Scalable simulation of nonequilibrium quantum dynamics via classically optimized unitary circuits

Luke Causer,[1,2] Felix Jung,[3] Asimpunya Mitra,[4,5] Frank Pollmann,[3,6] and Adam Gammon-Smith[1,2]

[1]*School of Physics and Astronomy, University of Nottingham, Nottingham, NG7 2RD, United Kingdom*
[2]*Centre for the Mathematics and Theoretical Physics of Quantum Non-Equilibrium Systems, University of Nottingham, Nottingham, NG7 2RD, United Kingdom*
[3]*Department of Physics, TFK, Technische Universität München, James-Franck-Straße 1, D-85748 Garching, Germany*
[4]*Department of Physics, Indian Institute of Technology Kharagpur, Kharagpur 721302, India*
[5]*Department of Physics, University of Toronto, Toronto, Ontario M5S 1A7, Canada*
[6]*Munich Center for Quantum Science and Technology (MCQST), Schellingstr. 4, 80799 München, Germany*

(Received 15 February 2024; revised 19 May 2024; accepted 18 June 2024; published 15 July 2024)

The advent of near-term digital quantum computers could offer us an exciting opportunity to investigate quantum many-body phenomena beyond that of classical computing. To make the best use of the hardware available, it is paramount that we have methods that accurately simulate Hamiltonian dynamics for limited circuit depths. In this paper, we propose a method to classically optimize unitary brickwall circuits to approximate quantum time evolution operators. Our method is scalable in system size through the use of tensor networks. We demonstrate that, for various three-body Hamiltonians, our approach produces quantum circuits that can outperform trotterization in both their accuracy and the quantum circuit depth needed to implement the dynamics, with the exact details being dependent on the Hamiltonian. We also explain how to choose an optimal time step that minimizes the combined errors of the quantum device and the brickwall circuit approximation.

DOI: 10.1103/PhysRevResearch.6.033062

I. INTRODUCTION

Quantum computers and quantum simulators provide the potential for a quantum advantage in complex computational problems [1], such as quantum chemistry [2–9] and quantum machine learning [10–12]. It is expected that one of the earliest applications will be in the field of quantum many-body physics [13–16]. Broadly speaking, the current methods to study quantum many-body systems fall into one of the three scenarios: (i) exact numerics for small enough system sizes where the Hamiltonian can be diagonalized [17,18], (ii) approximate numerics through low-rank approximations such as tensor networks for quantum states which have sufficiently small (area-law) entanglement [19–23], or (iii) exact results for models which are simple enough to allow for a comprehensive analytical treatment [24–32]. The limiting factor for (i) and (ii) is the curse of dimensionality; the Hilbert space grows exponentially in the number of degrees of freedom (e.g., the number of qubits). Digital quantum computers, however, would allow us to faithfully simulate the dynamics of the system for large times through tunable quantum gates [14,33]. Such simulations could pave the way for a more thorough understanding of many important quantum phenomena such as thermalization and ergodicity [34–36].

Despite the prospects of quantum advantage [37,38], modern-day quantum computers are still subject to strong noise, limiting the depths of the quantum circuits that can be simulated before the onset of decoherence. This provides us with a strong incentive to engineer optimal quantum circuits designed to achieve a specific task with minimal circuit depth. Various approaches to achieve this have been proposed for the simulation of quantum many-body dynamics. The most standard approach for simulating dynamics is to "trotterize" the Hamiltonian [39,40], which can be implemented on a quantum computer [14,41–43]. However, to achieve large times with a desired accuracy, this can require a circuit depth that scales at least linearly in time [44,45]. One proposed method to overcome this complexity is to, at each time step, compress the quantum state as a unitary circuit of fixed depth [46–48]. However, while this approach reduces the effects of noise from the quantum computer, it can also reduce the accuracy of the simulation. Furthermore, it requires a costly optimization task at each time step.

An alternative approach is to leverage classical computing to find optimal quantum circuits that can simulate quantum dynamics with a higher accuracy and smaller circuit depth when compared to trotterization. On one hand, this could extend the times for which accurate simulations can be achieved. On the other hand, unlike the approach described previously, would not require a costly optimization task at each time step. This approach was proven effective for the transverse field Ising model in Refs. [49,50], which optimized two-body brickwall circuits using gradient descent, and in Ref. [51], which made use of tensor networks for a restricted circuit ansatz. A similar approach was taken in Ref. [52] for the Heisenberg model on a chain, and a ladder with next-nearest-neighbor interactions, which optimized over arbitrary two-qubit gates using automatic differentiation.

In this paper, we propose a tensor network (TN) approach to classically optimize brickwall (BW) unitary circuits. Our method approximates the time propagator for Hamiltonian dynamics at small times as a matrix product operator (MPO) [53–55]. This MPO is then used along with the *polar decomposition* [46,56] to variationally optimize a unitary BW circuit on a classical computer by maximising their "overlap." The advantage of using an MPO to approximate the time propagator is that it can reach a desired accuracy with resources which scale only linearly in system size [57–59]. The classically optimized circuit can thus be used as part of a larger calculation on a digital quantum computer to reach a total time larger than what is accessible on a classical computer.

Our paper is structured as follows. In Sec. II, we give a brief introduction to simulating quantum dynamics with trotterization. The optimization procedure is then explained in Sec. III. We demonstrate our approach in Sec. IV for various three-body Hamiltonians. Using numerous metrics, we compare our results to trotterization to demonstrate the practical advantage of classically optimized BW circuits. We show that the optimized circuits outperform trotterization in two ways: (i) they approximate the true time evolution operator with less error than trotterization, and (ii) they can be implemented on digital quantum computers with less two-body entangling layers, thus reducing the error from decoherence. We also benchmark convergence for our method. In Sec. V, we explore the trade-off between optimization errors and decoherence errors from the device. We use these considerations to explain how to estimate the optimal time step which minimizes the compounded error. We give our conclusions in Sec. VI.

II. TROTTERIZATION

We consider locally interacting one-dimensional Hamiltonians, $\hat{H}$. Although our optimization method will be general, we will focus on local three-body interactions,

$$\hat{H} = \sum_{j=1}^{N} \hat{O}^{(1)}_{j} + \sum_{j=1}^{N-1} \hat{O}^{(2)}_{j,\,j+1} + \sum_{j=1}^{N-2} \hat{O}^{(3)}_{j,\,j+1,\,j+2}, \tag{1}$$

where $\hat{O}^{(r)}_{j_1,\ldots,j_r}$ describes some local operator which acts on lattice sites $j_1, \ldots, j_r$. We assume each lattice site has the same finite physical dimension d. While, in principle, everything in this paper is applicable to arbitrary d, we will focus on the case of $d = 2$ (e.g., qubits). One area of interest is quench dynamics: starting from some out-of-equilibrium initial state, $|\psi_0\rangle$, we study the dynamics $|\psi_t\rangle = U(t)\,|\psi_0\rangle$, where $U(t) = e^{-it\hat{H}}$ is the time evolution operator. Calculating $|\psi_t\rangle$ for arbitrary t can often be difficult due to the complexity of $U(t)$, which can be exponential in system size.

This difficulty could be overcome by simulating the Hamiltonian dynamics on a quantum computer. We first use the product formula to slice the exponential into n parts, $U(t) = U(\Delta t)^n$ for $\Delta t = t/n$ [14]. The unitary for the smaller time step Δt can then be approximated by some other unitary ansatz that can be efficiently implemented on the quantum device, $V(\Delta t)$, such as the Trotter-Suzuki decomposition (see below for details) [39,40]. The error of this unitary is

$$||U(\Delta t) - V(\Delta t)|| \leqslant \mathcal{O}(\Delta t^{k+1}), \tag{2}$$

where $k \geqslant 1$ is the *order* of the approximation and $||\cdot||$ is the operator norm. It can then be shown that the total simulation error goes as [60,61]

$$||U(t) - V(\Delta t)^n|| \leqslant n||U(\Delta t) - V(\Delta t)|| = \mathcal{O}(t\Delta t^k). \tag{3}$$

While only approximate, this approach offers the advantage that the error is well controlled through the choice of time step Δt.

A. Trotter-Suzuki decomposition

A standard approach for approximating $U(\Delta t)$ is to decompose it into a product of *local* unitary operators, $U^{(r)}_{j_1,\ldots,j_r}(\Delta t) = e^{-i\Delta t\hat{O}^{(r)}_{j_1,\ldots,j_r}}$. This is achieved through the Trotter-Suzuki decomposition [39,40]: for two operators $\hat{A}$ and $\hat{B}$, the first-order Trotter-Suzuki decomposition allows us to write

$$e^{-i\Delta t(\hat{A}+\hat{B})} = e^{-i\Delta t\hat{B}}e^{-i\Delta t\hat{A}} + \mathcal{O}(\Delta t^2). \tag{4}$$

In essence, this allows us to achieve the evolution by first evolving by $\hat{A}$, then followed by $\hat{B}$. This has an error of order $\mathcal{O}(t^2)$. This is easily improved on through the second-order Trotter-Suzuki decomposition,

$$e^{-i\Delta t(\hat{A}+\hat{B})} = e^{-i\Delta t\hat{A}/2}e^{-i\Delta t\hat{B}}e^{-i\Delta t\hat{A}/2} + \mathcal{O}(\Delta t^3). \tag{5}$$

Like the first-order decomposition, this evolves the state in a Floquet manner. It first does a partial evolution over $\hat{A}$, followed by a full evolution over $\hat{B}$, and then a partial evolution over $\hat{A}$. While this can be generalized to higher orders with more complex sequences of evolutions, here we will only consider up to the second order.

The Trotter-Suzuki decomposition allows for a way to approximately decompose $U(\Delta t)$ into a sequential application of local unitary operators. For the Hamiltonian Eq. (1), one must first split the Hamiltonian into various components which are each a sum over commuting operators. An easy way to achieve this is to first write the Hamiltonian as $\hat{H} = \hat{H}^{(1)} + \hat{H}^{(2)} + \hat{H}^{(3)}$, with each the sum over all operators which act locally on (r) qubits. The first term, $\hat{H}^{(1)}$, is already a sum over commuting operators. The second and third terms need to be split further,

$$\hat{H}^{(2)} = \sum_{j} \hat{O}^{(2)}_{2j-1,2j} + \sum_{j} \hat{O}^{(2)}_{2j,2j+1} := \hat{H}^{(2)}_{1} + \hat{H}^{(2)}_{2}, \tag{6}$$

$$\begin{aligned}\hat{H}^{(3)} = &\sum_{j} \hat{O}^{(3)}_{3j-2,3j-1,3j} + \sum_{j} \hat{O}^{(3)}_{3j-1,3j,3j+1} \\ &+ \sum_{j} \hat{O}^{(3)}_{3j,3j+1,3j+2} := \hat{H}^{(3)}_{1} + \hat{H}^{(3)}_{2} + \hat{H}^{(3)}_{3}.\end{aligned} \tag{7}$$

The Hamiltonian is now composed of six individual sums. The Trotter-Suzuki decompositions can be applied five times in total to obtain a product over local unitary gates. Since each sum contains commuting operators, this can be exactly split into the product of unitaries, e.g., $e^{-\Delta t\hat{H}^{(1)}} = \prod_{j} U^{(1)}_{j}(\Delta t)$.

Writing the full product of unitaries is rather cumbersome. As such, it is convenient to write the full evolution

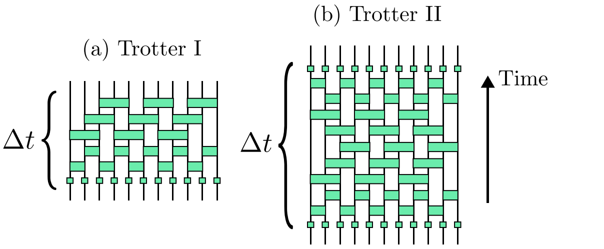


FIG. 1. Simulating time evolution with trotterization. The time evolution operator $U(\Delta t) = e^{-i\Delta t\hat{H}}$ can be approximated using the Trotter-Suzuki decomposition, as illustrated for local three-body interactions, see Eq. (1), and $N = 11$ qubits. The lines indicate the direction of time for each qubit, and the green rectangles are local unitary operators. (a) The first-order Trotter-Suzuki decomposition first evolves the system by all the single-body unitaries, followed by commuting two-body unitaries and finally three-body unitaries. Unitary operators which act on different qubits can be applied in parallel. Each unitary is the exponential of operators in Eq. (1) with time Δt. (b) The second-order Trotter-Suzuki decomposition is done symmetrically in time: all layers of unitaries are the exponential of operators with time $\Delta t/2$, apart from the middle layer, which is done with time Δt.

diagrammatically, as shown in Figs. 1(a) and 1(b) for the first- and second-order decompositions respectively. The lines show the evolution of a particular qubit (which are arranged in spatial order), the green rectangles show the action of a unitary gate on the given qubits. For the first-order decomposition, we first evolve by all single-body unitaries in parallel. The next two layers demonstrate the application of two-body unitaries. Notice that unitary gates that commute (act on different lattice sites) are applied in parallel. Finally, we evolve over the three-body unitaries with three layers of gates. For the second-order decomposition, we evolve in a symmetric way: each gate is evolved for a time $\Delta t/2$, except for the middle layer of gates, which are evolved for the full time Δt.

III. CLASSICAL OPTIMIZATION OF UNITARY CIRCUITS

Trotterization provides a systematic way to approximate the time-evolution operator for a many-body quantum system by decomposing it into the product of local unitary operators. One drawback to this approach is that it still relies on many-body unitaries which might not be simple to implement on current quantum devices. For example, the Trotter-Suzuki decomposition for the three-body Hamiltonian Eq. (1) would require us to implement three-body unitaries. However, the native gates on popular platforms commonly only include two-body entangling gates, such as controlled-NOT (CNOT) gates. While these provide a foundation for a universal set of quantum gates (i.e., any many-body unitary can be implemented to arbitrary accuracy), there is no promise that the three-body unitaries can be well implemented using a shallow quantum circuit.

In this section, we explain how one can optimize a BW unitary circuit with two-site gates and depth M to approximate $U(\Delta t)$ (for small Δt) in a way that is scalable with system size N. They are arranged compactly such that each layer has a unitary gate acting on each lattice site (except for lattice sites at the edges), see Figs. 2(a) and 2(b) for visualizations of circuit depths $M = 3$ and 4. Note that, while there is no guarantee that a BW circuit with a fixed depth M can well approximate $U(\Delta t)$, the operator entanglement of $U(\Delta t)$ for any local Hamiltonian is bounded by Lieb-Robinson bounds [62] and thus it is reasonable to expect that for sufficiently small times, $U(\Delta t)$ can be well-approximated by a BW circuit with a small depth [63]. Additionally, unlike the Trotter circuits, there is no direct way to determine an optimal BW circuit for general $\hat{H}$. However, they have the distinct advantage that they can be easily implemented on current quantum computers with a number of layers of two-body entangling gates that is constant in M, see Appendix B.

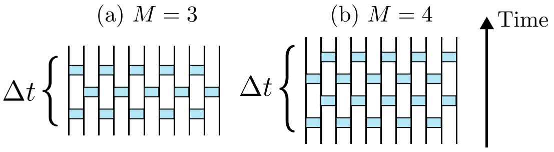


FIG. 2. Unitary brickwall circuits. A unitary BW circuit can be used as a variational ansatz for the time-evolution operator $U(\Delta t)$, as shown for $N = 11$ qubits and a circuit depth of (a) $M = 3$ and (b) $M = 4$. The lines indicate the direction of time for the circuit, and each rectangle is a two-qubit unitary matrix.

We will optimize the BW unitary circuits to maximize the "overlap" with the true time evolution operator $U(\Delta t)$, which we approximate using an MPO. Our method is reminiscent of standard variational matrix-product state methods and is not limited by system size (for example, Refs. [64–66]): we find an optimal update for a single gate in the BW circuit, keeping all other gates fixed. This is done for each gate in the circuit, which we sweep through multiple times until we find convergence. While there is freedom in deciding the order to update the gates, the strategy we found to work best is given in Fig. 3. We sweep back and forth through the lattice sites $2 \leqslant j \leqslant N-1$. For each lattice site, we update every gate which acts on the lattice site. Every sweep from left-to-right (or right-to-left) will update each unitary gate twice, apart from those on the edges, which are only updated once.

A. MPO approximation of $U(\Delta t)$

In general, it is possible to write $U(\Delta t)$ as a (d^N, d^N)-matrix. This is illustrated diagrammatically as a TN in Fig. 4(a), where each leg corresponds to a physical dimension d. However, this quickly becomes intractable to calculate due to its exponential cost with N. One strategy to overcome this cost is to instead approximate $U(\Delta t)$ with an MPO. This is a decomposition into a TN with a rank-4 tensor for each lattice

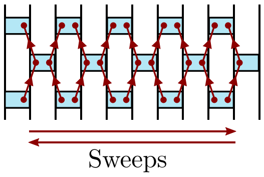


FIG. 3. Sweeping order. We sweep back-and-forth through the lattice sites $j = 2$ to $N - 1$. At each lattice site, we sequentially optimize each unitary gate which acts on site j in the order shown in the diagram.

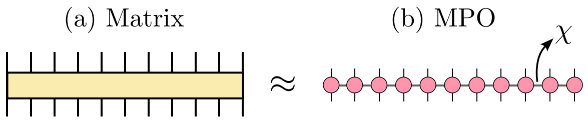


FIG. 4. Matrix product operators. (a) The time propagator $U(\Delta t) = e^{-i\Delta t\hat{H}}$ can be represented as a full-rank matrix. (b) For small times, $U(\Delta t)$ can be well approximated with an MPO, where the pink circles represent rank-4 tensors. The virtual bond dimension, $\chi \ll d^N$, can be used to control the accuracy of the approximation.

site,

$$U_{\text{MPO}} = \sum_{n_1\cdots n_N}\sum_{m_1\cdots m_N} T^{m_1\cdots m_N}_{n_1\cdots n_N} \left|m_1\cdots m_N\right\rangle \left\langle n_1\cdots n_N\right|, \tag{8}$$

with the matrix coefficients determined by a matrix product,

$$T^{m_1\cdots m_N}_{n_1\cdots n_N} = \text{Tr}\big[M^{(1)}_{n_1 m_1}\cdots M^{(N)}_{n_N m_N}\big]. \tag{9}$$

Each $M^{(j)}_{n_j m_j}$ is a (χ, χ) matrix, where χ is the virtual *bond dimension* which controls the accuracy of the approximation. When the labels n_j and m_j are unspecified, $M^{(j)}_{n_j m_j}$ is a rank-4 tensor with two physical dimensions and two bond dimensions which connect the tensor to the tensors of neighboring lattice sites, see Fig. 4(b). Each circle represents a tensor, the open legs the physical dimensions, and the connected legs the bond dimensions.

The MPO is described by $\mathcal{O}(N\chi^2 d^2)$ parameters, which is compared to the $\mathcal{O}(d^{2N})$ parameters of the full-rank matrix. The caveat of this is that the MPO requires a bond dimension $\chi \sim \mathcal{O}(d^N)$ to attain full accuracy for arbitrary $U(\Delta t)$. Nevertheless, for small times, $U(\Delta t)$ can be approximated to some accuracy with a bond dimension which is independent of system size, $\chi \sim \mathcal{O}(d^{\gamma\Delta t})$ [57–59]. This fact will be used throughout the paper.

Our method will assume that we have access to a high accuracy MPO approximation of $U(\Delta t)$. For system sizes $N \leqslant 12$, we construct the MPO by calculating $U(\Delta t)$ as a full-rank matrix to numerical precision, and then decompose it as an MPO using singular value decompositions (SVDs), see Ref. [66] for details. For $N \geqslant 12$, we construct the approximation by evolving an MPO using trotterization [67], although we note there are other methods which might be more effective, e.g., Refs. [68–70].

We start with the identity matrix $U_{\text{MPO}}(\tau = 0) = \hat{\mathbb{1}}$. To ensure the MPO approximation is accurate [71], we evolve it using one-hundred second-order Trotter steps with time step $\Delta\tau = \Delta t/100$, which we denote by $U_{\text{Trotter}}(\Delta\tau)$. At each evolution step, the time evolution operator is approximated by

$$U_{\text{MPO}}(\tau + \Delta\tau) \approx U_{\text{Trotter}}(\Delta\tau)U_{\text{MPO}}(\tau). \tag{10}$$

The application of each gate will give a new MPO with a bond dimension $\chi_{\text{exact}} = r\chi$, where $r > 1$ is some integer which depends on the circuit. Thus the bond dimension of the resulting MPO will grow exponentially in the number of applications of the trotterized circuit. To prevent this, at each time step, we approximate the MPO with another MPO with a smaller bond dimension χ' such that $\chi \leqslant \chi' \leqslant \chi_{\text{exact}}$. In practice, this is achieved by compressing the MPO using SVDs after each application of unitary gate [66]. We allow the bond dimension to grow dynamically, with its value determined through the SVD. We use an SVD truncation error of $\epsilon = 10^{-16}$ to ensure that the truncation error is minimal [72]. Notice that this setup allows for each individual bond dimension between lattice sites to take its own value. Assuming a maximal bond dimension χ, the cost of constructing $U(\Delta t)$ as an MPO is $\mathcal{O}(N\Delta t\Delta\tau^{-1}\chi^3 d^8)$.

B. The cost function

Suppose we want to update the unitary gate at depth $m \leqslant M$ which acts on a lattice site l, which for simplicity we call G. The objective is to choose the gate G while keeping all other gates fixed such that the BW circuit best approximates the MPO. To this end, we use the squared Frobenius norm

$$\begin{aligned}\mathcal{F}^{(2)}(U_{\text{BW}}(\Delta t)) &= ||U_{\text{MPO}}(\Delta t) - U_{\text{BW}}(\Delta t)||^2_F\\ &= \text{Tr}[U^\dagger_{\text{MPO}}(\Delta t)U_{\text{MPO}}(\Delta t)]\\ &\quad + \text{Tr}[U^\dagger_{\text{BW}}(\Delta t)U_{\text{BW}}(\Delta t)]\\ &\quad - 2\text{Re}(\text{Tr}[U^\dagger_{\text{MPO}}(\Delta t)U_{\text{BW}}(\Delta t)])\end{aligned} \tag{11}$$

as a cost function. To find the optimal choice for the gate, G_{opt}, one needs to minimize Eq. (11) with respect to G,

$$G_{\text{opt}} = \text{argmin}_{G\in\mathcal{U}(d^2)}\{\mathcal{F}^{(2)}(U_{\text{BW}}(\Delta t))\}, \tag{12}$$

where $\mathcal{U}(d^2)$ denotes the group of unitary matrices with dimensions (d^2, d^2). The first term in Eq. (11) depends only on the MPO, which has no variational parameters, and is thus constant. The second term is the trace of the product of the BW circuit with its Hermitian adjoint. Since all gates are unitary, the product of any gate with its Hermitian adjoint is the identity matrix, as shown in Fig. 5(a). It follows that this term is constant, $\text{Tr}[U^\dagger_{\text{BW}}(\Delta t)U_{\text{BW}}(\Delta t)] = 2^N$. Note that this is true for *all* choices of unitary gates, G. The final term is the trace of the BW circuit with the Hermitian adjoint of the MPO. This is shown in Fig. 5(b). The gate which is being optimized is colored green. Since the previous two terms are both constants, it follows that minimizing the squared Frobenius norm is equivalent to

$$G_{\text{opt}} = \text{argmax}_{G\in\mathcal{U}(d^2)}\{\text{Re}(\text{Tr}[U^\dagger_{\text{MPO}}(\Delta t)U_{\text{BW}}(\Delta t)])\}. \tag{13}$$

C. Updating the gates

We now explain how to find the unitary gate, G_{opt}, which maximizes $\text{Re}(\text{Tr}[U^\dagger_{\text{MPO}}(\Delta t)U_{\text{BW}}(\Delta t)])$. We first contract all tensors and gates in $\text{Tr}[U^\dagger_{\text{MPO}}(\Delta t)U_{\text{BW}}(\Delta t)]$ except for the gate G. It is convenient to group together all tensors and gates which act on lattice sites $j < l$ (but not those which act on $j = l$) to give the tensor L_{l-1}. Similarly, we group together all tensors and gates which act on lattice sites $j > l$ (but not those which act on $j = l$) to give the tensor R_{l+1}. For more details, see Sec. III D. The gives the TN shown in Fig. 5(c), which can be fully contracted to give the tensor $\tilde{G}^\dagger$. The exact contraction will differ depending on l, m and M. The cost of contracting the TN in Fig. 5(c) is at most $\mathcal{O}(\chi M d^{4+2\lceil M/2\rceil} + \chi^2 d^{2+2\lceil M/2\rceil})$. Note that, while the cost of contracting the network is exponential in the circuit depth M,

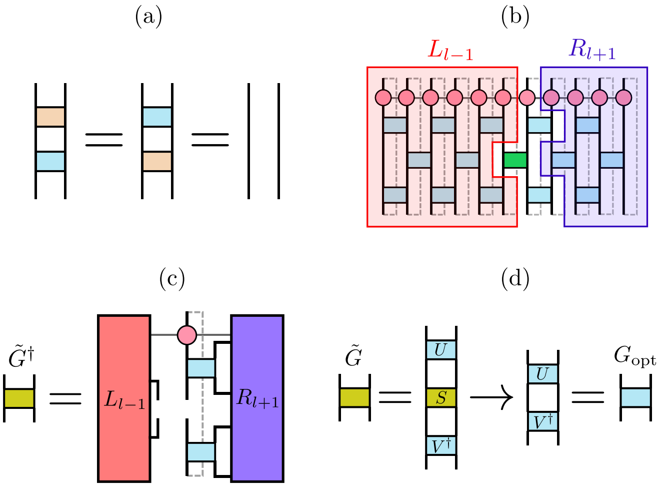


FIG. 5. Updating unitary gates. The procedure for updating a gate G illustrated for a circuit with depth $M = 3$. The target gate is at site $l = 7$ and depth $m = 2$. The optimal update for the gate is determined through Eq. (12). (a) A unitary gate (blue rectangle), multiplied with its Hermitian adjoint (orange rectangle), gives the identity matrix. (b) The TN representation of $\mathrm{Tr}[U^\dagger_{\rm MPO}(\Delta t)U_{\rm BW}(\Delta t)]$. The green rectangle is the target gate. The tensors and gates which act on lattice sites $j < l$ (but not on $j = l$) can be contracted to give tensor L_{l-1}. Similarly, the tensors and gates which act on lattice sites $j > l$ (but not on $j = l$) can be grouped to give the tensor R_{l+1}. (c) The tensor $\tilde{G}^\dagger$ can be found by partially contracting over the TN in (c), excluding the gate G. (d) The choice of *unitary* gate which maximizes $\mathrm{Re}(\mathrm{Tr}[U^\dagger_{\rm MPO}(\Delta t)U_{\rm BW}(\Delta t)])$ can be found through the polar decomposition. In practice, we perform an SVD on the tensor $\tilde{G} = USV^\dagger$, and discard the matrix of singular values S. Multiplying U and $V^\dagger$ gives the optimal unitary gate $G_{\rm opt} = UV^\dagger$.

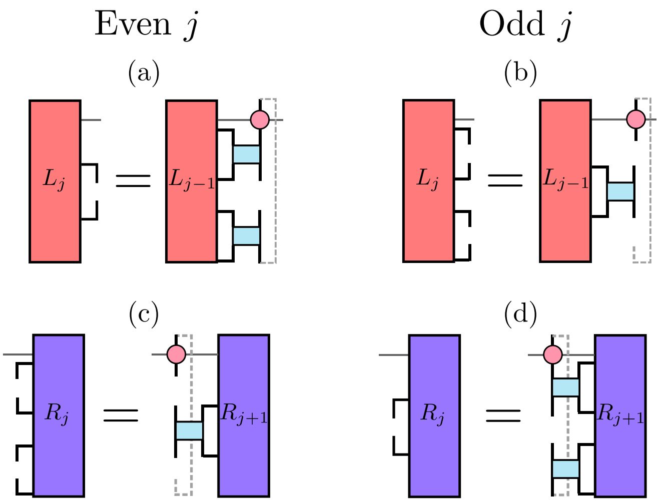


FIG. 6. Contracting the environment. Building the environment blocks, illustrated for a circuit depth $M = 3$ circuit. [(a) and (b)] Building the left block L_j from the previous L_{j-1} for even and odd j, respectively. [(c) and (d)] Building the right block R_j from the previous R_{j+1} for even and odd j, respectively.

in practice one wants to find a good approximation with as minimal depth as possible.

The unitary gate $G_{\rm opt}$ which maximizes $\mathrm{Re}(\mathrm{Tr}[U^\dagger_{\rm MPO}(\Delta t)U_{\rm BW}(\Delta t)])$ can then be found through the *polar decomposition* of $\tilde{G}$ [46,56]. In practice, this is found by performing an SVD on $\tilde{G} = USV^\dagger$, where U and $V^\dagger$ are unitary matrices, and S is a diagonal matrix of singular values. The polar decomposition can then be retrieved by replacing S with identity, i.e., $G_{\rm opt} = UV^\dagger$. This is shown in Fig. 5(d), and is achieved with a computational cost $\mathcal{O}(d^6)$.

D. Contracting the environment

As is often done in TN methods, it is convenient and efficient to recycle partially contracted networks [66]. For this method, it will be useful to save each of the blocks L_j and R_j to memory for two reasons. The first is the blocks L_{k-1} and R_{k+1} will be the same for each of the M gates which act on site k. They can be calculated and then recycled to update each of the M gates. The second reason is that the blocks L_j and R_j can be used iteratively to calculate the blocks at all sites j. That is, L_j can be calculated using L_{j-1} and R_j can be calculated using R_{j+1}. This procedure is illustrated in Fig. 6 for a circuit depth $M = 3$, although the exact calculations will differ for other circuit depths. The left column shows how to construct the blocks for even j, and the right column for odd j. The block L_{j-1} is contracted with all the gates which act on both sites $j-1$ and j, along with the adjoint of the MPO tensor at site j to form the block L_j. Similarly, the block R_{j+1} is contracted with the gates which act on both sites j and $j+1$, along with the adjoint of the MPO tensor at site j to give R_j. The cost of expanding such blocks is at most $\mathcal{O}(\chi^2 d^{2+2\lceil M/2\rceil} + \chi M d^{4+2\lceil M/2\rceil})$, and storing to memory is at most $\mathcal{O}(\chi d^{2\lceil M/2\rceil})$ per block. See Table I for a complete description of all the computational costs in our method.

IV. RESULTS

We now benchmark our approach for numerous models of interest for system sizes $N = 8$ to 512. To assess the quality of the dynamics, we first measure the error of the BW circuit with respect to the high-accuracy MPO approximation, and investigate violations of conserved quantities. We then calculate the number of CNOT layers needed to implement the

TABLE I. The computational complexity for each of the steps in the optimization method. N is the number of lattice sites; d is the local dimension of the system; Δt is the time step of unitary $U(\Delta T) = e^{-i\Delta t\hat{H}}$; $\Delta\tau$ is the time step in trotterization; χ is the bond dimension of the MPO $U(\Delta t)$; M is the depth of the BW circuit.

Estimating $U(\Delta t)$ as an MPO
$\mathcal{O}(N\Delta t\,\Delta\tau^{-1}\chi^3 d^8)$
Updating the environment
$\mathcal{O}(\chi^2 d^{2+2\lceil M/2\rceil}) + \mathcal{O}(\chi M d^{4+2\lceil M/2\rceil})$
Updating the gates (per gate)
$\mathcal{O}(\chi^2 d^{2+2\lceil M/2\rceil}) + \mathcal{O}(\chi M d^{4+2\lceil M/2\rceil})$
Cost per sweep
$\mathcal{O}(N\chi^2 d^{2+2\lceil M/2\rceil}) + \mathcal{O}(N\chi M d^{4+2\lceil M/2\rceil})$

circuits on a quantum computer, which we compare to trotterization. Finally, we investigate the convergence properties of our method to demonstrate its practicality.

A. Error metrics

We first consider the performance of the unitary circuits with respect to the time step, Δt. One instructive measure is the error density of the circuit with respect to the MPO [73],

$$\delta(\Delta t) = \sqrt{2 - \mathrm{Re}(\mathrm{Tr}[U_{\mathrm{MPO}}^{\dagger}(\Delta t)U_{\mathrm{BW}}(\Delta t)])^{1/N}}. \tag{14}$$

For small Δt, the error density for first and second-order trotterization will scale as $\mathcal{O}(\Delta t^2)$ and $\mathcal{O}(\Delta t^3)$ respectively; this will provide both a qualitative and quantitative criterion to compare our results to. It is also important to note that we must achieve a scaling better than $\mathcal{O}(\Delta t)$ to have meaningful results: if the error of the gates goes as $\mathcal{O}(\Delta t)$, then the total error for time t is $\mathcal{O}(\Delta t)$, and using a smaller Δt will yield no improvements.

A second quantity to consider is the Frobenius norm of the commutator of the BW circuit with some conserved quantities $\hat{O}$,

$$\mathcal{C}(U_{\mathrm{BW}}(\Delta t), \hat{O}) = \frac{||[U_{\mathrm{BW}}(\Delta t), \hat{O}]||_{\mathrm{F}}}{||\hat{O}||_{\mathrm{F}}}, \tag{15}$$

which is normalized by the Frobenius norm of the operator $\hat{O}$. Note that unlike $\delta(\Delta t)$—which calculates the error with respect to a high accuracy *approximation* of the true time propagator—this quantity can be calculated *exactly* if $\hat{O}$ has an efficient MPO representation. For example, Eq. (15) can be used with any local Hamiltonian, $\hat{O} = \hat{H}$, to investigate the conservation of energy.

1. Cluster Ising model

The first model we consider is a cluster Ising (CI) model,

$$\hat{H}_{\mathrm{CI}} = -(1+g)^2 \sum_{j=1}^{N} \hat{X}_j - 2(1-g^2) \sum_{j=1}^{N-1} \hat{Z}_j \hat{Z}_{j+1} + (g-1)^2 \sum_{j=1}^{N-2} \hat{Z}_j \hat{X}_{j+1} \hat{Z}_{j+2}, \tag{16}$$

where $g \in [-1, 1]$ is a tuning parameter which interpolates between a symmetry protected topological phase for $g < 0$ and topologically trivial phase for $g > 0$ [74,75]. The operators $\hat{X}_j$ and $\hat{Z}_j$ are the usual Pauli operators for two-level systems, acting at the lattice site j. This choice of Hamiltonian is motivated by recent works [76] which show how the ground state of Eq. (16) can be prepared on quantum simulators.

We apply our variational approach to Eq. (16). We choose a random initial unitary circuit with each gate random, but close to identity. For $g \leqslant 0$, we find the initial circuit plays little role in convergence: the method is able to consistently converge to some optimal value.

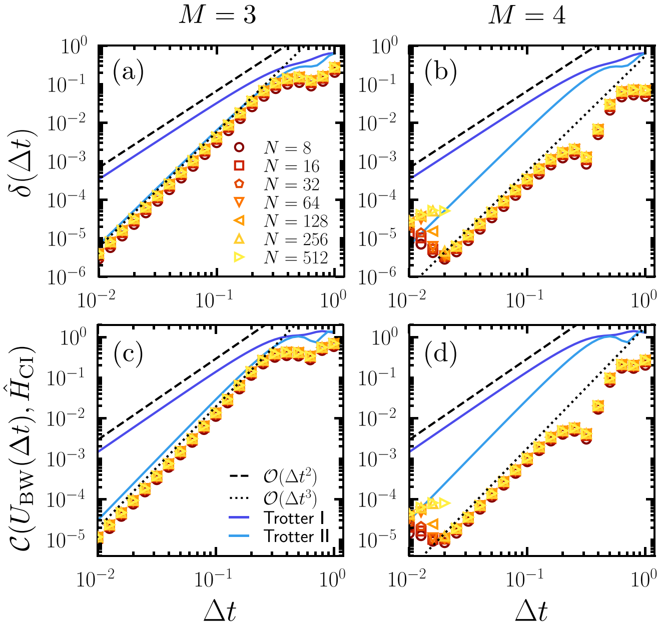


FIG. 7. Error scaling for the CI model. [(a) and (b)] The error density of the BW circuit with respect to the MPO approximation, $\delta(\Delta t)$ and [(c) and (d)] the Frobenius norm of the commutator with the Hamiltonian, Eq. (15). The data points show the results from variational optimization for system sizes $N = 8$ to 512. The solid lines show the same results for first and second-order trotterization with $N = 8$. The dashed lines show the scaling $\mathcal{O}(\Delta t^2)$ and the dotted lines show the scaling $\mathcal{O}(\Delta t^3)$. The left column is for depth $M = 3$, and the right column is for depth $M = 4$. All results are for $g = -0.75$.

Figure 7 show the results for the symmetry protected phase with depths $M = 3$ (left panels) and $M = 4$ (right panels). The top panels show the error density, $\delta(\Delta t)$, and the bottom panels show the error in the conservation of energy, $\mathcal{C}(U_{\mathrm{BW}}(\Delta t), \hat{H}_{\mathrm{CI}})$. Similar results for the trivial phase are found in Appendix A. In each case, we find the minimal depth needed to obtain at least a scaling $\mathcal{O}(\Delta t^2)$ is $M \geqslant 3$. Interestingly, there are no circuit depths which yield results comparable to first-order trotterization, shown by the darker solid line (which has scaling $\mathcal{O}(\Delta t^2)$, shown by the dashed line). Instead, the circuit depth $M = 3$ has scaling $\mathcal{O}(\Delta t^3)$ (dotted line), matching the results of second-order trotterization (lighter solid line). Indeed, we find the variational optimization is even able to give prefactor improvements to trotterization, but with a drastic reduction in the number of CNOT layers required, see Sec. IV B.

We also observe $\mathcal{O}(\Delta t^3)$ behavior for the circuit depth $M = 4$. However, it should be noted that the extra BW layer yields an order of magnitude improvement in the error and conservation of energy. Furthermore, we observe a dip in the error around $\Delta t \sim 0.4$ which is consistent with system size. While we are not able to offer an explanation for this behavior, the key observation is that additional layers might provide significant improvement. Notice that, for small times $\Delta t \sim 10^{-2}$, the optimization method can struggle to converge to the expected scaling. Nevertheless, the errors are on the order of numerical precision.

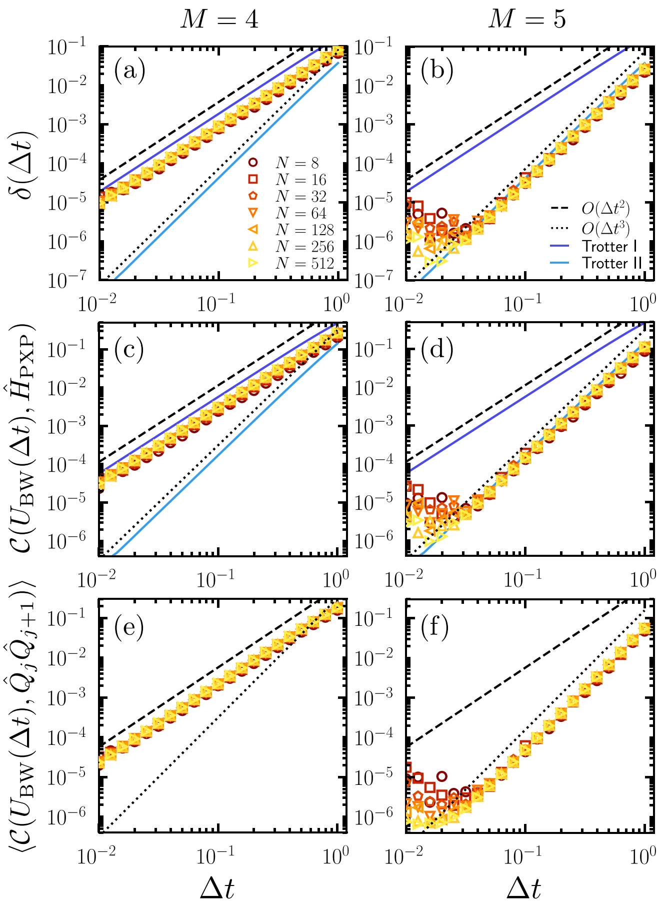


FIG. 8. Error scaling for the PXP model. Error metrics for the PXP model with circuit depths $M = 4$ (left panels) and $M = 5$ (right panels). [(a) and (b)] The error density $\delta(\Delta t)$ as a function of Δt. [(c) and (d)] Violation of the conservation of energy, $\mathcal{C}(U_{\mathrm{BW}}(\Delta t), \hat{H}_{\mathrm{PXP}})$. [(e) and (f)] Violation of the conserved quantities, $\langle \mathcal{C}(U_{\mathrm{BW}}(\Delta t), \hat{Q}_j\hat{Q}_{j+1})\rangle$. The data points and curves are as defined in Fig. 7.

2. The PXP model

The second model we consider is the PXP model [77–79],

$$\hat{H}_{\mathrm{PXP}} = \hat{X}_1\hat{P}_2 + \sum_{j=2}^{N-1} \hat{P}_{j-1}\hat{X}_j\hat{P}_{j+1} + \hat{P}_{N-1}\hat{X}_N, \quad (17)$$

where $\hat{P}_j = \frac{1}{2}(\hat{\mathbb{1}} - \hat{Z}_j)$ is the local projection onto the spin down state, and $\hat{\mathbb{1}}$ is the identity operator. The PXP model is an idealized *kinetically constrained model* for realising the dynamics of cold atom platforms [80] in a strong coupling limit, and has recently received much attention due to its connection to quantum many-body scars [79,81]. Furthermore, unlike the CI model considered in the previous section, the PXP model is nonintegrable [79].

Figure 8 shows the results of our optimization method. We use the variational approach with an annealing strategy which reduces the time Δt of the MPO from an initial time of $\Delta t = 1.0$ to the target time. The first column is for depth $M = 4$ (which was the minimal depth needed to reach order $k = 1$), and the second column shows depth $M = 5$. The first row shows the error density $\delta(\Delta t)$. For a depth of $M = 4$, we are able to obtain an $\mathcal{O}(\Delta t^2)$ scaling, with prefactor improvements on first-order trotterization. For a depth of $M = 5$, this can be improved to an $\mathcal{O}(\Delta t^3)$ scaling, matching the results of second-order trotterization. Notice that, as was the case for the CI model, the variational approach can sometimes struggle to converge for small Δt.

The second row of Fig. 8 shows the error in the conservation of energy, $\mathcal{C}(U_{\mathrm{BW}}(\Delta t), \hat{H}_{\mathrm{PXP}})$. This value scales in the same way as the error density, and also closely matches the results of trotterization. The PXP model also conserves the quantities $\hat{Q}_j\hat{Q}_{j+1}$ for all j, with $\hat{Q}_j = \hat{\mathbb{1}} - \hat{P}_j$. We calculate how well this quantity is conserved through the lattice site average

$$\langle \mathcal{C}(U_{\mathrm{BW}}(\Delta t), \hat{Q}_j\hat{Q}_{j+1})\rangle$$
$$= (N-1)^{-1}\sum_{j=1}^{N-1} \mathcal{C}(U_{\mathrm{BW}}(\Delta t), \hat{Q}_j\hat{Q}_{j+1}), \quad (18)$$

shown in the bottom row of Fig. 8. The error in these conserved quantities is qualitatively the same as the error density and the error in the conservation of energy. However, it is important to note that the conserved quantities of the PXP models are a result of the kinetic constraints in Eq. (17) and are *exactly* conserved by trotterization.

3. The next-nearest-neighbor Ising model

The final model we consider is the next-nearest-neighbor Ising (NNNI) model,

$$\hat{H}_{\mathrm{NNNI}} = g_x\sum_{j=1}^{N}\hat{X}_j + g_{zz}\sum_{j=1}^{N-1}\hat{Z}_j\hat{Z}_{j+1} + g_{z1z}\sum_{j=1}^{N-2}\hat{Z}_j\hat{Z}_{j+2}. \quad (19)$$

Its second-order trotterization can be written as a BW unitary circuit through a decomposition into SWAP gates and $R_{zz}(2\Delta t) = e^{-i\Delta t\hat{Z}\otimes\hat{Z}}$ gates, as illustrated in Fig. 9(a). By grouping gates together (shown by the dashed lines), the second-order decomposition becomes a BW circuit with depth $M = 6$. For $g_{zz} = 0$, this can be done for $M = 5$ by removing the gates responsible for the nearest-neighbor interactions in Eq. (19).

Unlike the two previous models, we find that the optimization for the NNNI model is highly susceptible to becoming stuck in local minima. Because of this, optimising the circuits to achieve a scaling that beats $\mathcal{O}(\Delta t)$ is extremely difficult, and only happens on rare occasions for some choice of initial guess. This effect becomes more apparent for larger system sizes. As such, this model serves as an interesting example where the solution is known explicitly, but the variational optimization method is unable to reliably obtain a good result. Instead, we propose an initial guess which uses the structure of the circuit in Fig. 9(a), but only for the SWAP gates, see Fig. 9(b). All other gates are set to identity. With this choice of initial guess, we find the error metrics converge to those of the second-order trotterization solution, see Figs. 9(c) and 9(d) for the error density and Figs. 9(e) and 9(f) for the conservation of energy. These results could motivate the future investigation of strategies to systemically suggest more optimal choices of initial guesses. Notice that in this instance, we observe little improvement when compared to the results of trotterization.

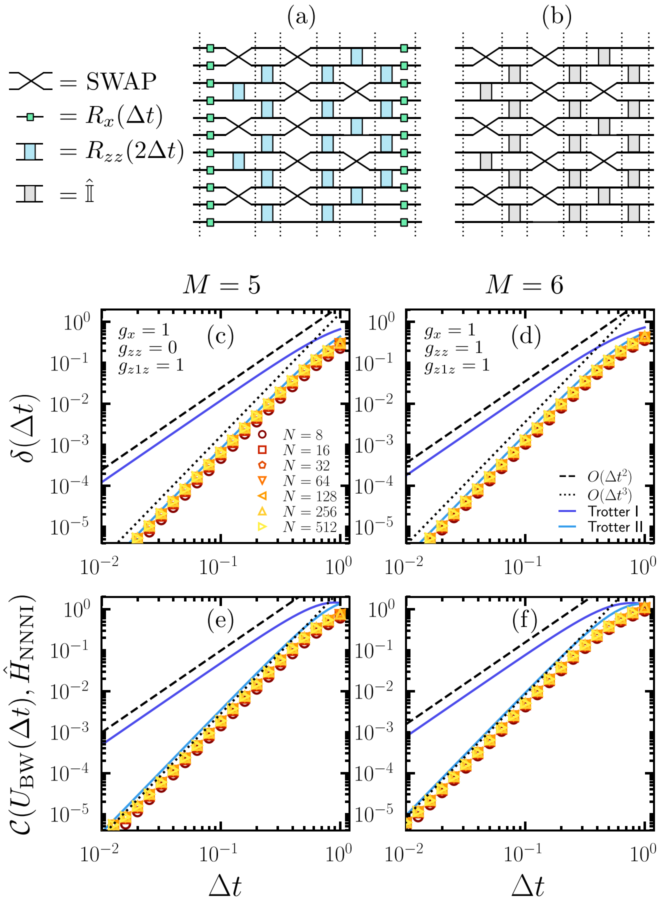


FIG. 9. Error scaling for the NNNI model. (a) A second-order trotter decomposition for the NNNI model can be written as a BW circuit composed of SWAP gates, $R_x(\Delta t) = e^{-i\Delta t\hat{X}/2}$ gates and $R_{zz}(2\Delta t) = e^{-i\Delta t\hat{Z}\otimes\hat{Z}}$ gates. The dashed lines show where gates can be compressed to give one layer of the BW circuit. (b) As an initial guess for variational optimization with circuit depth $M = 6$, we use a similar circuit where the SWAP gates are placed in the same positions as (a). The remainder of the gates are set to identity. For depth $M = 5$, the initial guess is the same, but with the final layer removed. [(c) and (d)] The error density, $\delta(\Delta t)$. [(e) and (f)] The error in the conservation of energy, $\mathcal{C}(U_{\mathrm{BW}}(\Delta t), \hat{H}_{\mathrm{NNNI}})$. (c) and (e) show results for $g_x = g_{z1z} = 1$ and $g_{zz} = 0$ with circuit depth $M = 5$, while (d) and (f) show results for $g_x = g_{zz} = g_{z1z} = 1$ and $M = 6$. The data points and curves are as defined in Fig. 7.

B. Number of CNOT layers

While two-qubit gates are required to create entanglement, they are often the operations which take the longest to implement on quantum computers, and thus are most responsible for decoherence. For many popular platforms, such as IBM's quantum processors, they are implemented using CNOT gates. Because of this, it is essential to implement quantum dynamics using as few CNOT layers as possible. We discuss how both the BW circuits and trotterized circuits can be implemented using just single qubit operations and CNOT gates in Appendix B, but summarize the results here and in Table II.

Each layer of BW unitary gates can be decomposed into a quantum circuit with three layers of CNOT gates. It follows that a BW circuit of depth M has at most $3M$ layers of CNOT gates. The trotterized circuits for each of the models requires more care. The first and second-order trotterized circuits for the CI model can be implemented with 16 CNOT layers and 28 CNOT layers respectively. On the contrary, a BW circuit of depth $M = 3$ gives a second-order scaling while requiring only nine CNOT layers. This yields an enhancement factor of ~3.1 when compared to second-order trotterization, and can even be implemented with fewer CNOT layers than first-order trotterization. A circuit depth of $M = 4$ also has a second-order scaling, but with order-of-magnitude improvements on second-order trotterization. This can be implemented with 12 CNOT layers.

TABLE II. We compare the circuits from trotterization to ones from our optimization method. The maximal number of CNOT layers required to implement a single time step of the trotterized circuits and BW circuits for any system size is shown in the second column. The final column shows the error density $\delta(\Delta t)$ (rounded to three significant figures), for time step $\Delta t = 0.1$ and system size $N = 8$. Results are shown for the cluster Ising (CI) model, the PXP model, and a next-nearest-neighbor Ising (NNNI) model. The order of the approximation k is given for each circuit. Note that the error density scales as $\mathcal{O}(\Delta t^{k+1})$ for small Δt. The CI model is for $g = -0.75$ and the NNNI model is for $g_x = g_{zz} = g_{z1z} = 1$.

	Circuit	CNOT Layers	Error, $\delta(\Delta t = 0.1)$
CI	Trotter $k = 1$	16	3.03×10^{-2}
	Trotter $k = 2$	28	6.32×10^{-3}
	Brickwall $k = 2$	9	3.03×10^{-3}
PXP	Trotter $k = 1$	42	1.87×10^{-3}
	Trotter $k = 2$	70	4.23×10^{-5}
	Brickwall $k = 1$	12	6.67×10^{-4}
	Brickwall $k = 2$	15	3.06×10^{-5}
NNNI	Trotter $k = 1$	13	1.78×10^{-2}
	Trotter $k = 2$	15	1.81×10^{-3}
	Brickwall $k = 2$	18	1.04×10^{-3}

For the PXP model, we found that circuit depths $M = 4$ and 5 gave first- and second-order scaling respectively, requiring a total of 12 and 15 layers of CNOTs. On the contrary, first-order trotterization requires 42 CNOT layers (assuming there are only CNOT connections between neighboring qubits), giving an enhancement factor of 3.5. Similarly, second-order trotterization requires 70 CNOT layers, giving an enhancement factor of ~4.6.

The final model we considered was the NNNI model. Here, both the BW circuits and the trotterized circuit gave a second-order scaling with the same number of layers of two-qubit gates. Naively, one might assume they can be implemented with the same number of CNOT gates. However, it is important to note that the two-qubit gates in the trotterized circuit only include SWAP gates and $R_{zz}(\Delta t)$ gates, while the classically optimized BW circuits allow for arbitrary two-qubit gates. Any general two-qubit gate (including SWAP gates) can be implemented with three CNOT gates, while the $R_{zz}(\Delta t)$ requires only two CNOTs. Because of this, the trotterized circuit can be implemented with fewer CNOT layers than the classically optimized circuits. For $g_{zz} = 0$ ($g_{zz} \neq 0$), the optimized BW circuit requires 15 (18) CNOT layers, while the trotterized circuit requires only 13 (15) layers. Note that, while the BW circuit requires more CNOT layers to implement the dynamics, it yields an improvement in the error density, see Table II.

C. Convergence of optimization

We now benchmark the performance of the variational optimization. In particular, we test the number of sweeps required for convergence. After each sweep l, we measure the Frobenius norm of the difference between our BW circuit ansatz and the MPO approximation, $\mathcal{F}_l^{(1)} = \sqrt{\mathcal{F}_l^{(2)}}$, where $\mathcal{F}_l^{(2)}$ is Eq. (11) after the l-th sweep. We define convergence to be when the relative difference in the Frobenius norm between successive sweeps falls below some value ϵ

$$\frac{\mathcal{F}_{l-1}^{(1)} - \mathcal{F}_l^{(1)}}{\mathcal{F}_{l-1}^{(1)}} < \epsilon. \tag{20}$$

In practice, we use $\epsilon = 10^{-6}$.

We first check the number of sweeps required for convergence as a function of time, with a fixed depth M, which we choose to be the minimum depth that is required to yield an error density that scales better than $\mathcal{O}(\Delta t)$. For the CI model, this is a circuit depth of $M = 3$, and for the PXP model, a circuit depth of $M = 4$. This is shown in Figs. 10(a) and 10(b)] for the CI and PXP models respectively with system sizes $N = 64,\ 128,$ and 256. Each data point shows the average over one hundred independent runs. Notice that in both cases, the number of sweeps required to converge looks to be independent of system size. Furthermore, for small times $\Delta t \lesssim 1$, the number of sweeps needed to converge appears to go as a power law, $O(\Delta t^{-\alpha})$; for the CI model, there is an exponent $\alpha = 2$, and for the PXP model, an exponent of $\alpha \approx 2.5$. While we cannot offer a compelling explanation for the exact values of these exponents, we speculate they are related to the transport properties of the models. Nevertheless, the key point is that this scales subexponentially. Notice that the number of sweeps deviate from the scaling at large times, where the error density no longer scales as a polynomial (cf. Fig. 7). Furthermore, while it might seem counterintuitive that the number of sweeps needed for convergence decreases with larger time step Δt, note that the accumulated error for evolving to a total time t with a small time step Δt goes as $\mathcal{O}(t\Delta t^k)$, with $k = 2$ for the CI model and $k = 1$ for the PXP model. It follows that decreasing the time step will result in a more accurate evolution.

We also investigate convergence for a fixed time step t and variable circuit depth M. Figures 10(c) and 10(d) show the average number of sweeps required for convergence as a function of M for system sizes $N = 8,\ 16,$ and 32 for the CI model and PXP model respectively. While there is no obvious trend, it is clear that the number of sweeps needed for convergence scales subexponetially in the circuit depth. However, we emphasize that while the number of sweeps might not scale exponentially, the cost required for updating unitary gates does. We also show the average error density for the converged circuits in Figs. 10(e) and 10(f), for the same systems as before. Notice that while increasing the circuit depth consistently provides an improvement to the error, it looks to be diminishing. The most substantial improvements are found when increasing the circuit depth changes the scaling of the error density for small times, i.e., increasing the depth M increases the exponent k for $\delta(\Delta t) \sim \mathcal{O}(\Delta t^{k+1})$. This is evident from panel (f), where circuit depths $M = 3, 4, 5$ gives orders $k = 0, 1, 2$, respectively. In practice, we find that is it more effective to optimize for a shallow circuit which also yields a desirable scaling.

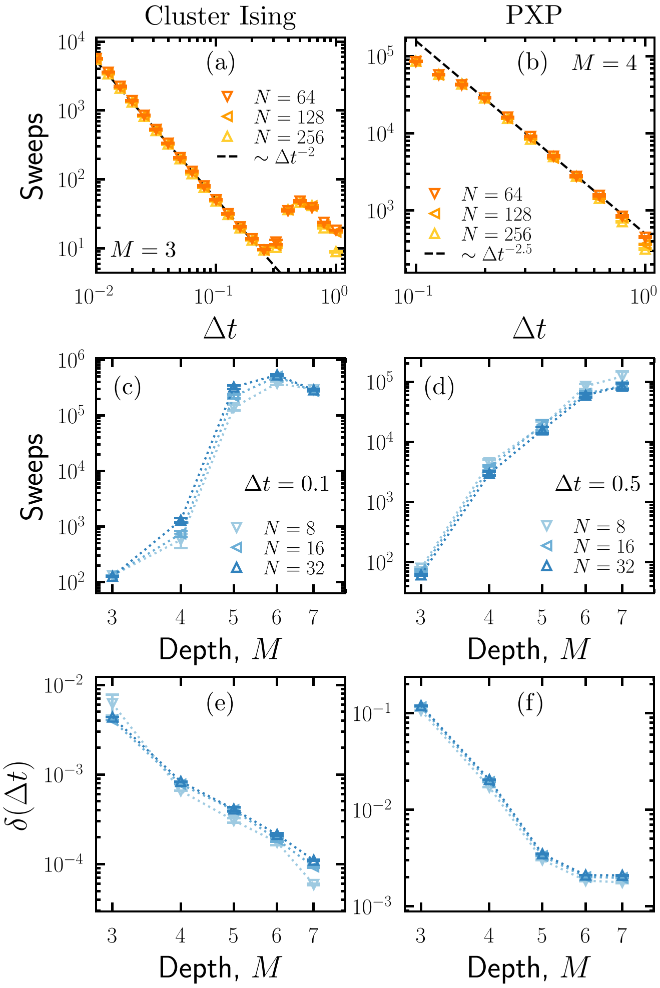


FIG. 10. Convergence of variational optimization. The left panels are for the CI model with $g = -0.5$, and the right panels are for the PXP model. [(a) and (b)] The number of sweeps required to converge for a fixed circuit depth M as a function of time, Δt. Data are shown for $N = 64,\ 128,$ and 256 with a circuit depth of $M = 3$ for the CI model, and $M = 4$ for the PXP model. For small times, the number of sweeps required to converge goes as some power law, $\Delta t^{-\alpha}$, with $\alpha = 2$ for the CI model and $\alpha \approx 2.5$ for the PXP model, shown by the dashed lines. [(c) and (d)] The number of sweeps required to converge for a fixed time Δt as a function of circuit depth, M. Data is shown for $N = 8, 16, 32$ with a time of $\Delta t = 0.1$ for the CI model, and $t = 0.5$ for the PXP model. [(e) and (f)] The error density, $\delta(\Delta t)$, averaged over all runs for the same systems as (c) and (d). Each data point is the mean of 100 independent runs.

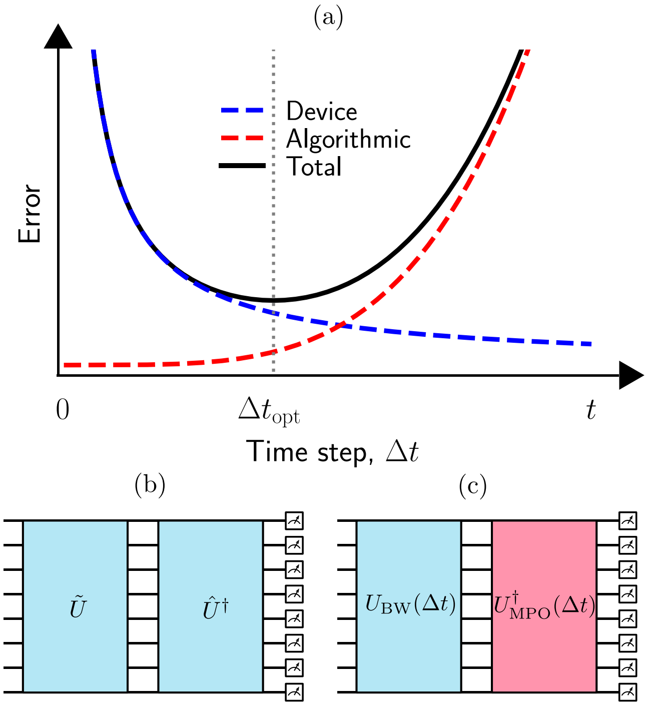


FIG. 11. Optimal time step. (a) The optimal time step for quantum simulation can be determined through considerations of the errors native to the quantum device (blue dashed line) and the algorithmic errors from the optimization of BW circuits (red dashed line). Small time steps, Δt, require a large circuit depth to reach time t, and accumulate a large error from the device. On the contrary, the BW circuit is unable to well approximate the true time evolution operator for large time steps. The optimal time step $\Delta t_{\rm opt}$ is determined through a combination of the two. (b) The native error of a quantum computer can be measured by running a BW circuit U forwards and then in reverse, and taking measurements. In practice, the quantum computer will add noise to the evolution, and we can consider this the same as evolving by some noisy circuits $\tilde{U}$ and $\hat{U}^\dagger$. The error can be calculated from the discrepancy between the initial state and the measurements; see the main text for details. (c) The algorithmic error can be calculated with a similar setup, where this time we evolve using the BW circuit, $U_{\rm BW}(\Delta t)$, but then evolve using the adjoint of the MPO, $U^\dagger_{\rm MPO}(\Delta t)$.

V. OPTIMAL TIME STEP

We have explained how to optimize a BW circuit to efficiently simulate quantum dynamics for some time Δt, which can then be used repeatedly to achieve some target time t that is unobtainable on a classical computer. However, it is not immediately clear what time step Δt should be chosen to simulate the dynamics. On one hand, we have shown that choosing a smaller time step can reduce the *algorithmic error* for evolving to time t. Indeed, if a BW circuit of depth M gives a kth-order approximation (e.g., $M = 3$ for the CI model has order $k = 2$), then the cost of evolving to time t is $\mathcal{O}(t\Delta t^k)$. This error is illustrated by the dashed red line in Fig. 11. This entices us to pick a small time step, resulting in a deeper quantum circuit. On the other hand, quantum computers are never completely isolated from their environment. This introduces errors which are native to the *device*. We assume that this error is exponential in the circuit depth, as shown by the blue dashed line in Fig. 11. This indicates that our simulations would benefit from a shallower quantum circuit. The total simulation error is a combination of the two, shown by the black curve, and the optimal time step is the one which minimizes this error. In what follows, we explain how to estimate each of these errors to guide us in determining an optimal time step. We note that a similar analysis was done for trotterization in Ref. [82].

A. Measures of fidelity on quantum devices

One way to estimate the error native to a quantum computer is to simulate a unitary BW circuit, followed by the same circuit, but in reverse. For a perfect quantum computer, the resulting operation is the identity matrix, $UU^\dagger = \hat{\mathbb{1}}$, and the initial state will be retrieved with certainty. However, in practice, each layer of the quantum circuit will incur some random error native to the device, and the circuit described above will not yield the identity matrix. This allows us to estimate a layer-dependent error of the device through the *fidelity* of the state.

Let us assume the system is prepared in a state such that each qubit is an eigenstate of the Z basis, e.g., $|\psi\rangle = \bigotimes_{j=1}^N |\sigma_j\rangle$ with $\sigma_j = \pm 1$. We can then evolve the system using a unitary BW circuit, followed by the same circuit in reverse. Following this, one can then take a measurement for each qubit in the Z-basis. This is shown in Fig. 11(b). Note that, in practice, the error might depend on the unitary circuit, and one should use a circuit which resembles the circuits that simulate the dynamics of the Hamiltonian. The fidelity of the device for the total circuit goes as

$$F^{\rm d} = \frac{1}{2^N}\left\langle \sum_{\boldsymbol{\sigma}} |\langle \boldsymbol{\sigma}|\hat{U}^\dagger\tilde{U}|\boldsymbol{\sigma}\rangle|^2 \right\rangle_{\rm noise}, \tag{21}$$

where $|\boldsymbol{\sigma}\rangle = |\sigma_1\cdots\sigma_N\rangle$, and $\tilde{U}$ and $\hat{U}$ are the unitary circuit U subject to two independent realizations of the random noise of the quantum computer. The $\langle\cdot\rangle_{\rm noise}$ indicates an average over random noise. In practice, one can estimate Eq. (21) by simulating the circuit shown in Fig. 11(b) a finite number of times (sometimes referred to as *shots*) each with a random initial state.

While the fidelity provides an unbiased way to estimate the native errors of quantum simulation, in practice it is exponentially difficult to measure in the number of qubits: the number of shots needed for a measurement with some desired accuracy goes as $\mathcal{O}(e^N)$ [83]. Typically, one is interested in studying the evolution of local quantities, such as the expectation value of local observables. As such, it could be beneficial to instead determine an error using only local measurements. As before, we can prepare the initial state of the system to be a product state of Z-basis eigenstates, and then evolve it forwards and backwards. We then take measurements in the Z basis. However, this time, we calculate a *local fidelity*,

$$\tilde{F}^{\rm d} = \frac{1}{N2^N}\left\langle \sum_{\boldsymbol{\sigma}}\sum_{j=1}^{N} \langle \boldsymbol{\sigma}|\hat{U}^\dagger\tilde{U}\hat{\sigma}_j\tilde{U}^\dagger\hat{U}|\boldsymbol{\sigma}\rangle \right\rangle_{\rm noise}, \tag{22}$$

where $\hat{\sigma}_j = (\hat{\mathbb{1}} + {\rm sgn}(\sigma_j)\hat{Z}^j)/2$ is a projector onto the state σ_j (i.e., the jth spin in the initial state $\boldsymbol{\sigma}$). Notice that Eq. (22) can

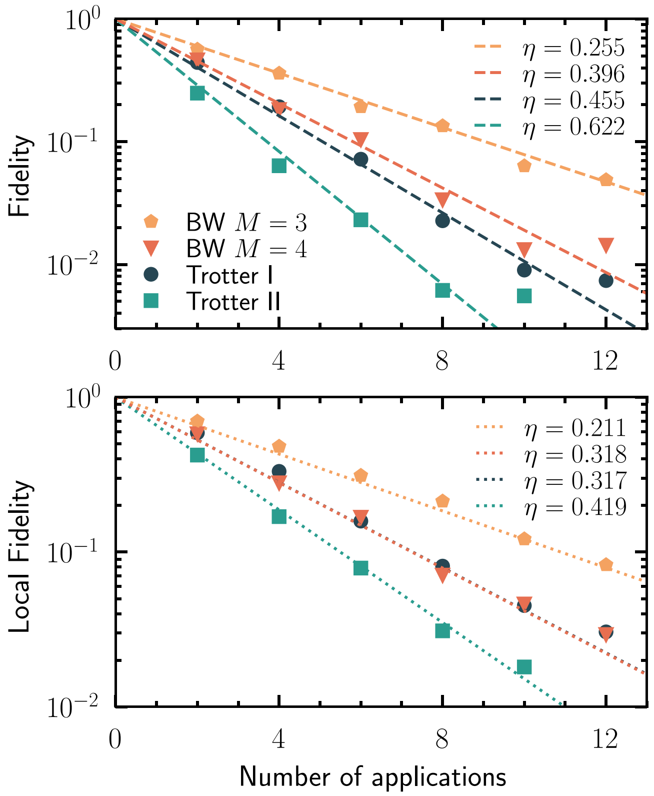


FIG. 12. Fidelity of the quantum device. We run the forwards-backwards setup from Fig. 11(b) to measure the errors on the device ibm_brisbane, which was accessed on the 11th of November 2023. Results are for the CI model with $g = -0.5$, $N = 5$ qubits (each starting from spin down), and time step $\Delta t = 0.25$. The top panel shows the normalized fidelity as a function of the total number applications of each circuit. The bottom panel shows the same, but for the normalized local fidelity. Data are shown for first-order trotterization, second-order trotterization, and optimized BW circuits with depths $M = 3$ and 4. The dashed/dotted line shows the exponential function with exponent $-\eta$, with η given in the legends. Each data point is averaged over 8192 shots.

be measured with the same set up as Eq. (21). Furthermore, the definition can easily be extended to consider multiple-site observables.

We demonstrate our approach in Fig. 12 using the ibm_brisbane device, where we simulate the dynamics of the CI Ising for $N = 5$ qubits using a time-step of $\Delta t = 0.25$, with first and second-order trotterization, and with BW circuits with depths $M = 3$ and 4. We first evolve forwards in time by multiple applications of time step Δt, and then backwards in time, with the same number of applications of time step $-\Delta t$, as described in the setup above. While technical limitations only allowed us to run the circuits with the initial state where all spins are $\sigma_j = -1$, in practice, one should average over many initial conditions.

The top panel shows the normalized fidelity $(F^d - 2^{-N})/(1 - 2^{-N})$ as a function of the number of total applications of each circuit (which is twice the number of forward applications). The bottom panel shows the same, but for the normalized local fidelity, $2\tilde{F}^d - 1$ [84]. The dashed/dotted lines show the exponential fit $e^{-\eta x}$ for each type of circuit, where x is the number of applications. It is clear that the fidelity decays approximately exponentially for both measures. While the number of applications of each circuit is the same, it is important to note that each circuit will differ in its quantum circuit depth. We expect the most significant source of error to come from the number of CNOT layers. Indeed, the BW circuit with $M = 3$ outperforms both first and second-order trotterization while attaining an error density on par with second-order trotterization (cf. Fig. 7). The BW circuit with depth $M = 4$ has a native error similar to that of first-order trotterization, but with an error density which outperforms second-order trotterization by a significant prefactor.

B. Measures of fidelity for classically optimized circuits

The error density from Sec. IV allows us to understand the algorithmic error of a single time step with respect to the true evolution operator. However, it cannot be used comparatively with the fidelity measures of the previous section. We can instead use a forwards and backwards scheme as was done in the previous section, but this time, with the forwards evolution done with the BW circuit approximation, and the backwards evolution done with the true dynamics, $U^\dagger(\Delta t)$, which in practice we approximate with an MPO, $U(\Delta t) \approx U_{\text{MPO}}(\Delta t)$. That is, we calculate

$$F^{\text{a}}_{\Delta t} = \frac{1}{2^N}\sum_{\boldsymbol{\sigma}} |\,\langle\boldsymbol{\sigma}|U^\dagger_{\text{MPO}}(\Delta t)U_{\text{BW}}(\Delta t)|\boldsymbol{\sigma}\rangle\,|^2. \tag{23}$$

This is illustrated in Fig. 11(c).

While we are unable to bound the fidelity using the error density (or the Frobenius norm), it is possible to bound the fidelity using the operator norm [85]. If we have $||U(\Delta t) - U_{\text{BW}}(\Delta t)|| \leqslant \mathcal{E}(\Delta t)$, then it also follows that $1 - F^{\text{a}}_{\Delta t} \leqslant \mathcal{E}(\Delta t)^2$, see Appendix C. For the local fidelity, we can calculate

$$\tilde{F}^{\text{a}}_{\Delta t} = \frac{1}{N2^N}\sum_{\boldsymbol{\sigma}}\sum_{j=1}^{N}\langle\boldsymbol{\sigma}|U^\dagger_{\text{BW}}(\Delta t)U_{\text{MPO}}(\Delta t)\,\hat{\sigma}_j \times U^\dagger_{\text{MPO}}(\Delta t)U_{\text{BW}}(\Delta t)|\boldsymbol{\sigma}\rangle. \tag{24}$$

We explain how to calculate Eqs. (23) and (24) using TNs in Appendix C. It is important to note that Eqs. (23) and (24) do not give the infidelity (or a bound on the infidelity) for the evolution of arbitrary state $|\psi\rangle$. Instead, they provide the infidelity averaged over all possible states. Nevertheless, they will serve as a useful estimate of the algorithmic error for a single time step.

Equations (23) and (24) provide us with a way to estimate the fidelity of the time-evolved state with a BW circuit for the time step, Δt. However, in practice, we would like to calculate the fidelity for n applications of the circuit, i.e., the fidelity of evolving to time $t = n\Delta t$ using the product formula. While calculating this exactly for large N and large t is generally not possible, we can again bound it by the operator norm error, $F^{\text{a}}_{n\Delta t} \geqslant 1 - n^2\mathcal{E}(\Delta t)^2$. We estimate that the total infidelity for n applications of the circuit to go as

$$I^{\text{a}}_{n\Delta t} \approx n^2\big(1 - F^{\text{a}}_{\Delta t}\big) \tag{25}$$

and similarly for the local infidelity. We expect that $1 - F^{\text{a}}_{\Delta t} \sim \mathcal{O}(\Delta t^{2k+2})$ for a kth-order approximation and small Δt, which is numerically verified for the CI model in Appendix C. Appendix C also demonstrates that while Eq. (25) can overestimate the true error for large n, it can still give a reasonable approximation.

C. The total error

We can now use our estimates of the fidelity to estimate a total fidelity for simulating a total time t with n time steps on a quantum device. We expect the fidelity to be multiplicative,

$$F_{n\Delta t} \approx e^{-\eta n}\big[1 - n^2\big(1 - F^{\mathrm{a}}_{\Delta t}\big)\big], \tag{26}$$

where η is the exponential decay rate of the fidelity due to native errors from the quantum device. For some target simulation time, t, one can estimate the infidelity $I_{n\Delta t} = 1 - F_{n\Delta t}$ for a range of n, and use it to predict what choice of $\Delta t = t/n$ gives minimal infidelity. While the values of η recorded from the `ibm_brisbane` are too large to reliably predict an optimal time, we show an example of calculating the optimal time step Δt_{opt} with fictional values of η in Appendix C.

VI. CONCLUSIONS

In this paper, we have proposed a method to classically optimize BW circuits to efficiently simulate the dynamics of quantum many-body systems, building on the works of Refs. [49–52,86]. Like Refs. [51,52], we make use of MPO approximations of the time evolution operator to variationally update unitary gates in the BW circuits, allowing the method to efficiently scale with system size. The previous works chose particular parametrizations of the two-qubit gates to enforce unitarity, which were optimized using gradient descent approaches. However, our method makes use of the polar decomposition to find efficient and optimal unitary gates [46]. We successively apply our method to multiple Hamiltonians with neighboring three-body interactions (whereas previous works were limited to just two-body interactions), demonstrating that the dynamics of many-body Hamiltonians can be efficiently implemented using a BW structure with a shallow depth. This is significant because BW circuits can easily be implemented on current digital quantum computers [46].

The first model we considered is a cluster Ising model. We successfully demonstrated that dynamics of the model could be implemented with an accuracy that outperforms second-order trotterization with a BW circuit depth of $M = 3$, with significantly less CNOT layers required. We then tested our method on the PXP model, a kinetically constrained model relevant to cold atom experiments. Here, we were also able to find circuits with depths $M = 4$ and 5 which outperform both first and second-order trotterization, respectively. Like the cluster Ising model, this is done with drastically less CNOT layers. However, this came with the trade-off that the conserved quantities of the PXP model are violated (whereas trotterization exactly conserves these quantities). Finally, we tested our model on a next-nearing-neighbor Ising model. Here, we were only able to find slight improvements on the accuracy when compared to second-order trotterization, but requires a small increase in the number of CNOT layers. For this model, we found that our method often suffered from local minima; to overcome this, we used an initial ansatz inspired by the second-order trotterization. This could motivate future studies to find strategies to overcome these optimization difficulties.

We have shown that our optimization method can generate BW circuits which are more accurate and efficient than those produced by first- and second-order trotterization, reducing the amount of algorithmic error. This could allow us to simulate dynamics using larger time steps, and thus allow us to simulate dynamics for longer times before the onset of decoherence. Furthermore, we show that the same circuits can be implemented on quantum computers using fewer layers of CNOTs than trotterization. This reduces the accumulated native errors of the quantum device, allowing us to simulate more time steps, again increasing the possible simulation window

We have also explained how the errors from optimization and those from the quantum computer can be used to guide us in our choice of time step to minimize the total error. We first implemented the optimized BW circuits and trotterized circuits on IBM's quantum processor `ibm_brisbane`, demonstrating that optimized BW circuits can achieve a more precise dynamics while being subject to less noise. We then proposed that the optimal time step could be estimated through the infidelity of a single time step (with respect to the true time evolution operator), and the errors native to the quantum simulator. This assumed that the algorithmic error from the optimization of BW circuits grew linearly with the number of applications. Indeed, for trotterization it is known that this overestimates the error at large times [87]. This estimation can be improved by finding tighter bounds on the errors [88–90]. Similar advancements for BW circuits could facilitate a better choice of time step. Alternatively, one could use adaptive methods to find the optimal choice of time step [91,92].

One avenue for future works could be to investigate how to generalize the method to two dimensions. While BW circuits are a natural ansatz for one-dimensional systems, two-dimensional systems can allow for more complex arrangements. This could allow for the possibility to also optimize the geometry of the circuits. Furthermore, the use of TNs allowed our method to scale well with system size. Unfortunately, two-dimensional TNs, such as projected-entangled pair states [93], are known to suffer from a poor scaling with system size due to the inability to exactly contract them in an efficient manner [94,95]. Nevertheless, there are promising developments in two-dimensional TNs which might allow for this to soon be a possibility [96–99]. We hope to report on this at a later date.

The code for optimising unitary brickwall circuits can be found at [100].

ACKNOWLEDGMENTS

L.C. was supported by an EPSRC Doctoral prize from the University of Nottingham. A.M. acknowledges the support of the DAAD-WISE scholarship from the Deutscher Akademischer Austausch Dienst, and the KVPY fellowship from the Government of India. At the University of Toronto, A.M. is supported by the Lachlan Gilchrist Fellowship. F.P.

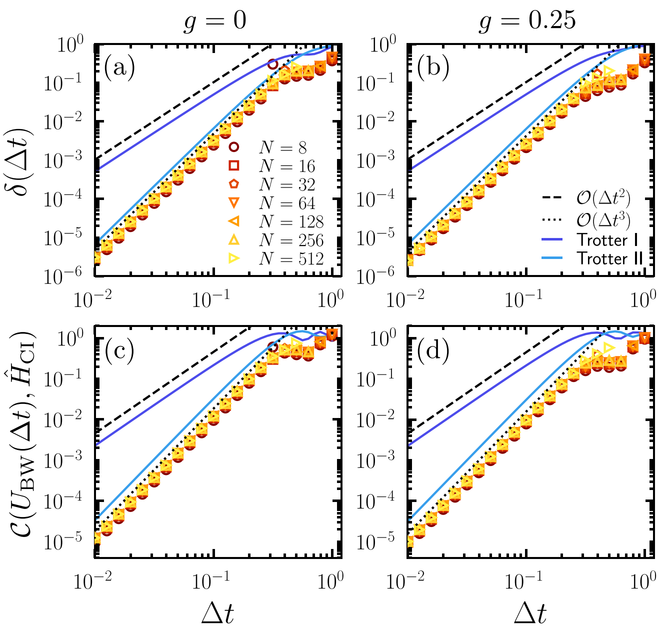


FIG. 13. Error scaling in the CI model. [(a) and (b)] The error density $\delta(\Delta t)$ and [(c) and (d)] the error in the conservation of energy $\mathcal{C}(U_{\mathrm{BW}}(\Delta t), \hat{H}_{\mathrm{CI}})$ for the CI model with circuit depth $M = 3$. The left column is at the tricritical point $g = 0$, and the right column is in the trivial phase $g = 0.25$. All quantities are defined as in Fig. 7.

acknowledges support from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy–EXC–2111–390814868, the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 851161), as well as the Munich Quantum Valley, which is supported by the Bavarian state government with funds from the Hightech Agenda Bayern Plus. A.G.-S. acknowledges support from a research fellowship from The Royal Commission for the Exhibition of 1851. We acknowledge access to the University of Nottingham Augusta HPC service.

APPENDIX A: ERROR SCALING IN THE TRIVIAL PHASE OF THE CLUSTER ISING MODEL.

The variational optimization method proposed in Sec. III worked well for CI in the symmetry protected phase ($g < 0$) and had little dependence on initial state. Conversely, the approach can struggle for $g \gtrsim 0$, often getting stuck in local minimum. A reliable strategy to combat this is to anneal the parameter g over the range $[-1, 1]$, using the result from a previous choice of g as an initial seed. In practice, we start from a value of $g = -1.0$ and optimize a random initial guess using our variational method. We then iteratively reduce the parameter $g \leftarrow g - 0.05$ and optimize using the previous result as the initial guess. This is repeated until the target value of g is obtained. The results of this approach are shown in Fig. 13 for $g = 0$ and $g = 0.25$ with circuit depth $M = 3$. The top panel shows the error density and the bottom panel shows the error in the conservation of energy. The results closely match that of Fig. 7.

APPENDIX B: IMPLEMENTING BRICKWALL AND TROTTERIZED CIRCUITS ON QUANTUM COMPUTERS

The first unitary gate we consider is a general two-qubit gate, such as those in unitary BW circuits. Any two-qubit gate can be decomposed into a circuit of CNOT gates and single-qubit gates, which requires at most three CNOTs. This is shown in Fig. 14(a). The single-qubit gates need to be carefully chosen to make this decomposition valid; see Refs. [43,101] for more details. While the general decomposition can be achieved with three CNOTs, some gates can be implemented with less. For example, the two-qubit rotation gate

$$R_{zz}(\Delta t) = e^{-i\frac{\Delta t}{2}\hat{Z}\otimes\hat{Z}} \tag{B1}$$

can be implemented with a single-qubit rotation gate $R_z(\Delta t) = e^{-i\frac{\Delta t}{2}\hat{Z}}$ placed on the target qubit in-between two identical CNOTs [60]. Notice that this gate is used in the trotterization for both the CI model and the NNNI model.

Three-qubit entangling gates require more care to implement. For rotations of Pauli strings on three qubits, it is well understood how to efficiently implement such gates using only single-qubit gates and CNOTs [102]. For example, the rotation gate

$$R_{zxz}(\Delta t) = e^{-i\frac{\Delta t}{2}\hat{Z}\otimes\hat{X}\otimes\hat{Z}}, \tag{B2}$$

is shown in Fig. 14(c). We first note that

$$R_{zxz}(\Delta t) = (\hat{\mathbb{1}} \otimes H \otimes \hat{\mathbb{1}})R_{zzz}(\Delta t)(\hat{\mathbb{1}} \otimes H \otimes \hat{\mathbb{1}}), \tag{B3}$$

with $R_{zzz}(\Delta t) = e^{-i\frac{\Delta t}{2}\hat{Z}\otimes\hat{Z}\otimes\hat{Z}}$. The $R_{zzz}(\Delta t)$ gate can then be decomposed into four CNOTs and an $R_z(\Delta t)$ gate [60].

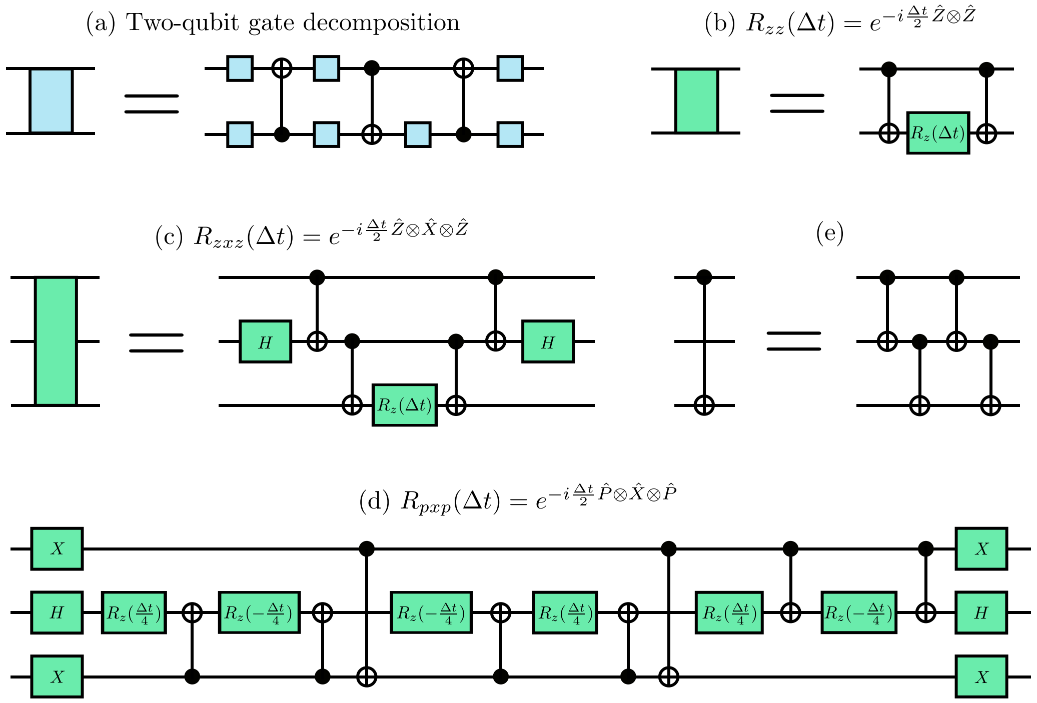
To implement the three-body gates in the trotterized circuits for the PXP model, we first note that

$$\begin{aligned} R_{pxp}(\Delta t) &= e^{-i\frac{\Delta t}{2}\hat{P}\otimes\hat{X}\otimes\hat{P}} \\ &= (\hat{X} \otimes H \otimes \hat{X})R_{qzq}(\Delta t)(\hat{X} \otimes H \otimes \hat{X}), \end{aligned} \tag{B4}$$

where $\hat{Q} = \hat{\mathbb{1}} - \hat{P}$ and $R_{qzq}(\Delta t) = e^{-i\frac{\Delta t}{2}\hat{Q}\otimes\hat{Z}\otimes\hat{Q}}$. We can then write

$$\begin{aligned} R_{qzq}(\Delta t) = (\hat{Q} \otimes R_z(\Delta t) \otimes \hat{Q}) + (\hat{P} \otimes \hat{\mathbb{1}} \otimes \hat{Q}) \\ + (\hat{Q} \otimes \hat{\mathbb{1}} \otimes \hat{P}) + (\hat{P} \otimes \hat{\mathbb{1}} \otimes \hat{P}). \end{aligned} \tag{B5}$$

This can be considered to be a three-qubit gate where the first and last qubit act as control gates for the rotation gate on the central qubit. The remainder of the circuit makes use of six $R_z(\Delta t)$ gates and eight CNOT gates to implement the trotterized PXP dynamics [60], and is shown in Fig. 14(d). While we will not give the complete details of how the circuit is derived, one can carefully check that this correctly implements the gate $R_{pxp}(\Delta t)$. Notice that the circuit has two CNOT gates which act on next-nearest-neighboring qubits. If the quantum computer simulating the dynamics has connections between these qubits, then the circuit can be used as presented. However, if the quantum computer only has connections between neighboring qubits, then these CNOT gates must be decomposed into CNOTs acting only on neighboring qubits; the most optimal known decomposition for this is shown in Fig. 14(e), and requires four CNOTs [103]. This increases the number of CNOTs required for the $R_{pxp}(\Delta t)$ gate to fourteen.



(a) Two-qubit gate decomposition

(b) $R_{zz}(\Delta t) = e^{-i\frac{\Delta t}{2}\hat{Z}\otimes\hat{Z}}$

(c) $R_{zxz}(\Delta t) = e^{-i\frac{\Delta t}{2}\hat{Z}\otimes\hat{X}\otimes\hat{Z}}$

(e)

(d) $R_{pxp}(\Delta t) = e^{-i\frac{\Delta t}{2}\hat{P}\otimes\hat{X}\otimes\hat{P}}$

$R_z(\Delta t)$
H
$R_z(\Delta t)$
H
X
H
X
$R_z(\frac{\Delta t}{4})$
$R_z(-\frac{\Delta t}{4})$
$R_z(-\frac{\Delta t}{4})$
$R_z(\frac{\Delta t}{4})$
$R_z(\frac{\Delta t}{4})$
$R_z(-\frac{\Delta t}{4})$
X
H
X

FIG. 14. Decompositions of unitary gates. (a) A decomposition for any two-qubit unitary gate in terms of single-qubit unitaries and three CNOT gates. (b) The two-qubit rotation gate $R_{zz}(\Delta t) = e^{-i\frac{\Delta t}{2}\hat{Z}\otimes\hat{Z}}$ can be implemented using only two CNOT gates, and a single-qubit rotation $R_z(\Delta t) = e^{-i\frac{\Delta t}{2}\hat{Z}}$. (c) The three-qubit gate $R_{zxz}(\Delta t) = e^{-i\frac{\Delta t}{2}\hat{Z}\otimes\hat{X}\otimes\hat{Z}}$ can be decomposed as four CNOT gates, an $R_z(\Delta t)$ gate and two Hadamard gates, denoted by H. (d) The PXP gate, $R_{pxp}(\Delta t) = e^{-i\frac{\Delta t}{2}\hat{P}\otimes\hat{X}\otimes\hat{P}}$, can be decomposed into eight CNOT gates, six $R_z(\pm\Delta t/4)$ rotation gates, two Hadamard gates and four Pauli-X gates, denoted by X. (e) A CNOT gate which acts on two next-nearest-neighboring qubits can be decomposed into 4 CNOTs which act only on neighboring qubits.

1. The number of CNOT layers required for brickwall circuits and trotterization

We can now calculate the number of CNOT layers required to implement both the BW circuits and trotterized circuits for each of the models considered in Sec. IV. BW circuits require three CNOT layers per BW layer. For the CI model, a BW circuit with depths $M = 3$ and $M = 4$ both give a second-order scaling, requiring 9 and 12 CNOT layers. respectively. First-order trotterization for the CI model requires two layers of $R_{zz}(\Delta t)$ gates and three layers of $R_{zzz}(\Delta t)$ gates, with each layer requiring two and four CNOTs respectively. This gives a total of 16 CNOT layers for the whole circuit. Second-order trotterization requires four layers of $R_{zz}(\Delta t)$ gates and five layers of $R_{zzz}(\Delta t)$ gates, giving a total of 28 CNOT layers.

For the PXP model, first- and second-order scaling are achieved with circuit depths $M = 4$ and $M = 5$ respectively for the BW circuit. These require a total of 12 and 15 CNOT layers. First-order trotterization requires three layers of $R_{pxp}(\Delta t)$ gates, each of which can be implemented using fourteen layers of *neighbouring* CNOTs, giving a total of 42 CNOT layers. Similarly, second-order trotterization requires five layers of $R_{pxp}(\Delta t)$ gates, giving a total of 70 CNOT layers. If next-nearest-neighbors connections are available, then this can be reduced to 24 and 40 CNOT layers, respectively.

BW circuits with depths $M = 5$ or $M = 6$ are required to yield a second-order scaling for the NNNI model, which can be implemented with a total of 15 and 18 CNOT layers, respectively. The optimal second-order trotterization is shown in Fig. 9(a). Note that while the structure of the circuit looks similar to that of the BW circuit, the two-qubit entangling gates here are exclusively SWAP gates (which require three CNOTs) and $R_{zz}(\Delta t)$ gates. $R_{zz}(\Delta t)$ can be implemented using only two CNOTs, compared to the more general two-qubit gate which requires three CNOTs. This gives an improvement, only requiring a total of 13 CNOT layers if $g_{zz} = 0$, or 15 CNOT layers if $g_{zz} \neq 0$.

APPENDIX C: NATIVE AND ALGORITHMIC ERRORS FOR SIMULATING DYNAMICS

In this Appendix, we provide further details for the discussion in Sec. V. We first prove that the infidelity of a unitary approximation is bounded by its operator norm error. We then show how the infidelity averaged over all initial basis states can be calculated using TNs, and how this error propagates after multiple applications of the unitary approximation. Finally, we use our results to determine optimal time steps for simulating dynamics.

1. Error bounds on the infidelity

Given the exact unitary operator, $U(\Delta t) = e^{-i\Delta t\hat{H}}$ and some unitary approximation, $V(\Delta t)$, we want to find an upper bound on the infidelity

$$I_\psi = 1 - |\langle\psi|U^\dagger(\Delta t)^n V(\Delta t)^n|\psi\rangle|^2 \tag{C1}$$

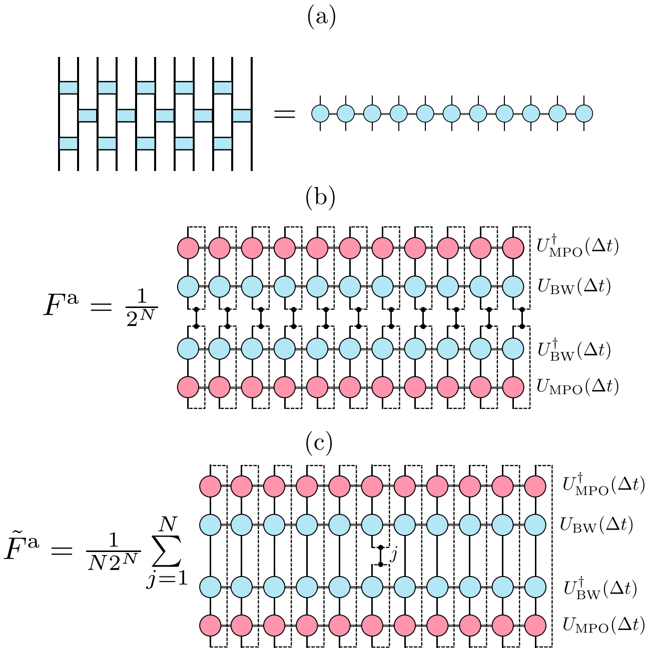

(a)
=
(b)
$F^{\mathrm{a}} = \frac{1}{2^N}$
$U^\dagger_{\mathrm{MPO}}(\Delta t)$
$U_{\mathrm{BW}}(\Delta t)$
$U^\dagger_{\mathrm{BW}}(\Delta t)$
$U_{\mathrm{MPO}}(\Delta t)$
(c)
$\tilde{F}^{\mathrm{a}} = \frac{1}{N2^N}\sum_{j=1}^{N}$
j
$U^\dagger_{\mathrm{MPO}}(\Delta t)$
$U_{\mathrm{BW}}(\Delta t)$
$U^\dagger_{\mathrm{BW}}(\Delta t)$
$U_{\mathrm{MPO}}(\Delta t)$

FIG. 15. Fidelity calculations as TNs. (a) A BW circuit with circuit depth M can be written as an MPO with bond dimension $\chi = 2^{\lceil M/2 \rceil}$. (b) The fidelity Eq. (23) can be calculated as a TN; the (almost) exact time propagator, and its adjoint, are represented by an MPO with the pink circles. The BW circuit is an MPO with blue circles. The black dots are three-point delta functions. (b) The local fidelity at site j is calculated in the same way, but the delta functions are only placed at site j (with a sum over j).

for some normalized wavefunction $|\psi\rangle$. We first assume that the error of the approximation is $||U(\Delta t) - V(\Delta t)|| = \mathcal{E}(\Delta t)$, where $\| \cdot \|$ is the operator norm of a matrix. This implies that $\|[U(\Delta t) - V(\Delta t)]\,|\psi\rangle\,\| \leqslant \mathcal{E}(\Delta t) \|\,|\psi\rangle\,\|$ for all $|\psi\rangle$, where $\|\,|\psi\rangle\,\| = \langle\psi|\psi\rangle$ is the Euclidean vector norm. We can then bound the error for the product formula,

$$\begin{aligned}
&\|U(\Delta t)^n - V(\Delta t)^n\| \\
&\quad= \left\| \sum_{m=0}^{n-1} U(\Delta t)^m [U(\Delta t) - V(\Delta t)] V(\Delta t)^{n-m-1} \right\| \\
&\quad\leqslant \sum_{m=0}^{n-1} \|U(\Delta t)^m [U(\Delta t) - V(\Delta t)] V(\Delta t)^{n-m-1}\| \\
&\quad\leqslant \sum_{m=0}^{n-1} \|U(\Delta t)^m|| \, \|U(\Delta t) - V(\Delta t)\| \, \|V(\Delta t)^{n-m-1}\| \\
&\quad= \sum_{m=0}^{n-1} \|U(\Delta t) - V(\Delta t)\| = n\mathcal{E}(\Delta t).
\end{aligned} \tag{C2}$$

The first equality can be shown by induction, and the inequalities follow from the properties of the operator norm. It then follows that

$$\begin{aligned}
&\mathrm{Re}(\langle\psi|U^\dagger(\Delta t)^n V(\Delta t)^n|\psi\rangle) \\
&\quad= 1 - \frac{1}{2}||[U(\Delta t)^n - V(\Delta t)^n]\,|\psi\rangle\,||^2 \geqslant 1 - \frac{1}{2}n^2\mathcal{E}(\Delta t)^2.
\end{aligned} \tag{C3}$$

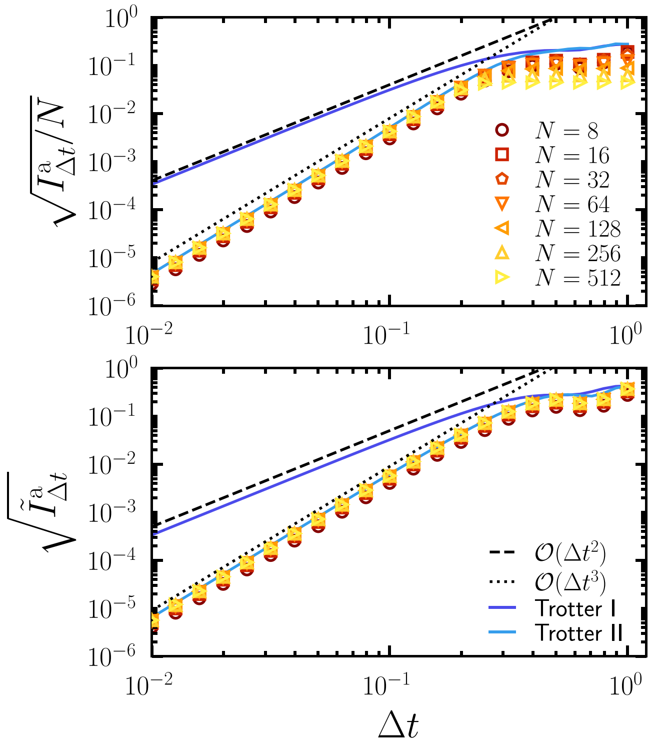


FIG. 16. The algorithmic infidelity. The square root of the algorithmic infidelity scaled by system size, $\sqrt{I^{\mathrm{a}}_{\Delta t}/N}$, (top panel) and the square root of the algorithmic local infidelity, $\sqrt{\tilde{I}^{\mathrm{a}}_{\Delta t}}$ (bottom panel), as functions of the time step, Δt. Results are for the CI model with $g = -0.75$ and circuit depth $M = 3$. The data points and curves are as defined in Fig. 7.

This allows us to bound the infidelity by the error of the unitary approximation,

$$I_\psi \leqslant 1 - \left(1 - \frac{1}{2}n^2\mathcal{E}(\Delta t)^2\right)^2 \leqslant n^2\mathcal{E}(\Delta t)^2. \tag{C4}$$

Notice that this holds for all normalized $|\psi\rangle$.

2. Estimating the state-averaged infidelity using TNs

The algorithmic fidelity Eq. (23) and the local fidelity Eq. (24) can be well estimated using the MPO approximation $U(\Delta t) \approx U_{\mathrm{MPO}}(\Delta t)$ and the BW circuit $U_{\mathrm{BW}}(\Delta t)$. It is first convenient to write the BW circuit as an MPO, which can be achieved exactly using bond dimension $\chi = 2^{\lceil M/2 \rceil}$ (although in practice it might be advantageous to compress this into an MPO with smaller bond dimension). This is shown in Fig. 15(a). To calculate the fidelity, we can write

$$\begin{aligned}
F^{\mathrm{a}}_{\Delta t} &= \frac{1}{2^N} \sum_{\sigma_1\cdots\sigma_N} \sum_{\sigma'_1\cdots\sigma'_N} \langle\sigma_1\cdots\sigma_N|U^\dagger_{\mathrm{MPO}}(\Delta t) U_{\mathrm{BW}}(\Delta t)|\sigma'_1\cdots\sigma'_N\rangle \\
&\quad\times \sum_{\phi_1\cdots\phi_N} \sum_{\phi'_1\cdots\phi'_N} \langle\phi_1\cdots\phi_N|U^\dagger_{\mathrm{BW}}(\Delta t) U_{\mathrm{MPO}}(\Delta t)|\phi'_1\cdots\phi'_N\rangle \\
&\quad\times \prod_{j=1}^{N} \sum_{k_j} \delta_{\sigma_j,\sigma'_j,k_j} \delta_{\phi_j,\phi'_j,k_j},
\end{aligned} \tag{C5}$$

where $\delta_{\sigma_j,\sigma_j',k_j}$ is the three-point delta functions for spins σ_j, σ_j' and k_j. The delta functions enforce the condition $\sigma_j = \sigma_j' = \phi_j = \phi_j'$. Despite the fact that Eq. (C5) appears more complicated than Eq. (23), it is very convenient to write the fidelity in this way if we want to calculate it as a TN. The TN representation is shown in Fig. 15(b), where the black dots represent the three-point delta functions.

The algorithmic local fidelity can be calculated similarly. Here, we write

$$\tilde{F}^{\mathrm{a}}_{\Delta t} = \frac{1}{N2^N}\sum_{j=1}^{N}\sum_{\sigma_1\cdots\sigma_N}\sum_{\sigma_1'\cdots\sigma_N'}\langle\sigma_1\cdots\sigma_j\cdots\sigma_N|U^{\dagger}_{\mathrm{MPO}}(\Delta t)U_{\mathrm{BW}}(\Delta t)|\sigma_1'\cdots\sigma_j'\cdots\sigma_N'\rangle\sum_{\phi_j,\phi_j'}\langle\sigma_1'\cdots\phi_j\cdots\sigma_N'|U^{\dagger}_{\mathrm{BW}}(\Delta t)U_{\mathrm{MPO}}(\Delta t)|$$
$$\times\ \sigma_1\cdots\phi_j'\cdots\sigma_N\rangle\sum_{k_j}\delta_{\sigma_j,\sigma_j',k_j}\delta_{\phi_j,\phi_j',k_j}. \tag{C6}$$

Notice that we only need to introduce the delta functions at spin j. This is shown as a TN in Fig. 15(c).

Figure 16 shows the square root of infidelity and local infidelity for optimized BW circuits with depth $M = 3$, calculated using Eqs. (C5) and (C6). The plot for the fidelity is scaled by system size. The infidelities are obtained using $I^{\mathrm{a}}_{\Delta t} = 1 - F^{\mathrm{a}}_{\Delta t}$ and $\tilde{I}^{\mathrm{a}}_{\Delta t} = 1 - \tilde{F}^{\mathrm{a}}_{\Delta t}$. As was the case for the error density, both of these quantities scale as $\mathcal{O}(\Delta t^3)$ respectively (dotted line), and outperform the results of both first- and second-order trotterization (solid lines).

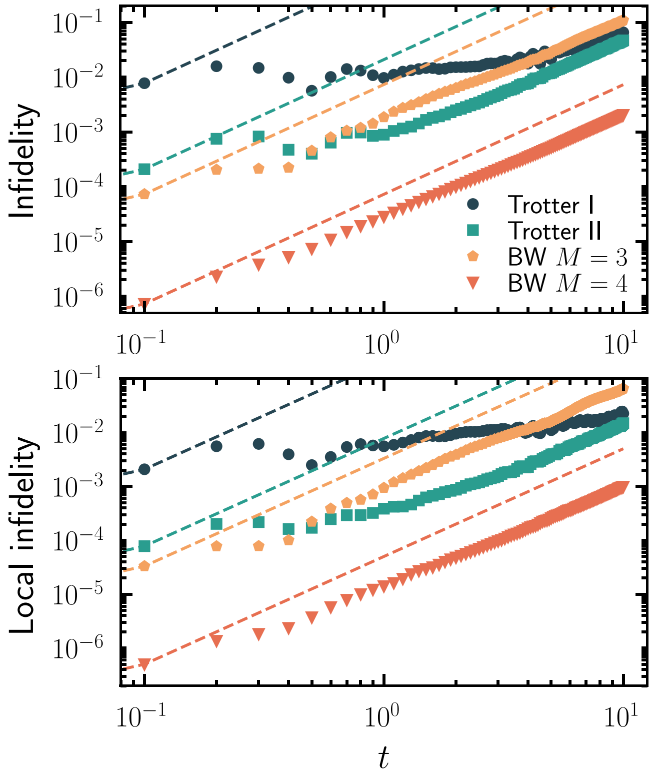


FIG. 17. The propagated algorithmic infidelity. The (top) infidelity and (bottom) local infidelity of $U_{\mathrm{BW}}(\Delta t)^n$ when compared to the exact time evolution operator $U_{\mathrm{BW}}(n\Delta t)$ as a function of $t = n\Delta t$. The results are shown for the CI model with $g = -0.75$ and $N = 8$, with time step $\Delta t = 0.1$. Results for both BW circuits and trotterization are shown. The dashed lines show the estimate $I^{\mathrm{a}}_{n\Delta t} \approx n^2 I^{\mathrm{a}}_{\Delta t}$.

3. State-averaged infidelity of the product formula

We now comment on the propagation of algorithmic errors when using the product formula $U(t = n\Delta t) \approx U_{\mathrm{BW}}(\Delta t)^n$. In the main text, the estimation of the infidelity $I^{\mathrm{a}}_{n\Delta t} \approx n^2 I^{\mathrm{a}}_{\Delta t}$ was motivated by the bound from the error. For small system sizes, we are able to calculate both $U(n\Delta t)$ and $U_{\mathrm{BW}}(\Delta t)^n$ exactly and decompose them as MPOs. This allows us to use

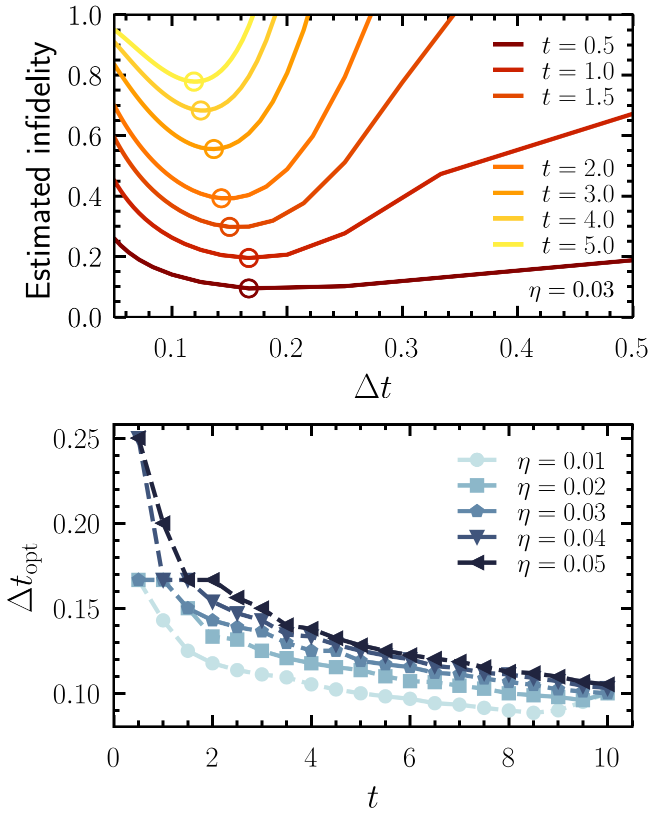


FIG. 18. Predicting an optimal time step. The algorithmic infidelity and the error from the device can be used to predicted an optimal time step. (a) The estimated infidelity using Eq. (26) as a function of time step Δt for the CI model with $g = -0.5$, $N = 5$, and $M = 3$. Each curve shows the estimated different total simulated time, t. The markers indicate the optimal time step Δt_{opt}. The curves are shown for fictional $\eta = 0.03$. (b) The optimal time step Δt_{opt} as a function of t for various fictional η.

Eqs. (C5) and (C6) and thus calculate the infidelity using the product formulas.

Figure 17 shows the results for the CI model with $g = -0.75$ and $N = 8$ with time step $\Delta t = 0.1$. We show both BW circuits with depths $M = 3$ and $M = 4$, and also first- and second-order trotterization. The dashed lines show the estimate. Notice that at early times, both BW circuit depths outperform both orders of trotterization. However, this is superseded at later times, when both first- and second-order trotterization perform better than the BW with depth $M = 3$. While this is surprising, it is consistent with recent results [90] which have shown that the time-propagated Trotter error can have stricter bounds than originally expected [14], albeit for systems with nearest-neighbor interactions. It is important to note that, however, the BW circuit with depth $M = 3$ is still subject to significantly less noise than both first- and second-order trotterization due to its shallower quantum circuit implementation. The same is true for depth $M = 4$, which has a smaller time-propagated infidelity than both orders of trotterization. In each case, the estimate over-estimates the true error. Nevertheless, it serves as a useful estimate for both BW circuits.

4. Predicting an optimal time step

We now demonstrate how the algorithmic infidelity and the infidelity measured from the quantum device can be used to predict an optimal time step to minimize the total error. Note that, unfortunately, the errors measured from the `ibm_brisbane` device are too large to do our analysis. We instead use Eq. (26) with some fictional choices of η.

Figure 18(a) shows Eq. (26) as a function of time step, Δt, for various total simulation times, t, and $\eta = 0.03$. Notice that for small time steps Δt, the estimated infidelity is large due to the number of applications of BW circuit that is needed. On the contrary, for large Δt, the error is large because the BW approximation has substantial errors. The optimal time steps, shown by the circle markers, is the choice which balances the two errors. This is shown in Fig. 18(b) as a function of t for multiple η. Notice that the optimal time step decreases with t. While we have also observed this is the case for other choices of g, the exact details might depend on the problem.

[1] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Comput. **26**, 1484 (1997).

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

[3] I. Kassal, J. D. Whitfield, A. Perdomo-Ortiz, M.-H. Yung, and A. Aspuru-Guzik, Simulating chemistry using quantum computers, Annu. Rev. Phys. Chem. **62**, 185 (2011).

[4] P. J. J. O'Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, E. Jeffrey, E. Lucero, A. Megrant, J. Y. Mutus, M. Neeley, C. Neill, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. C. White, P. V. Coveney, P. J. Love, H. Neven, A. Aspuru-Guzik, and J. M. Martinis, Scalable quantum simulation of molecular energies, Phys. Rev. X **6**, 031007 (2016).

[5] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, Nature (London) **549**, 242 (2017).

[6] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. D. Sawaya, S. Sim, L. Veis, and A. Aspuru-Guzik, Quantum chemistry in the age of quantum computing, Chem. Rev. **119**, 10856 (2019).

[7] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, Quantum computational chemistry, Rev. Mod. Phys. **92**, 015003 (2020).

[8] F. Arute *et al.* (Google AI Quantum and Collaborators), Hartree-fock on a superconducting qubit quantum computer, Science **369**, 1084 (2020).

[9] B. Bauer, S. Bravyi, M. Motta, and G. K.-L. Chan, Quantum algorithms for quantum chemistry and quantum materials science, Chem. Rev. **120**, 12685 (2020).

[10] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature (London) **549**, 195 (2017).

[11] D. Peral-García, J. Cruz-Benito, and F. J. García-Peñalvo, Systematic literature review: Quantum machine learning and its applications, Comput. Sci. Rev. **51**, 100619 (2024).

[12] A. Zeguendry, Z. Jarir, and M. Quafafou, Quantum machine learning: A review and case studies, Entropy **25**, 287 (2023).

[13] R. P. Feynman, Simulating physics with computers, Int. J. Theor. Phys. **21**, 467 (1982).

[14] S. Lloyd, Universal quantum simulators, Science **273**, 1073 (1996).

[15] C. Zalka, Simulating quantum systems on a quantum computer, Proc. R. Soc. London A **454**, 313 (1998).

[16] J. Preskill, Quantum Computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[17] J. M. Zhang and R. X. Dong, Exact diagonalization: The Bose–Hubbard model as an example, Eur. J. Phys. **31**, 591 (2010).

[18] J.-H. Jung and J. D. Noh, Guide to exact diagonalization study of quantum thermalization, J. Korean Phys. Soc. **76**, 670 (2020).

[19] F. Verstraete and J. I. Cirac, Matrix product states represent ground states faithfully, Phys. Rev. B **73**, 094423 (2006).

[20] F. Verstraete, M. M. Wolf, D. Perez-Garcia, and J. I. Cirac, Criticality, the area law, and the computational power of projected entangled pair states, Phys. Rev. Lett. **96**, 220601 (2006).

[21] M. B. Hastings, An area law for one-dimensional quantum systems, J. Stat. Mech.: Theor. Exp. (2007) P08024.

[22] J. Eisert, M. Cramer, and M. B. Plenio, Colloquium: Area laws for the entanglement entropy, Rev. Mod. Phys. **82**, 277 (2010).

[23] A. M. Dalzell and F. G. S. L. Brandão, Locally accurate MPS approximations for ground states of one-dimensional gapped local hamiltonians, Quantum **3**, 187 (2019).

[24] B. Bertini, P. Kos, and T. Prosen, Exact spectral form factor in a minimal model of many-body quantum chaos, Phys. Rev. Lett. **121**, 264101 (2018).

[25] B. Skinner, J. Ruhman, and A. Nahum, Measurement-induced phase transitions in the dynamics of entanglement, Phys. Rev. X **9**, 031009 (2019).

[26] L. Piroli, B. Bertini, J. I. Cirac, and T. Prosen, Exact dynamics in dual-unitary quantum circuits, Phys. Rev. B **101**, 094304 (2020).

[27] B. Buča, K. Klobas, and T. Prosen, Rule 54: exactly solvable model of nonequilibrium statistical mechanics, J. Stat. Mech. (2021) 074001.

[28] L. Zadnik and M. Fagotti, The folded Spin-1/2 XXZ Model: I. diagonalisation, jamming, and ground state properties, SciPost Phys. Core **4**, 010 (2021).

[29] L. Zadnik, K. Bidzhiev, and M. Fagotti, The folded spin-1/2 XXZ model: II. Thermodynamics and hydrodynamics with a minimal set of charges, SciPost Phys. **10**, 099 (2021).

[30] B. Bertini, P. Kos, and T. Prosen, Exact spectral statistics in strongly localized circuits, Phys. Rev. B **105**, 165142 (2022).

[31] F. Fritzsch, R. Ghosh, and T. Prosen, Boundary chaos: Exact entanglement dynamics, SciPost Phys. **15**, 092 (2023).

[32] B. Bertini, C. D. Fazio, J. P. Garrahan, and K. Klobas, Exact quench dynamics of the floquet quantum east model at the deterministic point, Phys. Rev. Lett. **132**, 120402 (2024).

[33] F. Tacchino, A. Chiesa, S. Carretta, and D. Gerace, Quantum computers as universal quantum simulators: State-of-the-art and perspectives, Adv. Quantum Tech. **3**, 1900052 (2020).

[34] L. D'Alessio, Y. Kafri, A. Polkovnikov, and M. Rigol, From quantum chaos and eigenstate thermalization to statistical mechanics and thermodynamics, Adv. Phys. **65**, 239 (2016).

[35] J. M. Deutsch, Eigenstate thermalization hypothesis, Rep. Prog. Phys. **81**, 082001 (2018).

[36] T. Mori, T. N. Ikeda, E. Kaminishi, and M. Ueda, Thermalization and prethermalization in isolated quantum systems: a theoretical overview, J. Phys. B **51**, 112001 (2018).

[37] J. Preskill, Quantum computing and the entanglement frontier, arXiv:1203.5813.

[38] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell *et al.*, Quantum supremacy using a programmable superconducting processor, Nature (London) **574**, 505 (2019).

[39] H. F. Trotter, On the product of semi-groups of operators, Proc. Am. Math. Soc. **10**, 545 (1959).

[40] M. Suzuki, Decomposition formulas of exponential operators and lie exponentials with some applications to quantum mechanics and statistical physics, J. Math. Phys. **26**, 601 (1985).

[41] A. A. Zhukov, S. V. Remizov, W. V. Pogosov, and Y. E. Lozovik, Algorithmic simulation of far-from-equilibrium dynamics using quantum computer, Quantum Inf. Processing **17**, 223 (2018).

[42] H. Lamm and S. Lawrence, Simulation of nonequilibrium dynamics on a quantum computer, Phys. Rev. Lett. **121**, 170501 (2018).

[43] A. Smith, M. S. Kim, F. Pollmann, and J. Knolle, Simulating quantum many-body dynamics on a current digital quantum computer, npj Quantum Inf. **5**, 106 (2019).

[44] J. Haah, M. B. Hastings, R. Kothari, and G. H. Low, Quantum algorithm for simulating real time evolution of lattice hamiltonians, SIAM J. Comput. **52**, FOCS18 (2023).

[45] A. M. Childs and Y. Su, Nearly optimal lattice simulation by product formulas, Phys. Rev. Lett. **123**, 050503 (2019).

[46] S.-H. Lin, R. Dilip, A. G. Green, A. Smith, and F. Pollmann, Real- and imaginary-time evolution with compressed quantum circuits, PRX Quantum **2**, 010342 (2021).

[47] M. Benedetti, M. Fiorentini, and M. Lubasch, Hardware-efficient variational quantum algorithms for time evolution, Phys. Rev. Res. **3**, 033083 (2021).

[48] N. Astrakhantsev, S.-H. Lin, F. Pollmann, and A. Smith, Time evolution of uniform sequential circuits, Phys. Rev. Res. **5**, 033187 (2023).

[49] R. Mansuroglu, T. Eckstein, L. Nützel, S. A. Wilkinson, and M. J. Hartmann, Variational hamiltonian simulation for translational invariant systems via classical pre-processing, Quantum Sci. Technol. **8**, 025006 (2023).

[50] A. Kotil, R. Banerjee, Q. Huang, and C. B. Mendl, Riemannian quantum circuit optimization for hamiltonian simulation, J. Phys. A: Math. Theor. **57**, 135303 (2024).

[51] C. Mc Keever and M. Lubasch, Classically optimized hamiltonian simulation, Phys. Rev. Res. **5**, 023146 (2023).

[52] M. S. J. Tepaske, D. Hahn, and D. J. Luitz, Optimal compression of quantum many-body time evolution operators into brickwall circuits, SciPost Phys. **14**, 073 (2023).

[53] I. P. McCulloch, Infinite size density matrix renormalization group, revisited, arXiv:0804.2509.

[54] G. K.-L. Chan, A. Keselman, N. Nakatani, Z. Li, and S. R. White, Matrix product operators, matrix product states, and ab initio density matrix renormalization group algorithms, J. Chem. Phys. **145**, 014102 (2016).

[55] C. Hubig, I. P. McCulloch, and U. Schollwöck, Generic construction of efficient matrix product operators, Phys. Rev. B **95**, 035129 (2017).

[56] G. Evenbly and G. Vidal, Algorithms for entanglement renormalization, Phys. Rev. B **79**, 144108 (2009).

[57] S. Michalakis, Stability of the area law for the entropy of entanglement, arXiv:1206.6900.

[58] K. Van Acoleyen, M. Mariën, and F. Verstraete, Entanglement rates and area laws, Phys. Rev. Lett. **111**, 170501 (2013).

[59] L. Colmenarez and D. J. Luitz, Lieb-robinson bounds and out-of-time order correlators in a long-range spin chain, Phys. Rev. Res. **2**, 043047 (2020).

[60] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10th ed. (Cambridge University Press, USA, 2011).

[61] S. Zhuk, N. Robertson, and S. Bravyi, Trotter error bounds and dynamic multi-product formulas for hamiltonian simulation, arXiv:2306.12569.

[62] E. H. Lieb and D. W. Robinson, The finite group velocity of quantum spin systems, Commun. Math. Phys. **28**, 251 (1972).

[63] B. Bertini, P. Kos, and T. Prosen, Operator entanglement in local quantum circuits I: Chaotic Dual-Unitary circuits, SciPost Phys. **8**, 067 (2020).

[64] S. R. White, Density matrix formulation for quantum renormalization groups, Phys. Rev. Lett. **69**, 2863 (1992).

[65] S. Rommer and S. Östlund, Class of ansatz wave functions for one-dimensional spin systems and their relation to the density matrix renormalization group, Phys. Rev. B **55**, 2164 (1997).

[66] U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, Ann. Phys. **326**, 96 (2011), January 2011 Special Issue.

[67] F. Verstraete, J. J. García-Ripoll, and J. I. Cirac, Matrix product density operators: Simulation of finite-temperature and dissipative systems, Phys. Rev. Lett. **93**, 207204 (2004).

[68] J. Haegeman, J. I. Cirac, T. J. Osborne, I. Pižorn, H. Verschelde, and F. Verstraete, Time-dependent variational principle for quantum lattices, Phys. Rev. Lett. **107**, 070601 (2011).

[69] M. P. Zaletel, R. S. K. Mong, C. Karrasch, J. E. Moore, and F. Pollmann, Time-evolving a matrix product state with long-ranged interactions, Phys. Rev. B **91**, 165112 (2015).

[70] M. V. Damme, J. Haegeman, I. McCulloch, and L. Vanderstraeten, Efficient higher-order matrix product operators for time evolution, arXiv:2302.14181.

[71] By this, we mean that the MPO has an accuracy much greater than that which can be achieved by the BW ansatz we are optimising.

[72] See Ref. [66] for a definition of the truncation error.

[73] In practice, we expect that $\mathrm{Tr}[U_{\mathrm{MPO}}^{\dagger}(\Delta t)U_{\mathrm{BW}}(\Delta t)] = (2 - \delta(\Delta t)^2)^N$; Eq. (14) follows directly from this.

[74] R. Verresen, R. Moessner, and F. Pollmann, One-dimensional symmetry protected topological phases and their transitions, Phys. Rev. B **96**, 165124 (2017).

[75] R. Verresen, N. G. Jones, and F. Pollmann, Topology and edge modes in quantum critical chains, Phys. Rev. Lett. **120**, 057001 (2018).

[76] A. Smith, B. Jobst, A. G. Green, and F. Pollmann, Crossing a topological phase transition with a quantum computer, Phys. Rev. Res. **4**, L022020 (2022).

[77] P. Fendley, K. Sengupta, and S. Sachdev, Competing density-wave orders in a one-dimensional hard-boson model, Phys. Rev. B **69**, 075106 (2004).

[78] I. Lesanovsky, Many-body spin interactions and the ground state of a dense Rydberg lattice gas, Phys. Rev. Lett. **106**, 025301 (2011).

[79] C. J. Turner, A. A. Michailidis, D. A. Abanin, M. Serbyn, and Z. Papić, Weak ergodicity breaking from quantum many-body scars, Nat. Phys. **14**, 745 (2018).

[80] A. Browaeys and T. Lahaye, Many-body physics with individually controlled Rydberg atoms, Nat. Phys. **16**, 132 (2020).

[81] C. J. Turner, A. A. Michailidis, D. A. Abanin, M. Serbyn, and Z. Papić, Quantum scarred eigenstates in a Rydberg atom chain: Entanglement, breakdown of thermalization, and stability to perturbations, Phys. Rev. B **98**, 155134 (2018).

[82] K. W. Kuper, J. P. Pajaud, K. Chinni, P. M. Poggi, and P. S. Jessen, Trotter errors and the emergence of chaos in quantum simulation, arXiv:2212.03843.

[83] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, and J. Tennyson, The variational quantum eigensolver: A review of methods and best practices, Phys. Rep. **986**, 1 (2022).

[84] The fidelities are normalised such that the value of the fidelity is one initially, and is zero after all memory of the initial conditions is lost (and the state of the system can be assumed to be random).

[85] C. Yi and E. Crosson, Spectral analysis of product formulas for quantum simulation, npj Quantum Inf. **8**, 37 (2022).

[86] R. Mansuroglu, F. Fischer, and M. J. Hartmann, Problem-specific classical optimization of hamiltonian simulation, Phys. Rev. Res. **5**, 043035 (2023).

[87] Q. Zhao, Y. Zhou, A. F. Shaw, T. Li, and A. M. Childs, Hamiltonian simulation with random inputs, Phys. Rev. Lett. **129**, 270502 (2022).

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

[89] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu, Theory of trotter error with commutator scaling, Phys. Rev. X **11**, 011020 (2021).

[90] D. Layden, First-order trotter error from a second-order perspective, Phys. Rev. Lett. **128**, 210501 (2022).

[91] H. Zhao, M. Bukov, M. Heyl, and R. Moessner, Making trotterization adaptive and energy-self-correcting for NISQ devices and beyond, PRX Quantum **4**, 030319 (2023).

[92] T. N. Ikeda, H. Kono, and K. Fujii, Measuring Trotter error and its application to precision-guaranteed Hamiltonian simulations, arXiv:2307.05406.

[93] F. Verstraete and J. I. Cirac, Renormalization algorithms for quantum-many body systems in two and higher dimensions, arXiv:cond-mat/0407066.

[94] P. Corboz, R. Orús, B. Bauer, and G. Vidal, Simulation of strongly correlated fermions in two spatial dimensions with fermionic projected entangled-pair states, Phys. Rev. B **81**, 165104 (2010).

[95] M. Lubasch, J. I. Cirac, and M.-C. Bañuls, Algorithms for finite projected entangled pair states, Phys. Rev. B **90**, 064425 (2014).

[96] H.-J. Liao, J.-G. Liu, L. Wang, and T. Xiang, Differentiable programming tensor networks, Phys. Rev. X **9**, 031041 (2019).

[97] M. P. Zaletel and F. Pollmann, Isometric tensor network states in two dimensions, Phys. Rev. Lett. **124**, 037201 (2020).

[98] S.-H. Lin, M. P. Zaletel, and F. Pollmann, Efficient simulation of dynamics in two-dimensional quantum spin systems with isometric tensor networks, Phys. Rev. B **106**, 245102 (2022).

[99] C. Geng, H.-Y. Hu, and Y. Zou, Differentiable programming of isometric tensor networks, Mach. Learn.: Sci. Technol. **3**, 015020 (2022).

[100] L. Causer, lcauser/classically-optimised-circuits: v1: arXiv release (2023), 10.5281/zenodo.10359964.

[101] B. Kraus and J. I. Cirac, Optimal creation of entanglement using a two-qubit gate, Phys. Rev. A **63**, 062309 (2001).

[102] J. D. Whitfield, J. Biamonte, and A. Aspuru-Guzik, Simulation of electronic structure hamiltonians using quantum computers, Mol. Phys. **109**, 735 (2011).

[103] V. Shende, S. Bullock, and I. Markov, Synthesis of quantum-logic circuits, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **25**, 1000 (2006).