Lawrence Berkeley National Laboratory

Title

Algebraic compression of quantum circuits for Hamiltonian evolution

Permalink

https://escholarship.org/uc/item/80p908pr

Authors

Kökcü, Efekan
Camps, Daan
Oftelie, Lindsay Bassman
et al.

Algebraic Compression of Quantum Circuits for Hamiltonian Evolution

Efekan Kökcü,[1] Daan Camps,[2] Lindsay Bassman,[2] J. K. Freericks,[3]
Wibe A. de Jong,[2] Roel Van Beeumen,[2] and Alexander F. Kemper[1]

[1]*Department of Physics, North Carolina State University, Raleigh, North Carolina 27695, USA*
[2]*Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*
[3]*Department of Physics, Georgetown University, 37th and O Sts. NW, Washington, DC 20057 USA*
(Dated: March 29, 2022)

Unitary evolution under a time-dependent Hamiltonian is a key component of simulation on quantum hardware. Synthesizing the corresponding quantum circuit is typically done by breaking the evolution into small time steps, also known as Trotterization, which leads to circuits whose depth scales with the number of steps. When the circuit elements are limited to a subset of $SU(4)$ — or equivalently, when the Hamiltonian may be mapped onto free fermionic models — several identities exist that combine and simplify the circuit. Based on this, we present an algorithm that compresses the Trotter steps into a single block of quantum gates using algebraic relations between adjacent circuit elements. This results in a fixed depth time evolution for certain classes of Hamiltonians. We explicitly show how this algorithm works for several spin models, and demonstrate its use for adiabatic state preparation of the transverse field Ising model.

I. INTRODUCTION

Quantum computers were initially conceived of as a tool for simulating quantum systems [1], and indeed, this is seen as one of the most promising applications for near-term quantum processors [2]. The main challenge for implementing such simulations on a quantum computer is to generate a circuit—usually comprised of single- and two-qubit gates—that mimics the effect of the time evolution operator $U(t)$ on qubits, which is termed *unitary synthesis*. Unitary synthesis in the most generic case is exponentially hard due to exponential growth of the Hilbert space dimension, i.e. size of the matrix $U(t)$, with the system size [3, 4]. Though there are ways to ameliorate this problem by leveraging the algebra generated by the terms in the Hamiltonian [5], generating an exact circuit remains classically NP-hard for interacting fermion models. In practice, an approximate circuit implementation of the unitary is often sufficient and can even be imposed by technological constraints as we discuss next.

The challenges are exacerbated by the fact that near-term quantum computers are small and noisy [2], which greatly limits how deep quantum circuits can be before their results are indistinguishable from random noise. This means that circuit synthesis techniques must also attempt to optimize quantum circuits to be as shallow as possible; such circuit optimization is an NP-hard problem for general quantum circuits [6, 7].

Generating a circuit is particularly difficult for dynamic simulations which simulate a quantum system evolving in time. This is because a separate circuit must be generated for each time-step, and for a general Hamiltonian, the circuit depth increases with increasing simulation time [8–10].

Here, we present an algorithm for compressing circuits for simulations of time-evolution to a constant depth, independent of increasing simulation time, for a particular class of Hamiltonians. The common feature of these Hamiltonians is that they can be mapped to free fermionic Hamiltonians [11]; equivalently, they produce a Hamiltonian algebra that scales polynomially in the number of qubits [5]. This property allows us to circumvent the no-fast-forwarding theorem [8, 12, 13].

Our approach uses a first order Trotter-Suzuki decomposition of the time-evolution operator of a time-dependent Hamiltonian $\mathcal{H}(t)$ for the unitary synthesis step. The unitary operator $U(t)$ for this process obeys

$$\partial_t U(t) = -i\mathcal{H}(t)U(t) \qquad (1)$$

with $U(t = 0) = \mathcal{I}$, i.e. the identity operator. This equation can be approximately solved by Trotter decomposition with a time step size Δt:

$$U(t) = \lim_{\Delta t \to 0} e^{-i\Delta t \mathcal{H}(t_N)} \cdots e^{-i\Delta t \mathcal{H}(t_2)} e^{-i\Delta t \mathcal{H}(t_1)}, \quad (2)$$

where $t_n = (n-1)\Delta t$, and $t_N = t$ (kept constant while taking the limit). After approximating each small time step $e^{-i\Delta t \mathcal{H}(t_i)}$, the right hand side of Eq. (2) often has a straightforward circuit implementation that approximates $U(t)$. The accuracy of the Trotter decomposition depends on the size of Δt. This is the trade-off with the Trotter decomposition: the smaller the time step, the longer the circuit. Moreover, regardless of the choice of Δt, the circuit grows with increasing simulation time.

In order to deal with the circuits that grow as a function of simulation time, several approaches exist, each with inherent limitations. We limit the discussion to those methods that can be executed on Noisy Intermediate Scale Quantum (NISQ) hardware, putting aside the others [14–19]. General purpose transpilers can be used to shorten the growing-depth circuits, but these do not typically take advantage of the algebraic structure of the problem, and are limited to how much compression can be achieved, ultimately placing a limit on the largest feasible simulation time. Constant depth circuits can be obtained variationally by optimizing a generic circuit via a

hybrid classical-quantum algorithm [20]. However, these approximate circuits are only applicable to time evolution of a particular initial state and their error grows with simulation time. Alternatively, a more general constant depth circuit may be found by using the distance from a specific target unitary as the cost function [21] — this approach does not scale favorably in the number of qubits due to the necessarily large number of parameters in the classical optimization. Finally, constructive approaches to circuit synthesis that rely on the algebraic structure of the unitaries have been proposed [3, 5, 22], but these require a similar classical optimization step.

Our proposed method, while not applicable to all problems, is a constructive proof of the circuit ansatz used in Ref. [21]. Because of the constructive nature, it scales to 1,000s of qubits, which is out of the reach of previous methods [5, 21] that require optimization. This work is accompanied by a dual paper [23] which focuses on the underlying matrix structures of the circuit elements, and on the efficient and numerically stable implementation of the circuit compression algorithms for each model discussed below. We show that this compression technique scales polynomially in the number of qubits, and is thus applicable for the foreseeable future on quantum hardware. Our codes are made available as part of the fast free fermion compiler (F3C) [24, 25] on https://github.com/QuantumComputingLab. F3C is build based on the QCLAB toolbox [26, 27] for creating and representing quantum circuits.

We demonstrate the power of our compressed circuits by showing how they enable adiabatic state preparation (ASP), where a system is evolved under a slowly varying Hamiltonian in order to prepare a non-trivial ground state. The system is initialized in the ground state of the initial Hamiltonian, which is presumed to be trivial to prepare, and ends in the (non-trivial) ground state of the final Hamiltonian given slow enough variation of the Hamiltonian. Given the constant depth nature of the circuit for different times, the state preparation for long times, is ultimately accomplished in a single step. We apply this to preparing the ground state of the transverse field Ising model (TFIM) on 5 qubits.

The paper is organized as follows. We define an object called a "block" and prove the existence of a constructive compression method for circuits made only of blocks in Sec. II. In Sec. III, we propose specific mappings between the circuit elements of Trotter-decomposed time evolution operators and blocks for the Kitaev chain, the TFIM, the transverse field XY (TFXY) model, as well as a nearest neighbor free fermion model. We further prove that the three properties of blocks hold for each, thus showing that a compressed circuit exists for these models. We apply our method to ground state construction of the TFIM with open boundary conditions in Sec. IV, where we show that compressed Trotter circuits are superior to traditional Trotter circuits due to their ability to simulate as slow time evolution as desired, which is significant for adiabatic evolution. We finish by discussing

the potential implications of our results in a broad sense in Sec. V.

II. BLOCKS AND THE COMPRESSION THEOREM

In this section, we will define and prove generic mathematical tools necessary to show that certain models have a Trotter expansion that can be compressed down to a fixed depth circuit. To compress the Trotter expansion, we map certain elements in the Trotter step into a mathematical structure called a "block". These block structures are model dependent and are required to have certain simplification rules given in Def. 1. We follow with lemmas and theorems that use these simplification rules to compress a Trotter circuit that is fully expressed via blocks into a fixed depth circuit.

Let $B_i(\vec{\theta})$ be functions of $\vec{\theta}$, where the index can be any positive integer $i = 1, 2, 3, \ldots$ and $\vec{\theta}$ are multiple real numbers. Each $B_i(\vec{\theta})$ will be mapped to quantum circuit elements in a model dependent way such that the parameters $\vec{\theta}$ will be the parameters for the gates, and the index will separate them in terms of which qubits they act on. The number of parameters and index to qubit(s) mapping vary from model to model, as it will be seen in Sec. III. For example, we will have $B_i(\theta) \equiv \exp(-i\theta X_i X_{i+1})$ for odd i and $B_i(\theta) \equiv \exp(-i\theta Y_i Y_{i+1})$ for even i as the mapping for an open one-dimensional (1D) Kitaev chain in Sec. III A, and show that this mapping satisfies the three defining properties listed below.

Definition 1 (Block). *Define a "block" $B_i(\vec{\theta})$ as a structure that satisfies the following three properties:*

1. ***the fusion property:*** *for any set of parameters $\vec{\alpha}$ and $\vec{\beta}$, there exist $\vec{a}$ such that*

$$B_i(\vec{\alpha})\, B_i(\vec{\beta}) = B_i(\vec{a}), \tag{3}$$

2. ***the commutation property:*** *for any set of parameters $\vec{\alpha}$ and $\vec{\beta}$*

$$B_i(\vec{\alpha})\, B_j(\vec{\beta}) = B_j(\vec{\beta})\, B_i(\vec{\alpha}), \qquad |i - j| > 1, \tag{4}$$

3. ***the turnover property:*** *for any set of parameters $\vec{\alpha}$, $\vec{\beta}$ and $\vec{\gamma}$ there exist $\vec{a}$, $\vec{b}$ and $\vec{c}$ such that*

$$B_i(\vec{\alpha})\, B_{i+1}(\vec{\beta})\, B_i(\vec{\gamma}) = B_{i+1}(\vec{a})\, B_i(\vec{b})\, B_{i+1}(\vec{c}). \tag{5}$$

Henceforth, we drop the parameters $\vec{\theta}$ when referring to a block and simply write B_i for convenience. The reader should assume that each block in an expression may have different parameters unless it is stated otherwise.

Using Def. 1, we now define three important "block" structures, which will be used to obtain and prove the fixed depth circuits. We start with a cascade of blocks.
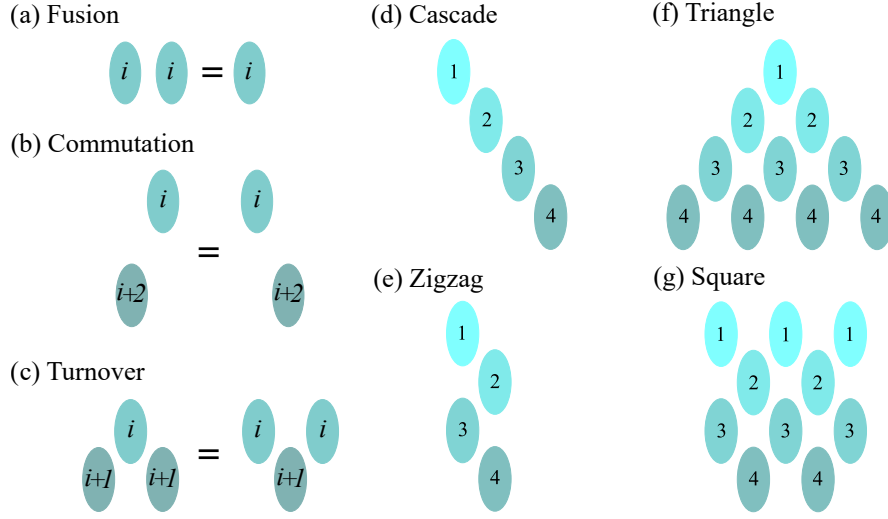
FIG. 1. Panels **(a)**, **(b)**, **(c)** illustrate the defining properties of a block given in Def. 1. Panels **(d)**, **(e)**, **(f)**, **(g)** illustrate the block structures cascade, zigzag and triangle defined in Defs. 1 to 3, and square defined in Thm. 2.

Definition 2 (Cascade). *Define a "cascade" of blocks to be*

$$C_{i,j} := \prod_{k=i}^{j} B_k = B_i \, B_{i+1} \, \cdots \, B_j, \qquad (6)$$

where B_i are blocks as defined in Def. 1 and $i < j$.

Lemma 1. *Let B_m be a block as defined in Def. 1 and $C_{j,n}$ a cascade of blocks as defined in Def. 2 with $j \leq m < n$. Then the transformation*

$$C_{j,n} \, B_m = B_{m+1} \, C_{j,n}, \qquad (7)$$

can be performed by one turnover operation.

Proof. Using Def. 2, we obtain

$$C_{j,n} B_m = \left(\prod_{k=j}^{n} B_k \right) B_m,$$

$$= \left(\prod_{k=j}^{m+1} B_k \right) B_m \left(\prod_{k=m+2}^{n} B_k \right),$$

$$= \left(\prod_{k=j}^{m-1} B_k \right) B_m \, B_{m+1} \, B_m \left(\prod_{k=m+2}^{n} B_k \right).$$

Next, applying the turnover property to $B_m \, B_{m+1} \, B_m$

and using the commutation property, yields

$$C_{j,n} B_m = \left(\prod_{k=j}^{m-1} B_k \right) B_{m+1} \, B_m \, B_{m+1} \left(\prod_{k=m+2}^{n} B_k \right),$$

$$= B_{m+1} \left(\prod_{k=j}^{m-1} B_k \right) \left(\prod_{k=m}^{n} B_k \right),$$

$$= B_{m+1} \left(\prod_{k=j}^{n} B_k \right),$$

$$= B_{m+1} \, C_{j,n},$$

which completes the proof. $\square$

Next, we define a triangle of blocks, comprised of multiple cascades, and prove that an extra block can be fully compressed into a triangle.

Definition 3 (Triangle). *Define a "triangle" of blocks as*

$$T_i := \prod_{k=0}^{i-1} C_{i-k,i} = C_{i,i} \, C_{i-1,i} \, \cdots \, C_{1,i}, \qquad (8)$$

where $C_{i,j}$ are cascades of blocks as defined in Def. 2.

Theorem 1. *Let B_m be a block as defined in Def. 1 and T_n a triangle of blocks as defined in Def. 3 with $m \leq n$. Then B_m can be fully merged into T_n, i.e.,*

$$T_n \, B_m = T_n, \qquad (9)$$

via $n - m$ turnover operations and 1 fusion operation.

Proof. Using Def. 3, we obtain

$$T_n B_m = \left(\prod_{k=0}^{n-1} C_{n-k,n} \right) B_m = C_{n,n} \, C_{n-1,n} \, \cdots \, C_{1,n} \, B_m.$$
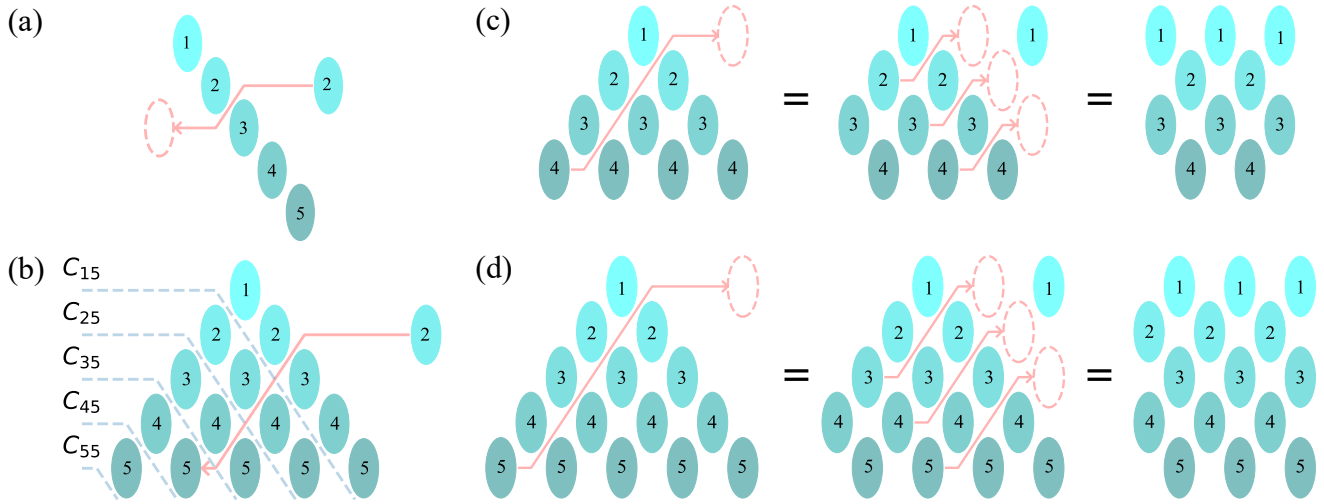
FIG. 2. Panel **(a)** illustrates Lemma 1. As it can be seen here, this lemma can be interpreted as a block shifting one position down while passing through a cascade from right to left. Panel **(b)** includes the diagrammatic representation of Thm. 1. Block B_2 on the right side of a triangle with height 5 i.e. T_5, finds its way all the way down to 5th level passing through cascades and finally merges with the block that is at the end of the path, and results with a triangle with height 5 made by blocks with different parameters at the end. Panels **(c)** and **(d)** illustrate the transformation described in Thm. 2 respectively for triangles with even and odd height. As described, one starts by pushing the leftmost cascade all the way to the right, and continue by pushing every other cascade until reaching to the longest cascade $C_{1,n}$. This results in the square circuits shown in the rightmost part of both panels.

Applying Lemma 1 $n - m$ times, yields

$$T_n B_m = C_{n,n} C_{n-1,n} \cdots C_{1,n} B_m,$$
$$= C_{n,n} C_{n-1,n} \cdots C_{2,n} B_{m+1} C_{1,n},$$
$$\vdots$$
$$= C_{n,n} C_{n-1,n} \cdots C_{n-m+1,n} B_n C_{n-m,n} \cdots C_{1,n},$$

where each time we swap a block with a cascade, we perform 1 turnover. Hence, we have done $n - m$ turnover operations in order to move the extra block B_m to B_n. Next, we can merge B_n into the triangle $C_{n,n} C_{n-1,n} \cdots C_{n-m+1,n}$, i.e.,

$$C_{n-m+1,n} B_n = \left(\prod_{k=n-m+1}^{n} B_k \right) B_n,$$
$$= \left(\prod_{k=n-m+1}^{n-1} B_k \right) B_n B_n,$$
$$= \left(\prod_{k=n-m+1}^{n-1} B_k \right) B_n = C_{n-m+1,n},$$

where we applied the fusion property once. Finally,

$$T_n B_m = C_{n,n} C_{n-1,n} \cdots C_{n-m+1,n} C_{n-m,n} \cdots C_{1,n},$$
$$= \prod_{k=0}^{n-1} C_{n-k,n} = T_n,$$

via $n - m$ turnover and 1 fusion operation. □

Thm. 1 shows that a triangle and any finite set of blocks can be merged into a triangle. Here we define another block structure, called zigzag of blocks, which corresponds to a time step of a Trotter decomposition of the Hamiltonian evolution after mapping to blocks. We use Thm. 1 to explicitly show that a zigzag can efficiently be compressed into a triangle of appropriate size. We also show that a triangle can be merged with another triangle efficiently, using the fact that a triangle is a finite set of blocks.

Definition 4 (Zigzag). *Define a "zigzag" of blocks as*

$$L_{i,j} := \left(\prod_{\substack{k=i \\ \text{odd}}}^{j} B_k \right) \left(\prod_{\substack{k=i \\ \text{even}}}^{j} B_k \right), \tag{10}$$

where B_k are blocks as defined in Def. 1 and $i < j$; note all even numbered blocks are to the right of odd numbered blocks.

Corollary 1. *Let $C_{1,n}$, T_n, and $L_{1,n}$ be a cascade, triangle, and zigzag of blocks as defined in Defs. 2 to 4, respectively. Then a triangle with height n and a cascade (zigzag) with height n can be merged into a triangle with height n, i.e.*

$$T_n C_{1,n} = T_n, \qquad T_n L_{1,n} = T_n, \tag{11}$$

with $n(n-1)/2$ turnover and n fusion operations.

Proof. In both cases, we are merging one block from every height $1, 2, 3, .., n$, therefore we use the result of Thm. 1

for each of these n blocks. Each block will require 1 fusion, therefore we need to apply n fusion operations. The number of turnover operations is

$$\sum_{k=1}^{n}(n-k) = \sum_{k=1}^{n-1} k = \frac{n(n-1)}{2},$$

which concludes the proof. □

Corollary 2. *Two triangles with height n can be merged into a triangle with height n, i.e.*

$$T_n T_n = T_n, \tag{12}$$

via $n(n^2-1)/6 = O(n^3)$ turnover and $n(n-1)/2 = O(n^2)$ fusion operations.

Proof. There are $n(n-1)/2$ blocks in T_n. As a result of Thm. 1 this merging will require $n(n-1)/2$ fusion operations. Using the definition of T_n, we see that we are effectively merging the triangle with cascades with heights 1 to n. Using the result of Cor. 1, we get the number of turnover operations required for this merger to be

$$\sum_{k=1}^{n} \frac{k(k-1)}{2} = \frac{n(n^2-1)}{6}, \tag{13}$$

which concludes the proof. □

Using Cors. 1 and 2 as induction steps, we provide the two following compression techniques for Trotter decompositions involving blocks only:

Corollary 3 (Compression for time-dependent Hamiltonians). *A product of r zigzags with height n for $r > 1$ will lead to a triangle with height n*

$$\prod_{k=1}^{r} L_{1,n} = T_n, \tag{14}$$

via $O(n^2 r)$ turnover and $O(nr)$ fusion operations.

Proof. A zigzag can be viewed as a triangle with identity operations in place of the missing blocks. Therefore for $r = 1$, the result should be a triangle. Using Cor. 1, we know that adding another zigzag to the triangle each time will require $O(n^2)$ turnover and $O(n)$ fusion operations. Doing this r times will require $O(n^2 r)$ turnover and $O(nr)$ fusion operations. □

Corollary 4 (Compression for time-independent Hamiltonians). *Consider r identical zigzags with height n. Using the fact that they are same, they can be compressed into a triangle with height n*

$$(L_{1,n})^r = T_n \tag{15}$$

where we use power instead of the product symbol to emphasize that the parameters are the same, via $O(n^3 \log_2 r)$ turnover and $O(n^2 \log_2 r)$ fusion operations.

Proof. Given that zigzags (with added identity gates) are triangles , the left hand side of (15) can also be written as $(T_n)^r$. We can replace it with $((T_n)^2)^{r/2}$ and turn it into $(T_n)^{r/2}$ via $O(n^2)$ fusion and $O(n^3)$ turnover operations using Cor. 2. We need to repeat this $O(\log_2 r)$ many steps in total, leading to this compression being performed via $O(n^3 \log_2 r)$ turnover and $O(n^2 \log_2 r)$ fusion operations. □

Cors. 3 and 4 are the main results used for compression. For a time-independent Hamiltonian, one can choose either of the two compression algorithms. The performance depends on the number of Trotter steps needed, and the number of qubits. Depending on the particular problem at hand, one or the other may be preferable; the $O(n^2 r)$ scaling of Cor. 3 is a better choice for larger systems, whereas Cor. 4 is a better choice for a large number of Trotter steps. For time-dependent Hamiltonians, each Trotter time step will have different coefficients, meaning the corresponding zigzags will not all be identical. In this case, Cor. 3 is the only choice for compression.

So far we have proved that we can compress any finite expansion given in terms of blocks into a triangle, a block structure with depth $O(n)$ where n is the height of the triangle. When translated into a quantum circuit, this compression corresponds to a compression of a long circuit into a short, fixed depth circuit. For NISQ devices, it is crucial to have short depth circuits due to the fact that the noise increases for longer circuit depths. In the rest of this section, we will show that the depth of the triangle can be reduced in half, leading to better quantum circuits for NISQ devices.

Lemma 2. *For $i < a \le b \le j$,*

$$C_{a,b} C_{i,j} = C_{i,j} C_{a-1,b-1}. \tag{16}$$

Proof. $C_{a,b} = B_a B_{a+1} \cdots B_b$ is formed by blocks with indices that are between i and j, hence each of them can be passed from left side of $C_{i,j}$ to right side by decrementing the index by one. □

Theorem 2. *A triangle with height n can be further optimized to a square with height n which is defined as*

$$S_n = \prod_{\substack{k=n-1\downarrow \\ k \text{ even}}}^{2} C_{k,n} \prod_{\substack{k=n\downarrow \\ k \text{ odd}}}^{1} C_{1,k} \tag{17}$$

for odd n and

$$S_n = \prod_{\substack{k=n-1\downarrow \\ k \text{ odd}}}^{1} C_{k,n} \prod_{\substack{k=n-1\downarrow \\ k \text{ odd}}}^{1} C_{1,k} \tag{18}$$

for even n, where down arrow $\downarrow$ indicates that the product is done in the decreasing order for index k. S_n has the same number of blocks as T_n, however it has two-thirds the depth of T_n which is important for quantum computers.

Proof. First, consider the case for odd n. The definition of a triangle is:

$$T_n = \prod_{k=0}^{n-1} C_{n-k,n} = C_{n,n}\, C_{n-1,n} \cdots C_{1,n} \qquad (19)$$

We will carry all $C_{\text{odd},n}$ cascades, i.e. $C_{n,n}$, $C_{n-2,n}, \ldots C_{5,n}$, $C_{3,n}$ to the right side of $C_{1,n}$ by using Lemma 2 in the given order. In that order, $C_{m,n}$ will pass through $m-1$ cascades and become $C_{1,n-m+1}$ at the end, thereby leaving all the $C_{\text{even},n}$ cascades behind. This leads to a square formation for odd n as defined in the theorem.

For even n, we will carry all $C_{\text{even},n}$ cascades, i.e. $C_{n,n}$, $C_{n-2,n}, \ldots, C_{4,n}$, $C_{2,n}$. In that order, $C_{m,n}$ will pass through $m-1$ cascades and become $C_{1,n-m+1}$ at the end, thereby leaving all the $C_{\text{odd},n}$ cascades behind. This leads to a square formation for even n as defined in the theorem. $\qquad\square$

In summary, triangles are useful to compress a Trotter expansion if the Trotter steps can be represented with blocks. After the compression, we are left with a triangle circuit. This can then be further simplified to a square (with same number of blocks as the triangle), but with half the total circuit depth. This leads to the following overall compression algorithm:

1. Find a block mapping, and represent the circuit via blocks.

2. Compress all blocks one by one into a triangle.

3. Transform the triangle into a square.

In the next section, we provide block mappings for Trotter decompositions of certain Hamiltonians.

III. TIME EVOLUTION OF CERTAIN SPIN MODELS

In this section, we show that the operations outlined in Sec. II are applicable to time evolution for several commonly found models in physics and quantum information theory. Each of these models has a Hamiltonian written as a sum of Pauli operators for n qubits (or n spin-½ particles)

$$\mathcal{H}(t) = \sum_j H_j(t)\sigma^j, \qquad (20)$$

where $H_j(t)$ are time- and site-dependent coefficients and σ^j are Pauli string operators, *i.e.*, elements of the n-site Pauli group $\mathcal{P}_n = \{I, X, Y, Z\}^{\otimes n}$. The common feature of the models for which Thm. 1 applies is that they can all be mapped onto free fermionic models after a Jordan-Wigner transformation [11] — or, equivalently that the algebra generated by the operators in the Hamiltonian

scales polynomially in the system size [5]. This suggests that there may be other models that share this property that will be amenable to this approach.

We next consider the Trotter decomposition of the time evolution operator given in Eq. (2) for those models. To generate a circuit for the full $U(t)$, we must convert each exponential $\exp(-i\Delta t\mathcal{H}(t_i))$ into a set of gates. The most straightforward way to map these exponentials into a set of gates is to first decompose the Hamiltonian in each exponential into components that each contain mutually commuting operators. Trotterization can once again be employed to approximate each exponential as a product of exponentials of each component of the Hamiltonian. To the first order in Δt, this approximation is written as

$$e^{-i\Delta t\mathcal{H}(t_i)} = \prod_j e^{-i\Delta t H(t_i)\sigma^j} + O(\Delta t^2) \qquad (21)$$

where j multiplies over the different components of the Hamiltonian. Though higher order approximation schemes exist [28], this first order approximation is sufficient for our method. Once the time evolution is Trotter-decomposed into a series of exponentials of single Pauli strings, a quantum circuit can then be constructed from single- and two-qubit operations [29, 30]. After finding a block mapping to exponentials of single Pauli strings that occur in Eq. (21) , the operations given in Sec. II can be used to recombine the Trotter steps into a square, which can then be transformed into a fixed depth circuit for the models discussed below.

A. 1D Kitaev Chain

The Hamiltonian for the 1D Kitaev chain with open boundary conditions is given by

$$\mathcal{H} = \sum_{\text{odd } i} a_i X_i X_{i+1} + \sum_{\text{even } i} b_i Y_i Y_{i+1}. \qquad (22)$$

The time evolution can be approximated using the Trotter decomposition with a time step Δt; for a 5-site chain this can be written explicitly as

$$e^{-i\Delta t\mathcal{H}} = e^{-i\Delta t\, a\, X_1 X_2} e^{-i\Delta t\, c\, X_3 X_4} e^{-i\Delta t\, b\, Y_2 Y_3} e^{-i\Delta t\, d\, Y_4 Y_5}. \qquad (23)$$

Note that other choices are also available as multiplication of these exponentials in any order are all identical to each other within $O(\Delta t^2)$ error.

Consider the mapping $B_i \equiv B_i(\theta) = \exp(-i\theta X_i X_{i+1})$ for odd i and $B_i \equiv B_i(\theta) = \exp(-i\theta Y_i Y_{i+1})$ for even i. With this mapping, the Trotter step given in Eq. (23) can be written in terms of B_i matrices

$$e^{-i\Delta t\mathcal{H}} = B_1(\Delta t\, a)\, B_3(\Delta t\, c)\, B_2(\Delta t\, b)\, B_4(\Delta t\, d). \qquad (24)$$

Here we show that these operators are blocks and satisfy the properties given in Def. 1. Fusion is easily satisfied

$$e^{-iaX_iX_{i+1}}e^{-ibX_iX_{i+1}} = e^{-i(a+b)X_iX_{i+1}},$$
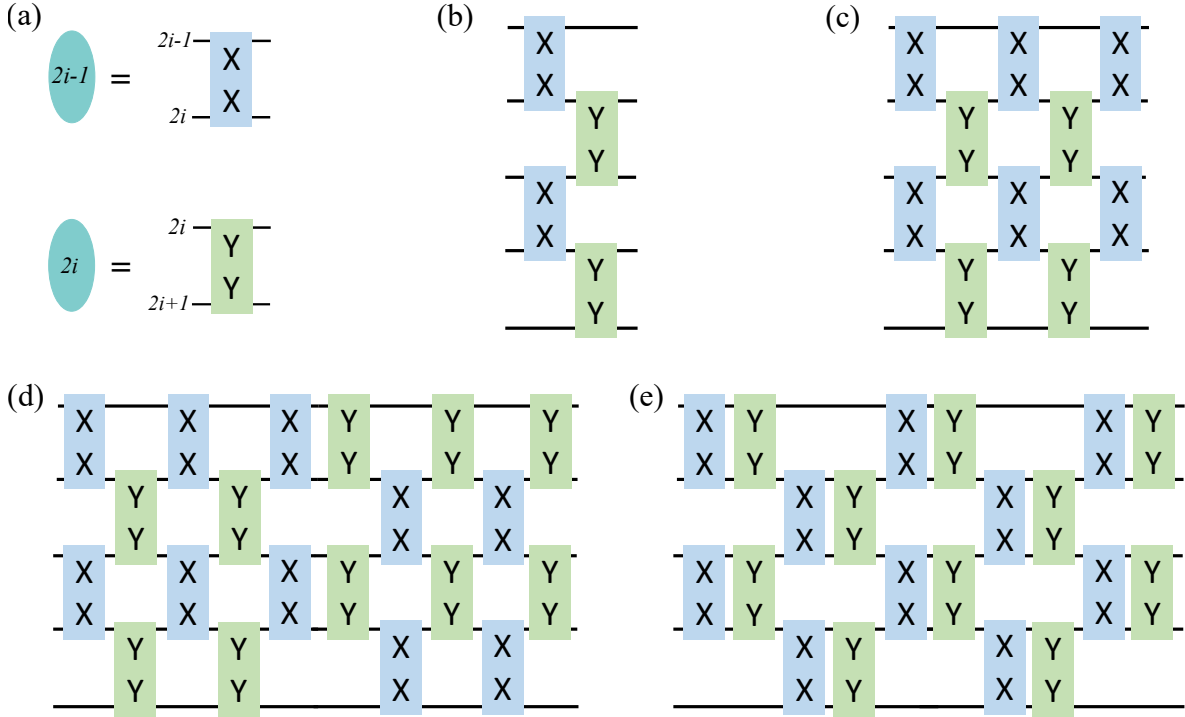$$e^{-iaY_iY_{i+1}}\, e^{-ibY_iY_{i+1}}\ = e^{-i(a+b)Y_iY_{i+1}}, \qquad (25)$$

FIG. 3. **(a)** block mapping of 1D Kitaev chain, **(b)** the Trotter-decomposed operator for a single time step given in Eq. (23) and **(c)** full time evolution circuit after Trotter compression. Each block XX indicates $\exp(-iaXX)$ for some a (and similar for YY). **(d)** Full time evolution of the XY model can be written as two consecutive evolutions of Kitaev chains. **(e)** These two evolutions can be combined to make XX and YY operations adjacent to each other to reduce the CNOT count by one half.

and independent blocks that do not share a spin commute due to the fact that they act on different spins. This leaves us with the proof of the turnover property of blocks; we need to demonstrate that

$$e^{-iaX_iX_{i+1}}e^{-ibY_{i+1}Y_{i+2}}e^{-icX_iX_{i+1}}$$
$$= e^{-i\alpha Y_{i+1}Y_{i+2}}e^{-i\beta X_iX_{i+1}}e^{-i\gamma Y_{i+1}Y_{i+2}} \quad (26)$$

where i is an odd number, and α, β, γ are functions of a, b, c. To show this, we note that the exponentiated operators form an $\mathfrak{su}(2)$ algebra. This is easily seen from their commutation relations,

$$[X_iX_{i+1}, Y_{i+1}Y_{i+2}] = 2i\, X_iZ_{i+1}Y_{i+2},$$
$$[X_iZ_{i+1}Y_{i+2}, X_iX_{i+1}] = 2i\, Y_{i+1}Y_{i+2}, \quad (27)$$
$$[Y_{i+1}Y_{i+2}, X_iZ_{i+1}Y_{i+2}] = 2i\, X_iX_{i+1}.$$

With this, we can leverage a standard Euler angle decomposition in $\mathfrak{su}(2)$. Denoting the three directions in $\mathfrak{su}(2)$ as $\hat{x}$, $\hat{y}$, and $\hat{z}$, any element in $\mathfrak{su}(2)$ can be written as a rotation around $\hat{x}$ followed by a rotation around $\hat{z}$ followed by another rotation around $\hat{x}$. In Eq. (26), the directions are $\hat{x} \equiv X_iX_{i+1}$, $\hat{z} \equiv Y_{i+1}Y_{i+2}$ on the left hand side, and vice versa on the right. Therefore for any $a, b, c \in \mathbb{R}$, there exist $\alpha, \beta, \gamma \in \mathbb{R}$ and vice versa such that Eq. (26) is satisfied.

The coefficients can be calculated by mapping 2×2 Pauli matrices to the exponents X_iX_{i+1}, $Y_{i+1}Y_{i+2}$ due

to the isomorphism between the algebra generated by the exponents and $\mathfrak{su}(2)$. Mapping X to X_iX_{i+1} and Z to $Y_{i+1}Y_{i+2}$, Eq. (26) becomes

$$\begin{pmatrix} \cos a & -i\sin a \\ -i\sin a & \cos a \end{pmatrix}\begin{pmatrix} e^{-ib} & 0 \\ 0 & e^{ib} \end{pmatrix}\begin{pmatrix} \cos c & -i\sin c \\ -i\sin c & \cos c \end{pmatrix}$$
$$= \begin{pmatrix} e^{-i\alpha} & 0 \\ 0 & e^{i\alpha} \end{pmatrix}\begin{pmatrix} \cos \beta & -i\sin \beta \\ -i\sin \beta & \cos \beta \end{pmatrix}\begin{pmatrix} e^{-i\gamma} & 0 \\ 0 & e^{i\gamma} \end{pmatrix}, \quad (28)$$

which may be solved to obtain the following equations

$$\tan(\alpha + \gamma) = \tan(b)\frac{\cos(a-c)}{\cos(a+c)},$$
$$\tan(\alpha - \gamma) = -\tan(b)\frac{\sin(a-c)}{\sin(a+c)}, \quad (29)$$

and their inverses

$$\tan(a + c) = \tan(\beta)\frac{\cos(\alpha-\gamma)}{\cos(\alpha+\gamma)},$$
$$\tan(a - c) = -\tan(\beta)\frac{\sin(\alpha-\gamma)}{\sin(\alpha+\gamma)}. \quad (30)$$

α and γ can be solved directly from Eq. (29), and then β can be solved from one of the relations given in Eq. (30). Although we provide the analytical expressions here, in

the code [24, 25] we use another method based on Givens rotations applied on 2×2 matrices, explained in detail in Ref. [23]. This method is far more efficient and precise compared to calculating the angles via Eqs. (29) and (30) because the analytic expressions rely on repeated use of trigonometric and inverse trigonometric functions.

Having shown that the elements of the Kitaev chain circuit satisfy the properties of a block, Thm. 1 and Cors. 3 and 4 say that the Trotter circuit for this model can be compressed into a triangle of blocks; and Thm. 2 states that it can be further simplified into a square as given in Fig. 3(c). Considering the n spin Kitaev chain, one Trotter step would have $n-1$ 2-qubit XX and YY rotations, leading to $n-1$ blocks with the mapping we provided. This would lead to $n(n-1)/2$ blocks in the final square circuit. Depending on the hardware, this means we either need $n(n-1)/2$ 2-qubit rotations, or considering the circuit implementation of XX and YY rotations via CNOT gates, we need $n(n-1)$ CNOT gates.

B. XY Model

The Hamiltonian for the 1D XY Model with open boundary conditions is given by

$$\mathcal{H} = \sum_{i=1}^{n-1} \left(a_i X_i X_{i+1} + b_i Y_i Y_{i+1} \right) \tag{31}$$

where n is the number of spins. It can be shown that this Hamiltonian is the sum of two Kitaev chains

$$\mathcal{H} = \mathcal{H}_{\text{even X}} + \mathcal{H}_{\text{odd X}}, \tag{32}$$

where

$$\mathcal{H}_{\text{even X}} = \sum_{\text{even } i} a_i X_i X_{i+1} + \sum_{\text{odd } i} b_i Y_i Y_{i+1},$$
$$\mathcal{H}_{\text{odd X}} = \sum_{\text{odd } i} a_i X_i X_{i+1} + \sum_{\text{even } i} b_i Y_i Y_{i+1}. \tag{33}$$

One can check that these chains are independent from each other, i.e. $[\mathcal{H}_{\text{even X}}, \mathcal{H}_{\text{odd X}}] = 0$, by observing

$$[X_i X_{i+1}, Y_i Y_{i+1}] = 0,$$
$$[X_i X_{i+1}, X_{i+1} X_{i+2}] = 0, \tag{34}$$
$$[Y_i Y_{i+1}, Y_{i+1} Y_{i+2}] = 0.$$

Therefore the evolution of the XY model can be written as two separate evolutions of Kitaev chains, as given in Fig. 3(d). In fact, Eq. (34) also shows that any two terms in the separate evolution circuits commute with each other, and therefore can be moved freely. Considering the fact that XX and YY when individually implemented require 4 CNOTs in total whereas their combination requires only 2 CNOTs, it is advantageous to bring XX and YY rotations together for hardware that uses CNOTs. Eq. (34) allows us to transform the circuit in Fig. 3(d) to Fig. 3(e) and reduce number of CNOTs by one half.

C. Transverse Field Ising Model

The Hamiltonian for the 1D TFIM with open boundary conditions is given by

$$\mathcal{H} = \sum_{i=1}^{n-1} a_i X_i X_{i+1} + \sum_{i=1}^{n} b_i Z_i. \tag{35}$$

The time evolution is again factored into Trotter steps of length Δt; for a 4-site chain this can be implemented as shown in Fig. 4.

Consider the mapping $B_{2i-1} \equiv B_{2i-1}(\theta) = \exp(-i\theta Z_i)$ and $B_{2i} \equiv B_{2i}(\theta) = \exp(-i\theta X_i X_{i+1})$. With this mapping, a Trotter step for this model can be written as a product of B_i matrices. Here we show that these operators satisfy the block properties from Def. 1. Note that this is an example where not all blocks are 2-qubit operators, which shows that blocks can vary in size. The fusion operation is satisfied by this mapping

$$e^{-iaX_i X_{i+1}} e^{-ibX_i X_{i+1}} = e^{-i(a+b)X_i X_{i+1}},$$
$$e^{-iaZ_i} e^{-ibZ_i} = e^{-i(a+b)Z_i}. \tag{36}$$

The commutation of blocks with an odd index i is clear since they do not act on a common spin: B_{2i-1} acts on qubit i and B_{2i+1} acts on qubit $i+1$. Blocks with even i on the other hand should be checked explicitly, because B_{2i} acts on spins i and $i+1$, and B_{2i+2} acts on spins $i+1$ and $i+2$, i.e., they both act on spin $i+1$. However, since $[X_i X_{i+1}, X_{i+1} X_{i+2}] = 0$, commutation of independent blocks applies for even indexed blocks as well. This leaves us with the turnover property,

$$e^{-iaX_i X_{i+1}} e^{-ibZ_i} e^{-icX_i X_{i+1}}$$
$$= e^{-i\alpha Z_i} e^{-i\beta X_i X_{i+1}} e^{-i\gamma Z_i}. \tag{37}$$

To show this, we note that similar to the Kitaev chain, the exponents form an $\mathfrak{su}(2)$ algebra

$$[Z_i, X_i X_{i+1}] = 2i\, Y_i X_{i+1},$$
$$[Y_i X_{i+1}, Z_i] = 2i\, X_i X_{i+1}, \tag{38}$$
$$[X_i X_{i+1}, Y_i X_{i+1}] = 2i\, Z_i.$$

Therefore, by the arguments given in the discussion for the Kitaev chain, it can be shown that the turnover property is satisfied for the blocks as defined for the TFIM as well. The relation between the sets of angles are also the same as in Eq. (29) due to the fact that it is the same switch between Euler decompositions of $\mathfrak{su}(2)$.

Having shown that the elements of the TFIM circuit satisfy the properties of a block (Def. 1), it follows from Thm. 1 that the Trotter circuit for this model can be compressed into a triangle of blocks.

Considering the n spin TFIM, one Trotter step would have $n-1$ 2-qubit XX and n 1-qubit Z rotations, leading to $2n-1$ blocks with the mapping we provided. This would lead to $n(2n-1)$ blocks or n Trotter steps in the
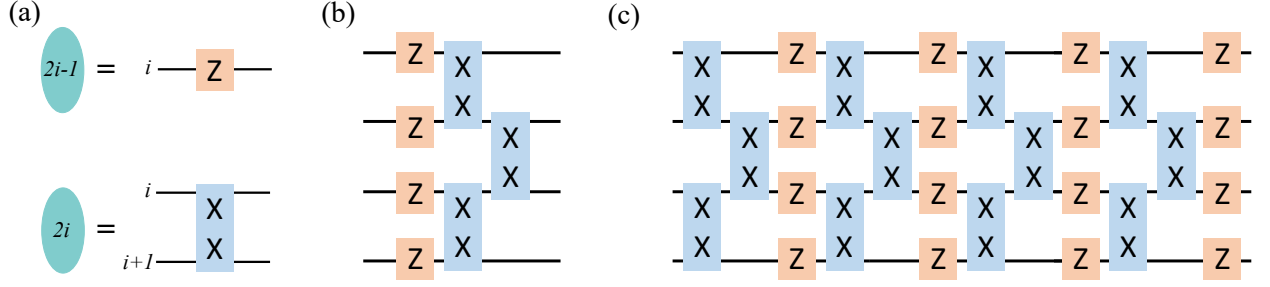
FIG. 4. **(a)** block mapping of the 1D TFIM. As can be seen, blocks are not always 2 qubit operators. **(b)** The Trotter-decomposed operator for a single time step of the evolution under the 1D TFIM Hamiltonian. **(c)** Full time evolution circuit after Trotter compression. Each block XX indicates $\exp(-iaXX)$ for some a, and each block Z indicates $\exp(-ibZ)$ for some b.

final square circuit. Depending on the hardware, this means we either need $n(n-1)$ XX rotations, or considering the circuit implementation of XX rotations via CNOT gates, we need $2n(n-1)$ CNOT gates.

D. Transverse Field XY Model

The Hamiltonian for the 1D transverse field XY (TFXY) model with open boundary conditions is given by

$$\mathcal{H} = \sum_{i=1}^{n-1}(a_i X_i X_{i+1} + b_i Y_i Y_{i+1}) + \sum_{i=1}^{n} c_i Z_i. \quad (39)$$

Here, the verification of the block properties is somewhat more complex. First, we define the B_i operator as

$$B_i \equiv e^{-ia\,Z_i} e^{-ib\,Z_{i+1}} e^{-ic\,X_i X_{i+1}} \\ e^{-id\,Y_i Y_{i+1}}\, e^{-if\,Z_i} e^{-ig\,Z_{i+1}} \quad (40)$$

which can be shown diagrammatically as in Fig. 5(a). A Trotter step for this model can be written in terms of the B_i matrices, since the exponents in Eq. (40) involve all the terms in the Hamiltonian. We proceed to prove that the mapping (40) satisfies all block properties.

Commutation. First, we prove that the B_i operators satisfy the commutation relation given in Def. 1. This property follows from the observation that if two blocks are not on top of each other nor nearest neighbors to each other, they do not act on a shared spin. For example, B_i acts on qubits i and $i+1$, and B_{i+2} acts on qubits $i+2$ and $i+3$. Therefore they act on different spaces, and commute.

Fusion. To prove the fusion property, it suffices to show that the expression given in Eq. (40) represents a generic element of a Lie group. The algebra formed by the exponents in Eq. (40) and their commutators is the Hamiltonian algebra for the 2 site TFXY model [5] (here we suppress $i=1$ for convenience):

$$\mathfrak{g}(\text{TFXY}_2) = \text{span } i\{X_1 X_2, Y_1 Y_2, X_1 Y_2, Y_1 X_2, Z_1, Z_2\}. \quad (41)$$

We can use an involution $\theta : \mathfrak{g} \to \mathfrak{g}$ to form a Cartan decomposition $\mathfrak{g} = \mathfrak{k} \oplus \mathfrak{m}$ [31]. Specifically, the involution $\theta(g) = -X_1 X_2\, g^T\, X_1 X_2$ (which counts the number of Z matrices), where $g \in \mathfrak{g}(\text{TFXY}_2)$, yields

$$\mathfrak{k} = \text{span } i\{Z_1, Z_2\}, \\ \mathfrak{m} = \text{span } i\{X_1 X_2, Y_1 Y_2, X_1 Y_2, Y_1 X_2\}. \quad (42)$$

We can choose a maximal Abelian subalgebra $\mathfrak{h} \in \mathfrak{m}$ as

$$\mathfrak{h} = \text{span } i\{X_1 X_2, Y_1 Y_2\}. \quad (43)$$

The "KHK theorem" [31] states that any element in the group $G \in \exp(g)$ for $g \in \mathfrak{g}(\text{TFXY}_2)$ can be written as

$$G = e^{-ia\,Z_1} e^{-ib\,Z_2} e^{-ic\,X_1 X_2} \\ e^{-id\,Y_1 Y_2}\, e^{-if\,Z_1} e^{-ig\,Z_2}, \quad (44)$$

which is precisely Eq. (40). Thus, two blocks can be fused because the right hand side of Eq. (40) is a generic element in the Lie group $\exp(\mathfrak{g}(\text{TFXY}_2))$.

Turnover. To show that the blocks B_i satisfy the turnover property, we consider the Hamiltonian algebra of the 3 spin TFXY model,

$$\mathfrak{g}(\text{TFXY}_3) = \text{span } i\{X_1 X_2, Y_1 Y_2, X_1 Y_2, Y_1 X_2, X_2 X_3, \\ Y_2 Y_3, X_2 Y_3, Y_2 X_3, Z_1, Z_2, Z_3, \\ X_1 Z_2 X_3, X_1 Z_2 Y_3, Y_1 Z_2 X_3, \\ Y_1 Z_2 Y_3\}. \quad (45)$$

Using the involution $\theta(g) = Z_3 g Z_3$, where $g \in \mathfrak{g}(\text{TFXY}_3)$, we partition $\mathfrak{g}(\text{TFXY}_3)$ into

$$\mathfrak{k} = \text{span } i\{X_1 X_2, Y_1 Y_2, X_1 Y_2, Y_1 X_2, Z_1, Z_2, Z_3\}, \\ \mathfrak{m} = \text{span } i\{X_2 X_3, Y_2 Y_3, X_2 Y_3, Y_2 X_3, X_1 Z_2 X_3, \\ X_1 Z_2 Y_3, Y_1 Z_2 X_3, Y_1 Z_2 Y_3\}. \quad (46)$$

We note that $\mathfrak{k}$ is simply

$$\mathfrak{k} = (I_3 \otimes \mathfrak{g}(\text{TFXY}_2)) \oplus \text{span } i\{Z_3\}, \quad (47)$$

where I_3 is the identity operator for the 3rd spin. The relation between $\mathfrak{k}$ and $\mathfrak{g}(\text{TFXY}_2)$ together with Eq. (44)
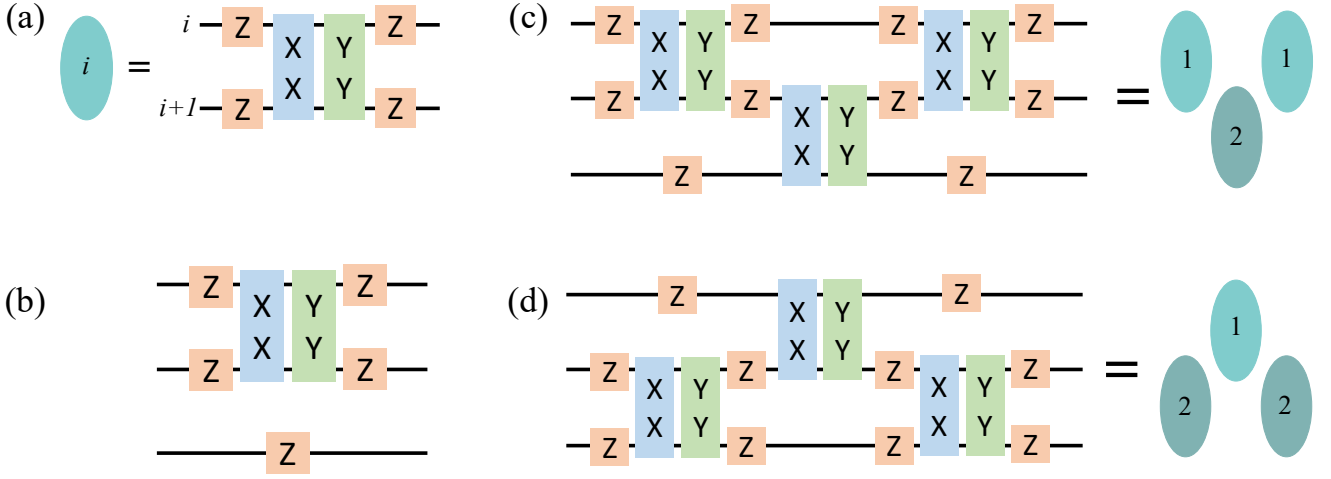
FIG. 5. **(a)** block structure for the 1D TFXY model. **(b)** A generic circuit element that can represent any element in the group $\exp(\mathfrak{k})$ where $\mathfrak{k}$ results from Cartan decomposition of $\mathfrak{g}(\mathrm{TFXY}_3)$. It can be seen that this circuit represent a generic element via Eq.(47). **(c)** Circuit structure for a generic element of the group $\exp(\mathfrak{g}(\mathrm{TFXY}_3))$ obtained via Cartan decomposition corresponding to the involution $\theta(g) = Z_3\, g\, Z_3$ and Cartan subalgebra given in Eq. (48). As it can be seen, with the block definition, this structure corresponds to $B_1B_2B_1$. **(d)** We could have chosen $\theta(g) = Z_1\, g\, Z_1$ instead, and come up with a generic representation of an element in the group $\exp(\mathfrak{g}(\mathrm{TFXY}_3))$ which corresponds to $B_2B_1B_2$. Considering that both panel (c) and panel (d) are generic representations of the same group, this proves that the structure given in panel (a) satisfies the turnover property.

allows us to represent any element in the group $\exp(\mathfrak{k})$ as in Fig. 5(b). Therefore by choosing the following Cartan subalgebra

$$\mathfrak{h} = \mathrm{span}\, i\{X_2X_3, Y_2Y_3\}, \qquad (48)$$

the KHK theorem implies that any element G in the group $\exp(\mathfrak{g}(\mathrm{TFXY}_3))$ can be written as $G = \exp(k_1)\exp(h)\exp(k_2)$ where $k_1, k_2 \in \mathfrak{k}$ and $h \in \mathfrak{h}$. This may be represented by the circuit given in Fig. 5(c), which is a V shaped 3 block structure, with each block representing the operator in Eq. (40).

The Cartan decomposition is not unique — we could have alternatively used the involution $\theta(g) = Z_1\, g\, Z_1$. This results in a swap of the 1st and 3rd spins. Thus, any element in the group $\exp(\mathfrak{g}(\mathrm{TFXY}_3))$ can be equally well represented by an inverted circuit as shown in Fig. 5(d). This means that we have found two equivalent block structures that both represent a generic element in the group $\exp(\mathfrak{g}(\mathrm{TFXY}_3))$; these can then be transformed into one another, which proves the turnover property for the block mapping of Eq. (40).

The above proofs for both fusion and turnover properties for the block structure Eq. (40) are based on the algebraic properties of $\mathfrak{g}(\mathrm{TFXY}_2)$ and $\mathfrak{g}(\mathrm{TFXY}_3)$, and do not show how to calculate the coefficients after the fusion or turnover. Here we provide another Lie algebra based method to calculate the coefficients after the fusion and turnover operations.

Fusion calculation. We already proved that the B_i matrices given in Eq. (40) form a Lie group. The Lie algebra associated with that group turned out to be the Hamiltonian algebra of a 2 site TFXY model (41). We

can use a different basis for that algebra, and rewrite it as

$$\begin{aligned}\mathfrak{g}(\mathrm{TFXY}_2) = \mathrm{span}\, i\{&(X_1X_2 \pm Y_1Y_2)/2, \\ &(X_1Y_2 \pm Y_1X_2)/2, \\ &(Z_1 \pm Z_2)/2\}.\end{aligned} \qquad (49)$$

We can split it into two algebras $\mathfrak{g}_+$ and $\mathfrak{g}_-$ given by

$$\begin{aligned}\mathfrak{g}_- = \mathrm{span}\, i\{&(X_1X_2 - Y_1Y_2)/2, (Z_1 + Z_2)/2, \\ &(X_1Y_2 + Y_1X_2)/2\},\end{aligned} \qquad (50)$$

and

$$\begin{aligned}\mathfrak{g}_+ = \mathrm{span}\, i\{&(X_1X_2 + Y_1Y_2)/2, (Z_1 - Z_2)/2, \\ &(X_1Y_2 - Y_1X_2)/2\}.\end{aligned} \qquad (51)$$

A simple check on the commutation relations for the basis elements yields that both algebras are isomorphic to $\mathfrak{su}(2)$, and both are independent, i.e. $[\mathfrak{g}_+, \mathfrak{g}_-] = 0$. Renaming $(X_iX_{i+1} \pm Y_iY_{i+1})/2 \equiv Q_i^\pm$ and $(Z_i \pm Z_{i+1})/2 \equiv R_i^\pm$ for convenience, one can rewrite Eq. (40) as a multiplication of two independent $SU(2)$ elements

$$\begin{aligned}B_i \equiv &\, e^{-i(a+b)\, R_i^+} e^{-i(a-b)\, R_i^-} e^{-i(c+d)\, Q_i^+} \\ &\, e^{-i(c-d)\, Q_i^-} e^{-i(f+g)\, R_i^+} e^{-i(f-g)\, R_i^-} \\ = &\, \big(e^{-i(a+b)\, R_i^+} e^{-i(c-d)\, Q_i^-} e^{-i(f+g)\, R_i^+}\big) \\ &\, \big(e^{-i(a-b)\, R_i^-} e^{-i(c+d)\, Q_i^+} e^{-i(f-g)\, R_i^-}\big).\end{aligned} \qquad (52)$$

Then, the fusion operation will boil down to multiplication of two set of $SU(2)$ matrices represented by an Euler decomposition, which only requires the $\mathfrak{su}(2)$ turnover

that we provided in Eq. (29).Each $SU(2)$ multiplication will require one $\mathfrak{su}(2)$ turnover operation, therefore fusion of two TFXY blocks require two $\mathfrak{su}(2)$ turnover operations in total.

Turnover calculation. This proof entirely relies on the fact that the TFXY block Eq. (40) can be written as a combination of TFIM blocks, i.e. Z and XX rotations. This can be achieved by plugging

$$Y_iY_{i+1} = e^{i\frac{\pi}{4}Z_i}e^{i\frac{\pi}{4}Z_{i+1}}X_iX_{i+1}e^{-i\frac{\pi}{4}Z_i}e^{-i\frac{\pi}{4}Z_{i+1}}, \quad (53)$$

into Eq. (40). Therefore a V shaped TFXY block structure $B_iB_{i+1}B_i$ can be transformed to a TFIM triangle with height 3 that acts on qubits i, $i+1$ and $i+2$ given as
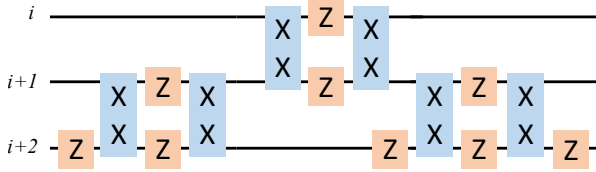
where we deliberately grouped some gates together. All three groups can be rewritten as a TFXY block by applying the TFXY fusion operation we explained above once. For example, the leftmost group is a product of the block $B_{i+1} \equiv \exp(ia\,Z_{i+2})\exp(ib\,X_{i+1}X_{i+2})$, which is a TFXY block with some parameters equal to 0, with another block $B_{i+1} \equiv \exp(icZ_{i+1})\exp(idZ_{i+2})\exp(ieX_{i+1}X_{i+2})$. Therefore, we end up with a Λ shaped TFXY block structure. The reverse can be done in the same way by compressing Λ into an upside down triangle instead.

To form the TFIM triangle from the V shaped TFXY block structure, 26 $\mathfrak{su}(2)$ turnover operations must be applied. To combine the 2-qubit pieces into a TFXY block, 3 TFXY merge operations must be applied. Each TFXY merge requires 2 $\mathfrak{su}(2)$ turnovers, leading to a total of 32 $\mathfrak{su}(2)$ turnover operations for one TFXY turnover operation.

The constructive proofs we provide here use trigonometric and inverse trigonometric function evaluations frequently, which are inefficient and inaccurate compared to simple linear algebra operations. Thus, we additionally provide a linear algebra based method to evaluate the angles for the fusion and turnover operations of a TFXY model in Ref. 23. However, we include the Lie algebra based proof here as well because it provides additional insight from the algebraic structure of the TFXY model.

Since the mapping of the 1D TFXY model in Eq. (40) satisfies the properties listed in Def. 1 it can be compressed into a triangle according to Cor. 1.

For an n spin TFXY model, one Trotter step has $n-1$ XX and YY rotations with n 1-qubit Z rotations, leading to $n-1$ blocks with the mapping we provided for the TFXY model. This leads to $n(n-1)/2$ blocks in the final square circuit. Depending on the hardware, this means we either need $n(n-1)$ 2-qubit rotations (2 per block), or considering the circuit implementation of ad-

jacent XX and YY rotations via 2 CNOT gates [32], we need $n(n-1)$ CNOT gates.

E. Generalized TFXY and Free Fermions

The algebra generated by the elements in Eq. (41) and the generic group representation in Eq. (44) imply that we can generalize the results for the TFXY model to the following Hamiltonian (which can be called generalized TFXY Hamiltonian)

$$\mathcal{H} = \sum_{i=1}^{n-1}(a_iX_iX_{i+1} + b_iY_iY_{i+1} + c_iX_iY_{i+1} + d_iY_iX_{i+1}) + \sum_{i=1}^{n}f_iZ_i. \quad (54)$$

with the same block mapping from Eq. (40). Via the Jordan–Wigner transformation, this Hamiltonian can be mapped to

$$\mathcal{H} = \sum_{i=1}^{n-1}(\alpha_i\hat{c}_i^\dagger\hat{c}_{i+1} + \alpha_i^*\hat{c}_{i+1}^\dagger\hat{c}_i + \beta_i\hat{c}_i\hat{c}_{i+1} + \beta_i^*\hat{c}_{i+1}^\dagger\hat{c}_i^\dagger) + \sum_{i=1}^{n}\gamma_i\hat{c}_i^\dagger\hat{c}_i. \quad (55)$$

where $\hat{c}_i(\hat{c}_i^\dagger)$ is the fermion annihilation (creation) operator on site i. Therefore, using the block mapping in Eq. (40), we can compress Trotter decompositions under any free, one-dimensional Hamiltonian with nearest neighbor hopping and open boundary conditions.

F. Comparison of the Models

We introduced three different block mappings with two types of fusion and turnover operations. The turnover operation for the blocks given in the Kitaev chain, XY model and TFIM use Euler decompositions of $\mathfrak{su}(2)$, whereas the ones for TFXY and generalied TFXY models use algebraical properties of $\mathfrak{g}(\text{TFXY}_3)$. The $\mathfrak{su}(2)$ turnover is approximately 20 times more efficient than the $\mathfrak{g}(\text{TFXY}_3)$ turnover, because the former requires diagonalization of a 2×2 matrix whereas the latter requires simultaneous diagonalization of four 2×2 matrices [23], although in practice this only matters when the system size grows large.

Although some models share the same type of fusion and turnover properties, the resulting circuits vary in the number of CNOT or 2-qubit XX/YY rotation gates required for n qubits, as shown in Tab. I

Because all of these models can be derived from TFXY model, the circuits for the Kitaev chain, XY model, and TFIM do not have to be compressed by the model-specific block mappings we discussed above; one can consider

Block Mapping	$XX + YY$	#CNOT	Turnover Type
Kitaev Chain	$\frac{n(n-1)}{2}$	$n(n-1)$	$\mathfrak{su}(2)$
XY Model	$n(n-1)$	$n(n-1)$	$\mathfrak{su}(2)$
TFIM	$n(n-1)$	$2n(n-1)$	$\mathfrak{su}(2)$
TFXY	$n(n-1)$	$n(n-1)$	$\mathfrak{g}(\text{TFXY}_3)$

TABLE I. Number of 2-qubit gates required for the constant depth circuits on n qubits for the models discussed. Here, $XX + YY$ indicates the total number of XX and YY rotations.

them as specific instances of the TFXY model and use the TFXY block mapping instead. For the TFIM, if one uses the TFXY block structure, the height of the Trotter step zigzag (Def. 4) reduces by half, leading to 4 or 8 times fewer turnover operations depending on whether the system is time-dependent or not due to Cors. 3 and 4. Although the TFXY turnover is 20 times more expensive than the $\mathfrak{su}(2)$ turnover, and this leads to $3 - 5$ times more time spent for the compression, using TFXY block structure leads to half the number of CNOT gates in the resulting circuits, which proves particularly useful for quantum computers that rely on the CNOT 2-qubit gate (such as transmon-based hardware).

IV. EXAMPLE: ADIABATIC STATE PREPARATION

An important step of many static and dynamic simulations problems on quantum computers involves preparing the ground state of the relevant Hamiltonian. In general, this ground state is non-trivial to prepare on the qubits, even in cases where the state may be known analytically. In this example, we demonstrate how our compressed circuits can be used to efficiently prepare non-trivial ground states on the quantum computer with high-accuracy via adiabatic state preparation (ASP) [33, 34]. In ASP, qubits are prepared in the ground state of some initial Hamiltonian $\mathcal{H}_I$, which is presumed to be trivial to prepare. The system is then evolved under a parameter-dependent Hamiltonian $\mathcal{H}(s) = (1 - s)\mathcal{H}_I + s\mathcal{H}_P$ as s is varied from 0 to 1. Here, $\mathcal{H}_P$ is the problem Hamiltonian, whose ground state is desired and in general is non-trivial to prepare. If s is varied slowly enough, the system will remain in the ground state of the instantaneous Hamiltonian according to the adiabatic theorem [35]. Just how slow this variation needs to be is dependent upon the size of the energy gap between the ground state and first excited state.

Evolving the system under $\mathcal{H}(s)$ as s is varied from 0 to 1 can be viewed as evolving the system under a time-dependent Hamiltonian. To keep this evolution adiabatic, the total time T used to evolve s from 0 to 1 should be chosen large enough to ensure that the system evolves slow enough. To simulate this time-dependent Hamiltonian $\mathcal{H}(s)$, the time interval T is split into N

time steps of size Δt ($N = T/\Delta t$) and Trotter decomposition can be used which yields more accurate results for smaller Δt. Both requirements (large T for adiabaticity and small Δt for accuracy of Trotterization) lead to a large number of Trotter steps i.e. large N.

Assuming that the instantaneous Hamiltonian $\mathcal{H}(s)$ remains within the class of free-fermionic models while varying s, we can apply our compression techniques to efficiently produce minimal, fixed-depth circuits for such an adiabatic evolution. Our technique offers an enormous advantage in ASP, as the ability to generate fixed-depth circuits in an efficient amount of time for an arbitrarily large number of time-steps allows us to vary s arbitrarily slowly, thereby guaranteeing adiabatic evolution with one simple fixed-depth circuit. As shown in the companion paper [23], the compression can be performed efficiently, and thus poses no limitation on the adiabaticity, except for floating point resolution of Δt. Without our compression techniques, the quantum circuits generated via standard Trotter decomposition have a depth growing linearly with number of Trotter steps. Considering that ASP requires a large number of Trotter steps i.e. large N, these circuits rapidly become too large to produce accurate results on NISQ devices [36].

As an example, we demonstrate ASP of a spatially uniform, 5-spin TFIM with open boundary conditions, comparing performance between compressed circuits and circuits generated with standard Trotter decomposition (i.e., uncompressed circuits). The parameter-dependent Hamiltonian is given by $\mathcal{H}(s) = (1 - s)\mathcal{H}_I + s\mathcal{H}_P$ where $\mathcal{H}_I = h_z \sum_{i=1}^{n} Z_i$ and $\mathcal{H}_P = J_P \sum_{i=1}^{n-1} X_i X_{i+1} + h_z \sum_{i=1}^{n} Z_i$, where $n = 5$ is the number of qubits.

Varying the parameter s from 0 to 1 is equivalent to the following time-dependent Hamiltonian:

$$\mathcal{H}_{ASP}(t) = J(t) \sum_{i=1}^{n-1} X_i X_{i+1} + h_z \sum_{i=1}^{n} Z_i \qquad (56)$$

where $J(0) = 0$, and $J(t)$ is increased linearly to a final time $t = T$ such that $J(T) = J_P$ as defined in the problem Hamiltonian.

The qubits are initialized in the ground state of $\mathcal{H}_I = \mathcal{H}_{ASP}(0)$, which in this case is simply all qubits in the spin-up orientation, a trivial state to prepare on the quantum computer. Next, we evolve the qubits under the time-dependent Hamiltonian $\mathcal{H}_{ASP}$ from $t = 0$ to $t = T$ using Trotter decomposition to break this evolution into small time-steps of size Δt. If T is large enough to achieve an adiabatic evolution, and if Δt is small enough to minimize Trotter error, the final state of the qubits after this evolution will be the ground state of our problem Hamiltonian $\mathcal{H}_P = \mathcal{H}_{ASP}(T)$, which in general is non-trivial to prepare. Finally, we end the simulation by evolving the system under the time-independent problem Hamiltonian $\mathcal{H}_P = \mathcal{H}_{ASP}(T)$. If adiabatic simulation was successful, the system will remain in the stationary ground state of the problem Hamiltonian for the entirety of this final evolution.
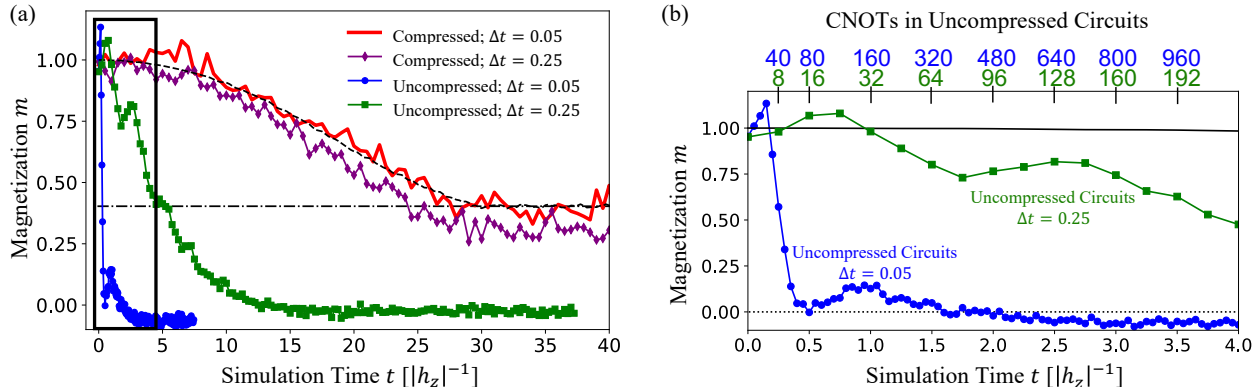
FIG. 6. Simulation results from *ibmq_brooklyn* comparing compressed circuits versus uncompressed circuits derived from standard Trotter decomposition. A 5-qubit system is evolved under a time-dependent Hamiltonian that is adiabatically varied from an Ising model to a transverse field Ising model (TFIM) from time $t = 0$ to time $t = 30$, and then evolved under the final (time-independent) TFIM for the remaining time. (a) Simulation results from compressed circuits (solid red and purple diamond lines) and uncompressed circuits (blue circle and green square lines) for two different values of Trotter step-size compared to the numerically exact results (black dashed line). The black dashed-dot line gives the expected magnetization of the final TFIM, which is maintained through the final evolution ($t > 30$). The solid red curve shows higher accuracy than the purple diamond curve due to smaller Trotter error. Results from the uncompressed circuits quickly decay to random noise, giving a magnetization of 0. (b) A zoomed-in view of the uncompressed circuit results within the portion of the plot in panel (a) in the solid black box. CNOT counts for each uncompressed circuit throughout the simulation are denoted on the top horizontal axis (the upper row of numbers corresponds to the blue circle line, the lower row of numbers corresponds to the green square line). The compressed circuits contain 20 CNOTs for all time-steps.

We use *ibmq_brooklyn*, a quantum processor that relies on CNOT as a 2-qubit gate, to generate our results. To compress the Trotter circuit, we applied TFXY block structure rather than TFIM block structure to minimize the number of CNOTs as given in Tab. I. Fig. 6 shows simulation results with $h_z = -1$, $J_P = -2$, and $T = 30$ using compressed circuits (red solid and purple diamond lines) and uncompressed circuits (green square and blue circle lines). Both plots show the average magnetization along the z-direction of the 5-qubit system versus time as it evolved under $\mathcal{H}_{ASP}$. The average magnetization for an n-spin system is given by $\langle m(t) \rangle \equiv \frac{1}{n} \sum_i \sigma_i^z(t)$. The dashed black line gives the exact system magnetization of the instantaneous Hamiltonian $\mathcal{H}_{ASP}(t)$ (computed numerically via exact diagonalization), and the black dashed-dot line shows the expected magnetization of the final Hamiltonian $\mathcal{H}_P = \mathcal{H}_{ASP}(T)$. If the system is truly varied adiabatically, its magnetization should reach this value at $T = 30$ and stay at this value throughout the remainder of the evolution under the problem Hamiltonian.

The solid red and purple diamond curves in Fig. 6(a) show simulation results from compressed circuits executed on *ibmq_brooklyn* using two different values for Δt used for the Trotter decomposition. We applied readout error mitigation and zero-noise extrapolation to these results. The solid red curve is in remarkably good agreement with the exactly computed values, while the purple diamond curve stabilizes at a final magnetization that is slightly smaller than expected; this discrepancy is due to

Trotter error. As is clear from Fig. 6, $\Delta t = 0.05$ is sufficiently small, while $\Delta t = 0.25$ leads to non-negligible Trotter error. While $\Delta t = 0.05$ appears sufficiently small in terms of Trotter error, we note that our compressed circuits allow for Δt to be arbitrarily small.

In comparison, the blue circle curve in Fig. 6(a) shows the magnetization from the same quantum processor using uncompressed circuits and a time-step of $\Delta t = 0.05$. We see an immediate loss of accuracy in the results as the uncompressed circuits quickly become too deep for NISQ hardware. The magnetization quickly decays towards 0, which indicates the results are simply random noise. To improve upon these results, we increased Δt, such that fewer time-steps are required to reach the same total simulation time, and which in turn means the circuit depths of the uncompressed circuits grows more slowly with the simulation time [Fig. 6(b)]. This is shown by the green square curve, which shows hardware results from uncompressed circuits with a time-step of $\Delta t = 0.25$. While its performance appears slightly better, these results, too, eventually decay to a magnetization of zero well before the adiabatic evolution is complete.

Fig. 6(b) shows a zoomed-in view of the portion of the plot in panel a surrounded by the solid black box. Quantum hardware simulation results for the uncompressed circuits for both Δt values are shown, with the number of CNOTs in the circuit used for each time-step given on the top horizontal axis in the corresponding color (the upper line gives CNOT count for $\Delta t = 0.05$, while the lower line gives CNOT counts for $\Delta t = 0.25$). The num-

ber of CNOTs per circuit for $\Delta t = 0.05$ grows much faster with total simulation time than for $\Delta t = 0.25$, which is why the green (upper dashed) curve maintains higher accuracy at initial simulation times. For reference, our compressed circuits only contain 20 CNOTs for all time-steps for this 5-qubit system.

V. DISCUSSION AND OUTLOOK

We have demonstrated a new circuit compression method leading to a fixed depth circuit for the time evolution of free fermionic models, even for a time-dependent Hamiltonian. It relies only on (space and time) local operations of the circuit elements, i.e. it does not require a full time evolution operator to begin the compression. Rather, it can locally compress 2-qubit gates, piece by piece, which makes it naturally amenable to handling Trotter decomposition. The method obviates the need to work with large matrices, which extends the range of applicability and enabled very efficient compression of the circuit elements[23]. This is in contrast to alternative fixed-depth methods based on the works of Reck et al. and Clements et al. [37, 38] which arrive at the triangle (Fig. 1(f)) and square (Fig. 1(g)) circuit structures respectively – with blocks replaced by interferometer operators – by taking a global approach to the problem and decomposing the full unitary operator in the single-particle sector (an $n \times n$ matrix) using Givens rotations.

While we have focused on the generalized TFXY model and its descendants, the approach here extends to any model that has similarly structured frustration graphs [11], or (equivalently) can be mapped onto a free fermionic Hamiltonian with nearest neighbor hopping. We should note that the compression method cannot be universally applied, as is evident from the No-Fast-Forwarding theorem [12], but it is possible that there are other classes of models where the compression method works, such as those outlined by Gu et al. [13]. Finally, even if the entire circuit for time evolution does not follow the required block structure, sub-circuits can still be optimized in this way (if the sub-circuits do satisfy the block structure).

Free fermionic time-evolution can be used for ground state preparation via adiabatic evolution, which we have chosen as an example application in this work. More generally, it can be shown that any generic Bogolyubov transformation [39, 40], i.e. unitary transformations of one particle creation-annihilation operators, can be written as a free fermionic time-evolution. Therefore the compressed circuits can be used to generate fermionic Gaussian States as a Variational Quantum Eigensolver (VQE) ansatz for the ground state of any free fermionic Hamiltonian, or as a starting approximation for interacting systems. This was recently demonstrated for Slater determinants[41], Hartree-Fock anzatse[42], and generic fermionic Gaussian states[43].

Since our method does not require the Hamiltonian to be translationally invariant or time independent, the class of free fermionic Hamiltonians that can be compressed is relatively broad. Thus, it may be used to study free fermionic systems after a quench from a clean to a disordered system which show complex dynamics of the entanglement entropy [44]. Certain topological models, e.g. the staggered free fermionic system of the Rice-Mele model [45], are also accessible. There, the adiabatic evolution is characterized by the Chern number of the system; although NISQ quantum computers can measure the Chern number in certain cases [46], adiabatic evolution requires a compressed circuit. Free fermionic systems also emerge as mean field approximations of many physical systems of interest such as superconductivity and charge density waves [47, 48]. With our method, a fixed depth circuit can be generated to simulate *non-equilibrium* mean field theories.

Broadening our perspective beyond time evolution of quantum systems, we have revealed a new method for manipulating and compressing circuit elements that have three relatively simple to check properties. For the time evolution problem discussed in this work, the blocks are composed of 1- and 2-qubit sets of operations that are derived from particular models— but this is not a requirement, and the circuit operations work equally well for larger unitaries as long as the properties of blocks are satisfied. The blocks also do not have to be identical (as can be seen in the Kitaev chain example) and they do not even have to be operating on the same number of qubits (as can be seen in the TFIM example above). This suggests that the ideas outlined above could be incorporated into transpiler software, and that the method we proposed is far more general than Givens rotation based methods listed above.

We anticipate that the ideas developed here can be extended and applied beyond time evolution. In particular, we are investigating whether circuits that arise from the Quantum Approximate Optimization Algorithm may be treated in this fashion. Similarly, an extension to higher dimensions may be possible, enabling simulations on quantum hardware far beyond the current limitations.

[1] R. P. Feynman, International Journal of Theoretical Physics **21**, 467 (1982).

[2] J. Preskill, Quantum **2**, 79 (2018).

[3] N. Khaneja and S. J. Glaser, Chemical Physics **267**, 11 (2001).

[4] B. Drury and P. Love, Journal of Physics A: Mathematical and Theoretical **41**, 395305 (2008).

[5] E. Kökcü, T. Steckmann, J. Freericks, E. F. Dumitrescu, and A. F. Kemper, arXiv preprint arXiv:2104.00728 (2021).

[6] D. Herr, F. Nori, and S. J. Devitt, npj Quantum Information **3**, 35 (2017).

[7] A. Botea, A. Kishimoto, and R. Marinescu, in *Eleventh Annual Symposium on Combinatorial Search* (2018).

[8] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Comm. Math. Phys. **270**, 359 (2007).

[9] A. M. Childs and R. Kothari, Quantum Inf. Comput. **10**, 669 (2010), 0908.4398.

[10] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu, Physical Review X **11** (2021).

[11] A. Chapman and S. T. Flammia, Quantum **4**, 278 (2020).

[12] Y. Atia and D. Aharonov, Nature communications **8**, 1 (2017).

[13] S. Gu, R. D. Somma, and B. Şahinoğlu, arXiv preprint arXiv:2105.07304 (2021).

[14] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Physical Review Letters **114**, 090502 (2015).

[15] G. H. Low and I. L. Chuang, Physical Review Letters **118**, 010501 (2017).

[16] G. H. Low and N. Wiebe, arXiv:1805.00675 (2018).

[17] A. Kalev and I. Hen, arXiv:2006.02539 (2020).

[18] J. Haah, M. B. Hastings, R. Kothari, and G. H. Low, SIAM Journal on Computing , FOCS18 (2021).

[19] C. Cirstoiu, Z. Holmes, J. Iosue, L. Cincio, P. J. Coles, and A. Sornborger, npj Quantum Information **6**, 1 (2020).

[20] B. Commeau, M. Cerezo, Z. Holmes, L. Cincio, P. J. Coles, and A. Sornborger, arXiv preprint arXiv:2009.02559 (2020).

[21] L. Bassman, R. Van Beeumen, E. Younis, E. Smith, C. Iancu, and W. A. de Jong, arXiv preprint arXiv:2103.07429 (2021).

[22] H. N. S. Earp and J. K. Pachos, Journal of mathematical physics **46**, 082108 (2005).

[23] D. Camps, E. Kökcü, L. Bassman, W. A. de Jong, A. Kemper, R. Van Beeumen, *et al.*, arXiv preprint arXiv:2108.03283 (2021).

[24] D. Camps and R. Van Beeumen, F3C (2021), version 0.1.0.

[25] R. Van Beeumen and D. Camps, F3C++ (2021), version 0.1.0.

[26] D. Camps and R. Van Beeumen, QCLAB (2021), version 0.1.2.

[27] R. Van Beeumen and D. Camps, QCLAB++ (2021), version 0.1.2.

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

[29] J. D. Whitfield, J. Biamonte, and A. Aspuru-Guzik, Molecular Physics **109**, 735 (2011).

[30] A. Jayashankar, A. M. Babu, H. K. Ng, and P. Mandayam, Phys. Rev. A **101**, 042307 (2020).

[31] R. Raczka and A. O. Barut, *Theory of group representations and applications* (World Scientific Publishing Company, 1986); R. Gilmore, *Lie groups, physics, and geometry: an introduction for physicists, engineers and chemists* (Cambridge University Press, 2008); B. Hall, *Lie groups, Lie algebras, and representations: an elementary introduction*, Vol. 222 (Springer, 2015).

[32] G. Vidal and C. M. Dawson, Physical Review A **69**, 010301 (2004).

[33] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, Science **309**, 1704 (2005).

[34] R. Barends, A. Shabani, L. Lamata, J. Kelly, A. Mezzacapo, U. Las Heras, R. Babbush, A. G. Fowler, B. Campbell, Y. Chen, *et al.*, Nature **534**, 222 (2016).

[35] M. Born and V. Fock, Zeitschrift für Physik **51**, 165 (1928).

[36] A. Smith, M. Kim, F. Pollmann, and J. Knolle, npj Quantum Information **5**, 1 (2019).

[37] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, Physical review letters **73**, 58 (1994).

[38] W. R. Clements, P. C. Humphreys, B. J. Metcalf, W. S. Kolthammer, and I. A. Walmsley, Optica **3**, 1460 (2016).

[39] J. G. Valatin, Nuovo Cim. **7**, 843 (1958).

[40] N. N. Bogolyubov, Nuovo Cim. **7**, 794 (1958).

[41] I. D. Kivlichan, J. McClean, N. Wiebe, C. Gidney, A. Aspuru-Guzik, G. K.-L. Chan, and R. Babbush, Phys. Rev. Lett. **120**, 110501 (2018).

[42] Z. Jiang, K. J. Sung, K. Kechedzhi, V. N. Smelyanskiy, and S. Boixo, Phys. Rev. Applied **9**, 044036 (2018).

[43] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, S. Boixo, M. Broughton, B. B. Buckley, D. A. Buell, *et al.*, Science **369**, 1084 (2020).

[44] R. Lundgren, F. Liu, P. Laurell, and G. A. Fiete, Phys. Rev. B **100**, 241108 (2019).

[45] M. J. Rice and E. J. Mele, Phys. Rev. Lett. **49**, 1455 (1982).

[46] X. Xiao, J. Freericks, and A. Kemper, arXiv preprint arXiv:2101.07283 (2021).

[47] C. A. Balseiro and L. M. Falicov, Phys. Rev. B **20**, 4457 (1979).

[48] A. Altland and B. D. Simons, *Condensed matter field theory* (Cambridge university press, 2010).