSP-based approach to simulating dynamics governed by time-independent Hamiltonians [66] approximates the operator e^{-iHt}[2] using a special form of block of encoding of H (also known as the *Szegedy quantum walk operator* or the *iterate*) denoted by W_H. This is known as *qubitization*.[3] The operator W_H is designed in such a way that upon its repeated application, one recovers block encodings for *higher powers of* H (which does not occur for a general block encoding U_H). This can be achieved if for all Hamiltonian eigenstates $|\lambda\rangle$, the applications of W_H to states of the form $|0\rangle_a|\lambda\rangle_s$ produce a state belonging to the subspace $\mathrm{span}\{|0\rangle_a|\lambda\rangle_s, |\perp^\lambda\rangle_{as}\}$:

$$\begin{aligned} W_H|0\rangle_a|\lambda\rangle_s &= \lambda\,|0\rangle_a|\lambda\rangle_s + \sqrt{1-\lambda^2}|\perp^\lambda\rangle_{as}\,, \\ W_H|\perp^\lambda\rangle_{as} &= -\sqrt{1-\lambda^2}|0\rangle_a|\lambda\rangle_s + \lambda\,|\perp^\lambda\rangle_{as}\,. \end{aligned} \tag{9}$$

The walk operator W_H can be written as

$$W_H = (R_0 \otimes \mathbb{1}_s)\,(S \otimes \mathbb{1}_s)\,U_H\,, \tag{10}$$

[2]Without loss of generality, we rescale the Hamiltonian and the evolution time as $H \to \alpha H$ and $t \to t/\alpha$ to ensure that $||H||_2 \leq 1$.

[3]Confusingly, *qubitization* often refers to the entire process of simulating time evolution based on QSP and repeated calls to W_H [66]. Moreover, sometimes *qubitization* is used as an umbrella term for all quantum simulation algorithms based on the usage of BE. In the present paper, *qubitization* refers solely to the process of constructing W_H.

where $R_0 = \left(2\,|0\rangle_a\langle 0|_a - \mathbb{1}_a\right)$ implements reflection with respect to the state $|0\rangle_a$, if there exists an operator S satisfying [66]

$$(\langle 0|_a \otimes \mathbb{1}_s)\,(S \otimes \mathbb{1}_s)\,U_H\,(|0\rangle_a \otimes \mathbb{1}_s) = H/\alpha\,, \tag{11a}$$

$$(\langle 0|_a \otimes \mathbb{1}_s)\,[(S \otimes \mathbb{1}_s)\,U_H]^2\,(|0\rangle_a \otimes \mathbb{1}_s) = \mathbb{1}_s\,. \tag{11b}$$

Finding an operator S that satisfies the above relations for a given U_H is generally a difficult task. In situations where S is unknown, one can always construct W_H by adding an additional ancillary qubit and performing two controlled calls to U_H [65]. While this procedure is fully general, it introduces a large overall coefficient in the cost to construct W_H. Importantly, for all BE methods considered in this work, the operator S was determined and the costly general procedure was avoided. For a more careful treatment of this subject, including the motivation for and construction of the iterate, we refer the reader to Refs. [47, 66].

2.2 Linear Combination of Unitaries

The Linear Combination of Unitaries (LCU) method enables the construction of BEs of operators by breaking an operator T acting on some system Hilbert space $\mathcal{H}_s$ into a sum of into unitary operators U_i, each with known implementation:

$$T = \sum_{i=0}^{K-1} c_i U_i\,, \tag{12}$$

where the coefficients c_i are chosen to be real and positive. To implement LCU, one adds an ancillary qubit register with $\lceil \log K \rceil$ qubits and defines a *select* oracle (denoted as SEL) as follows:

$$\mathrm{SEL} \equiv \sum_{i=0}^{K-1} |i\rangle\langle i| \otimes U_i\,. \tag{13}$$

SEL implements each unitary U_i conditioned on the state $|i\rangle$ encoded as a binary string on the a ancillary qubits.

One also defines a *prepare* oracle (denoted as PREP) as follows:

$$\mathrm{PREP}\left|0^{\otimes a}\right\rangle = \frac{1}{\sqrt{c}} \sum_{i=0}^{K-1} \sqrt{c_i}\,|i\rangle\,, \tag{14}$$

where $c \equiv \sum_{i=1}^{K-1} |c_i|$. Given the definitions of these oracles, one can verify that the quantum circuit given in Fig. 3 implements the action of the

operator T on some input state $|\psi\rangle \in \mathcal{H}_s$ and thus prepares a BE of T. This result can be summarized with the following lemma (see for example [27],[63] for a proof of this result):

Lemma 1 (LCU block encoding)**.** *Let $U_0 \ldots, U_{K-1}$ be K unitary operators acting on a system Hilbert space $\mathcal{H}_s$ and let $T = \sum_{i=0}^{K-1} c_i U_i$, with $(c_0, \ldots, c_{K-1}) \in \mathbb{C}^K$, be an operator also acting on $\mathcal{H}_s$. Then, the unitary given by $\left(PREP^\dagger \otimes \mathbb{1}_s\right) SEL \left(PREP \otimes \mathbb{1}_s\right)$, where $\mathbb{1}_s \in \mathcal{H}_s$ is the identity operator, is a $(c, \log K)$ BE of the operator T, where $c \equiv \sum_{i=0}^{K-1} |c_i|$. The elementary gate complexity of constructing this BE asymptotically scales as $\mathcal{O}(K \log K)$.*

2.3 Simulating time evolution using Generalized QSP

In this section, we review an improved version of the original QSP method, known as Generalized QSP (GQSP). Provided access to some unitary U, QSP and GQSP allow one to construct circuits implementing the action of polynomials of U. In the context of simulating time evolution, one can choose U to be the Hamiltonian walk operator and use GQSP to implement polynomial approximations of the time evolution operator.

The main result of Ref. [71], relevant to the problem of simulating time evolution, is given by:

Theorem 2 (Corollary 5 from Ref. [71])**.** *$\forall P \in \mathbb{C}[x]$ with deg(P) = d, if*

$$\forall x \in \mathbb{T}, \quad |P(x)|^2 \leq 1\,, \tag{15}$$

where $\mathbb{T} = \{x \in \mathbb{C} : |x| = 1\}$ is the unit circle in the complex plane, then $\exists\, \vec{\theta}, \vec{\phi} \in \mathbb{R}^{d+1}$ and $\gamma \in \mathbb{R}$ such that:

$$\left(\prod_{j=1}^{d} R(\theta_j, \phi_j, 0)\, C_U\right) R(\theta_0, \phi_0, \gamma) = \begin{pmatrix} P(U) & * \\ * & * \end{pmatrix}, \tag{16}$$

where $R(\theta, \phi, \gamma)$ represents an SU(2) rotation and $C_U = (|0\rangle\langle 0| \otimes U) + (|1\rangle\langle 1| \otimes \mathbb{1})$ is a 0-controlled application of U (the signal operator*),*

Furthermore, Ref. [71] also provides an efficient classical algorithm to compute the required rotation angles $\vec{\theta}, \vec{\phi}, \gamma$ for a desired polynomial $P(x)$.

For the purpose of simulating time evolution, we note that the eigenvalues of the operator $W_H^{(\lambda)}$, defined by restricting the action of W_H to the subspace spanned by $\text{span}\{|0\rangle_a|\lambda\rangle_s, |\perp^\lambda\rangle_{as}\}$, are given by $e^{\pm i \arccos(\lambda)}$. Thus we set the unitary to which controlled calls are made in the GQSP circuit to be $U = W_H$ (see Fig. 4 adapted from Ref. [71]) and seek an implementation of the function given by $e^{-it\cos(x)}$ to achieve the desired polynomial transformation given by $P(W_H^{(\lambda)}) = e^{-it\lambda}$. This is achieved through the Jacobi-Anger expansion (see for example Refs. [66, 71]), given by:

$$e^{-it\cos(x)} = \sum_{k=-\infty}^{\infty} (-i)^k J_k(t) e^{ikx}, \tag{17}$$

where J_k is the k^{th} Bessel function of the first kind. It has been previously established the coefficients in this Fourier series decay exponentially and in particular, the number of SU(2) rotations required to achieve an ϵ-close approximation to this polynomial transformation is asymptotically bounded by $\mathcal{O}(t + \log(1/\epsilon)/\log\log(1/\epsilon))$ (see Refs. [66, 71]). We conclude by noting that, while using QSP for Hamiltonian simulation achieves the same asymptotically optimal scaling, using GQSP requires half as many controlled calls to W_H [16, 71].

2.4 QSVT

While QSP and GQSP are powerful algorithmic tools, in certain cases, the required polynomial transformations can be achieved using the simpler Quantum Singular Value Transformation (QSVT) [41]. The QSVT method is very similar to the QSP and GQSP methods in that it allows one to implement general matrix functions of some Hermitian operator A by performing repeated calls to its BE U_A. The major difference of QSVT has to do with the classes of matrix functions that can be implemented. In particular, QSVT implements real functions with definite parity, whereas QSP implements complex functions with the real and imaginary parts having opposite parity. Only requiring the real part of the matrix polynomial leads to several advantages. First, the fundamental building block is now the BE U_A, rather than a controlled call to the walk operator W_A, as in QSP or GQSP. Second, the conditions on implementable real polynomial are relaxed relative to the complex polynomials implemented using QSP. Third, the phases that parameterize the circuit can be solved using a straightforward optimization procedure for large polynomials using e.g. QSP-

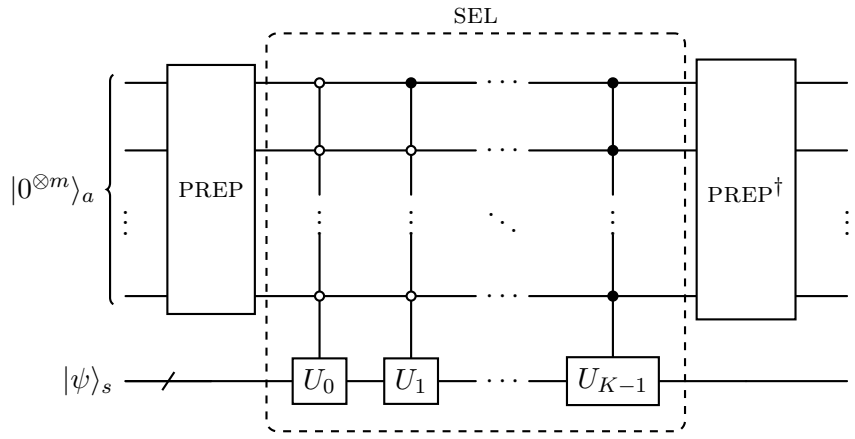

Figure 3: The Linear Combination of Unitaries (LCU) circuit [27] to implement the BE of an operator $T = \sum_{i=0}^{K-1} c_i U_i$, a linear combination of unitary operators with known implementations. Operators, PREP and SEL are defined in Eq. (14) and Eq. (13), correspondingly.

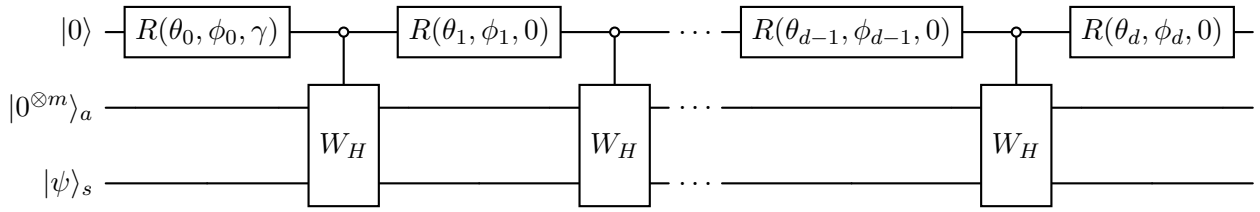

Figure 4: Circuit diagram for simulating time evolution using GQSP, adapted from Ref. [71]. Each $R(\theta_0, \phi_0, \gamma)$ is an SU(2) rotation, with the individual angle parameters shown in the diagram chosen to construct a polynomial of W_H in the context of this work.

PACK [1]. The QSVT theorem for Hermitian matrices is as follows [63]:

Theorem 3 (QSVT for Hermitian matrices with real polynomials)**.** *Let $A \in \mathbb{C}^{2^n \times 2^n}$ be a normalized Hermitian operator with a $(1, m)$ block encoding U_A. Given a degree d polynomial $P_{\mathrm{Re}}(x) \in \mathbb{R}[x]$ such that*

1. *P_{Re} has parity $d \pmod 2$,*
2. *$|P_{\mathrm{Re}}(x)| \leq 1 \ \ \forall x \in [-1, 1]$,*

then there exist symmetric phase factors $(\phi_0, \ldots, \phi_d) \in \mathbb{R}^{d+1}$ such that the circuit presented in Fig. 5 implements a $(1, m+1)$ block-encoding of $P_{\mathrm{Re}}(A)$.

One important property of QSVT circuits is that their controlled versions can be implemented with a small overhead [41]. For even polynomials, only the single-qubit R_z gates in the $\mathrm{CR}_{\tilde{\phi}_j}$ circuits (see Fig. 6) acting on the signal qubit need to be controlled. For odd polynomials, in addition to controlling the single-qubit R_z gates, one must also control a single call to the BE U_A. This property can be used to combine different QSVT circuits using LCU without introducing a large overall constant prefactor, which is important when using LCU to combine BEs of different local terms in the Hamiltonian constructed using QSVT, as explained in Sec. 3.4.

2.5 QETU

QETU is similar to the QSVT method described in Sec. 2.4 in that it allows one to implement a broad class of matrix functions of some Hermitian operator A by performing repeated calls to some fundamental building block. The major difference of QETU is that, instead of performing repeated calls to a BE U_A as with QSVT, the building block for QETU is the time-evolution operator $\mathrm{e}^{-i\tau A}$ for some value of τ. In a broad sense, QETU replaces the BE U_A in the QSVT circuit in Fig. 5 with the controlled time-evolution operator $\mathrm{e}^{-i\tau A}$. The QETU circuit for implementing even functions is shown in Fig. 7. Taken from Ref. [33], the QETU theorem is:

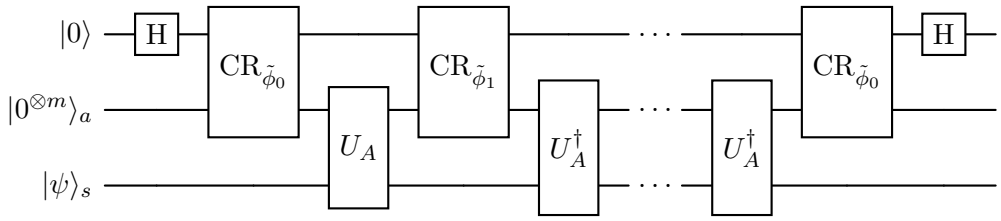

Figure 5: Circuit diagram for QSVT, reproduced from Ref. [63], using symmetric phases $(\tilde{\phi}_0, \tilde{\phi}_1, \ldots, \tilde{\phi}_1, \tilde{\phi}_0)$. The Hadamard gates serve to isolate the real part of the matrix function being implemented. The circuit for CR_ϕ is shown in Fig. 6.

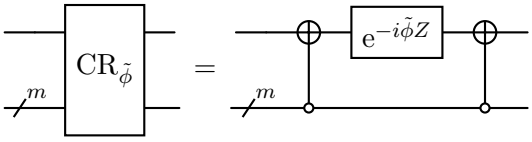

Figure 6: Diagram for the CR_ϕ circuit appearing in the QSVT circuit in Fig. 5.

Theorem 4 (QETU)**.** *Let $U = \mathrm{e}^{-i\tau A}$ be a unitary operator, where A is an n-qubit Hermitian operator and τ is real parameter. For any real even polynomial $F(x)$ of degree d satisfying $|F(x)| \leq 1, \forall x \in [-1, 1]$, one can find a sequence of symmetric phase factors $(\varphi_0, \ldots, \varphi_d) \in \mathbb{R}^{d+1}$ such that the circuit in Fig. 7 denoted by $\mathcal{U}$ satisfies $(\langle 0| \otimes \mathbb{1})\mathcal{U}(|0\rangle \otimes \mathbb{1}) = F(\cos(\tau A/2))$.*

In practice, the QETU circuit is used for approximately implementing $f(A)$ by realizing a transformation $F(\cos(\tau A/2))$, where $F(x)$ is a polynomial approximation to $F(x) \equiv f(\frac{2}{\tau}\arccos(x))$.

Similar to QSVT, the phases $\{\tilde{\varphi}_j\}$ appearing in the circuit in Fig. 7 can be calculated using QSPPACK [1]. As described in Appendix B of Ref. [33], the phases in the QETU circuit $\{\tilde{\varphi}_j\}$ are related to the phases calculated using QSPPACK $\{\tilde{\phi}_j\}$ by $\tilde{\varphi}_j = \tilde{\phi}_j + (2 - \delta_{j\,0})\pi/4$.

Completely analogous to the case with QSVT circuits, controlled calls to QETU circuits can also be implemented for a small additional overhead. This property can be used to combine BEs of different local terms in the Hamiltonian that were constructed using QETU; the procedure for constructing BEs using QETU is given in Sec. 3.5.

3 Block encodings for a bosonic lattice field theory

In this section we describe how to construct block encodings for certain types of bosonic lattice field theories, and how to use those to simulate time evolution using GQSP. Our choice of GQSP is guided by the fact that, while this algorithm has the same asymptotic complexity as QSP or QSVT, but it comes with a smaller constant prefactor [16, 71].

The general strategy for using GQSP to simulate time evolution of the Hamiltonian in Eq. (1) is achieved with the following steps:

1. Construct the BEs $U_{f_k(\hat{\xi}_k)}$ for the individual terms of the form $f_k(\hat{\xi}_k)$.[4]
2. Construct $U_{f_1(\hat{\pi}_i)} = \widehat{\mathcal{F}}_i^\dagger U_{f_1\left(\hat{\pi}_i^{(D)}\right)} \widehat{\mathcal{F}}_i$ using the Quantum Fourier transform circuit.
3. Use LCU to combine these site-local BEs to obtain the BE for the full Hamiltonian U_H.
4. Construct the qubitized walk operator W_H.
5. Simulate time evolution using the GQSP circuit in Fig. 4.

After introducing the Hamiltonian of the scalar bosonic lattice field theory and discuss the digitization of the local bosonic Hilbert space, we present three different techniques for preparing BEs for a broad class of operators appearing in site-local Hamiltonians encountered in various formulations of lattice field theories. In all three techniques the BE of $f_k(\hat{\xi}_k)$ will start with an encoding

[4] Since the functional form of f_{k_ℓ} is the same $\forall \ell \in \{0, 1, \ldots, N-1\}$ and fixed $k \in \{0, 1, 2\}$, in what follows we will drop the ℓ subscript on f_k when the statement is true / applies $\forall \ell$.

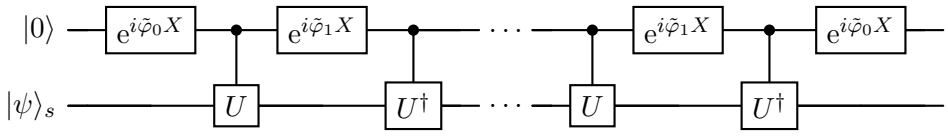

Figure 7: Circuit diagram for QETU. The top qubit is the control qubit and the bottom register is the state that the matrix function is applied to. Here $U = \mathrm{e}^{-i\tau A}$ is the time evolution circuit. If the control qubit is measured to in the zero state, one prepares the normalized quantum state $F(\cos(\tau A/2))|\psi\rangle/||F(\cos(\tau A/2))|\psi\rangle||$ for some even polynomial $F(x)$. For symmetric phase factors $(\tilde{\varphi}_0, \tilde{\varphi}_1, \ldots, \tilde{\varphi}_1, \tilde{\varphi}_0) \in \mathbb{R}^{d+1}$, the function $F(x)$ is a real even polynomial of degree d. The probability of measuring the control qubit in the zero state is $p = ||F(\cos(\tau A/2))|\psi\rangle||^2$.

of a basic building block, which is then combined to form $f_k(\hat{\xi}_k)$.

The first techniques uses QSVT, and starts from the simplest building block possible, namely the operator $\hat{\xi}_k$. Since this operator is not unitary, it needs to be block-encoded. As we will see, this encoding can be done with resources that are linear in $\mathsf{n_q}$, but comes with a large prefactor. Constructing the final function $f_k(\hat{\xi}_k)$ adds a constant multiplicative factor on the resource requirement.

The second method uses QETU, and directly uses the unitary operator $\mathrm{e}^{-i\tau\hat{\xi}_k}$ as the basic building block. The building block of QETU can still be implemented with resources that are linear in $\mathsf{n_q}$, but removes the large pre-factors arising in the previous method. The downside is that constructing $f_k(\hat{\xi}_k)$ introduces an exponential dependence on $\mathsf{n_q}$. As we will see, despite the exponential dependence on $\mathsf{n_q}$, the absence of the large pre-factor make this method more efficient for $\mathsf{n_q} \lesssim 6$.

The final method takes a different approach, and chooses a more complicated building block, which requires resources with exponential scaling in $\mathsf{n_q}$ for its construction. This building block is chosen such to make the construction of $f_k(\hat{\xi}_k)$ as easy as possible, and is in fact trivial. This gives better scaling than the second method, and in fact outpeforms all methods $\mathsf{n_q} \lesssim 11$. We call this final method Linear Operators Via Exponentiation and Linear Combination of Unitaries (LOVE-LCU).

The comparison of different BEs is summarized in Table 1.

3.1 General Form of Hamiltonians of Interest

A common approach to formulating quantum field theories in a way suitable for solving those on quantum computers amounts to 1) discretizing the real space so that the fields were defined on a finite subset $\mathcal{V}$ of sites of some lattice, e.g., $\mathcal{V} \subset a\mathbb{Z}^d \subset \mathbb{R}^d$, subject to an appropriate boundary condition; 2) digitizing quantum degrees of freedom residing on lattice sites, i.e., truncating the infinite-dimensional local Hilbert space to a finite-dimensional one. While alternative approaches are also subject to active investigation [19, 20, 60], the aforementioned paradigm has two major advantages: the locality of the discretized Hamiltonian (for local quantum field theories) and linear growth of the Hamiltonian norm with problem size.[5] Both of these are of great utility from the quantum computing perspective. While fermionic lattice systems have been subject to active investigation in the literature (for recent studies, see Ref. [39, 82] and references therein), in this work we focus on lattice bosons, which are an essential ingredient in nuclear [81] and high-energy particle physics [12].

Many physically relevant Hamiltonians, including scalar field theories [49, 58], dual basis formulations of U(1) [8, 13] and SU(2) [31] gauge theories, are (at least partially) formulated in terms of conjugate operators defined on each lattice site. For this work we use the notation of a scalar field theory where we denote by $\hat{\varphi}_i$ and $\hat{\pi}_i$ the "field" and "momentum" operators at site i, respectively, satisfying the canonical commutation relations $[\hat{\varphi}_i, \hat{\pi}_j] = i\delta_{ij}$. The general idea is to exploit the fact that, for a particular highly efficient digitization scheme of such operators (described in detail below), the local operators $\hat{\varphi}_i$ and $\hat{\pi}_i$ take on simple forms and can therefore be used as low cost building blocks in the QSVT or QETU circuits. These can then be used to prepare low cost BEs of the individual terms appearing in the Hamiltonian; and the BE for the full Hamiltonian can be prepared by using LCU to combine the BEs of the site-local terms.

[5] Here we do not touch upon the issue of renormalization which may affect the Hamiltonian norm via adjustment of coupling constants.

Algorithm	Gate Complexity	Ancillary qubits	Indefinite parity	Multivariable functions
Standard LCU	$\mathcal{O}(\mathsf{n_q^d})$	$\mathcal{O}(d \log \mathsf{n_q})$	Yes	Yes
QSVT	$\mathcal{O}(d \cdot \mathsf{n_q} \log \mathsf{n_q})$	$\mathcal{O}(\log \mathsf{n_q})$	No	No
QETU with $\mathrm{e}^{i\tau\hat{\xi}}$	$\mathcal{O}(2^{\mathsf{n_q}} \cdot \mathsf{n_q})$	1	No	No
QETU with $\mathrm{e}^{i2\arccos\hat{\xi}}$	$\mathcal{O}(d \cdot 2^{\mathsf{n_q}})$	1	No	No
LOVE-LCU	$\mathcal{O}(2^{\mathsf{n_q}})$	1	Yes	Yes

Table 1: Properties of BE methods considered in this work. Gate count and ancillary qubit count are for constructing a BE of a degree d function $f(\hat{\xi})$ of an n-qubit diagonal Hermitian operator $\hat{\xi}$ that is a sum of $\mathcal{O}(n)$ single Z-gates (for operators considered in this work, $\hat{\xi} \in \{\hat{\varphi}_i, \hat{\pi}^{(D)}, \hat{\varphi}_i - \hat{\varphi}_j\}$). The QSVT- and QETU-based methods require f to have definite parity and be a function of a single variable. Standard LCU and LOVE-LCU can be applied to functions with indefinite parity.

To be concrete, we consider Hamiltonians of the form

$$\hat{H} = \sum_{i=1}^{\mathsf{N_\Lambda}} \left(f_0\left(\hat{\varphi}_i\right) + f_1\left(\hat{\pi}_i\right) \right) + \sum_{\langle ij \rangle} f_2\left(\hat{\varphi}_i - \hat{\varphi}_j\right), \tag{18}$$

where $\mathsf{N_\Lambda}$ is the number of lattice sites and $\sum_{\langle ij \rangle}$ is a sum over all adjacent sites i and j. A discussion of applying our methods to more general forms of the Hamiltonian is given in Sec. 5.

To *digitize* the theory, the local bosonic Hilbert space at each lattice site j is represented using $\mathsf{n_q}$ qubits. Following Refs. [10, 49, 58], we work in the eigenbasis of the digitized field operators $\hat{\varphi}_j$, i.e., we identify the computational basis states $|k\rangle_j$ at site j with the eigenstates $|\varphi_j^{(k)}\rangle$. The eigenvalues of the field operator $\hat{\varphi}_j$ are chosen as

$$\begin{gathered} \hat{\varphi}_j |\varphi_j^{(k)}\rangle = (-\varphi_{\max} + k\delta\varphi) |\varphi_j^{(k)}\rangle, \\ k = 0, \ldots, 2^{\mathsf{n_q}} - 1, \end{gathered} \tag{19}$$

where $\delta\varphi = 2\varphi_{\max}/(2^{\mathsf{n_q}} - 1)$. The free parameter $\varphi_{\max}$ can be chosen to minimize the digitization errors and is theory dependent. Generally, by choosing $\varphi_{\max}$ to scale exponentially with $\mathsf{n_q}$ [8, 10, 31, 49, 53, 58], the digitization errors decrease exponentially with $\mathsf{n_q}$. This fact implies that the number of qubits per site $\mathsf{n_q}$ can generally be kept small, which is important to keep in mind for the discussion moving forward.

Exploiting the fact that $\hat{\pi}_j$ is conjugate to $\hat{\varphi}_j$, the momentum operators are digitized as

$$\begin{gathered} \hat{\pi}_j^{(D)} |\pi_j^{(k)}\rangle = (-\pi_{\max} + k\delta\pi) |\pi_j^{(k)}\rangle, \\ k = 0, \ldots, 2^{\mathsf{n_q}} - 1, \end{gathered} \tag{20}$$

where

$$\hat{\pi}_j^{(D)} = \hat{\mathcal{F}}_j \hat{\pi}_j \hat{\mathcal{F}}_j^\dagger \tag{21}$$

is the conjugate momentum operator in the momentum basis, $\hat{\mathcal{F}}_j$ denotes the usual discrete Fourier transform (FT) at site j, $\pi_{\max} = \pi/\delta\varphi$ and $\delta\pi = 2\pi_{\max}/(2^{\mathsf{n_q}} - 1)$. The Hamiltonian in this digitizaiton is then written through functions of the diagonal operators $\hat{\varphi}_j$ and $\hat{\pi}_j^{(D)}$ as

$$\begin{aligned} \hat{H} = \sum_{i=1}^{\mathsf{N_\Lambda}} \Big(f_0\left(\hat{\varphi}_i\right) + \hat{\mathcal{F}}_i^\dagger f_1(\hat{\pi}_i^{(D)}) \hat{\mathcal{F}}_i \Big) \\ + \sum_{\langle ij \rangle} f_2\left(\hat{\varphi}_i - \hat{\varphi}_j\right). \end{aligned} \tag{22}$$

3.2 Implementation of field operators and their functions

The operators $\hat{\varphi}_j$ and $\hat{\pi}_j^{(D)}$ are $2^{\mathsf{n_q}}$-dimensional diagonal operators with evenly spaced entries on the diagonal and hence can be expressed as a linear combination of $\mathsf{n_q}$ single-qubit Pauli-Z gates [58, 88]. For the field operator $\hat{\varphi}$ for example (dropping the site indices for brevity), one can write

$$\hat{\varphi} = -\frac{\varphi_{\max}}{2^{\mathsf{n_q}} - 1} \sum_{m=0}^{\mathsf{n_q}-1} 2^{\mathsf{n_q}-1-m} Z_m, \tag{23}$$

where Z_m is a Pauli-Z gate acting on the m^{th} qubit of the local Hilbert space considered. Aside from the overall prefactor, the operator $\hat{\pi}^{(D)}$ has an identical gate representation. Given that the operators $\hat{\xi}_k$ are diagonal, one can readily determine complicated functions $f_k(\hat{\xi}_k)$ using classical resources by simply applying the function f_k to the diagonal elements of $\hat{\xi}_k$.

3.3 LCU to Block Encode Individual Terms

In this section, we review the cost of using standard LCU techniques to BE the individual $f_k(\hat{\xi}_k)$

terms. For a more detailed analysis, see Ref. [47] and references therein.

Consider first using LCU to BE $f_0(\hat{\varphi})$ and $f_1(\hat{\pi}^{(D)})$ From Eq. (23), we see that a general degree d function of $\hat{\varphi}$ (or $\hat{\pi}^{(D)}$) is a sum of $\mathcal{O}(\mathsf{n}_\mathsf{q}^d)$ Pauli strings. It is important to note that the number of Pauli strings is upper bounded by $2^{\mathsf{n}_\mathsf{q}}$. This, combined with Lemma 1, implies that the asymptotic gate cost to BE $f_0(\hat{\varphi})$ and $f_1(\hat{\pi}^{(D)})$ is either $\mathcal{O}(\mathsf{n}_\mathsf{q}^d \log \mathsf{n}_\mathsf{q})$ or $\mathcal{O}(\mathsf{n}_\mathsf{q} 2^{\mathsf{n}_\mathsf{q}})$, depending on the degree d and value of n_q used.

Turning to the final term, for a degree d function, $f_2(\hat{\varphi}_i - \hat{\varphi}_j)$ is a sum of $\mathcal{O}(\mathsf{n}_\mathsf{q}^d)$ terms. In this case, the number of Pauli strings is upper bounded by $2^{2\mathsf{n}_\mathsf{q}}$. Using similar arguments, the asymptotic gate cost to BE $f_2(\hat{\varphi}_i - \hat{\varphi}_j)$ is either $\mathcal{O}(\mathsf{n}_\mathsf{q}^d \log \mathsf{n}_\mathsf{q})$ or $\mathcal{O}(\mathsf{n}_\mathsf{q} 2^{2\mathsf{n}_\mathsf{q}})$.

Lastly, we discuss the scale factor when using LCU. The scale factor is the sum of the absolute values of the coefficients of the unitary terms being added together. Given a particular $f_k(\hat{\varphi}_k)$, the scale factor can be readily determined using using Eq. (23). The scale factors for operators considered in Sec. 4 are derived in Appendix C.

3.4 QSVT to Block Encode Individual Terms

In the context of quantum simulation, the QSVT algorithm is typically utilized as a way of constructing polynomial functions of Hamiltonian operator [41, 63]. In this section we use it to construct the functions $f_k(\hat{\xi}_k)$ and demonstrate that QSVT can also be used as a highly-efficient tool for constructing block encodings. The high level flow of this technique is depicted in Fig. 1. For this discussion we assume the functions f_k to have even parity.

The building block in the QSVT circuit is the BE of the simple operator $\hat{\xi}_k$. From Eq. (23) we see that each $\hat{\xi}_k$ is written as a sum of $\mathcal{O}(\mathsf{n}_\mathsf{q})$ Pauli-Z gates, each of which are unitary. The BE $U_{\hat{\xi}_k}$ can therefore be constructed with LCU using $\mathcal{O}(\mathsf{n}_\mathsf{q} \log \mathsf{n}_\mathsf{q})$ gates and $\mathcal{O}(\log \mathsf{n}_\mathsf{q})$ ancillary qubits.

If f_k is a polynomial function, QSVT can prepare the BE U_{f_k} using a constant number of calls to $U_{\hat{\xi}_k}$. For non-polynomial functions, one can express the function $f_k(\hat{\xi}_k)$ as a (potentially infinite) Chebyshev series of the operator $\hat{\xi}_k$, and then use QSVT to construct the BEs U_{f_k}.

To determine the Chebyshev expansion, we write

$$\sum_{j=0} c_{2j} T_{2j}\left(\frac{\hat{\xi}_k}{\alpha}\right) = \frac{1}{\beta} f_k(\hat{\xi}_k)\,, \tag{24}$$

where α is the scale factor used in the BE of $\hat{\xi}_k$ and β normalizes the function $f_k(\hat{\xi}_k)$ which can be obtained classically. The upper limit of the sum has been left blank to stress the series can be a sum of a finite or infinite number of terms. The coefficients c_{2j} can be then be found using the orthogonality relations of Chebyshev polynomials. The gate complexity depends on the convergence of the Chebyshev polynomial approximation, and for a degree-d polynomial the gate complexity is therefore given by using $\mathcal{O}(d\,\mathsf{n}_\mathsf{q} \log \mathsf{n}_\mathsf{q})$. The scale factor of the BE is $\beta = \|f_k(\hat{\xi}_k)\|$.

We conclude by pointing out that using $\hat{\varphi}_i$ and $\hat{\pi}_i$ as building blocks to construct a BE of the Hamiltonian is not a unique choice. One could equivalently use, e.g., $\hat{\varphi}_i^2$ and $\hat{\pi}_i^2$, and modify the Chebyshev polynomial used accordingly. However, because the number of terms, as well as the length of the largest Pauli string, in $\hat{\varphi}_i^n$ (and $\hat{\pi}_i^n$) scales as $\mathcal{O}(\mathsf{n}_\mathsf{q}^n)$, using the simpler building blocks $\hat{\varphi}_i$ and $\hat{\pi}_i$ will result in better asymptotic scaling with n_q. We leave investigations of alternative constructions for future work.

3.5 QETU to Block Encode Individual Terms

While QETU was initially proposed to eliminate the need for block encoding in early fault-tolerant quantum algorithms [33], Ref. [54] has investigated its utility for preparing functions of Hermitian operators, using Gaussian states as an example. Below, we show that QETU can be utilized for constructing block encodings for Hamiltonians of the form (18).

Preparing BEs using QETU is conceptually similar to using QSVT. Following the procedure outlined in Fig. 1, the building block used is $\mathrm{e}^{-i\tau\hat{\xi}_k}$, which is the simplest unitary operator containing information about $\hat{\xi}_k$. By performing repeated calls to $\mathrm{e}^{-i\tau\hat{\xi}_k}$, QETU can prepare a BE of $f_k(\hat{\xi}_k)$.

The remainder of this section is split into two parts. We first describe the technical details of using QETU to prepare general matrix functions. From there, we discuss the computational cost of preparing BEs of $f_k(\hat{\xi}_k)$.

3.5.1 Technical details

We begin by discussing QETU for a general function $f(\hat{A})$ of a Hermitian matrix $\hat{A}$ (not necessarily diagonal) with known eigenvalues $\hat{A}|a_j\rangle = a_j|a_j\rangle$, and will specialize to $f_k(\hat{\xi}_k)$ later. We will call the minimum and maximum eigenvalues $a_{\min}$ and $a_{\max}$, respectively. QETU allows to start from an implementation of the operator $\mathrm{e}^{-i\tau\hat{A}}$ and implement a BE of the function $F(\cos(\tau\hat{A}/2))$. Since one needs to shift the spectrum of the operator $\tau\hat{A}$ to lie in the range $[0, 2\pi]$, we therefore define a shifted operator

$$\hat{A}_{\mathrm{sh}} = c_1\hat{A} + c_2\mathbb{1}\,, \tag{25}$$

where

$$c_1 = \frac{\pi}{a_{\max} - a_{\min}}\,, \quad c_2 = -c_1 a_{\min}\,, \tag{26}$$

and require $\tau \leq 2$. In order to implement the desired function $f_k(\hat{A})/\beta$ (where β is required to normalize the function to allow a BE), one needs to choose

$$F(x) = \frac{1}{\beta} f\left(\frac{\frac{2}{\tau}\arccos(x) - c_2}{c_1}\right). \tag{27}$$

Note that the direct implementation of QETU applies only in cases when $F(x)$ has definite parity. Since the function in Eq. (27) does not generally have definite parity, one would need LCU to combine even and odd functions together to obtain the general function $F(\hat{A})$. However, as shown in Ref. [54], the need for LCU to add the even and odd pieces can, in certain cases, be avoided altogether by choosing by choosing τ in such a way as to make $F(x)$ have definite parity. To see this, note that if one chooses $\tau = \pi/c_2$, the function $F(x)$ becomes

$$\begin{aligned} F(x)\Big|_{\tau=\pi/c_2} &= \frac{1}{\beta} f\left(-\frac{2}{\pi}\frac{c_2}{c_1}\arcsin(x)\right) \\ &= \frac{1}{\beta} f\left(\frac{2a_{\min}}{\pi}\arcsin(x)\right) \end{aligned} \tag{28}$$

where we have used $\arcsin(x) = \pi/2 - \arccos(x)$. Because $\arcsin(x)$ has odd parity, the function $F(x)$ has the same parity as the function $f(x)$. This means that LCU is not required if the original function $f_k(\hat{A})$ one aims to implement has definite parity. Note if $\tau > 2$, this technique cannot be applied and one must construct the even and odd pieces separately. Importantly, all operators $\hat{\xi}_k$ considered in this work are digitized such that their minimum and maximum eigenvalues are equal, which implies $\tau = 2$.

Next, we consider the scale factor β, which is given by $\beta = \max_{x\in[-1,1]} |F(x)|$. Using the fact that $\arcsin(x) \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, we can rewrite the scale factor as

$$\beta = \max_{y\in[-a_{\min}, a_{\min}]} |f(y)|\,. \tag{29}$$

We will use this to determine the scale factor for the BEs of interest in the next section.

The final observation we make regarding $F(x)$ is that, for general functions $f(x)$, its first derivative diverges at $x = \pm 1$ due to the presence of the $\arcsin(x)$ (or $\arccos(x)$) term. Chebyshev approximations of functions with m continuous derivatives on $x \in [-1, 1]$ are known to converge polynomially to the true function with the typical rate $(1/n_{\mathrm{Ch}})^m$, where n_{Ch} is the number of Chebyshev polynomials used in the approximation [84]. This is to be contrasted with the exponential convergence of Chebyshev series to infinitely differentiable functions. If one needed to reproduce $F(x)$ for all $x \in [-1, 1]$, this poor convergence could lead to a large gate cost.

As was shown in Ref. [54] this issue can be overcome by exploiting the fact that the spectrum of the operator $\hat{A}_{\mathrm{sh}}$ is known exactly and therefore one only has to reproduce $F(x)$ at those finitely many points. If $\hat{A}_{\mathrm{sh}}$ is represented using $\mathsf{n_q}$ qubits, there are $\mathcal{O}(2^{\mathsf{n_q}})$ such values, and one can therefore reproduce the function at those points exactly by using $\mathcal{O}(2^{\mathsf{n_q}})$ Chebyshev polynomials. In this way, one avoids the poor polynomial convergence of the Chebyshev expansion, at the cost of introducing an exponential scaling with $\mathsf{n_q}$. This exponential scaling, however, does not pose a problem as long we work with small values of $\mathsf{n_q}$ and as previously discussed, this is precisely the case in lattice field theories owing to the fact that the digitization errors generally decrease exponentially with $\mathsf{n_q}$.

3.5.2 Cost to prepare block-encodings of bosonic operators

We now describe the cost of using QETU to prepare BEs of $f_k(\hat{\xi}_k)$, where, again, each f_k is assumed to have even parity. After constructing the shifted operator $\hat{\xi}_{k,\mathrm{sh}}$ according to Eq. (25), the building block in the QETU circuit is a simple

operator $e^{-i\tau\hat{\xi}_{k,\mathrm{sh}}}$. Because each $\hat{\xi}_{k,\mathrm{sh}}$ is still a diagonal operator with evenly spaced eigenvalues (or a Kronecker sum of two such operators in the case of $\hat{\xi}_2 \in \{\hat{\varphi}_i - \hat{\varphi}_j\}$), they can be written as a sum of $\mathcal{O}(\mathsf{n_q})$ single-qubit Pauli-Z gates, and $e^{-i\tau\hat{\xi}_{k,\mathrm{sh}}}$ can be implemented using only $\mathcal{O}(\mathsf{n_q})$ single-qubit R_z gates. A single controlled call to $e^{-i\tau\hat{\xi}_{k,\mathrm{sh}}}$ therefore requires $\mathcal{O}(\mathsf{n_q})$ of R_z and CNOT gates. As discussed, QETU can prepare the BE U_{f_k} exactly using a polynomial of degree equal to the number of times we sample the function f_k.

The operators $\hat{\varphi}_i$ and $\hat{\pi}_i^{(D)}$ are sampled $2^{\mathsf{n_q}}$ times, implying one must use a degree-$\mathcal{O}(2^{\mathsf{n_q}})$ polynomial to reproduce $F_0(\hat{\varphi}_i) \sim f_0(\arcsin(\hat{\varphi}_i))$ and $F_1(\hat{\pi}_i^{(D)}) \sim f_1(\arcsin(\hat{\pi}_i^{(D)}))$ exactly. Note that one can reduce the degree of the polynomial by a factor of two by exploiting the fact that the operators are sampled symmetrically about zero and that f_0 and f_1 are even functions. Therefore, the asymptotic gate cost of preparing the BEs U_{f_0} and U_{f_1} using QETU is $\mathcal{O}(\mathsf{n_q}2^{\mathsf{n_q}})$, independent of the degree d of the polynomial of f_0 and f_1. To determine the scale factor, note that the minimum eigenvalue of $\hat{\varphi}$ ($\hat{\pi}$) is $-\varphi_{\max}$ ($-\pi_{\max}$). Using Eq. (29), the scale factor for U_{f_0} (U_{f_1}) is therefore $\max_{x\in[-\varphi_{\max},\varphi_{\max}]}|f_0(x)|$ ($\max_{x\in[-\pi_{\max},\pi_{\max}]}|f_1(x)|$. For many systems of physical interest, including scalar field theories (considered in Sec. 4), the maximum value of f_0, f_1 occur at the largest argument they are evaluated. In this situation, the scale factors for U_{f_0} and U_{f_1} are given by the smallest possible values, namely $\|f_0(\hat{\varphi})\|$ and $\|f_1(\hat{\varphi})\|$, respectively.

The final operator we need to block-encode is $f_2(\hat{\varphi}_i - \hat{\varphi}_j)$. Because $(\hat{\varphi}_j - \hat{\varphi}_i)$ is a $2^{2\mathsf{n_q}}$-dimensional matrix, one would naively expect that $\mathcal{O}(2^{2\mathsf{n_q}})$ Chebyshev polynomials are needed to exactly reproduce $F_2(\hat{\varphi}_j - \hat{\varphi}_i)$ at all the $2^{2\mathsf{n_q}}$ values it is sampled. However, due to the symmetric digitization of $\hat{\varphi}$, the matrix $(\hat{\varphi}_j - \hat{\varphi}_i)$ only has $\mathcal{O}(2^{\mathsf{n_q}})$ unique values, and can be implemented exactly using $\mathcal{O}(2^{\mathsf{n_q}})$ Chebyshev polynomials. The asymptotic gate complexity of preparing the BE U_{f_2} using QETU is therefore $\mathcal{O}(\mathsf{n_q}2^{\mathsf{n_q}})$, independent of the degree d of the polynomial of f_2. Because the minimum eigenvalue of $(\hat{\varphi}_j - \hat{\varphi}_i)$ is $-2\varphi_{\max}$, the scale factor is $\max_{x\in[-2\varphi_{\max},2\varphi_{\max}]}$. As before, for many physical systems, the scale factor is given by the minimum value of $\|f_2(\hat{\varphi}_i - \hat{\varphi}_j)\|$.

Additional gate cost savings can be achieved if one uses the control-free version of QETU [33]. In practice, doing so reduces the number of R_z gates in the controlled call to $e^{-i\tau\hat{\xi}_{k,\mathrm{sh}}}$ by a factor of 2. A detailed procedure for implementing the control-free version of QETU for diagonal operators can be found in Ref. [54].

3.6 LOVE-LCU

In the previous section, we showed that, by employing $\mathrm{e}^{i\tau\hat{\xi}_k}$ as a building block, QETU can be used to construct BEs of $f_k(\hat{\xi}_k)$ with an asymptotic gate cost $\mathcal{O}(\mathsf{n_q}2^{\mathsf{n_q}})$. In this section, we study how the cost changes if different building blocks are utilized. First, we will argue that, in general, it is not possible to eliminate the exponential scaling with $\mathsf{n_q}$. With this fundamental limitation in mind, we show that using more complicated building blocks can reduce the asymptotic cost to prepare a BE. Specifically, the cost of preparing $f(\hat{\xi}_k)$ is reduced from $\mathcal{O}(\mathsf{n_q}2^{\mathsf{n_q}})$ to $\mathcal{O}(d2^{\mathsf{n_q}})$, where d is the degree of f. Finally, extending this logic further, we develop a conceptually simple framework that does not require QETU, and can prepare BEs of $f(\hat{\xi}_k)$ using $\mathcal{O}(2^{\mathsf{n_q}})$ gates, independent of the degree d of f.

3.6.1 Main idea

To begin the discussion, recall that the exponential scaling in $\mathsf{n_q}$ arises from the need to work with functions whose derivatives diverge, such as $\arccos(x)$ or $\arcsin(x)$. This result generalizes, and any building block that requires working with these functions will exhibit the same exponential scaling.

To understand what functions fall into this class, consider the general building block $\mathrm{e}^{-i\tau g(\hat{\xi}_k)}$. Using this building block, QETU returns $F(\cos(\frac{\tau g(\hat{\xi}_k)}{2}))$, which implies that the function we need to implement using Chebyshev polynomials is $F(x) \sim f(g^{-1}(\frac{2}{\tau}\arccos(x)))$. Any function $g(x)$ that results in $f(g^{-1}(\frac{2}{\tau}\arccos(x)))$ having divergent derivatives for $x \in [-1, 1]$ will consequently require $\mathcal{O}(2^{\mathsf{n_q}})$-degree Chebyshev polynomials to produce $F(x)$ exactly.

Consider first choosing $g(x) = x^m$ for $m = \{1, 2, \dots\}$. As was the case with $m = 1$, the function $F(x)$ diverges as $x = \pm 1$ for, and therefore requires $\mathcal{O}(2^{\mathsf{n_q}})$ Chebyshev polynomials to reproduce $F(x)$ exactly. Because each call to $\mathrm{e}^{-i\tau g(\hat{\xi}_k)}$ requires $\mathcal{O}(\mathsf{n_q}^m)$ gates, the cost of preparing a BE

of $f(\hat{\xi}_k)$ using $\mathrm{e}^{-i\tau g(\hat{\xi}_k)}$ as a building block is $\mathcal{O}(\mathsf{n}_\mathsf{q}^m 2^{n_q})$.

Consider now using non-polynomial $g(x)$. Further suppose that some $g(x)$ exists (which we discuss next) that results in $F(x) \sim f(g^{-1}(\frac{2}{\tau}\arccos(x)))$ being infinitely differentiable. While this choice may reduce the number of Chebyshev polynomials required, the cost of implementing $\mathrm{e}^{-i\tau g(\hat{\xi}_k)}$ for non-polynomial $g(x)$ is generally $\mathcal{O}(2^{\mathsf{n}_\mathsf{q}})$.

Taken together, these arguments imply that the cost of preparing BEs of $f_k(\hat{\xi}_k)$ using QETU generally scales exponentially with n_q. With this in mind, the rest of this section focuses on finding a $g(x)$ that reduces the cost from $\mathcal{O}(\mathsf{n}_\mathsf{q}2^{\mathsf{n}_\mathsf{q}})$ to $\mathcal{O}(2^{\mathsf{n}_\mathsf{q}})$.

3.6.2 Technical details

Our objective is to identify $g(x)$ such that $F(x) \sim f(g^{-1}(\frac{2}{\tau}\arccos(x)))$ is infinitely differentiable. The form of $F(x)$ suggests that $g(x) \sim \arccos(x)$ is a good starting point. The easiest way to accomplish this is to use the building block $\mathrm{e}^{-i2\arccos(\hat{\xi}_k/\alpha)}$, where $\alpha = \|\hat{\xi}_k\|$ to ensure the arccos is well defined. Setting $\tau = 1$ and replacing $\hat{A}$ in Thm 4 with $2\arccos(\hat{\xi}_k/\alpha)$, we observe that the QETU circuit now returns $F(\hat{\xi}_k/\alpha)$, which is only a function of $\hat{\xi}_k$. The procedure for determining the Chebyshev expansion of $f_k(\hat{\xi}_k)$ is now identical to that explained in Sec. 2.4; degree-d polynomial functions of $\hat{\xi}_k$ can be implemented using $\mathcal{O}(d)$ controlled calls to $\mathrm{e}^{-i2\arccos(\hat{\xi}_k/\alpha)}$.

Implementing $\mathrm{e}^{-i2\arccos(\hat{\xi}_k/\alpha)}$, however, is more complicated than $\mathrm{e}^{-i\tau\hat{\xi}_k}$. While the operator $\arccos(\hat{\xi}_k/\alpha)$ is still diagonal, and can easily be determined classically, it is an infinite degree Chebyshev polynomial in its argument $\hat{\xi}_k/\alpha$. This implies that its Pauli decomposition consists of $\mathcal{O}(2^{\mathsf{n}_\mathsf{q}})$ identity and R_z gates [88] and a controlled implementation of the operator $\mathrm{e}^{-i2\arccos(\hat{\xi}_k/\alpha)}$ requires $\mathcal{O}(2^{\mathsf{n}_\mathsf{q}})$ CNOT and R_z gates. On the other hand, constructing degree-d polynomial functions $f_k(\hat{\xi}_k)$ out of this building block only requires an additional multiplicative factor of d, giving an overall asymptotic scaling of $\mathcal{O}(d2^{\mathsf{n}_\mathsf{q}})$.

We thus see that choosing more complicated building blocks, such as $\mathrm{e}^{-i2\arccos(\hat{\xi}_k/\alpha)}$ which requires exponential (in n_q) resources to implement, can potentially make constructing the functions $f_k(\hat{\xi}_k)$ simpler. One such method that we propose in this section, which we call Linear Operators Via Exponentiation and Linear Combination of Unitaries (LOVE-LCU), uses $\mathrm{e}^{\pm i\arccos(f_k(\hat{\xi}_k)/\beta)}$ as a building block. Since

$$\frac{f_k(\hat{\xi}_k)}{\beta} = \frac{\mathrm{e}^{i\arccos\left(\frac{f_k(\hat{\xi}_k)}{\beta}\right)} + \mathrm{e}^{-i\arccos\left(\frac{f_k(\hat{\xi}_k)}{\beta}\right)}}{2}, \tag{30}$$

the construction of $f_k(\hat{\xi}_k)$ only needs LCU to combine these two terms. This can be implemented using a simple LCU circuit, shown in Fig. 8, with $\textsc{prep} = \mathrm{H} \otimes \mathbb{1}_s$ and

$$\begin{aligned}\textsc{sel} = |0\rangle\langle 0| \otimes \mathrm{e}^{i\arccos\left(\frac{f_k(\hat{\xi}_k)}{\beta}\right)} \\ + |1\rangle\langle 1| \otimes \mathrm{e}^{-i\arccos\left(\frac{f_k(\hat{\xi}_k)}{\beta}\right)}.\end{aligned} \tag{31}$$

The scale factor β can be chosen as the smallest possible value $\beta = \|f_k(\hat{\xi}_k)\|$.

The construction of the building block is very similar to before. The operators $\mathrm{e}^{\pm i\arccos\left(f_k(\hat{\xi}_k)/\beta\right)}$ are diagonal, which can be determined in a straightforward way using classical resources. It can then be decomposed into Pauli strings containing only the identity or Z-gates; in general, the decomposition will contain $\mathcal{O}(2^{\mathsf{n}_\mathsf{q}})$ Pauli strings. The circuits for the diagonal unitary matrices $\mathrm{e}^{\pm i\arccos(\hat{A})}$ can then be implemented using at most[6] $2^{\mathsf{n}_\mathsf{q}}-1$ R_z gates and $2^{\mathsf{n}_\mathsf{q}}-2$ CNOT gates [88], and the entire LOVE-LCU circuit requires $\mathcal{O}(2^{\mathsf{n}_\mathsf{q}})$ gates and a single ancillary qubit. This gate count can be further reduced by approximating the circuits for $\mathrm{e}^{i\arccos\hat{A}}$ in a systematic way by dropping rotation gates with small angles [62, 88]. LOVE-LCU can therefore prepare BEs of $f_k(\hat{\xi}_k)$ with resources that are independent of the degree d of the function f.

Another advantage of LOVE-LCU is that it is not restricted to BEs of single variables, but can also be applied to functions of multiple variables, *i.e.* the gate cost of using LOVE-LCU to construct a BE of the operator $f(\hat{\varphi}_1 - \hat{\varphi}_2)$ is the

[6] If the operator $\hat{A}$ has definite parity, for example $\hat{A} = \hat{\varphi}^n$ for some polynomial n, then only half of the Pauli strings will have non-zero coefficients. This can be understood by noting that $\arccos(x) = \pi/2 - \arcsin(x)$, which implies that $\arccos(x)$ has the same parity of the argument up to an overall constant shift.

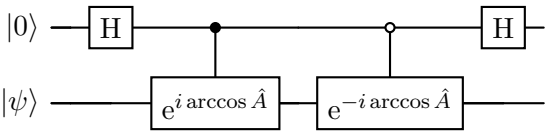

Figure 8: LCU circuit for block encoding of operator $\hat{A}$.

same as a general function $f(\hat{\varphi}_1, \hat{\varphi}_2)$. This implies that, instead of preparing BEs of $f_0(\hat{\varphi}_i)$, $f_0(\hat{\varphi}_j)$ and $f_2(\hat{\varphi}_i - \hat{\varphi}_j)$ separately and then adding them using another layer of LCU, one can directly use LOVE-LCU to prepare a BE of the sum $f_0(\hat{\varphi}_i) + f_0(\hat{\varphi}_j) + f_2(\hat{\varphi}_i - \hat{\varphi}_j)$.

The cost of LOVE-LCU can be further reduced by exploiting the structured form of the SEL oracle. In particular, SEL oracles of the form in Eq. (31) can be prepared with the techniques used in the control-free implementation of QETU. Rather than performing two controlled calls to $\mathrm{e}^{-i\arccos(f(\hat{\xi}_k/\alpha))}$, one can construct SEL for the cost of a single non-controlled circuit for $\mathrm{e}^{-i\arccos(f(\hat{\xi}_k/\alpha))}$ and an additional $2\mathsf{n_q}$ CNOT gates; the procedure for this control-free implementation can be found in Appendix B of Ref. [54].

While here we focused on using LOVE-LCU for block encoding diagonal operators, a similar construction can be applied to general Hermitian operators. The procedure is similar to the diagonal operator case; for an $\mathsf{n_q}$-qubit operator $\hat{A}$, one would evaluate $\mathrm{e}^{-i\arccos(\hat{A})}$ classically, and then exactly decompose the unitary operator into $\mathcal{O}(4^{\mathsf{n_q}})$ gates [83].

3.7 Constructing full Hamiltonian Block Encoding and Walk Operator

The previous sections discussed methods for constructing BEs of local terms in the Hamiltonian in Eq. (18). In this section, we discuss how to construct a BE of the full Hamiltonian $U_{\hat{H}}$. Furthermore, we show that there exists a simple operator S satisfying the conditions in Eqs. (11a) and (11b), which can be used to construct the walk operator $W_{\hat{H}}$ while avoiding the large overall prefactor when using the general qubitization procedure in Ref [66]. Here we only provide a high-level procedure; the detailed proofs are given in App. A.

For this discussion, we write the Hamiltonian in the general form $\hat{H} = \sum_\ell \hat{H}_\ell$, where $\hat{H}_\ell$ is a local term in the full Hamiltonian $\hat{H}$ that it is diagonalized by an efficiently implementable unitary (see Eq. (2)). Each $\hat{H}_\ell^{(D)}$ is then given by

$$\hat{H}_\ell^{(D)} = f_{k_\ell}(\hat{\xi}_{k_\ell}) \tag{32}$$

The BE of the full Hamiltonian $U_{\hat{H}}$ is constructed using a standard LCU procedure to add BEs of local terms $U_{\hat{H}_\ell}$. We denote by χ_ℓ the number of gates required to construct a given $U_{\hat{H}_\ell}$, and define $\chi_{\text{site}} = \max_\ell \chi_\ell$. Using LCU to add each of the $\mathcal{O}(\mathsf{N_\Lambda})$ local BEs $U_{\hat{H}_\ell}$ requires $\mathcal{O}(\log \mathsf{N_\Lambda})$ ancillary qubits, the register of which we denote by a_Λ. Since the local terms $f_{k_\ell}(\hat{\xi}_{k_\ell})$ for a given k_ℓ all have identical coefficients, the PREP oracle will have a relatively simple circuit implementation and will be sub-leading in the asymptotic gate cost. The SEL oracle is given by $\text{SEL} = \sum_\ell |\ell\rangle\langle\ell| \otimes U_{\hat{H}_\ell}$, where each $U_{\hat{H}_\ell}$ is controlled on $\mathcal{O}(\log \mathsf{N_\Lambda})$ qubits, each requiring $\mathcal{O}(\chi_{\text{site}} \log \mathsf{N_\Lambda})$ gates. The gate cost of the full SEL oracle, and therefore the gate cost of constructing $U_{\hat{H}}$, is $\mathcal{O}(\chi_{\text{site}} \mathsf{N_\Lambda} \log \mathsf{N_\Lambda})$.

We now describe how to construct the walk operator $W_{\hat{H}}$ in a way that avoids the expensive general qubitization procedure in Ref. [66]. The first step is to determine operators S for individual BEs $U_{\hat{H}_\ell}$ constructed using the methods in this work. Using this information, we then identify an S that can be used to construct $W_{\hat{H}}$ from the BE $U_{\hat{H}}$.

As shown in detail in Appendix A, for BEs $U_{\hat{H}_\ell}$ constructed using QSVT or QETU for even polynomials, $W_{\hat{H}_\ell}$ can be constructed by choosing S to be a single Z-gate acting on the signal and control qubit, respectively; this is due to the highly symmetric structure of QSVT and QETU circuits. Turning to LOVE-LCU, choosing S to be a single Z-gate acting on the ancillary qubit in the LOVE-LCU circuit also satisfies the relations in Eq. (11) necessary to construct $W_{\hat{H}_j}$. To see this, first observe that $\mathrm{Z}|0\rangle = |0\rangle$, which implies that Eq. (11a) is automatically satisfied. If we denote by U_A the circuit in Fig. 8, setting $U = \mathrm{e}^{i\arccos\hat{A}}$, we see

$$\begin{aligned}
&(\mathrm{Z}\otimes\mathbb{1}_s)U_A \\
&\quad= (\mathrm{HX}\otimes\mathbb{1}_s)\left(|0\rangle\langle 0|\otimes U + |1\rangle\langle 1|\otimes U^\dagger\right)(\mathrm{H}\otimes\mathbb{1}_s) \\
&\quad= (\mathrm{H}\otimes\mathbb{1}_s)\left(|1\rangle\langle 1|\otimes U + |0\rangle\langle 0|\otimes U^\dagger\right)(X\mathrm{H}\otimes\mathbb{1}_s) \\
&\quad= U_A^\dagger(Z\otimes\mathbb{1}_s),
\end{aligned} \tag{33}$$

where we have used the relations $\mathrm{ZH} = \mathrm{HX}$, $\mathrm{X}|0\rangle\langle 0| = |1\rangle\langle 1|\mathrm{X}$ and $\mathrm{X}|1\rangle\langle 1| = |0\rangle\langle 0|\mathrm{X}$, which satisfies the relation in Eq. (11b).

The BE $U_{\hat{H}}$ for the full Hamiltonian is constructed using LCU to combine local BEs $U_{\hat{H}_\ell}$, which, for BE methods used in this work, share a common S operator. By using a specific qubit organization of the individual $U_{\hat{H}_\ell}$'s, these properties imply that $W_{\hat{H}}$ for the full Hamiltonian can be constructed from $U_{\hat{H}}$ using the same common operator S, namely a single Z-gate. In particular, each $U_{\hat{H}_\ell}$ shares the same ancillary qubit register, denoted by a; if each $U_{\hat{H}_j}$ uses $n_{\text{anc}}^{(j)}$ ancillary qubits, then the register a contains $n_a = \max_j(n_{\text{anc}}^{(j)})$ qubits. Additionally, we make the choice that, if $n_{\text{anc}}^{(j)} < n_a$ for some j, the ancillary qubits used to construct $U_{\hat{H}_\ell}$ are $a_0, a_1, \ldots, a_{n_{\text{anc}}^{(j)}-1}$. Under these assumptions, it is shown in Appendix A that $W_{\hat{H}}$ can be constructed by choosing S to be a single Z, specifically $S = \mathbb{1}_{a_\Lambda} \otimes (\mathrm{Z} \otimes \mathbb{1})_a \otimes \mathbb{1}_s$.

We conclude by stressing that this simple construction of $W_{\hat{H}}$ is applicable to situations where one uses different methods to prepare different local BEs $U_{\hat{H}_\ell}$. This implies that one is free to choose the BE method that results in the lowest gate cost for each individual term $U_{\hat{H}_\ell}$.

4 Numerical demonstration

In Sec. 3 we described three main methods to block encode each local Hamiltonian $\hat{H}_\ell$, and provided the dependence of the required resources on the number of qubits $\mathsf{n_q}$ and degree of the polynomial of f_k. We did mention at the beginning of that section that some methods come with large pre-factors, which are not captured by the asymptotics provided. In this section we provide a more detailed numerical analysis of the resources required, including all required prefactors. Such an analysis requires making a detailed choice of the Hamiltonian, and we apply our techniques to the example of a scalar field theory Hamiltonian. Next, we compare the explicit gate cost of preparing BEs of local terms in the Hamiltonian using the methods described in Sec. 3. We then use the BE with lowest resource requirements and obtain explicit gate counts for using GQSP to time evolve a both a single and two-site system, and compare the cost to time evolution using 2$^{\text{nd}}$ and 4$^{\text{th}}$ order product formulas.

We consider a hypercubic lattice Λ of spatial dimension d and lattice spacing a. The Hamiltonians we consider can be written (using the notation from Sec. 3.1) as follows:

$$\hat{H} = \hat{H}_\varphi + \hat{H}_\pi\,, \tag{34a}$$

$$\hat{H}_\varphi = a^{\mathsf{d}} \left[\sum_{i=1}^{\mathsf{N}_\Lambda} V\left(\hat{\varphi}_i\right) + \sum_{\langle ij \rangle} \frac{(\hat{\varphi}_j - \hat{\varphi}_i)^2}{2a^2} \right], \tag{34b}$$

$$\hat{H}_\pi = a^{\mathsf{d}} \sum_{i=1}^{\mathsf{N}_\Lambda} \frac{1}{2}\hat{\pi}_i^2 = a^{\mathsf{d}} \sum_{i=1}^{\mathsf{N}_\Lambda} \frac{1}{2}\widehat{\mathcal{F}}_i^\dagger \left(\hat{\pi}_i^{(D)}\right)^2 \widehat{\mathcal{F}}_i\,, \tag{34c}$$

where $V\left(\hat{\varphi}_i\right)$ denotes the potential function. In this work, we consider the following potentials

$$V_1\left(\hat{\varphi}_i\right) = \frac{m^2}{2}\hat{\varphi}_i^2 + \frac{\lambda}{4!}\hat{\varphi}_i^4\,, \tag{35}$$

$$V_2\left(\hat{\varphi}_i\right) = g\cos(\hat{\varphi}_i)\,, \tag{36}$$

where $V_1(\hat{\varphi}_i)$ is the standard φ^4 scalar field theory potential, and the periodic potential $V_2(\hat{\varphi}_i)$ is relevant for dual basis formulations of compact U(1) gauge theories [8, 45], and a mixed-basis formulation of SU(2) gauge theory [31]. Setting the lattice spacing $a = 1$, the functions $f_k(\hat{\xi}_k)$ for this Hamiltonian are

$$f_0^{(1)}(\hat{\varphi}_i) = V_1(\hat{\varphi}_i) = \frac{m^2}{2}\hat{\varphi}_i^2 + \frac{\lambda}{4!}\hat{\varphi}_i^4\,, \tag{37a}$$

$$f_0^{(2)}(\hat{\varphi}_i) = V_2(\hat{\varphi}_i) = g\cos(\hat{\varphi}_i)\,, \tag{37b}$$

$$f_1(\hat{\pi}_i) = (\hat{\pi}_i^{(D)})^2\,, \tag{37c}$$

$$f_2(\hat{\varphi}_i - \hat{\varphi}_j) = \frac{1}{2}(\hat{\varphi}_j - \hat{\varphi}_i)^2\,. \tag{37d}$$

For all numerical results shown in this section, we set $m = 1, \lambda = 32$ and $g = 1$. To obtain the gate counts we use the basis gate-set containing CNOT, R_x, and R_z gates. There are several options for compiling multicontrolled gates. One choice is to use ancillary qubits as in Ref. [74] and then compile the resulting circuit down into CNOT, R_x, and R_z gates using, e.g., QISKIT transpilers. Another choice is to exploit the fact that all multicontrolled gates required for this work are diagonal matrices, which implies they can be compiled down into R_z and CNOT gates using the Walsh function formalism [88]. For the circuits studied in this work, we found that the latter choice resulted in smaller total gate counts, with significant reductions in some cases. All PREP oracles are implemented using QISKIT's exact state preparation algorithm. After compiling multicontrolled gates in this way, the final gate count in

terms of CNOT, R_x and R_z gates is obtained using QISKIT transpiler with optimization level 1.

Following the discussion in Sec. 3, the scale factor for $f_0^{(1)}(\hat{\varphi}_i)$, $f_1(\hat{\pi}_i)$, and $f_2(\hat{\varphi}_i - \hat{\varphi}_j)$ is the same for all BE methods considered (see Appendix C for a derivation of the scale factors when using vanilla LCU). Furthermore, the associated scale factors are given by their minimum values. As explained in more detail in Appendix C, the same is not true for $f_0^{(2)} = g\cos(\hat{\varphi})$; using standard LCU techniques results in a value of the scale factor $\sim 30\%$ larger than optimal. However, as we will show, LOVE-LCU has exponentially better complexity in $\mathsf{n_q}$ compared to standard LCU, and this slight difference in scale factor does not change our qualitative findings.

Figure 9 shows the number of rotation gates and ancillary qubits required to construct BEs of the functions in Eq. (37). Plots of CNOT gate counts are shown in App. B.

Considering first constructing BEs of $f_0^{(1)}(\hat{\varphi}_i)$ and $f_1(\hat{\pi}_i^{(D)})$, we find that, despite the asymptotically inefficient scaling with $\mathsf{n_q}$, using LOVE-LCU requires the fewest rotation gates for $\mathsf{n_q} \lesssim 11$; for $\mathsf{n_q} = 4$ qubits, LOVE-LCU requires only 13 rotation gates. For $\mathsf{n_q} \gtrsim 11$, due to the linear scaling with $\mathsf{n_q}$, using QSVT requires the fewest rotation gates. However, because digitization errors typically decrease exponentially with $\mathsf{n_q}$ [8, 58], realistic quantum simulations will not require so large a value of $\mathsf{n_q}$, implying that in some cases of practical interest LOVE-LCU will require the fewest rotation gates. Furthermore, QSVT requires $\mathcal{O}(\log \mathsf{n_q})$ ancillary qubits, while LOVE-LCU requires only a single ancillary qubit.

We now turn to preparing a BE of $f_2(\hat{\varphi}_i - \hat{\varphi}_j)$. Because this operator acts on two lattice sites and is of dimension $2^{2\mathsf{n_q}}$, the relative cost of the different methods are fundamentally different than the single qubit operators. In particular, the asymptotic gate cost of LOVE-LCU is now $\mathcal{O}(2^{2\mathsf{n_q}})$, which is the worst of all methods. Despite this fact, LOVE-LCU requires the fewest rotation gates for $\mathsf{n_q} \leq 5$. For $\mathsf{n_q} > 5$, using either QSVT or QETU with $\mathrm{e}^{-i\tau\hat{\varphi}}$ as a building block requires fewer gates than LOVE-LCU, with QSVT being the cheapest due to the $\mathcal{O}(\mathsf{n_q} \log \mathsf{n_q})$ asymptotic gate count scaling.

Because using $\mathsf{n_q} \sim 5$ is a much more realistic value for quantum simulations, it appears at first glance that using QSVT is the best method to prepare a BE of $f_2(\hat{\varphi}_i - \hat{\varphi}_j)$. However, LOVE-LCU has the advantage that it can prepare BEs of arbitrary $2\mathsf{n_q}$-qubit Hermitian operators for the same computational cost. This implies that one could use LOVE-LCU to prepare a BE of, e.g., $f_0^{(0)}(\hat{\varphi}_1) + f_0^{(0)}(\hat{\varphi}_2) + f_2(\hat{\varphi}_1 - \hat{\varphi}_2)$, for the same cost as just $f_2(\hat{\varphi}_1 - \hat{\varphi}_2)$. The same is not true for QSVT or the QETU based methods, which would require one to first prepare a BE of $f_0^{(0)}(\hat{\varphi}_1)$, $f_0^{(0)}(\hat{\varphi}_2)$, and $f_2(\hat{\varphi}_1 - \hat{\varphi}_2)$, and then add these operators using LCU. A comparison using different methods to prepare BE of the two-site Hamiltonian

$$\begin{aligned}\hat{H}_\varphi^{(2)} &= f_0^{(0)}(\hat{\varphi}_1) + f_0^{(0)}(\hat{\varphi}_2) + f_2(\hat{\varphi}_1 - \hat{\varphi}_2) \\ &= \sum_{j=1}^{2} \left(\frac{1}{2} m^2 \hat{\varphi}_j^2 + \frac{\lambda}{4!} \hat{\varphi}_j^2 \right) + \frac{1}{2} (\hat{\varphi}_1 - \hat{\varphi}_2)^2\end{aligned} \tag{38}$$

is shown in Fig. 10. Due to the need for an additional layer of LCU, we see that the cost of QSVT and QETU based methods has increased relative to standard LCU and LOVE-LCU methods; LOVE-LCU is now the best method for $\mathsf{n_q} \lesssim 6$, with QSVT requiring almost identical rotation gate counts for $\mathsf{n_q} = 6$.

The final operator we study is $\cos(\hat{\varphi})$. Using $\hat{A} = \cos(\hat{\varphi})$ is a special case of LOVE-LCU; the SEL oracle reduces to

$$\textsc{sel} = |0\rangle\langle 0| \otimes e^{-i\hat{\varphi}} + |1\rangle\langle 1| \otimes e^{i\hat{\varphi}}, \tag{39}$$

which can be implemented using the control-free procedure using $2\mathsf{n_q}$ of CNOT gates and $\mathsf{n_q}$ of R_z gates. While using QETU with $\mathrm{e}^{-i\tau\hat{\varphi}}$ can also achieve an asymptotic gate count of $\mathcal{O}(\mathsf{n_q})$, it will have a larger overall prefactor in the cost. Because $\cos(x)$ is an infinite degree polynomial, QSVT (or QETU using $\mathrm{e}^{-i2\arccos(\hat{\varphi}/\alpha)}$ as the building block), can only prepare approximate BEs of $\cos(\hat{\varphi})$; while one can use the efficient Jacobi-Anger expansion to do this, one must use a polynomial of degree $d = \mathcal{O}(\|\hat{\varphi}\| + \log(1/\epsilon)) = \mathcal{O}(2^{\mathsf{n_q}/2} + \log(1/\epsilon))$ to approximate to error $\mathcal{O}(\epsilon)$. Since $\cos(\hat{\varphi})$ is a sum of $\mathcal{O}(2^{\mathsf{n_q}})$ Pauli strings, the asymptotic gate cost of using LCU to BE $\cos(\hat{\varphi})$ requires $\mathcal{O}(\mathsf{n_q} 2^{\mathsf{n_q}})$ gates. From this discussion, we see that using LOVE-LCU has both the best asymptotic gate scaling as well as the smallest overall prefactor due to the simplicity of the SEL oracle. Despite the clear superiority of LOVE-LCU when preparing BEs of $\cos(\hat{\varphi})$, a comparison the gate count using LOVE-LCU and naive LCU to prepare a

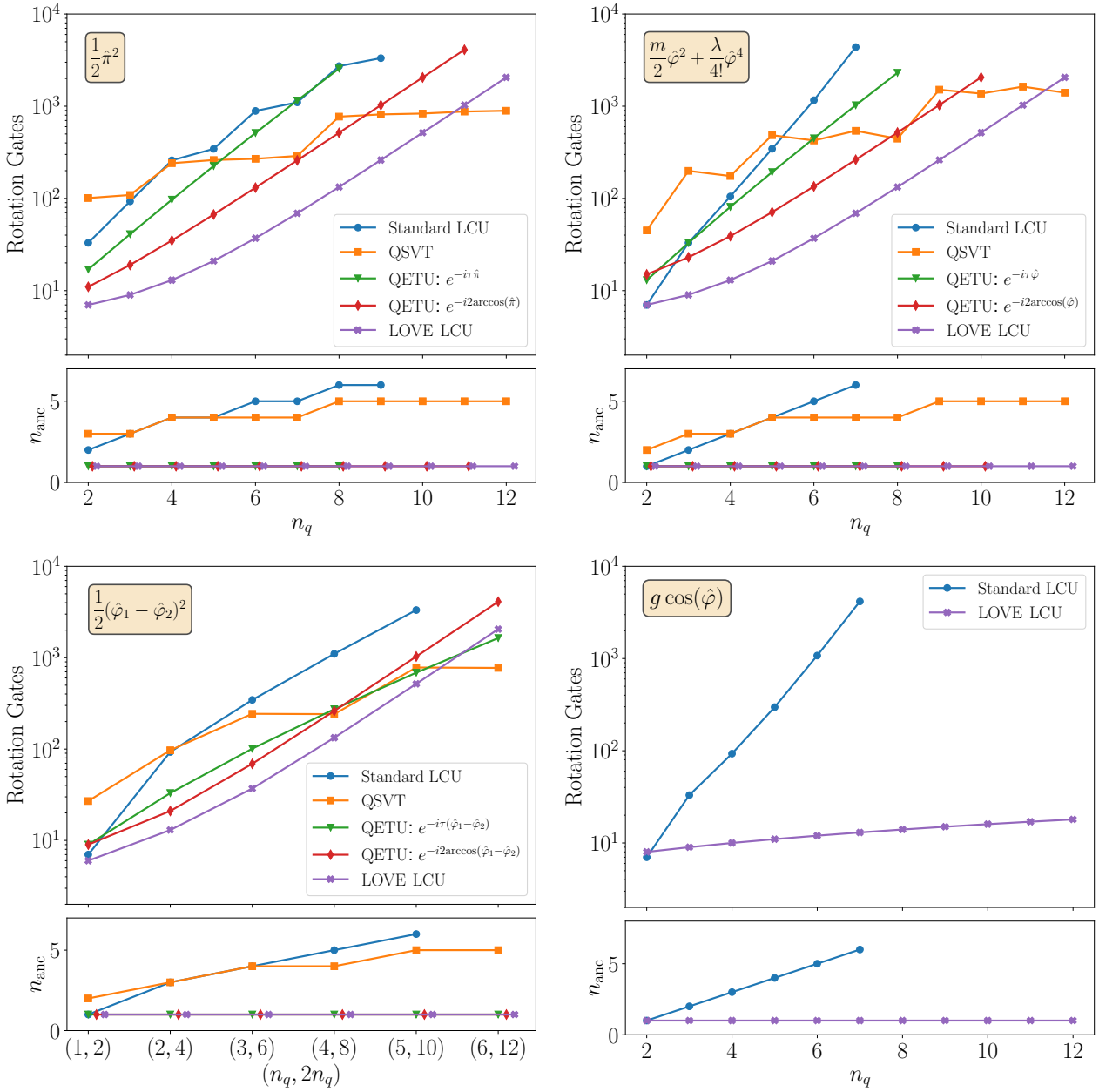

Figure 9: Rotation gate count and number of ancillary qubits required to block encode local bosonic operators. The top left, top right, bottom left, and bottom right plots show resource requirements to block encode $\frac{1}{2}\hat{\pi}^2$, $\frac{m}{2}\hat{\varphi}^2 + \frac{\lambda}{4!}\hat{\varphi}^4$, $\frac{1}{2}(\hat{\varphi}_1 - \hat{\varphi}_2)^2, g\cos(\hat{\varphi})$, respectively, for $m = 1, \lambda = 32, g = 1$. Different colored and shaped data points correspond to different methods to prepare the BE. Points with the same number of ancillary qubits have been shifted slightly for clarity. For the single-site operators $\hat{\pi}^2$ and $\frac{1}{m}\hat{\varphi}^2 + \frac{\lambda}{4!}\hat{\varphi}^4$, LOVE-LCU requires the fewest rotation gates for $\mathsf{n_q} \leq 11$. For the two-site operator $\frac{1}{2}(\hat{\varphi}_1 - \hat{\varphi}_2)^2$, LOVE-LCU requires the fewest rotation gates for $\mathsf{n_q} < 6$. LOVE-LCU constructs a BE of $\cos(\hat{\varphi})$ using the fewest gates for all $\mathsf{n_q}$.

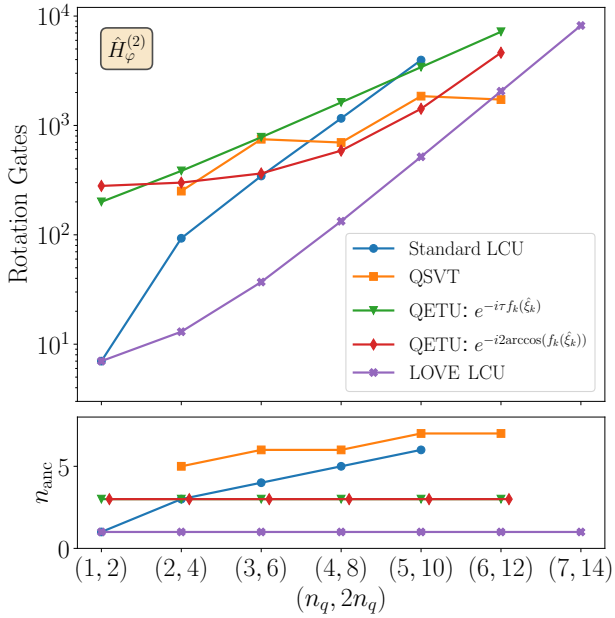

Figure 10: Rotation gate count and number of ancillary qubits required to block encode the two site Hamiltonian $\hat{H}_\varphi^{(2)}$ in Eq. (38). Different colored and shaped data points correspond to different methods to prepare the BE. Points with the same number of ancillary qubits have been shifted slightly for clarity. While QSVT and QETU based methods require using a final layer of LCU to add the local BEs $f_0^{(0)}(\hat{\varphi}_1), f_0^{(0)}(\hat{\varphi}_2), f_2(\hat{\varphi}_1 - \hat{\varphi}_2)$, LOVE-LCU does not. For this reason, LOVE-LCU requires only a single ancillary qubit. This advantage also leads to LOVE-LCU outperforming all other methods for $\mathsf{n_q} \lesssim 6$; for $\mathsf{n_q} > 6$, QSVT outperforms other methods due to the relative exponentially improved asymptotic scaling.

BE of $\cos(\hat{\varphi})$ is shown in the bottom right plot in Fig. 9; LOVE-LCU can prepare a BE of $\cos(\hat{\varphi})$ for $\mathsf{n_q} = 12$ using 18 rotation gates and a single ancillary qubit.

We now turn to the problem of simulating time evolution, considering both 2nd and 4th order PFs, and GQSP where the BE was constructed using LOVE-LCU. We study the gate cost to construct the time evolution operator as a function of the evolution time t and error ϵ; the error is defined as $\epsilon = \|U(t) - U_\epsilon(t)\|$, where $U(t)$ is the exact time evolution operator calculated using numerical exponentiation, and $U_\epsilon(t)$ is the approximate time evolution operator calculated using GQSP or PFs. Here we only show the rotation gate counts; CNOT gate counts are given in Appendix B.

The first system we study is the singe-site Hamiltonian

$$\begin{aligned} \hat{H}^{(1)} &= \mathcal{F}^\dagger f_1(\hat{\pi}^{(D)})\mathcal{F} + f_0^{(0)}(\hat{\varphi}) \\ &= \frac{1}{2}\hat{\pi}^2 + \frac{1}{2}m^2\hat{\varphi}^2 + \frac{\lambda}{4!}\hat{\varphi}^4. \end{aligned} \tag{40}$$

Figure 11 shows the number of rotation gates as a function of t and ϵ for $\mathsf{n_q} = 3$. We find that GQSP requires less rotation gates compared to using PFs for errors as large as $\epsilon \lesssim 10^{-2}$, which improves upon the work in Ref. [47] by ~ 5 orders of magnitude; the majority of the savings leading to this significant improvement are a factor of ~ 13 gate reduction in the construction of controlled calls to W_H, and a factor of two gate reduction from using GQSP as opposed to standard QSP. Another notable observation is that, in the region of error ϵ where using GQSP outperforms PFs, the 2^{nd} order PF outperforms the 4^{th} order PF. This is due to the fact that the 4^{th} order PF has not yet reached a small enough value of ϵ for the expected $(1/\epsilon)^{1/4}$ scaling to take effect. While GQSP algorithms are typically thought of as appropriate in the era of full fault-tolerance, our result for the single-site system suggests that for certain systems post-Trotter methods can outperform the PF-based ones sooner than expected.

To better understand how the relative costs change with the number of lattice sites, we now

Operator	Complexity from Ref. [46]	Complexity from QSVT
$\hat{\varphi}_i\hat{\varphi}_j$	$\mathcal{O}(\mathsf{n_q})$	$\mathcal{O}(\mathsf{n_q}\log\mathsf{n_q})$
$\hat{\varphi}^2$	$\mathcal{O}(\mathsf{n_q}^2)$	$\mathcal{O}\left(\mathsf{n_q}\log\mathsf{n_q}\right)$
$\hat{\pi}^2$	$\mathcal{O}(\mathsf{n_q}^2)$	$\mathcal{O}\left(\mathsf{n_q}\log\mathsf{n_q}\right)$
$\hat{\varphi}^4$	$\mathcal{O}(\mathsf{n_q}^2)$	$\mathcal{O}\left(\mathsf{n_q}\log\mathsf{n_q}\right)$

Table 2: Comparison of complexities to block-encode terms for $\mathsf{N}_\Lambda = \mathcal{O}(1)$ sites in scalar field theory Hamiltonian. The second column are complexities based on [46] which uses a "comparator" operator to reduce the complexity of LCU operations; The third column is the relevant QSVT method described in Sec. 3 (taking polynomials of degree $d \leq 4$).

consider the two-site Hamiltonian

$$\begin{aligned}\hat{H}^{(2)} &= \sum_{j=1}^{2}\left[\mathcal{F}_j^\dagger f_1(\hat{\pi}_j^{(D)})\mathcal{F}_j + f_0^{(0)}(\hat{\varphi}_j)\right] + f_2(\hat{\varphi}_1-\hat{\varphi}_2)\\ &= \sum_{j=1}^{2}\left(\frac{1}{2}\hat{\pi}_j^2 + \frac{1}{2}m^2\hat{\varphi}_j^2 + \frac{\lambda}{4!}\hat{\varphi}_j^2\right) + \frac{1}{2}\left(\hat{\varphi}_1-\hat{\varphi}_2\right)^2.\end{aligned} \tag{41}$$

Figure 12 shows the number of rotation gates as a function of t and ϵ for $\mathsf{n_q} = 3$. We see that GQSP now outperforms 2nd and 4th order PFs for $\epsilon \lesssim 5\times10^{-5}$ and $\epsilon \lesssim 10^{-7}$, respectively. While this result suggests one should use PFs for larger lattice sizes, it is likely that significant reductions in the GQSP gate count can be achieved by applying system specific optimizations for compiling the final LCU SEL and PREP oracles. Such gate reductions could make GQSP competitive with PFs for larger errors, and is an interesting direction for future work.

5 Discussion and Conclusion

In this work, we presented several new methods for preparing BEs for particular formulations of bosonic lattice field theories. The methods presented in this work relied on two properties of such digitizations, namely the simplicity of the local operators $\hat{\varphi}, \hat{\pi}^{(D)}$, and the fact that only a small number of qubits per site $\mathsf{n_q}$ are required to achieve the digitization errors necessary for realistic simulations.

Exploiting the simplicity of local operators $\hat{\varphi}$ and $\hat{\pi}^{(D)}$, we showed that QSVT can prepare BEs of degree d functions of local terms in the Hamiltonian using $\mathcal{O}(d\mathsf{n_q}\log\mathsf{n_q})$ gates. For polynomial functions with fixed degree d (relevant for simulating $\hat{\varphi}^4$ scalar field theories), this QSVT based method scales asymptotically as $\mathcal{O}(\mathsf{n_q}\log\mathsf{n_q})$ and improves upon the methods in Ref. [46] (see Table 2 for a comparison), and is identical to the scaling of preparing BEs of operators of the form $\sim(\hat{\varphi}_i-\hat{\varphi}_j)^2$ in Ref. [55].

Next, we considered construction of BEs with the aid of the QETU algorithm. While the simple operator $\mathrm{e}^{-i\tau\hat{\varphi}}$ is a natural building block for QETU circuits, because QETU generally requires approximating functions with discontinuous derivatives, we found that high degree polynomials are required to prepare BEs to a high precision. This poor convergence was overcome by exploiting the fact that the spectrum of the local operators are known exactly, and one can use QETU to prepare exact BEs by reproducing the function only at those values, which requires $\mathcal{O}(\mathsf{n_q}2^{\mathsf{n_q}})$ gates. Despite this technically inefficient scaling with $\mathsf{n_q}$, through explicit circuit constructions, we found that, due to QSVT having a relatively large prefactor in the gate cost, QETU outperforms QSVT for small values of $\mathsf{n_q}$. The comparison of different BEs considered in this work is summarized in Table 1.

Motivated by the observation that algorithms with inefficient scaling in $\mathsf{n_q}$ can require the fewest gates, we developed the conceptually simple LOVE-LCU approach. LOVE-LCU can prepare BEs of arbitrary Hermitian operators $\hat{A}$ acting on $\mathsf{n_q}$ qubits using only two controlled calls to $\mathrm{e}^{-i\arccos(\hat{A})}$, requiring a single ancillary qubit and $\mathcal{O}(2^{\mathsf{n_q}})$ ($\mathcal{O}(4^{\mathsf{n_q}})$) gates for diagonal (non-diagonal) operators. Our numerical investigations demonstrated that LOVE-LCU outperforms all other methods for preparing BEs of operators acting on $\lesssim 11$ qubits, at which point QSVT wins due to its relative exponentially improved gate scaling, albeit requiring more ancillary qubits than LOVE-LCU.

Our findings indicate that, while understanding asymptotic gate complexities can serve as a useful guide, explicit numerical investigations are essential to compare different BE techniques for a specific application as their performance depends heavily on the considered Hamiltonian, problem size, and desired precision. We note that one can view the different approaches to BEs considered in this work as belonging to the same family of QSP-based methods but with different building blocks,

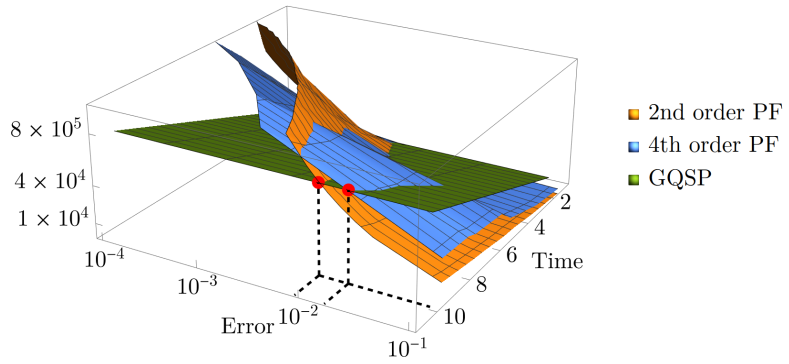

Figure 11: Rotation gate count as a function of error ϵ and simulation time t for simulating time-evolution of the single-site Hamiltonian in Eq. (40). The blue, orange, and green surfaces show results calculated using a 2^{nd} order PF, a 4^{th} order PF, and GQSP where the BE was constructing using LOVE-LCU. For $t = 10$, GQSP outperforms 2^{nd} and 4^{th} order PFs for $\epsilon \lesssim 10^{-2}$ and $\epsilon \lesssim 2 \times 10^{-2}$, respectively.

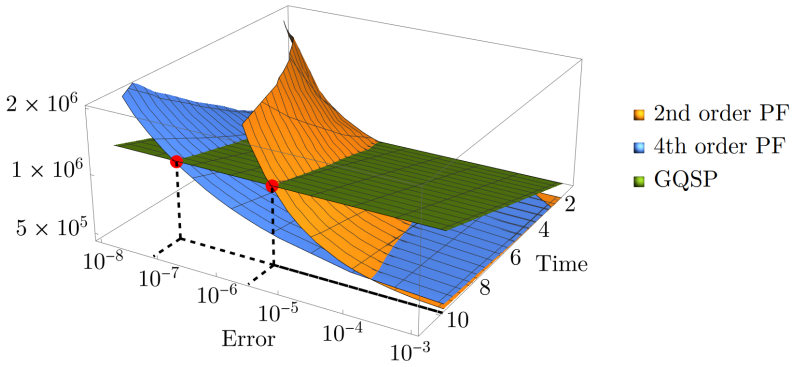

Figure 12: Rotation gate count as a function of error ϵ and simulation time t for simulating time-evolution of the two-site in Eq. (41). The blue, orange, and green surfaces show results calculated using a 2^{nd} order PF, a 4^{th} order PF, and GQSP where the BE was constructing using LOVE-LCU. For $t = 10$, GQSP outperforms 2^{nd} and 4^{th} order PFs for $\epsilon \lesssim 5 \times 10^{-6}$ and $\epsilon \lesssim 1 \times 10^{-7}$.

which thus change the functions of interest and their implementations. We observe that methods utilizing asymptotically inefficient building blocks can potentially outperform the asymptotically optimal ones in practically relevant regimes.

In addition to developing new methods for constructing BEs using these techniques, we also developed methods for efficiently constructing the walk operator $W_{\hat{H}}$, required for using QSP-based methods for Hamiltonian simulation. In particular, when BEs of local terms are constructed using any of the new methods presented in this work, and the full Hamiltonian is then constructed using a final LCU procedure to add these local BEs, the walk operator can be constructed as in Eq. (10) where S is a single Z-gate. Using GQSP combined with LOVE-LCU, we found that, for a single site anharmonic oscillator, GQSP outperforms PF methods for errors as small as $\epsilon \sim 10^{-2}$, which is ~ 5 orders of magnitude improved relative to previous work [47]. Note that because PF methods have better asymptotic scaling with the number of lattice sites than GQSP [47], the error threshold at which GQSP outperforms PFs is expected to be lower for larger lattice sizes. To better understand how the relative cost changes with the number of lattice sites, we studied the two-site system and found that GQSP now outperforms PF methods for errors $\epsilon \sim 10^{-7}$. This dramatic change, however, likely stems (at least in part) from using a compiler that is agnostic to the structure of the system studied, emphasizing the need for system specific circuit optimizations.

While we performed numerical gate count studies for a scalar field theory Hamiltonian, our methods can be directly applied to other bosonic theories formulated in terms of conjugate operators, including compact formulations of U(1) LGTs [8, 13] and the mixed-basis formulation of SU(2) LGT in Ref. [31]. Our methods can also be applied to LGTs that are not formulated in terms of conjugate variables, including the standard Kogut-Susskind Hamiltonian and Loop-String-Hadron formulations [51, 52, 76, 77]. Notably, this includes situations in which the Hamiltonian operator is non-local, i.e., contains an exponential number of Pauli operators. For example, the Hamiltonian operator of the gauge-fixed U(1) LGT [8, 13, 44, 45, 53] includes a term of the form $\cos\left(\sum_j^N B_j\right)$, where the magnetic field operator at site j, B_j, is analogous to the field operator $\hat{\varphi}$. While this operator is a sum of a number of Pauli strings exponential in the number of lattice sites, it can be readily encoded via the LOVE-LCU construction in Fig. 8 with a cost linear in the number of sites; this exponential reduction in cost allows one to avoid the usage of costly sparse oracle routines. More generally, the methods discussed in this work could be useful for block encoding Hamiltonians involving operator functions of multiple variables. One potential application is the first-quantized chemistry Hamiltonians, which can expressed in terms of position and derivative operators, such as $1/r$ and ∂_x^2, acting on fermionic wavefunctions.

Our work leads naturally to several interesting research directions. Regarding using QETU with $e^{-i\tau\hat{\varphi}_{\text{sh}}}$ as a building block, it was found in Ref. [54] that limiting the spectrum of $\hat{\varphi}_{\text{sh}}$ to be in the range $[\eta, \pi-\eta]$ and varying η allowed one to approximate Gaussian functions $\sim e^{-\hat{\varphi}^2}$ to a precision ϵ using only $\log(\mathsf{n_q}\log(1/\epsilon))$ gates. The same is not true, however, for polynomial operators of the form $\hat{\varphi}^n$ considered in this work; as $\eta \to \pi/2$ the scale factor of the BE goes to zero, leading to a significant increase in the cost of Hamiltonian simulation. It would be interesting to see if the method of varying η can achieve a $\mathcal{O}(\mathsf{n_q}\log(1/\epsilon))$ scaling for other functions that do not diverge for large argument, some relevant physical examples being central potentials of the form $1/r$, or $1/r^6$ and $1/r^{12}$ which appear in the Lennard-Jones potential, see, e.g., Ref. [55].

With an eye towards fault tolerance, it is essential to compile the circuits down to metrics suitable for the error correction protocols (e.g., T-gate counts) to assess the relative cost of the methods in this work. The QETU-based methods and LOVE-LCU inherently require the use of rotation gates, which implies they likely come with a large prefactor in the T-gate count. These methods, however, may serve useful for partial-fault tolerant implementations where only Clifford gates are implemented in a fault-tolerant manner, and rotation gates are implemented in a non-fault-tolerant way [2]. The QSVT-based methods are generally expected to require less T-gates, as one can use unary-LCU methods to reduce the cost of the SEL oracle, and the methods in Ref. [46] to construct a BE of $\hat{\varphi}$ avoiding the need for T-gates in the PREP oracle. It would be interesting to do a dedicated T-gate study using a combination the QSVT method

in this work and the efficient T-gate constructions in Refs. [6, 46].

While the total T-gate count is usually considered for the total cost of a fault-tolerant simulation (see, e.g., Refs. [32, 78]), there have been suggestions that another possible metric is instead T-gate *depth* [3]. This is similar in spirit to Amdahl's Law in classical computing, which states that the maximum possible speedup from parallelizing one's code is limited by components of the code that must be executed sequentially. In other words, Amdahl's Law describes the time a code takes to run in the limit of access to infinitely many classical computing nodes (barring communication time). By asking the same question in the context of using several fault-tolerant quantum computing nodes, T-gate depth is the metric for how parallelizable a circuit is. In scenarios where T-gate depth is the relevant cost, reductions in the depth of preparing BEs can be achieved using the modified LCU procedure in Ref. [18]. Using this or similar procedures, it is possible to imagine that the depth of constructing a BE of geometrically local Hamiltonians can be reduced to constant in the number of lattice sites. Such a method would result in the depth of QSP-based time evolution methods scaling linearly in the volume, which outperforms the volume scaling of the depth of PF based methods for geometrically local systems [28].

We conclude by pointing out that, while further improvements can likely be made within the paradigm of preparing BEs via QSP-based methods, dramatic reductions in the gate count will likely require considering new paradigms. Although our focus has been on constructing BEs using LCU for QSP-based time evolution, which typically requires long coherent circuits and full fault tolerance, our methods can be adapted for simulations relying solely on LCU with approaches like the Truncated Taylor Series [15]. To make these simulations more suitable for noisy intermediate-scale quantum (NISQ) devices, one promising direction is to explore recently developed randomized near-term implementations of LCU [23, 90], which could be combined with our constructions to enable more practical quantum simulations in the near term. Another promising avenue is to combine Trotter-based methods with near-optimal time evolution techniques [24, 35, 90], potentially leveraging the strengths of both approaches for improved efficiency. Additionally, the developing field of multi-variable QSP methods [43, 61, 70, 73, 79, 80] (this general class of methods includes multi-variable QETU), which leverages the structure of lattice field theories written as sums of many commuting variables, could result in significant cost reductions.

Acknowledgements

SH acknowledges support from the Berkeley Center for Theoretical Physics and the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02- 05CH11231. CWB and MK were supported by the DOE, Office of Science under contract DE-AC02-05CH11231, partially through Quantum Information Science Enabled Discovery (QuantISED) for High Energy Physics (KA2401032). This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Department of Energy Computational Science Graduate Fellowship under Award Number DE-SC0020347.

References

[1] QSPPACK. https://github.com/qsppack/QSPPACK.

[2] Yutaro Akahoshi, Kazunori Maruyama, Hirotaka Oshima, Shintaro Sato, and Keisuke Fujii. Partially fault-tolerant quantum computing architecture with error-corrected clifford gates and space-time efficient analog rotations. *PRX Quantum*, 5:010337, Mar 2024. DOI: 10.1103/PRXQuantum.5.010337. URL https://link.aps.org/doi/10.1103/PRXQuantum.5.010337.

[3] Matthew Amy, Dmitri Maslov, Michele Mosca, and Martin Roetteler. A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. *Trans. Comp.-Aided Des. Integ. Cir. Sys.*, 32(6): 818–830, jun 2013. ISSN 0278-0070. DOI: 10.1109/TCAD.2013.2244643. URL https://doi.org/10.1109/TCAD.2013.2244643.

[4] Lewis W. Anderson, Martin Kiffner, Tom O'Leary, Jason Crain, and Dieter Jaksch.

Solving lattice gauge theories using the quantum Krylov algorithm and qubitization. *Quantum*, 9:1669, 2025. DOI: 10.22331/q-2025-03-25-1669.

[5] Ryan Babbush, Dominic W. Berry, Yuval R. Sanders, Ian D. Kivlichan, Artur Scherer, Annie Y. Wei, Peter J. Love, and Alán Aspuru-Guzik. Exponentially more precise quantum simulation of fermions in the configuration interaction representation. *Quantum Sci. Technol.*, 3(1):015006, 2017. DOI: 10.1088/2058-9565/aa9463.

[6] Ryan Babbush, Craig Gidney, Dominic W. Berry, Nathan Wiebe, Jarrod McClean, Alexandru Paler, Austin Fowler, and Hartmut Neven. Encoding Electronic Spectra in Quantum Circuits with Linear T Complexity. *Phys. Rev. X*, 8(4):041015, 2018. DOI: 10.1103/PhysRevX.8.041015.

[7] João Barata, Niklas Mueller, Andrey Tarasov, and Raju Venugopalan. Single-particle digitization strategy for quantum computation of a ϕ^4 scalar field theory. *Phys. Rev. A*, 103(4):042410, 2021. DOI: 10.1103/PhysRevA.103.042410.

[8] Christian W. Bauer and Dorota M. Grabowska. Efficient representation for simulating U(1) gauge theories on digital quantum computers at all values of the coupling. *Phys. Rev. D*, 107(3):L031503, 2023. DOI: 10.1103/PhysRevD.107.L031503.

[9] Christian W. Bauer, Wibe A. de Jong, Benjamin Nachman, and Davide Provasoli. Quantum Algorithm for High Energy Physics Simulations. *Phys. Rev. Lett.*, 126(6):062001, 2021. DOI: 10.1103/PhysRevLett.126.062001.

[10] Christian W. Bauer, Marat Freytsis, and Benjamin Nachman. Simulating Collider Physics on Quantum Computers Using Effective Field Theories. *Phys. Rev. Lett.*, 127(21):212001, 2021. DOI: 10.1103/PhysRevLett.127.212001.

[11] Christian W. Bauer, So Chigusa, and Masahito Yamazaki. Quantum parton shower with kinematics. *Phys. Rev. A*, 109(3):032432, 2024. DOI: 10.1103/PhysRevA.109.032432.

[12] Christian W. Bauer et al. Quantum Simulation for High-Energy Physics. *PRX Quantum*, 4(2):027001, 2023. DOI: 10.1103/PRXQuantum.4.027001.

[13] Julian Bender and Erez Zohar. Gauge redundancy-free formulation of compact QED with dynamical matter for quantum and classical computations. *Phys. Rev. D*, 102(11):114517, 2020. DOI: 10.1103/PhysRevD.102.114517.

[14] Paul Benioff. The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. *J. Statist. Phys.*, 22(5):563–591, 1980. DOI: 10.1007/BF01011339.

[15] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Simulating Hamiltonian Dynamics with a Truncated Taylor Series. *Phys. Rev. Lett.*, 114(9):090502, 2015. DOI: 10.1103/PhysRevLett.114.090502.

[16] Dominic W. Berry, Danial Motlagh, Giacomo Pantaleoni, and Nathan Wiebe. Doubling the efficiency of Hamiltonian simulation via generalized quantum signal processing. *Phys. Rev. A*, 110(1):012612, 2024. DOI: 10.1103/PhysRevA.110.012612.

[17] Jan Lukas Bosse, Andrew M. Childs, Charles Derby, Filippo Maria Gambetta, Ashley Montanaro, and Raul A. Santos. Efficient and practical Hamiltonian simulation from time-dependent product formulas. *Nature Commun.*, 16(1):2673, 2025. DOI: 10.1038/s41467-025-57580-5.

[18] Gregory Boyd. Low-Overhead Parallelisation of LCU via Commuting Operators. 12 2023.

[19] R. Brower, S. Chandrasekharan, and U. J. Wiese. QCD as a quantum link model. *Phys. Rev. D*, 60:094502, 1999. DOI: 10.1103/PhysRevD.60.094502.

[20] Alexander J. Buser, Hrant Gharibyan, Masanori Hanada, Masazumi Honda, and Junyu Liu. Quantum simulation of gauge theory via orbifold lattice. *JHEP*, 09:034, 2021. DOI: 10.1007/JHEP09(2021)034.

[21] Daan Camps and Roel Van Beeumen. Fable: Fast approximate quantum circuits for block-encodings. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 104–113. IEEE, 2022.

[22] Daan Camps, Lin Lin, Roel Van Beeumen, and Chao Yang. Explicit Quantum Circuits

for Block Encodings of Certain Sparse Matrices. *SIAM J. Matrix Anal. Appl.*, 45(1): 801–827, 2024. DOI: 10.1137/22M1484298.

[23] Shantanav Chakraborty. Implementing any Linear Combination of Unitaries on Intermediate-term Quantum Computers. *Quantum*, 8:1496, 2024. DOI: 10.22331/q-2024-10-10-1496.

[24] Shantanav Chakraborty, Soumyabrata Hazra, Tongyang Li, Changpeng Shao, Xinzhao Wang, and Yuxin Zhang. Quantum singular value transformation without block encodings: Near-optimal complexity with minimal ancilla. 4 2025.

[25] So Chigusa and Masahito Yamazaki. Quantum simulations of dark sector showers. *Phys. Lett. B*, 834:137466, 2022. DOI: 10.1016/j.physletb.2022.137466.

[26] Andrew M Childs and Yuan Su. Nearly optimal lattice simulation by product formulas. *Physical review letters*, 123(5):050503, 2019.

[27] Andrew M. Childs and Nathan Wiebe. Hamiltonian Simulation Using Linear Combinations of Unitary Operations. *Quant. Inf. Comput.*, 12(11&12):0901–0924, 2012. DOI: 10.26421/QIC12.11-12-1.

[28] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu. Theory of Trotter Error with Commutator Scaling. *Phys. Rev. X*, 11(1):011020, 2021. DOI: 10.1103/physrevx.11.011020.

[29] Anthony Ciavarella, Natalie Klco, and Martin J. Savage. Trailhead for quantum simulation of SU(3) Yang-Mills lattice gauge theory in the local multiplet basis. *Phys. Rev. D*, 103(9):094501, 2021. DOI: 10.1103/PhysRevD.103.094501.

[30] Anthony N. Ciavarella and Christian W. Bauer. Quantum Simulation of SU(3) Lattice Yang Mills Theory at Leading Order in Large N. 2 2024.

[31] Irian D'Andrea, Christian W. Bauer, Dorota M. Grabowska, and Marat Freytsis. New basis for Hamiltonian SU(2) simulations. *Phys. Rev. D*, 109(7):074501, 2024. DOI: 10.1103/PhysRevD.109.074501.

[32] Zohreh Davoudi, Alexander F. Shaw, and Jesse R. Stryker. General quantum algorithms for Hamiltonian simulation with applications to a non-Abelian lattice gauge theory. *Quantum*, 7:1213, 2023. DOI: 10.22331/q-2023-12-20-1213.

[33] Yulong Dong, Lin Lin, and Yu Tong. Ground-State Preparation and Energy Estimation on Early Fault-Tolerant Quantum Computers via Quantum Eigenvalue Transformation of Unitary Matrices. *PRX Quantum*, 3 (4):040305, 2022. DOI: 10.1103/PRXQuantum.3.040305.

[34] Weijie Du and James P. Vary. Systematic input scheme for many-boson Hamiltonians via quantum walk. 7 2024.

[35] Paul K. Faehrmann, Mark Steudtner, Richard Kueng, Maria Kieferova, and Jens Eisert. Randomizing multi-product formulas for Hamiltonian simulation. *Quantum*, 6:806, 2022. DOI: 10.22331/q-2022-09-19-806.

[36] Roland C. Farrell, Ivan A. Chernyshev, Sarah J. M. Powell, Nikita A. Zemlevskiy, Marc Illa, and Martin J. Savage. Preparations for quantum simulations of quantum chromodynamics in 1+1 dimensions. II. Single-baryon β-decay in real time. *Phys. Rev. D*, 107(5):054513, 2023. DOI: 10.1103/PhysRevD.107.054513.

[37] Roland C. Farrell, Ivan A. Chernyshev, Sarah J. M. Powell, Nikita A. Zemlevskiy, Marc Illa, and Martin J. Savage. Preparations for quantum simulations of quantum chromodynamics in 1+1 dimensions. I. Axial gauge. *Phys. Rev. D*, 107(5):054512, 2023. DOI: 10.1103/PhysRevD.107.054512.

[38] Roland C. Farrell, Marc Illa, Anthony N. Ciavarella, and Martin J. Savage. Scalable Circuits for Preparing Ground States on Digital Quantum Computers: The Schwinger Model Vacuum on 100 Qubits. *PRX Quantum*, 5(2):020315, 2024. DOI: 10.1103/PRXQuantum.5.020315.

[39] Roland C. Farrell, Marc Illa, Anthony N. Ciavarella, and Martin J. Savage. Quantum simulations of hadron dynamics in the Schwinger model using 112 qubits. *Phys. Rev. D*, 109(11):114510, 2024. DOI: 10.1103/PhysRevD.109.114510.

[40] Richard P. Feynman. Simulating physics with computers. *Int. J. Theor. Phys.*, 21:467–488, 1982. DOI: 10.1007/BF02650179.

[41] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics.

In *51st Annual ACM SIGACT Symposium on Theory of Computing*, 6 2018. DOI: 10.1145/3313276.3316366.

[42] Joshua J. Goings, Alec White, Joonho Lee, Christofer S. Tautermann, Matthias Degroote, Craig Gidney, Toru Shiozaki, Ryan Babbush, and Nicholas C. Rubin. Reliably assessing the electronic structure of cytochrome P450 on today's classical computers and tomorrow's quantum computers. *Proc. Nat. Acad. Sci.*, 119(38):e2203533119, 2022. DOI: 10.1073/pnas.2203533119.

[43] Niladri Gomes, Hokiat Lim, and Nathan Wiebe. Multivariable QSP and Bosonic Quantum Simulation using Iterated Quantum Signal Processing. 8 2024.

[44] Dorota M. Grabowska, Christopher Kane, Benjamin Nachman, and Christian W. Bauer. Overcoming exponential scaling with system size in Trotter-Suzuki implementations of constrained Hamiltonians: 2+1 U(1) lattice gauge theories. 8 2022.

[45] Jan F. Haase, Luca Dellantonio, Alessio Celi, Danny Paulson, Angus Kan, Karl Jansen, and Christine A. Muschik. A resource efficient approach for quantum and classical simulations of gauge theories in particle physics. *Quantum*, 5:393, 2021. DOI: 10.22331/q-2021-02-04-393.

[46] Andrew Hardy et al. Optimized Quantum Simulation Algorithms for Scalar Quantum Field Theories. 7 2024.

[47] Siddharth Hariprakash, Neel S. Modi, Michael Kreshchuk, Christopher F. Kane, and Christian W. Bauer. Strategies for simulating the time evolution of Hamiltonian lattice field theories. *Phys. Rev. A*, 111(2):022419, 2025. DOI: 10.1103/PhysRevA.111.022419.

[48] Tatsuhiko N. Ikeda, Asir Abrar, Isaac L. Chuang, and Sho Sugiura. Minimum Trotterization Formulas for a Time-Dependent Hamiltonian. *Quantum*, 7:1168, 2023. DOI: 10.22331/q-2023-11-06-1168.

[49] Stephen P. Jordan, Keith S. M. Lee, and John Preskill. Quantum Computation of Scattering in Scalar Quantum Field Theories. *Quant. Inf. Comput.*, 14:1014–1080, 2014.

[50] Stephen P. Jordan, Hari Krovi, Keith S. M. Lee, and John Preskill. BQP-completeness of Scattering in Scalar Quantum Field Theory. *Quantum*, 2:44, 2018. DOI: 10.22331/q-2018-01-08-44.

[51] Saurabh V. Kadam, Indrakshi Raychowdhury, and Jesse R. Stryker. Loop-string-hadron formulation of an SU(3) gauge theory with dynamical quarks. *Phys. Rev. D*, 107(9):094513, 2023. DOI: 10.1103/PhysRevD.107.094513.

[52] Saurabh V. Kadam, Aahiri Naskar, Indrakshi Raychowdhury, and Jesse R. Stryker. Loop-string-hadron approach to SU(3) lattice Yang-Mills theory: Gauge invariant Hilbert space of a trivalent vertex. 7 2024.

[53] Christopher Kane, Dorota M. Grabowska, Benjamin Nachman, and Christian W. Bauer. Efficient quantum implementation of 2+1 U(1) lattice gauge theories with Gauss law constraints. 11 2022.

[54] Christopher F. Kane, Niladri Gomes, and Michael Kreshchuk. Nearly optimal state preparation for quantum simulations of lattice gauge theories. *Phys. Rev. A*, 110 (1):012455, 2024. DOI: 10.1103/PhysRevA.110.012455.

[55] Tyler Kharazi, Ahmad M. Alkadri, Jin-Peng Liu, Kranthi K. Mandadapu, and K. Birgitta Whaley. Explicit block encodings of boundary value problems for many-body elliptic operators. 7 2024.

[56] William M. Kirby, Sultana Hadi, Michael Kreshchuk, and Peter J. Love. Quantum simulation of second-quantized Hamiltonians in compact encoding. *Phys. Rev. A*, 104(4):042607, 2021. DOI: 10.1103/PhysRevA.104.042607.

[57] N. Klco, E. F. Dumitrescu, A. J. McCaskey, T. D. Morris, R. C. Pooser, M. Sanz, E. Solano, P. Lougovski, and M. J. Savage. Quantum-classical computation of Schwinger model dynamics using quantum computers. *Phys. Rev. A*, 98(3):032331, 2018. DOI: 10.1103/PhysRevA.98.032331.

[58] Natalie Klco and Martin J. Savage. Digitization of scalar fields for quantum computing. *Phys. Rev. A*, 99(5):052335, 2019. DOI: 10.1103/PhysRevA.99.052335.

[59] Natalie Klco, Jesse R. Stryker, and Martin J. Savage. SU(2) non-Abelian gauge field theory in one dimension on digital quantum computers. *Phys. Rev. D*, 101(7):074512, 2020. DOI: 10.1103/PhysRevD.101.074512.

[60] Michael Kreshchuk, William M. Kirby, Gary Goldstein, Hugo Beauchemin, and Peter J. Love. Quantum simulation of quantum field theory in the light-front formulation. *Phys. Rev. A*, 105(3):032418, 2022. DOI: 10.1103/PhysRevA.105.032418.

[61] Lorenzo Laneve and Stefan Wolf. On multivariate polynomials achievable with quantum signal processing. *Quantum*, 9:1641, 2025. DOI: 10.22331/q-2025-02-20-1641.

[62] Zhiyao Li, Dorota M. Grabowska, and Martin J. Savage. Sequency Hierarchy Truncation (SeqHT) for Adiabatic State Preparation and Time Evolution in Quantum Simulations. 7 2024.

[63] Lin Lin. Lecture Notes on Quantum Algorithms for Scientific Computation. 1 2022.

[64] Seth Lloyd. Universal Quantum Simulators. *Science*, 273(5278):1073, 1996. DOI: 10.1126/science.273.5278.1073.

[65] Guang Hao Low and Isaac L. Chuang. Optimal Hamiltonian Simulation by Quantum Signal Processing. *Phys. Rev. Lett.*, 118(1):010501, 2017. DOI: 10.1103/PhysRevLett.118.010501.

[66] Guang Hao Low and Isaac L. Chuang. Hamiltonian Simulation by Qubitization. *Quantum*, 3:163, 2019. DOI: 10.22331/q-2019-07-12-163.

[67] Hsuan-Hao Lu et al. Simulations of Subatomic Many-Body Physics on a Quantum Frequency Processor. *Phys. Rev. A*, 100(1):012320, 2019. DOI: 10.1103/PhysRevA.100.012320.

[68] Yu. I. Manin. The computable and the noncomputable. (Vychislimoe i nevychislimoe). Kibernetika. Moskva: "Sovetskoe Radio". 128 p. R. 0.45 (1980)., 1980.

[69] Mauro E. S. Morales, Pedro C. S. Costa, Giacomo Pantaleoni, Daniel K. Burgarth, Yuval R. Sanders, and Dominic W. Berry. Selection and improvement of product formulae for best performance of quantum simulation. 10 2022. DOI: 10.2478/qic-2025-0001.

[70] Hitomi Mori, Kaoru Mizuta, and Keisuke Fujii. Comment on "Multivariable quantum signal processing (M-QSP): prophecies of the two-headed oracle". *Quantum*, 8:1512, 2024. DOI: 10.22331/q-2024-10-29-1512.

[71] Danial Motlagh and Nathan Wiebe. Generalized Quantum Signal Processing. *PRX Quantum*, 5(2):020368, 2024. DOI: 10.1103/PRXQuantum.5.020368.

[72] Lento Nagano, Aniruddha Bapat, and Christian W. Bauer. Quench dynamics of the Schwinger model via variational quantum algorithms. *Phys. Rev. D*, 108(3):034501, 2023. DOI: 10.1103/PhysRevD.108.034501.

[73] Balázs Németh, Blanka Kövér, Boglárka Kulcsár, Roland Botond Miklósi, and András Gilyén. On variants of multivariate quantum signal processing and their characterizations. 12 2023.

[74] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 6 2012. DOI: 10.1017/cbo9780511976667.

[75] Bo Peng, Yuan Su, Daniel Claudino, Karol Kowalski, Guang Hao Low, and Martin Roetteler. Quantum simulation of boson-related Hamiltonians: techniques, effective Hamiltonian construction, and error analysis. *Quantum Sci. Technol.*, 10(2):023002, 2025. DOI: 10.1088/2058-9565/adbf42.

[76] Indrakshi Raychowdhury and Jesse R. Stryker. Solving Gauss's Law on Digital Quantum Computers with Loop-String-Hadron Digitization. *Phys. Rev. Res.*, 2(3): 033039, 2020. DOI: 10.1103/PhysRevResearch.2.033039.

[77] Indrakshi Raychowdhury and Jesse R. Stryker. Loop, string, and hadron dynamics in SU(2) Hamiltonian lattice gauge theories. *Phys. Rev. D*, 101(11):114502, 2020. DOI: 10.1103/PhysRevD.101.114502.

[78] Mason L. Rhodes, Michael Kreshchuk, and Shivesh Pathak. Exponential Improvements in the Simulation of Lattice Gauge Theories Using Near-Optimal Techniques. *PRX Quantum*, 5(4):040347, 2024. DOI: 10.1103/PRXQuantum.5.040347.

[79] Zane M. Rossi and Isaac L. Chuang. Multivariable quantum signal processing (M-QSP): prophecies of the two-headed oracle. *Quantum*, 6:811, 2022. DOI: 10.22331/q-2022-09-20-811.

[80] Zane M. Rossi, Jack L. Ceroni, and Isaac L. Chuang. Modular quantum signal processing in many variables. 9 2023.

[81] Martin J. Savage. Quantum computing for nuclear physics. *EPJ Web Conf.*, 296:01025, 2024. DOI: 10.1051/epjconf/202429601025.

[82] Kanav Setia and James D. Whitfield. Bravyi-Kitaev Superfast simulation of electronic structure on a quantum computer. *J. Chem. Phys.*, 148(16):164104, 2018. DOI: 10.1063/1.5019371.

[83] V. V. Shende, S. S. Bullock, and I. L. Markov. Synthesis of quantum-logic circuits. *IEEE Trans. Comput. Aided Design Integr. Circuits Syst.*, 25(6):1000–1010, 2006. DOI: 10.1109/tcad.2005.855930.

[84] Lloyd N. Trefethen. *Approximation Theory and Approximation Practice, Extended Edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2019. DOI: 10.1137/1.9781611975949. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611975949.

[85] James D. Watson. Randomly Compiled Quantum Simulation with Exponentially Reduced Circuit Depths. 11 2024.

[86] James D. Watson and Jacob Watkins. Exponentially Reduced Circuit Depths Using Trotter Error Mitigation. 8 2024.

[87] James D. Watson, Jacob Bringewatt, Alexander F. Shaw, Andrew M. Childs, Alexey V. Gorshkov, and Zohreh Davoudi. Quantum Algorithms for Simulating Nuclear Effective Field Theories. 12 2023.

[88] Jonathan Welch, Daniel Greenbaum, Sarah Mostame, and Alan Aspuru-Guzik. Efficient quantum circuits for diagonal unitaries without ancillas. *New Journal of Physics*, 16 (3):033040, mar 2014. DOI: 10.1088/1367-2630/16/3/033040. URL https://dx.doi.org/10.1088/1367-2630/16/3/033040.

[89] Christof Zalka. Simulating quantum systems on a quantum computer. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969):313–322, 1998.

[90] Pei Zeng, Jinzhao Sun, Liang Jiang, and Qi Zhao. Simple and High-Precision Hamiltonian Simulation by Compensating Trotter Error with Linear Combination of Unitary Operations. *PRX Quantum*, 6(1):010359, 2025. DOI: 10.1103/PRXQuantum.6.010359.

A Constructing the walk operator

In this appendix, we provide detailed proofs for the selection of the operator S (see Sec. 3) used in constructing the walk operator when the associated BE is implemented using QSVT (Lemma 5) or QETU (Lemma 6) for even polynomials. Specifically, we demonstrate that for both QSVT and QETU, the operator S can be chosen as a single Z-gate acting on the signal and control qubit, respectively, to satisfy the conditions in Eq. (11). We then consider the scenario where the BE for the full Hamiltonian $U_{\hat{H}}$ is constructed using LCU to combine several local $U_{\hat{H}_j}$ BEs, with each $U_{\hat{H}_j}$ constructed using either QSVT, QETU, or LOVE-LCU. We demonstrate that, as long as the circuit organization discussed in detail in Sec. 3.7 is used, the associated walk operator $W_{\hat{H}}$ can be constructed by choosing S to be a single Z-gate.

The following Lemma shows how to construct the walk operator given that that the BE was constructed using QSVT for even polynomials.

Lemma 5. *Let $\hat{A}$ and $f(\hat{A})$ be Hermitian operators acting on the Hilbert space $\mathcal{H}_s$, and, without loss of generality, assume $\|\hat{A}\| \leq 1$. Let U_A be a $(1, m)$ block encoding of $\hat{A}$, acting on the Hilbert space $\mathcal{H}_a \otimes \mathcal{H}_s$, where $\mathcal{H}_a$ is an m-qubit ancillary register. Let $U_{f(\hat{A})}$ be an $(\alpha, m+1)$ block encoding of $f(\hat{A})$, constructed using the even QSVT circuit (see Fig. 5) with the building block U_A, acting on the Hilbert space $\mathcal{H}_c \otimes \mathcal{H}_a \otimes \mathcal{H}_s$, where $\mathcal{H}_c$ is the single-qubit signal qubit. Then, the operator $S' = Z_c \otimes \mathbb{1}_a \otimes \mathbb{1}_s$ satisfies*

$$\left(\langle 0|_c \otimes \langle 0|_a \otimes \mathbb{1}_s\right) S' U_{f(\hat{A})} \left(|0\rangle_c \otimes |0\rangle_a \otimes \mathbb{1}_s\right) = f(\hat{A})/\alpha\,, \tag{42}$$

$$\left(\langle 0|_c \otimes \langle 0|_a \otimes \mathbb{1}_s\right) S' U_{f(\hat{A})} S' U_{f(\hat{A})} \left(|0\rangle_c \otimes |0\rangle_a \otimes \mathbb{1}_s\right) = \mathbb{1}_s\,. \tag{43}$$

Proof. Using the fact that $S' |0\rangle_c \otimes \mathbb{1}_{as} = |0\rangle_c \otimes \mathbb{1}_{as}$, we see that

$$\left(\langle 0|_c \otimes \langle 0|_a \otimes \mathbb{1}_s\right) S' U_{f(\hat{A})} \left(|0\rangle_c \otimes |0\rangle_a \otimes \mathbb{1}_s\right) = \left(\langle 0|_c \otimes \langle 0|_a \otimes \mathbb{1}_s\right) U_{f(\hat{A})} \left(|0\rangle_c \otimes |0\rangle_a \otimes \mathbb{1}_s\right) = f(\hat{A})/\alpha, \tag{44}$$

where the final equality in the above equation is the definition of the block encoding $U_{f(\hat{A})}$. The first relation is therefore satisfied.

To prove the second relation is satisfied, we show that $S' U_{f(\hat{A})} = U^\dagger_{f(\hat{A})} S'$. To see this, we use two circuit identities: $ZH = HX$ and $X_c \otimes \mathbb{1}_a \mathrm{CR}_{\tilde{\phi}} = \mathrm{CR}^\dagger_{\tilde{\phi}} X_c \otimes \mathbb{1}_a$ (see Fig. 6 for the circuit for $\mathrm{CR}_{\tilde{\phi}}$). The latter follows from the facts that X gates commute with the target of multi-controlled NOT gate and $XR_z(\theta) = R_z(\theta)^\dagger X$. Repeated use of these identities imply that commuting the Z gate in S' past the QSVT circuit flips the sign of the phases in each $\mathrm{CR}_{\tilde{\phi}}$. However, because the phases $\tilde{\phi}$ are symmetric, combined with the fact that one alternates calls to U_A and $U_A^\dagger$, flipping sign of the phases in each $\mathrm{CR}_{\tilde{\phi}}$ is equivalent to taking the Hermitian conjugate and thus $S' U_{f(\hat{A})} = U^\dagger_{f(\hat{A})} S'$. This implies that $S' U_{f(\hat{A})} S' U_{f(\hat{A})} = S' U_{f(\hat{A})} U^\dagger_{f(\hat{A})} S' = (S')^2 = \mathbb{1}_{cas}$ and so the second relation is satisfied.

□

The following Lemma shows how to construct the walk operator given that that the BE was constructed using QETU for even polynomials.

Lemma 6. *Let $\hat{A}$ and $f(\hat{A})$ be Hermitian operators acting on the Hilbert space $\mathcal{H}_s$. Let $U_{f(\hat{A})}$ be an $(\alpha, 1)$ block-encoding of $f(\hat{A})$, constructed using the QETU circuit for even polynomials (see Fig. 7) with the building block $\mathrm{e}^{-i\tau\hat{A}}$, acting on the Hilbert space $\mathcal{H}_c \otimes \mathcal{H}_s$ where $\mathcal{H}_c$ is the Hilbert space of the control qubit in the QETU circuit. Then, the operator $S' = Z_c \otimes \mathbb{1}_s$ satisfies*

$$\left(\langle 0|_c \otimes \mathbb{1}_s\right) S' U_{f(\hat{A})} \left(|0\rangle_c \otimes \mathbb{1}_s\right) = f(\hat{A})\,, \tag{45}$$

$$\left(\langle 0|_c \otimes \mathbb{1}_s\right) S' U_{f(\hat{A})} S' U_{f(\hat{A})} \left(|0\rangle_c \otimes \mathbb{1}_s\right) = \mathbb{1}_s\,. \tag{46}$$

Proof. Using the fact that $S' \left|0\right\rangle_c \otimes \mathbb{1}_s = \left|0\right\rangle_c \otimes \mathbb{1}_s$, we see that

$$\left(\left\langle 0\right|_c \otimes \mathbb{1}_s\right) S' U_{f(\hat{A})} \left(\left|0\right\rangle_c \otimes \mathbb{1}_s\right) = \left(\left\langle 0\right|_c \otimes \mathbb{1}_s\right) U_{f(\hat{A})} \left(\left|0\right\rangle_c \otimes \mathbb{1}_s\right) = f(\hat{A})\,, \tag{47}$$

where the final equality in the equation above is the definition of the block encoding $U_{\hat{H}}$. The first relation is therefore satisfied.

To show that the second relation is satisfied, we will demonstrate that $S' U_{f(\hat{A})} = U^\dagger_{f(\hat{A})} S'$. We denote by C_U the controlled call to the building block $\mathrm{e}^{-i\tau\hat{A}}$. Note that, due to the structure of alternating calls to C_U and $C_U^\dagger$, combined with the fact that the rotation angles $\{\tilde{\varphi}_j\}$ are symmetric, the circuit for $U^\dagger_{f(\hat{A})}$ is equivalent to negating the phases in the single-qubit R_x gates in the original QETU circuit. We will now show that commuting $Z_c \otimes \mathbb{1}_s$ past $U_{\hat{H}}$ has the effect of negating the signs of the phases in the R_x gates. We note the following circuit identities: commuting a Z gate past an $R_x(\theta)$ gate flips the sign of the rotation angle, *i.e.*, $Ze^{i\theta X} = e^{-i\theta X}Z$, and Z gates acting on the control qubit commute with controlled calls to $e^{\pm i\tau\hat{A}}$. This can be seen by noting that diagonal matrices commute, which implies $[Z, |0\rangle\langle 0|] = [Z, |1\rangle\langle 1|] = 0$. These identities imply that commuting the Z gate past the QETU circuit flips the sign of all rotation angles, and therefore $S' U_{f(\hat{A})} = U^\dagger_{f(\hat{A})} S'$. Using this result, we see that $S' U_{f(\hat{A})} S' U_{f(\hat{A})} = S' U_{f(\hat{A})} U^\dagger_{f(\hat{A})} S' = (S')^2 = \mathbb{1}_{cs}$, and so the second relation is satisfied. □

The following lemma shows how to construct the walk operator $W_{\hat{H}}$ for the full Hamiltonian from $U_{\hat{H}}$ when $U_{\hat{H}}$ is constructed using LCU to combine local $U_{\hat{H}_j}$'s, where each $U_{\hat{H}_j}$ shares a common S operator for constructing $W_{\hat{H}_j}$.

Lemma 7. *Let $\hat{H} = \sum_{j=0}^{M-1} \beta_j \hat{H}_j$ be a Hamiltonian acting on a Hilbert space $\mathcal{H}_s$ with $\beta_j \in \mathbb{R}^+$. Without loss of generality, assume $\|\hat{H}_j\| \leq 1, \forall j$. Let $U_{\hat{H}_j}$ be a $(1, m_j)$ block-encoding of $\hat{H}_j$, acting on the joint Hilbert space $\mathcal{H}_a \otimes \mathcal{H}_s$, where $\mathcal{H}_a$ is the Hilbert space of ancillary qubit register containing $\max_j(m_j)$ qubits. Furthermore, let $U_{\hat{H}} = (\textsc{prep}^\dagger \otimes \mathbb{1}_{as})\textsc{sel}(\textsc{prep} \otimes \mathbb{1}_{as})$ be a $\left(\|\vec{\beta}\|_1, \lceil \log_2 M \rceil + \max_j(m_j)\right)$ block-encoding of $\hat{H}$ acting on the joint Hilbert space $\mathcal{H}_{a_\Lambda} \otimes \mathcal{H}_a \otimes \mathcal{H}$, where $\mathcal{H}_{a_\Lambda}$ is the Hilbert space of the ancillary register containing $\lceil \log_2 M \rceil$ qubits, $\|\vec{\beta}\|_1 \equiv \sum_{j=0}^{M-1} |\beta_j|$, $\textsc{sel} = \sum_{j=0}^{M-1} (|j\rangle\langle j|)_{a_\Lambda} \otimes U_{\hat{H}_j}$ and $\textsc{prep}|0\rangle_{a_\Lambda} = \frac{1}{\sqrt{\|\vec{\beta}\|_1}} \sum_{j=0}^{M-1} \sqrt{\beta_j} |j\rangle_{a_\Lambda}$. Define $S' = \mathbb{1}_{a_\Lambda} \otimes S_a \otimes \mathbb{1}_s$. If for each $U_{\hat{H}_j}$ the operator S_a satisfies*

$$\left(\left\langle 0\right|_a \otimes \mathbb{1}_s\right) (S_a \otimes \mathbb{1}_s) U_{\hat{H}_j} \left(\left|0\right\rangle_a \otimes \mathbb{1}_s\right) = \hat{H}_j\,, \tag{48}$$

$$\left(\left\langle 0\right|_a \otimes \mathbb{1}_s\right) \left[S_a \otimes \mathbb{1}_s) U_{\hat{H}_j}\right]^2 \left(\left|0\right\rangle_a \otimes \mathbb{1}_s\right) = \mathbb{1}_s\,, \tag{49}$$

then

$$\left(\left\langle 0\right|_{a_\Lambda} \otimes \left\langle 0\right|_a \otimes \mathbb{1}_s\right) S' U_{\hat{H}} \left(\left|0\right\rangle_{a_\Lambda} \otimes \left|0\right\rangle_a \otimes \mathbb{1}_s\right) = \frac{\hat{H}}{\|\vec{\beta}\|_1}\,, \tag{50}$$

$$\left(\left\langle 0\right|_{a_\Lambda} \otimes \left\langle 0\right|_a \otimes \mathbb{1}_s\right) S' U_{\hat{H}} S' U_{\hat{H}} \left(\left|0\right\rangle_{a_\Lambda} \otimes \left|0\right\rangle_a \otimes \mathbb{1}_s\right) = \mathbb{1}_s\,. \tag{51}$$

Proof. First, note that

$$\begin{aligned} S' U_{\hat{H}} &= \left(\mathbb{1}_{a_\Lambda} \otimes S_a \otimes \mathbb{1}_s\right) \left(\textsc{prep}^\dagger \otimes \mathbb{1}_{as}\right) \left(\sum_{j=0}^{M-1} (|j\rangle\langle j|)_{a_\Lambda} \otimes U_{\hat{H}_j}\right) \left(\textsc{prep} \otimes \mathbb{1}_{as}\right) \\ &= \sum_{j=0}^{M-1} \left[\textsc{prep}^\dagger (|j\rangle\langle j|)_{a_\Lambda} \textsc{prep}\right] \otimes \left[(S_a \otimes \mathbb{1}_s) U_{\hat{H}_j}\right]. \end{aligned} \tag{52}$$

From Eq. (52) it immediately follows that the first relation is satisfied:

$$
\begin{aligned}
&(\langle 0|_{a_\Lambda} \otimes \langle 0|_a \otimes \mathbb{1}_s) S' U_{\hat{H}} \left(|0\rangle_{a_\Lambda} \otimes |0\rangle_a \otimes \mathbb{1}_s\right) \\
&\quad = (\langle 0|_{a_\Lambda} \otimes \langle 0|_a \otimes \mathbb{1}_s) \sum_{j=0}^{M-1} \left[\mathrm{PREP}^\dagger (|j\rangle\langle j|)_{a_\Lambda} \mathrm{PREP}\right] \otimes \left[(S_a \otimes \mathbb{1}_s) U_{\hat{H}_j}\right] \left(|0\rangle_{a_\Lambda} \otimes |0\rangle_a \otimes \mathbb{1}_s\right) \\
&\quad = \sum_{j=0}^{M-1} \langle 0|_{a_\Lambda} \left[\mathrm{PREP}^\dagger (|j\rangle\langle j|)_{a_\Lambda} \mathrm{PREP}\right] |0\rangle_{a_\Lambda} \otimes (\langle 0|_a \otimes \mathbb{1}_a) \left[(S_a \otimes \mathbb{1}_s) U_{\hat{H}_j}\right] \left(|0\rangle_a \otimes \mathbb{1}_a\right) \\
&\quad = \sum_{j=0}^{M-1} \langle 0|_{a_\Lambda} \left[\mathrm{PREP}^\dagger (|j\rangle\langle j|)_{a_\Lambda} \mathrm{PREP}\right] |0\rangle_{a_\Lambda} \hat{H}_j \\
&\quad = \frac{1}{\|\vec{\beta}\|_1} \sum_{j,k,l=0}^{M-1} \sqrt{\beta_k \beta_\ell} \langle k|j\rangle\langle j|l\rangle \hat{H}_j \\
&\quad = \frac{1}{\|\vec{\beta}\|_1} \sum_{j=0}^{M-1} \beta_j \hat{H}_j \\
&\quad = \frac{\hat{H}}{\|\vec{\beta}\|_1} .
\end{aligned}
\tag{53}
$$

The second relation also follows in a straightforward way:

$$
\begin{aligned}
&(\langle 0|_{a_\Lambda} \otimes \langle 0|_a \otimes \mathbb{1}_s) S' U_{\hat{H}} S' U_{\hat{H}} \left(|0\rangle_{a_\Lambda} \otimes |0\rangle_a \otimes \mathbb{1}_s\right) \\
&\quad = (\langle 0|_{a_\Lambda} \otimes \langle 0|_a \otimes \mathbb{1}_s) \left[\sum_{j=0}^{M-1} \left[\mathrm{PREP}^\dagger (|j\rangle\langle j|)_{a_\Lambda} \mathrm{PREP}\right] \otimes \left[(S_a \otimes \mathbb{1}_s) U_{\hat{H}_j}\right]\right]^2 \left(|0\rangle_{a_\Lambda} \otimes |0\rangle_a \otimes \mathbb{1}_s\right) \\
&\quad = (\langle 0|_{a_\Lambda} \otimes \langle 0|_a \otimes \mathbb{1}_s) \left[\sum_{j=0}^{M-1} \left[\mathrm{PREP}^\dagger (|j\rangle\langle j|)_{a_\Lambda} \mathrm{PREP}\right] \otimes \left[(S_a \otimes \mathbb{1}_s) U_{\hat{H}_j}\right]^2\right] \left(|0\rangle_{a_\Lambda} \otimes |0\rangle_a \otimes \mathbb{1}_s\right) \\
&\quad = \sum_{j=0}^{M-1} \langle 0|_{a_\Lambda} \mathrm{PREP}^\dagger (|j\rangle\langle j|)_{a_\Lambda} \mathrm{PREP} |0\rangle_{a_\Lambda} \left(\langle 0|_a \otimes \mathbb{1}_s\right) \left[(S_a \otimes \mathbb{1}_s) U_{\hat{H}_j}\right]^2 \left(|0\rangle_a \otimes \mathbb{1}_s\right)\Big) \\
&\quad = \frac{1}{\|\vec{\beta}\|_1} \sum_{j,k,l=0}^{M-1} \sqrt{\beta_k \beta_\ell} \langle k|j\rangle\langle j|l\rangle \\
&\quad = \frac{1}{\|\vec{\beta}\|_1} \sum_{j=0}^{M-1} \beta_j \\
&\quad = \mathbb{1}_s.
\end{aligned}
\tag{54}
$$

□

B CNOT gate counts for scalar field theory simulation

In this appendix, we present the associated CNOT gate counts from the numerical studies conducted in Sec. 4. All results are qualitatively similar to the rotation gate counts discussed in the main text.

Figure 13 shows the CNOT gate counts, associated with the rotation gate counts in Fig. 9, for BEs of the local terms $\frac{1}{2}\hat{\pi}^2$, $\frac{m}{2}\hat{\varphi}^2 + \frac{\lambda}{4!}\hat{\varphi}^4$, $\frac{1}{2}(\hat{\varphi}_1 - \hat{\varphi}_2)$, and $g\cos(\hat{\varphi})$.

Figure 14 shows the CNOT gate counts, associated with the rotation gate counts in Fig. 10, for BEs of the two-site $\hat{\varphi}$ Hamiltonian in Eq. (38).

Figure 15 shows the CNOT gate counts, associated with the rotation gate counts in Fig. 11, for simulating time evolution of the single site Hamiltonian in Eq. (40) using a 2nd order PF, a 4th order PF, and GQSP where W_H was constructed using LOVE-LCU.

Figure 16 shows the CNOT gate counts, associated with the rotation gate counts in Fig. 12, for simulating time evolution of the two-site Hamiltonian in Eq. (40) using a 2nd order PF, a 4th order PF, and GQSP where W_H was constructed using LOVE-LCU.

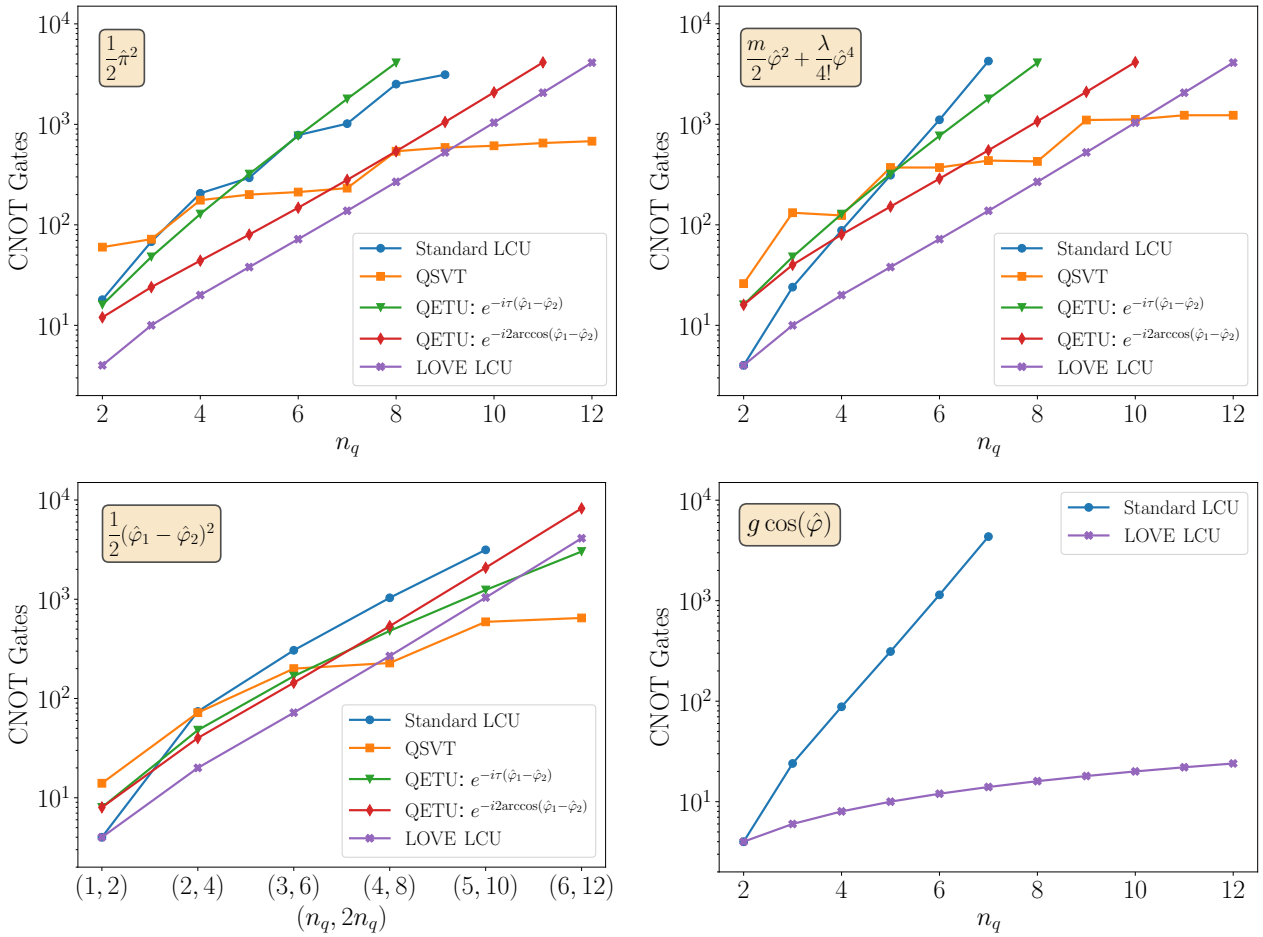

Figure 13: CNOT gate count and number of ancillary qubits required to block encode local bosonic operators. The top left, top right, bottom left, and bottom right plots show resource requirements to block encode $\frac{1}{2}\hat{\pi}^2$, $\frac{m}{2}\hat{\varphi}^2 + \frac{\lambda}{4!}\hat{\varphi}^4$, $\frac{1}{2}(\hat{\varphi}_1 - \hat{\varphi}_2)^2, g\cos(\hat{\varphi})$, respectively, for $m = 1, \lambda = 32, g = 1$. Different colored and shaped data points correspond to different methods to prepare the BE. For the single-site operators $\hat{\pi}^2$ and $\frac{m}{2}\hat{\varphi}^2 + \frac{\lambda}{4!}\hat{\varphi}^4$, LOVE-LCU requires the fewest rotation gates for $n_q \lesssim 9$. For the two-site operator $\frac{1}{2}(\hat{\varphi}_1 - \hat{\varphi}_2)^2$, LOVE-LCU requires the fewest rotation gates for $n_q \lesssim 4$. LOVE-LCU constructs a BE of $g\cos(\hat{\varphi})$ using the fewest gates for all n_q.

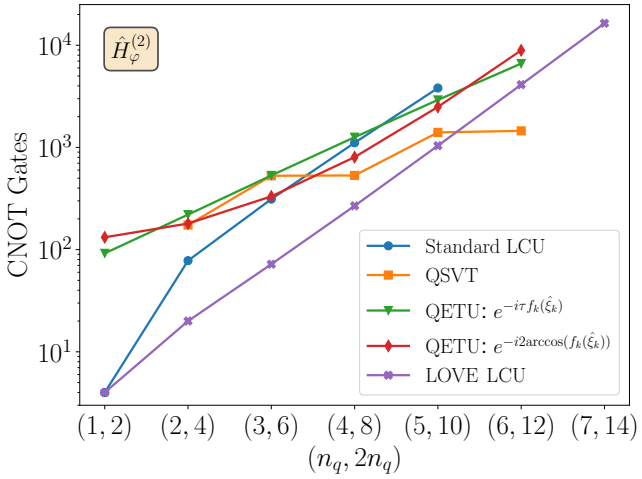

Figure 14: CNOT gate count required to block encode the two site Hamiltonian $\hat{H}_\varphi^{(2)}$ in Eq. (38). Different colored and shaped data points correspond to different methods to prepare the BE. Points with the same number of ancillary qubits have been shifted slightly for clarity. While QSVT- and QETU-based methods require using a final layer of LCU to add the local BEs $f_0^{(0)}(\hat{\varphi}_1), f_0^{(0)}(\hat{\varphi}_2), f_2(\hat{\varphi}_1 - \hat{\varphi}_2)$, LOVE-LCU does not. This advantage also leads to LOVE-LCU outperforming all other methods for $n_q \lesssim 5$; for $n_q > 5$, QSVT outperforms other methods due to the relative exponentially improved asymptotic scaling.

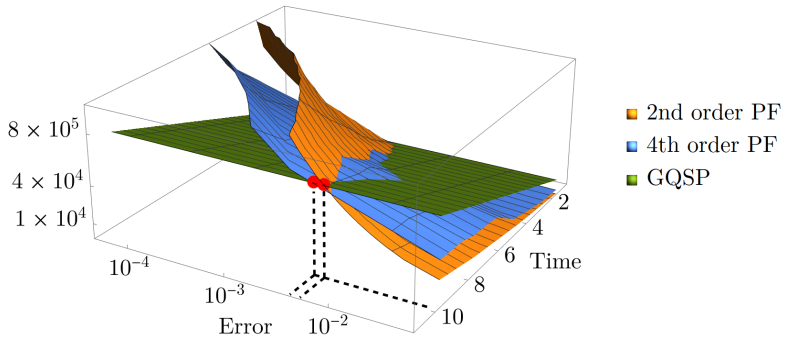

Figure 15: CNOT gate count as a function of error ϵ and simulation time t for simulating time-evolution of the single-site and two-site Hamiltonian in Eq. (40). The blue, orange, and green surfaces show results calculated using a 2^{nd} order PF, a 4^{th} order PF, and GQSP where the BE was constructing using LOVE-LCU. For $t = 10$, GQSP outperforms both 2^{nd} and 4^{th} order PFs for $\epsilon \lesssim 5 \times 10^{-3}$.

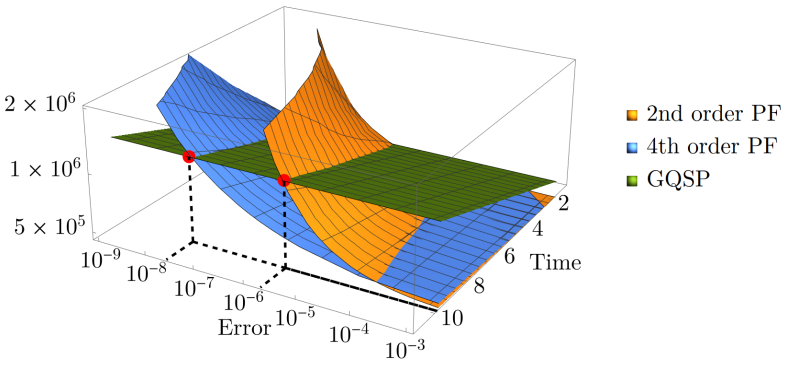

Figure 16: CNOT gate count as a function of error ϵ and simulation time t for simulating time-evolution of the two-site Hamiltonian in Eq. (41). The blue, orange, and green surfaces show results calculated using a 2^{nd} order PF, a 4^{th} order PF, and GQSP where the BE was constructing using LOVE-LCU. For $t = 10$, GQSP outperforms 2^{nd} and 4^{th} order PFs for $\epsilon \lesssim 1 \times 10^{-6}$ and $\epsilon \lesssim 5 \times 10^{-8}$.

C Scale factor for Block Encodings constructed using LCU

In this appendix, we derive the scale factors for block encodings of $f_k(\hat{\xi}_k)$, given in Eq. (37) in the main text, when prepared using standard LCU techniques.

To begin, we consider the scale factor for $\hat{\varphi}^d$; the result is analogous for $\left(\hat{\pi}^{(D)}\right)^d$. Using Eq. (23) we have

$$\begin{aligned}\hat{\varphi}^d &= (-\varphi_{\max})^d \left(\frac{2^{\mathsf{n_q}-1}}{2^{\mathsf{n_q}}-1}\right)^d \left(\sum_{m=0}^{\mathsf{n_q}-1} 2^{-m} Z_m\right)^d \\ &= (-\varphi_{\max})^d \left(\frac{2^{\mathsf{n_q}-1}}{2^{\mathsf{n_q}}-1}\right)^d \sum_{m_1,m_2,\ldots,m_d=0}^{\mathsf{n_q}-1} 2^{-(m_1+m_2+\cdots+m_d)} Z_{m_1} Z_{m_2} \ldots Z_{m_d}\,.\end{aligned} \tag{55}$$

Since $Z_{m_j}^2 = \mathbb{1}$, the sum over m_j includes multiple instances of the same Pauli string. Constructing a BE using LCU requires grouping these repeated terms and determining the total coefficient for each unique Pauli string. However, because the coefficient of each Pauli string in the sum has the same sign, no cancellations in the final values of the coefficients occur. This fact implies that, even if there are repeated Pauli strings in the sum, the scale factor can be found by simply adding the coefficients of each Pauli string in the sum. Using this fact, the scale factor is given by

$$\begin{aligned}\beta_{\hat{\varphi}^d} &= \varphi_{\max}^d \left(\frac{2^{\mathsf{n_q}-1}}{2^{\mathsf{n_q}}-1}\right)^d \sum_{m_1,m_2,\ldots,m_d=0}^{\mathsf{n_q}-1} 2^{-(m_1+m_2+\cdots+m_d)} \\ &= \varphi_{\max}^d \left(\frac{2^{\mathsf{n_q}-1}}{2^{\mathsf{n_q}}-1}\right)^d \left(\sum_{m=0}^{\mathsf{n_q}-1} 2^{-m}\right)^d \\ &= \varphi_{\max}^d \left(\frac{2^{\mathsf{n_q}-1}}{2^{\mathsf{n_q}}-1}\right)^d \left(\frac{2^{\mathsf{n_q}}-1}{2^{\mathsf{n_q}-1}}\right)^d \\ &= \varphi_{\max}^d\,,\end{aligned} \tag{56}$$

where, going from the second to the third line, we used the identity $\sum_{m=0}^{\mathsf{n_q}-1} 2^{-m} = (2^{\mathsf{n_q}}-1)/2^{\mathsf{n_q}-1}$. From this result we can directly see the scale factor for $f_1(\hat{\pi}^{(D)}) = \frac{1}{2}\left(\hat{\pi}^{(D)}\right)^2$ is

$$\beta_{f_1(\hat{\pi}^{(D)})} = \frac{1}{2}\pi_{\max}^2\,, \tag{57}$$

which is the smallest possible scale factor to BE this operator.

Next, we turn to $f_0^{(1)}(\hat{\varphi}) = \frac{m^2}{2}\hat{\varphi}^2 + \frac{\lambda}{4!}\hat{\varphi}^4$, whose Pauli decomposition is

$$\begin{aligned}f_0^{(1)}(\hat{\varphi}) &= \frac{m^2}{2}\hat{\varphi}^2 + \frac{\lambda}{4!}\hat{\varphi}^4 \\ &= (-\varphi_{\max})^2 \left(\frac{2^{\mathsf{n_q}-1}}{2^{\mathsf{n_q}}-1}\right)^2 \frac{m^2}{2} \sum_{m_1,m_2=0}^{\mathsf{n_q}-1} 2^{-(m_1+m_2)} Z_{m_1} Z_{m_2} + \\ &\quad + (-\varphi_{\max})^4 \left(\frac{2^{\mathsf{n_q}-1}}{2^{\mathsf{n_q}}-1}\right)^4 \frac{\lambda}{4!} \sum_{l_1,l_2,l_3,l_4=0}^{\mathsf{n_q}-1} 2^{-(l_1+l_2+l_3+l_4)} Z_{l_1} Z_{l_2} Z_{l_3} Z_{l_4}\,.\end{aligned} \tag{58}$$

Assuming $\lambda > 0$, we see that, similar to the previous case, the coefficient of each Pauli string in the decomposition of $f(\hat{\varphi})$ has the same sign, and the scale factor is simply the sum of the coefficients

$$\begin{aligned}\beta_{f_0^{(1)}(\hat{\varphi})} &= \frac{m^2}{2}\varphi_{\max}^2 \left(\frac{2^{\mathsf{n_q}-1}}{2^{\mathsf{n_q}}-1}\right)^2 \left(\sum_{m=0}^{\mathsf{n_q}-1} 2^{-m}\right)^2 + \frac{\lambda}{4!}\varphi_{\max}^4 \left(\frac{2^{\mathsf{n_q}-1}}{2^{\mathsf{n_q}}-1}\right)^4 \left(\sum_{m=0}^{\mathsf{n_q}-1} 2^{-m}\right)^4 \\ &= \frac{m^2}{2}\varphi_{\max}^2 + \frac{\lambda}{4!}\varphi_{\max}^4\,,\end{aligned} \tag{59}$$

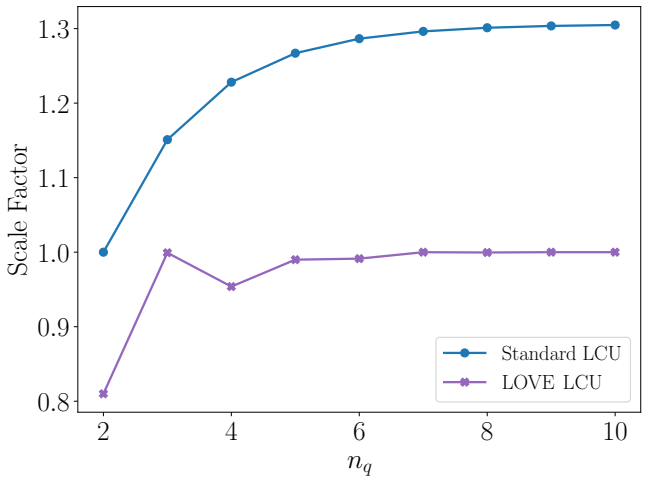

Figure 17: Scale factor of the BE of $\cos(\hat{\varphi})$ as a function of $\mathsf{n_q}$. The blue circles and purple crosses correspond to using standard LCU and LOVE-LCU to construct the BE, respectively. The value of $\varphi_{\max}$ was set to $\varphi_{\max} = \pi$ for all values of $\mathsf{n_q}$.

which saturates the lower bound given by the operator norm.

The situation is more complicated for $f_0^{(2)}(\hat{\varphi}) = g\cos(\hat{\varphi})$ as its Pauli decomposition consists of terms with opposite signs, requiring careful accounting for cancellations. For this reason, we study the scale factor numerically. Figure 17 presents the scale factor of the BE of $\cos(\hat{\varphi})$ constructed using both standard LCU and LOVE-LCU. For this comparison, we set $\varphi_{\max} = \pi$. With LOVE-LCU, the scale factor approaches the optimal value of 1, whereas with standard LCU, it remains generally larger and asymptotes to ~ 1.3.

Lastly, we consider $f_2(\hat{\varphi}_1 - \hat{\varphi}_2) = \frac{1}{2}(\hat{\varphi}_1 - \hat{\varphi}_2)^2$. To study the scale factor, we first rewrite it as $f_2(\hat{\varphi}_1 - \hat{\varphi}_2) = \frac{1}{2}(\hat{\varphi}_1^2 + \hat{\varphi}_2^2 - 2\hat{\varphi}_1\hat{\varphi}_2)$. While this operator is a sum of terms with different signs, this does not affect the determination of the scale factor since the terms with differing signs have no Pauli strings in common. This can be verified by directly examining the decomposition of $f_2(\hat{\varphi}_1 - \hat{\varphi}_2)$. If we assign $\hat{\varphi}_1$ to act on qubits $0, 1, \ldots, \mathsf{n_q} - 1$ and $\hat{\varphi}_2$ to act on qubits $\mathsf{n_q}, \mathsf{n_q} + 1, \ldots, 2\mathsf{n_q} - 1$, the decomposition is

$$\begin{aligned} f_2(\hat{\varphi}_1 - \hat{\varphi}_2) &= \frac{1}{2}(\hat{\varphi}_1^2 + \hat{\varphi}_2^2 - 2\hat{\varphi}_1\hat{\varphi}_2) \\ &= \frac{1}{2}\varphi_{\max}^2 \left(\frac{2^{\mathsf{n_q}-1}}{2^{\mathsf{n_q}}-1}\right)^2 \\ &\quad \times \left(\sum_{m_1,m_2=0}^{\mathsf{n_q}-1} 2^{-(m_1+m_2)} Z_{m_1} Z_{m_2} + \sum_{l_1,l_2=0}^{\mathsf{n_q}-1} 2^{-(l_1+l_2)} Z_{\mathsf{n_q}+l_1} Z_{\mathsf{n_q}+l_2} - 2\sum_{m,l=0}^{\mathsf{n_q}-1} 2^{-(m+l)} Z_m Z_{\mathsf{n_q}+l}\right). \end{aligned} \tag{60}$$

From this expression, we see that all Pauli strings with opposite signs are distinct. Thus, the scale factor can be determined by summing the coefficients. Doing so gives

$$\begin{aligned} \beta_{f_2(\hat{\varphi}_1-\hat{\varphi}_2)} &= \frac{1}{2}\varphi_{\max}^2 \left(\frac{2^{\mathsf{n_q}-1}}{2^{\mathsf{n_q}}-1}\right)^2 \left(\sum_{m_1,m_2=0}^{\mathsf{n_q}-1} 2^{-(m_1+m_2)} + \sum_{l_1,l_2=0}^{\mathsf{n_q}-1} 2^{-(l_1+l_2)} + 2\sum_{m,l=0}^{\mathsf{n_q}-1} 2^{-(m+l)}\right) \\ &= \frac{1}{2}\varphi_{\max}^2 \left(\frac{2^{\mathsf{n_q}-1}}{2^{\mathsf{n_q}}-1}\right)^2 4\left(\sum_{m=0}^{\mathsf{n_q}-1} 2^{-m}\right)^2 \\ &= 2\varphi_{\max}^2. \end{aligned} \tag{61}$$

Since the operator $\hat{\varphi}$ is sampled symmetrically from $-\varphi_{\max}$ to $\varphi_{\max}$, this scale factor attains its smallest possible value.