Optimized Quantum Program Execution Ordering to Mitigate Errors in Simulations of Quantum Systems

Teague Tomesh*‡**, Kaiwen Gui†**, Pranav Gokhale‡, Yunong Shi§, Frederic T. Chong¶‡,
Margaret Martonosi*, and Martin Suchara‖†
*Department of Computer Science, Princeton University, Princeton, USA
†Pritzker School of Molecular Engineering, University of Chicago, Chicago, USA
‡Super.tech, Chicago, USA
§Amazon Braket, New York, USA
¶Department of Computer Science, University of Chicago, Chicago, USA
‖Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, USA
**These authors contributed equally to this work; ttomesh@princeton.edu, kgui@uchicago.edu

2021 International Conference on Rebooting Computing (ICRC) | 978-1-6654-2332-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRC53822.2021.00013

***Abstract*—Simulating the time evolution of a physical system at quantum mechanical levels of detail — known as Hamiltonian Simulation (HS) — is an important and interesting problem across physics and chemistry. For this task, algorithms that run on quantum computers are known to be exponentially faster than classical algorithms; in fact, this application motivated Feynman to propose the construction of quantum computers. Nonetheless, there are challenges in reaching this performance potential.**

Prior work has focused on compiling circuits (quantum programs) for HS with the goal of maximizing either accuracy or gate cancellation. Our work proposes a compilation strategy that simultaneously advances both goals. At a high level, we use classical optimizations such as graph coloring and travelling salesperson to order the execution of quantum programs. Specifically, we group together mutually commuting terms in the Hamiltonian (a matrix characterizing the quantum mechanical system) to improve the accuracy of the simulation. We then rearrange the terms within each group to maximize gate cancellation in the final quantum circuit. These optimizations work together to improve HS performance and result in an average 40% reduction in circuit depth. This work advances the frontier of HS which in turn can advance physical and chemical modeling in both basic and applied sciences.

***Keywords*-quantum computing; compilation; program ordering; Hamiltonian simulation**

I. Introduction

The development of quantum computers is advancing rapidly. During the last decade, quantum computing (QC) systems comprised of tens of qubits were brought online for the first time [1]–[3]. Although information processing on these devices is noisy and error-prone, successful execution of several quantum algorithms — including a demonstration of quantum supremacy — has been achieved [1], [4]–[7].

This paper focuses on Hamiltonian simulation (HS) which is used for simulating the time evolution dynamics of a physical system at quantum mechanical detail [8]. This type of simulation was the original motivation behind Feynman's proposal of QC [9]. Lloyd [8] later proved that quantum computers are indeed efficient simulators of quantum systems, implying an exponential separation between quantum and classical algorithms for this problem. When at-scale, fault-tolerant QCs are available, they are expected to efficiently simulate the dynamics of classically intractable chemical and physical systems [10], [11]. HS offers the potential to reveal physical and chemical processes important to materials research, pharmaceuticals, and more [12]–[14].

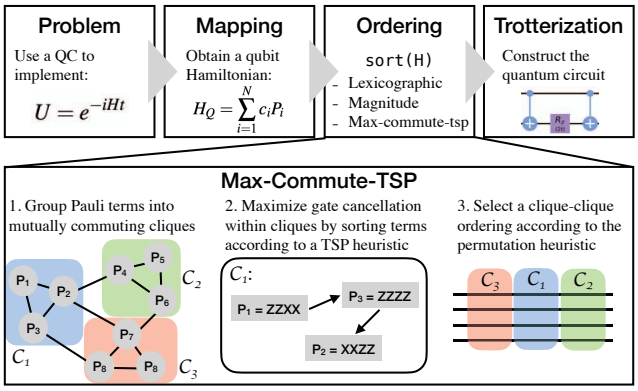

Figure 1: A summary of the Hamiltonian simulation compilation process and the *max-commute-tsp* ordering strategy.

While HS is promising in theory, key challenges in its uptake lie in demonstrating accurate and tractable HS execution on current QCs. For current Noisy Intermediate-Scale Quantum (NISQ) [15] computers, their limited scale precludes the execution of large HS instances, and their noisy operation impedes accuracy.

To implement HS on a quantum computer, one needs to specify a particular problem instance (e.g., selecting a specific molecule or defining an optimization objective) in terms of a Hamiltonian—a characteristic matrix describing the system of interest. Then, the quantum circuit (i.e., quantum program) that simulates the system is compiled via

978-1-6654-2332-8/21/$31.00 ©2021 IEEE
DOI 10.1109/ICRC53822.2021.00013

three steps: mapping, ordering, and Trotterization, shown in Fig. 1. The goal of the *mapping* step is to produce a Hamiltonian specified as a sum of Pauli terms (tensor products of Pauli matrices, more details in Sec. II-B) which act on the qubits of the quantum computer, although the specifics may vary depending on the use case.

The quantum computer executes the HS by sequentially simulating each Pauli term in the order that it appears in the summation produced during the *mapping* step. While any ordering is theoretically viable, some orderings have very poor performance or accuracy given current QC constraints. Thus, the purpose of the *ordering* step is to sort the Pauli terms with the goal of (i) minimizing the depth (i.e., quantum operation count or runtime) of the resulting quantum circuit and (ii) maximizing the accuracy of the simulation. This is analogous to the ordering of floating point operations or basic block scheduling performed by classical compilers [16]. Prior work has explored the impact that term order has on the quality of the simulation, and finding tighter bounds on the simulation error remains an open problem [17]–[20]. Additionally, the impact of term ordering on the gate count requirements for the simulation circuits has also been studied [21], [22].

Finally, the last step in the circuit compilation is *Trotterization* which iteratively constructs the quantum circuit according to the Suzuki-Trotter decomposition [23].

While prior works offered theoretical advances, their term ordering strategies focused solely on either accuracy or gate cancellation. Our work highlights the importance of *both* of these factors and applies classical optimization techniques to compile quantum circuits which are both short and accurate.

We improve the execution of HS circuits by developing methods to **mitigate both physical and algorithmic errors**. *Physical* errors stem from the fact that gate operations on NISQ processors are noisy. Two instances of the same HS problem compiled with different term orderings can result in quantum circuits of different lengths. The shorter circuit contains fewer noisy operations and will, therefore, have a higher probability of successful execution. Even in the fault-tolerant (i.e., error-corrected) regime, where gate operations are noiseless, compilations which produce shorter circuits are still desirable since depth is proportional to runtime. *Algorithmic* errors appear in the Trotterization step (covered in detail in Sec. II-B) because the continuous time evolution that we wish to simulate must be discretized before it can be implemented on a quantum computer. This discretization is only an approximation of the true evolution. Prior work as well as our results indicate that the order in which the Pauli terms are simulated can significantly impact the error of this approximation.

Our strategy (summarized in Fig. 1) for optimizing the program order of the simulation is based on two key insights. First, the algorithmic error associated with the Trotterization step is **due to non-commuting Pauli terms** within the Hamiltonian. Noticing this, we mitigate the algorithmic error by grouping together Pauli terms which commute with one another. We decide the grouping by constructing a graph that indicates which Pauli terms commute with one another, and then find a minimum clique cover on this graph. The second insight follows from the first, namely, once the Pauli terms are partitioned into mutually commuting groups, the **terms within each group can be rearranged without incurring any additional algorithmic error**. We choose to arrange the Pauli terms within each group to maximize the amount of gate cancellation in the final circuit. This is accomplished by finding a travelling salesperson path through each group which places similar Pauli terms next to each other, thus increasing the amount of gate cancellation.

These compilation techniques incorporate both classical optimizations and full-stack knowledge from application to hardware which improves the performance of Hamiltonian simulation. This approach has proven successful in previous work targeting the Variational Quantum Eigensolver (a quantum algorithm which can be considered a specialized case of HS [24], [25]) [26]–[28]. Furthermore, considering the physics at the hardware level — the QC must discretize the simulation to handle non-commuting Pauli terms — guides our optimizations at the compiler level: grouping together commuting Pauli terms. Going in the other direction, we use the compiler to optimize for short circuits which fit within the limits set by the capabilities of the underlying hardware.

In this paper we consider simulation and experiments for molecular Hamiltonians because they are a relevant and important application and also easily obtainable via the NIST Chemistry WebBook [29] and OpenFermion software package [30].

Our contributions in this work include:

- A new term ordering strategy, *max-commute-tsp*, which **simultaneously mitigates both physical and algorithmic errors**.
- Simulation and experimental results which demonstrate an average **40% reduction in circuit depth** and highlight the **importance of both gate cancellation and simulation accuracy** for good overall performance.
- A general, open-source implementation of HS which can be of use to the QC community as a challenging and practical benchmark.

The rest of the paper is organized as follows. An overview of HS is given in Sec. II. Prior ordering strategies and compilation techniques are discussed in Sec. III and a detailed description of *max-commute-tsp* is presented in Sec. IV. Our benchmarking methodology is given in Sec. V and Sec. VI contains the results of our simulations and the evaluations on trapped ion quantum computers. In Sec. VII we discuss our results as well as future work on analyzing HS performance in regimes beyond quantum chemistry.

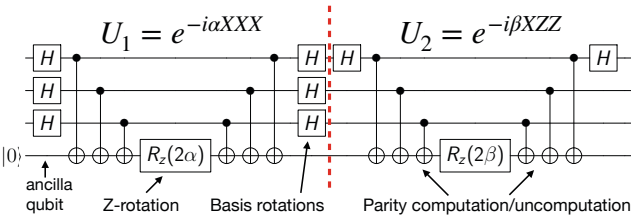

Figure 2: Dissecting a quantum circuit for simulating the evolution of $H = \alpha XXX + \beta XZZ$. The red dotted line separates the two subcircuits implementing each individual term. The full circuit implements the evolution unitary $U = U_2U_1 = e^{-i\beta XZZ}e^{-i\alpha XXX}$.

II. Hamiltonian Simulation

Simulating quantum systems is a general problem with important applications in physics, chemistry, and biology. Many supercomputer hours are spent simulating different molecules and materials each year [31]. With the recent development of quantum computers, and because they are efficient simulators of quantum systems, it is expected that these simulation problems will be among the most promising applications for quantum advantage [11]. However, it is likely that the simulation of classically-intractable systems will require a fault-tolerant quantum computer [10]. Despite this, Hamiltonian simulation is still used throughout many near-term NISQ algorithms for machine learning and other optimization applications. For the remainder of this paper, we choose to consider Hamiltonian simulation within the specific context of simulating molecular dynamics. Our compilation methods can be further applied wherever one wishes to simulate a Hamiltonian written as a sum of Pauli terms.

A. The Simulation Problem

To illustrate Hamiltonian simulation, assume we wish to study a time-dependent quantum system, $|\psi(t)\rangle$, such as an atom or molecule. The time evolution of this system will be described by a matrix, H, called the Hamiltonian whose eigenvalues are the allowed energy levels of the system. If the initial state of the system is $|\psi(0)\rangle$, then the state at a later time is given by the equation

$$|\psi(t)\rangle = U\,|\psi(0)\rangle = e^{-iHt}\,|\psi(0)\rangle\,. \tag{1}$$

The simulation problem is solved by computing the unitary evolution matrix U. In general, this is a difficult task because the size of the Hamiltonian grows exponentially with the size of the system.

B. Quantum Circuits for Hamiltonian Simulation

Exponential scaling with system size is characteristic of quantum mechanics, and it is one of the main reasons why classical methods for solving quantum problems are intractable. However, a quantum computer is capable of representing exponentially large state vectors and evolution matrices using polynomially many qubits and gate operations. Below, we describe the compilation process for constructing the simulation circuits.

Mapping – The molecular Hamiltonians considered here are typically written in terms of fermionic operators which must be mapped to operators that act on qubits [32], [33]. The Hamiltonian can then be written as a sum of Pauli terms:

$$H = \sum_{i=1}^{N} c_i P_i. \tag{2}$$

Here, $c_i \in \mathbb{R}$ and the Pauli terms, P_i, are tensor products of the Pauli matrices with length equal to the number of qubits.

$$P_i = \bigotimes_{n=1}^{n_q} m_n \quad \text{where} \quad m_n \in \{I, X, Y, Z\}$$

Ordering – A quantum computer simulates the evolution Eq. (1) under a Hamiltonian Eq. (2) by sequentially simulating each of the individual terms c_iP_i. Any ordering of the Pauli terms is valid, but some orderings are able to mitigate physical and algorithmic errors more effectively than others. Although the simulation error's dependence on term ordering has been studied extensively [18], [20], [34], finding tight bounds on the error remains an open problem [17], [35]. In addition, the gate requirements for Hamiltonian simulation circuits and numerous circuit optimizations have also been studied, and have shown that different orderings can have different amounts of gate cancellation [10], [22], [36].

In this work, we consider a number of different compilation methods including *lexicographic* [21], [22], *magnitude* [21], [22], *depleteGroups* [20], *random*, and our newly-proposed *max-commute-tsp* term orderings. An overview of each of these strategies is given in Sec. III and *max-commute-tsp* is discussed in detail in Sec. IV.

Trotterization – Once the Hamiltonian has been generated and an ordering is selected, a quantum circuit is constructed which implements the evolution unitary U in Eq. (1). Importantly, if all of the terms in H commute with one another, then U can immediately be written as a product of individual terms

$$U = e^{-iHt} = e^{-ic_1P_1t}e^{-ic_2P_2t}\dots e^{-ic_NP_Nt}$$
$$\text{if } P_jP_k = P_kP_j,\ \forall j,k \in [N].$$

In general, however, real-world Hamiltonians may contain non-commuting terms. In this case, we can use the Suzuki-Trotter decomposition [23] to break the evolution of non-commuting terms into many small time steps to approximate the total evolution unitary [37]:

$$e^{-iHt} \approx (e^{-ic_1P_1\Delta t}e^{-ic_2P_2\Delta t}\dots e^{-ic_NP_N\Delta t})^{t/\Delta t} + O(t\Delta t). \tag{3}$$

We denote $t/\Delta t = r$ as the Trotter number. The product of exponentials in Eq. (3) can be represented as a single quantum circuit, repeated r times, while the remaining

$O(t\Delta t) = O(t^2/r)$ terms, referred to as the Trotter error, is the algorithmic error associated with the Trotterization process. As $r \to \infty$ the Trotter error vanishes and the quantum simulation becomes exact, but this comes at a cost of increasing circuit depth.

Eq. (3) is an example of a first-order Trotter decomposition; higher order decompositions also exist which further improve the accuracy of the approximation (see Sec. 4.7 of [38]), however, for simplicity we consider only the first-order decomposition in this paper.

Fig. 2 shows a simulation circuit for an example Hamiltonian. Both terms require single-qubit gates to rotate each qubit into the computational basis. The parity of the qubits is then computed by performing a CNOT, controlled by a data qubit and targeting an ancilla qubit, for each Pauli matrix in the Pauli term. Another method for computing the parity uses a ladder of CNOTs between nearest neighbors to compute the parity. We choose to use the former method even though it introduces a single ancilla overhead because it allows for additional gate cancellations which are not possible with the ladder implementation.

III. Prior Term Ordering Strategies

Prior work has studied the use of commutativity at the circuit level to optimize the gate counts for the Quantum Approximate Optimization Algorithm (QAOA) [39] and more general quantum circuits [40]. The optimizations in [39] are complementary to the techniques discussed in this paper which operate at the abstraction level of Pauli terms to minimize errors before the application is compiled down into a quantum circuit. In addition, we list here a variety of ordering strategies which have been introduced and their impact on Trotter error or gate costs studied in previous work.

Lexicographic – Prior work has proposed and improved the *lexicographic* ordering which orders Pauli terms alphabetically to achieve high levels of gate cancellation [21], [22]. The *lexicographic* ordering can produce circuits with shorter depth since consecutive Pauli terms which act on the same qubit with the same Pauli matrix result in single- and two-qubit gates cancelling at the interface between the two terms (see Fig. 2). Although this ordering produces short circuits, it does nothing to mitigate Trotter (algorithmic) errors and therefore a larger Trotter number (i.e., longer circuits) will be required to attain high accuracy. Additionally, Sec. IV-B introduces a pathological example where the circuits produced by a *lexicographic* ordering are asymptotically equivalent to no gate cancellation at all.

Magnitude – Prior work also considered sorting the Pauli terms according to the magnitude of their coefficients in Eq. (2) in descending order [21], [22]. Interestingly, the *magnitude* ordering can produce simulation circuits with very low Trotter error, often outperforming the analytically computed bounds on the Trotter error. Tranter et al. [20] suggest that the superior accuracy of the *magnitude* strategy may be attributable to simulating the terms with large coefficients earlier in the circuit and so they cannot compound any errors that occur later on. Despite the low algorithmic errors of a *magnitude* ordering, the resulting circuits will be quite deep because this ordering does not utilize any information related to gate cancellations between terms.

DepleteGroups – The *depleteGroups* strategy was proposed by Tranter et al. [20], which also partitions the Pauli terms into groups where every term commutes with every other term within the group. Once the Pauli terms are grouped into mutually commuting cliques, the final ordering is produced by iteratively selecting the highest magnitude term from each clique until all the groups have been exhausted, which is the opposite of the approach described in Sec. IV.

Random – We also consider a *random* ordering of the Pauli terms to serve as a baseline for comparison. Random term orderings were also used by Childs et. al. [18] to prove stronger bounds on the size of the Trotter error.

IV. Max-Commute-TSP

As shown in Fig. 1, the *max-commute-tsp* ordering is composed of three parts. This algorithm sorts the Hamiltonian by first grouping the Pauli terms into mutually commuting cliques. It then orders the terms within each clique according to a travelling salesperson (TSP) heuristic before heuristically selecting a permutation of the cliques. The following sections describe each of these steps and also motivate why *max-commute-tsp* is able to mitigate both the physical and algorithmic errors in Hamiltonian simulation.

A. Term Grouping to Mitigate Trotter Errors

The first step in the *max-commute-tsp* ordering is to construct a commutation graph. The nodes of the commutation graph represent the Pauli terms of the Hamiltonian, and an edge exists between every pair of Pauli terms which commute with one another. The terms are then grouped into a minimum number of mutually commuting cliques that cover the entire graph.

Recall that a quantum computer is only able to carry out the Hamiltonian simulation by Trotterizing the continuous evolution [23]. This Trotterization step is required to deal with non-commuting terms in the Hamiltonian and incurs Trotter errors, as discussed in Sec. II. In practice, some of the Trotter errors may be avoided by grouping as many commuting terms together as possible.

Theoretical Analysis of Commutation Groupings – We illustrate the effectiveness of the group commutation ordering strategy with an example Hamiltonian containing two commuting cliques. The analysis can be generalized to more commutation groups by the reader. Consider a Hamiltonian $H = \sum_{i=1}^{k} \alpha_i H_i$, where a_i are real numbers and H_i are simple Hamiltonians that can be mapped to quantum circuits

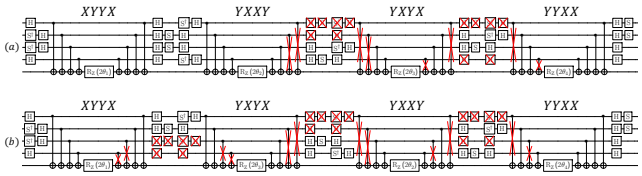

Figure 3: An example showing the gate cancellations that are available with (a) *lexicographic* and (b) TSP orderings.

(or diagonalized) directly. Suppose H can be divided into two commuting groups (cliques) $H_1^c = \sum_{m=1}^{p} H_m$ and $H_2^c = \sum_{m=p+1}^{k} H_m$, *i.e.*, $[H_m, H_n] = 0$ if $0 < m, n \leq p$ or $p < m, n \leq k$. We compare the Trotter error of a term grouping strategy against other orderings below.

For a *group commutation ordering* of H, it has been shown that the approximation error (in the additive form) of the Lie-Trotter formula is given by the variation-of-parameters formula [34], [41], [42],

$$\begin{aligned}\delta_{gc} &= e^{-it(H_1^c)}e^{-it(H_2^c)} - e^{-itH} \\ &= \int_0^t d\tau e^{-i(t-\tau)H}[e^{-i\tau H_1^c}, H_2^c]e^{\tau H_2^c}. \end{aligned} \tag{4}$$

Note that the error form also applies to general H^g where $H^g = H_1^g + H_2^g$. For simplicity, we denote the integral in Eq. (4) as $I(H_1^g, H_2^g)$ for general H_1^g, H_2^g. Thus, we can simply write the approximation error as $\delta_{gc} = I(H_1^c, H_2^c)$.

We are interested primarily in the operator norm (*i.e.*, spectral norm) of $||\delta_{gc}|| = ||I(H_1^c, H_2^c)||$, which gives the worst-case analysis of the error.

We can recursively apply the error formula in Eq. (4) to the Lie-Trotter formula for an *arbitrary ordering* of H. Let π be a permutation of the set $\{1, .., p\}$ that defines the ordering. First, we can approximate e^{-itH} by separating $H_{\pi(1)}$ from other terms:

$$\begin{aligned}\delta_1 &= e^{-itH_{\pi(1)}}e^{-it\sum_{m=2}^{k} H_{\pi(m)}} - e^{-itH} \\ &= I(H_{\pi(1)}, \sum_{m=2}^{k} H_{\pi(m)}).\end{aligned}$$

Then we can recursively repeat the process for the rest of the Hamiltonian and arrive at the following expression for the approximation error δ of the Lie-Trotter formula.

$$\begin{aligned}\delta_{ngc} &= e^{-itH_{\pi(1)}}e^{-itH_{\pi(2)}}...e^{-itH_{\pi(2)}} - e^{-itH} \\ &= I(H_{\pi(1)}, \sum_{m=2}^{p} H_{\pi(m)}) + e^{-itH_{\pi(1)}} I(H_{\pi(2)}, \sum_{m=3}^{p} H_{\pi(m)}) \\ &+ e^{-itH_{\pi(1)}}e^{-itH_{\pi(2)}} I(H_{\pi(3)}, \sum_{m=4}^{p} H_{\pi(m)}) + ... \\ &+ e^{-itH_{\pi(1)}}...e^{-itH_{\pi(j)}} I(H_{\pi(j+1)}, \sum_{m=j+2}^{p} H_{\pi(m)}) + ...\end{aligned}$$

Using the triangle inequality and the submultiplicativity of the operator norm, together with the fact that the operator norm of a unitary is 1, we have

$$\begin{aligned}||\delta_{ngc}|| \approx ||I(H_{\pi(1)}, \sum_{m=2}^{p} H_{\pi(m)}) + I(H_{\pi(2)}, \sum_{m=3}^{p} H_{\pi(m)}) \\ + ... + I(H_{\pi(j+1)}, \sum_{m=j+2}^{p} H_{\pi(m)}) + ...||.\end{aligned}$$

Also we know that $H_{\pi(k)}$ is either in H_1^c or H_2^c. Thus, we have

$$\begin{aligned}||\delta_{ngc}|| \approx || \sum_{\pi(j)\in[1,p]} I(H_{\pi(j)}, H_2^c)|| + \\ || \sum_{\pi(j)\in[p+1,k]} I(H_{\pi(j)}, H_1^c)||.\end{aligned}$$

Although we have no proof that $||\delta_{gc}|| < ||\delta_{ngc}||$ (because we do not have information about the full commutation relation and magnitude information in H), we can, however, make several observations why group commutation ordering is advantageous. First, $||\delta_{gc}||$ in general has a much lower upper bound than does the first term in $||\delta_{ngc}||$. In fact, the upper bound of $||\delta_{gc}||$ does not scale with the number of terms p while $||I(H_{\pi(1)}, \sum_{m=2}^{p} H_{\pi(m)})||$ is of $O(p)$. Second, $||\delta_{gc}||$ does not include the second term in $||\delta_{ngc}||$. Thus, there is strong evidence that *group commutation* ordering has an advantage over naive ordering in terms of eliminating Trotter errors.

Algorithms for Min-Clique-Cover – Partitioning the commutation graph of a quantum Hamiltonian has been proposed for other quantum computing applications, including Hamiltonian simulation [25] and minimizing the number of measurements required in variational algorithms such as the Variational Quantum Eigensolver [26]–[28].

The minimum clique cover problem is NP-Complete [43]. Fortunately, we do not require the exact min-clique-cover solution for the purpose of this work since approximate solutions are able to effectively mitigate both algorithmic and physical errors. The commutation graphs produced by real-world applications (especially molecular Hamiltonians) tend to be highly structured allowing for reasonably good solutions to be found in time scaling like $O(N^2)$ or $O(N^3)$ for a graph with N nodes [44]–[46].

B. Travelling Salesperson For Gate Cancellation

Because the terms within each clique commute with one another, they can be rearranged at will without incurring any additional Trotter error. Therefore, the Pauli terms within cliques can be sorted into an order which maximizes gate cancellation without the worry of degrading the accuracy of the simulation.

We frame the question of ordering the Pauli terms within cliques as an instance of the Travelling Salesperson problem. Recall the example simulation circuit in Fig. 2, we characterize the potential for gate cancellation between two terms using the following principles:

- Two CNOT gates with the same target and control qubits can be cancelled so long as no single-qubit gates lie between them.
- Sequential Pauli terms which act on the same qubits with the same Pauli matrix will be able to cancel the basis rotation single-qubit gates between them.

According to these gate cancellation principles, it will be favorable to place two Pauli terms next to each other if they operate on the same qubits with the same Pauli matrices. This can be framed as an instance of the Travelling Salesperson problem, where instead of distance travelled the cost function is the number of two-qubit gates in the final circuit.

We focus primarily on the cancellation of two-qubit gates, because they are the dominant source of errors and latency in the prevalent quantum platforms. Typically, CNOT gates have at least 10x lower fidelity and 2–5x longer duration than single-qubit gates [47], [48].

Defining the TSP Instance – The TSP objective is to order a set of k Pauli terms, $\{P_1, P_2, ..., P_k\}$, such that the number of CNOT gates in the resulting circuit is minimized.

Every Pauli term within the Hamiltonian corresponds to a subcircuit implemented on the quantum computer. Within a Pauli term, each non-I Pauli matrix generates two identical CNOTs (one for parity-compute, the other for parity-uncompute). For example, the Pauli term: XZZ in Fig. 2 has a total of 6 CNOTs in its subcircuit because it has three non-I Pauli matrices. These CNOTs are controlled on the m-th qubit (where $m =$ the index of the Pauli matrix within the Pauli term) and target the ancilla qubit. Without any gate cancellation, the number of CNOT gates required to implement the simulation circuit for a Hamiltonian, $H = \sum_{j=1}^{k} c_j P_j$, is

$$2\sum_{j=1}^{k} |P_j|_{\text{Ham}} = 2\sum_{j=1}^{k}\sum_{i=1}^{N} \mathbb{1}_{P_j[i]\neq I}, \tag{5}$$

where N denotes the width of the Pauli terms and Ham refers to Hamming weight.

A good permutation, however, can substantially reduce the number of required CNOTs compared to Eq. (5)'s upper bound because the CNOT gates between neighboring Pauli term subcircuits can cancel with one another. Using the gate cancellation principles listed above, we define the CNOT distance between two Pauli terms as

$$\begin{aligned} |P_1 - P_2|_{\text{CNOT}} &:= |P_1 - P_2|_{\text{Ham}} + \sum_{i\in[N]} \mathbb{1}_{I\ \neq P_1[i]\neq P_2[i]\neq I} \\ &= \sum_{i\in[N]} \mathbb{1}_{P_1[i]\neq P_2[i]}(1 + \mathbb{1}_{I\notin\{P_1[i],P_2[i]\}}). \end{aligned} \tag{6}$$

This distance reports the number of CNOTs needed to implement the transition from the parity-uncompute zone of the P_1 subcircuit through the parity-compute zone of the P_2 subcircuit, after all possible gates have been cancelled.

The TSP instance is then defined as: given a graph, with nodes representing Pauli terms and edges between nodes weighted according to the CNOT distance, find the shortest cycle which visits each vertex once. To be precise, we actually desire the shortest Hamiltonian path, e.g., we want to visit each Pauli term once, without returning to the start. This is accomplished by generating the TSP cycle and then deleting the most expensive edge in the path.

Approximating TSP – Solving TSP in the most general setting is NP-hard. Moreover, no polynomial-time algorithms exist which are guaranteed to approximate it to a constant ratio [49]. In the case of *metric graphs*, however, TSP can be efficiently 1.5-approximated via Christofides' algorithm, meaning that the approximation will return an ordering that requires at most 1.5x as many CNOTs as the optimal ordering [50]. In addition, Christofides' algorithm is fast, running in $O(k^3)$ time, and it is known to perform well in practice, often attaining near-optimal solutions [51].

We now prove that the graph defined by the $|P_1 - P_2|_{\text{CNOT}}$ distance function is a metric graph. We have already seen that it is symmetric in its arguments (i.e., the graph is undirected), so we need only show that it satisfies the triangle inequality:

$$|P_1 - P_2|_{\text{CNOT}} + |P_2 - P_3|_{\text{CNOT}} - |P_1 - P_3|_{\text{CNOT}} \geq 0.$$

Expanding this expression we obtain:

$$\begin{aligned} &|P_1 - P_2|_{\text{CNOT}} + |P_2 - P_3|_{\text{CNOT}} - |P_1 - P_3|_{\text{CNOT}} \\ &\quad= \sum_{i\in[N]} \mathbb{1}_{P_1[i]\neq P_2[i]}(1 + \mathbb{1}_{I\notin\{P_1[i],P_2[i]\}}) \\ &\qquad +\mathbb{1}_{P_2[i]\neq P_3[i]}(1 + \mathbb{1}_{I\notin\{P_2[i],P_3[i]\}}) \\ &\qquad -\mathbb{1}_{P_1[i]\neq P_3[i]}(1 + \mathbb{1}_{I\notin\{P_1[i],P_3[i]\}}). \end{aligned} \tag{7}$$

We will prove that each three-term expression in the sum is non-negative for each i, so that the full sum must also be non-negative. Note that the third term evaluates to 0, -1, or -2.

- If it is 0, the three-term expression is already non-negative since the first two terms are non-negative.

- If it is -1, then $P_1[i] \neq P_3[i]$, and one of the two is I. We must also have $P_1[i] \neq P_2[i]$ or $P_2[i] \neq P_3[i]$, so the first two terms must sum to at least 1. Thus, the three-term expression is non-negative.
- If it is -2, then $P_1[i] \neq P_3[i]$, and neither is I. Suppose that $P_1[i] = P_2[i]$; then the first term is +2, and the three-term expression is non-negative. Similarly, if $P_2[i] = P_3[i]$, then the second term is +2, and the three-term expression is non-negative. And if $P_1[i] \neq P_2[i] \neq P_3[i]$, then the first two terms are both at least +1, so the three-term expression is non-negative.

Thus, we conclude that Eq. (7) is a sum over non-negative numbers which proves that the triangle inequality holds for the CNOT distance. Therefore, our graph is metric, and Christofides' algorithm can be used to efficiently attain a 1.5-approximation to the optimal TSP.

1. XX|XXXXXXXX	1. XX|XXXXXXXX
2. XY|XXXZZZZZ	2. XZ|XXXXXXXZ
3. XZ|XXXXXXXZ	3. YY|XXXXXXZZ
4. YX|XXZZZZZZ	4. ZX|XXXXXZZZ
5. YY|XXXXXXZZ	5. ZZ|XXXXZZZZ
6. YZ|XZZZZZZZ	6. XY|XXXZZZZZ
7. ZX|XXXXXZZZ	7. YX|XXZZZZZZ
8. ZY|ZZZZZZZZ	8. YZ|XZZZZZZZ
9. ZZ|XXXXZZZZ	9. ZY|ZZZZZZZZ
(a) *Lexicographic*	(b) TSP

Figure 4: Pathological example of *lexicographic*'s suboptimality compared to TSP.

Advantage Over Lexicographic Ordering – Although the *lexicographic* ordering often leads to large amounts of gate cancellation, it does not achieve the optimality of TSP over the entire Hamiltonian. As an example, consider the 4-qubit Hamiltonian with lexicographically ordered strings $[XXXX, XXYY, XYXY, XYYX, YXXY, YXYX, YYXX, YYYY]$. These eight commuting strings arise in the Jordan-Wigner encoding for molecules, so this example is ubiquitous [26], [37]. Applying Eq. (5), we see that $8 \times 2 \times 4 = 64$ CNOTs are needed prior to gate cancellation. After CNOT cancellation, summing Eq. (6) along the lexicographic order gives a total of 40 CNOTs. Now consider the TSP order $[XXXX, XXYY, XYXY, XYYX, YXYX, YXXY, YYXX, YYYY]$, which flips the fifth and sixth terms from the lexicographic order. Under TSP, we are able to generate circuits with only 36 CNOTs. See Fig. 3 for an example of the gate cancellations that are available in these two cases. In summary, unoptimized to *lexicographic* to TSP have CNOT costs of $64 \rightarrow 40 \rightarrow 36$.

In certain cases, TSP can have an even greater factor of improvement over *lexicographic* ordering. As an example, consider the nine Pauli terms in Fig. 4, sorted by *lexicographic* and TSP orderings.

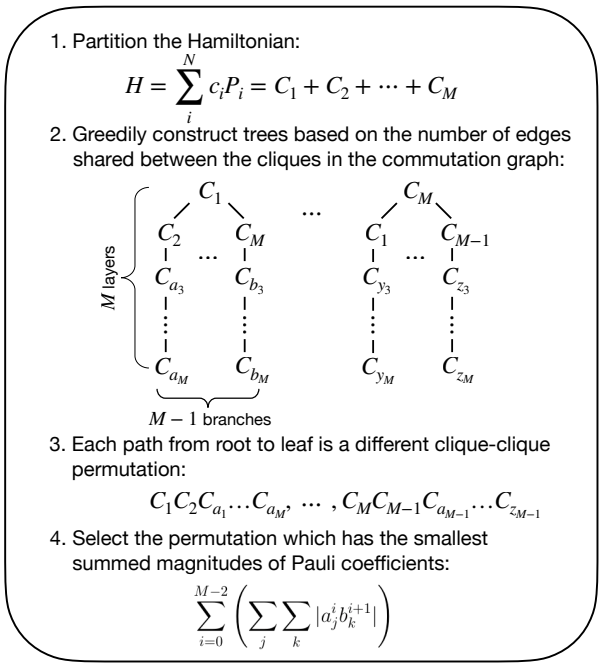

Figure 5: Polynomial time heuristic for selecting a clique-clique ordering. (1) The commutation graph, which was constructed to partition the Hamiltonian into M cliques, is (2) utilized to greedily construct many trees based on the number of edges shared between cliques. (3) Each traversal of a tree from root to leaf produces a different clique-clique permutation. (4) The permutation with the smallest commutator magnitude (Eq. (8)) is selected.

Without gate cancellation, $9 \times 10 \times 2 = 180$ CNOTs are required. Under the *lexicographic* order, gate cancellation yields 112 CNOTs. However, with reordering into the TSP route, only 62 CNOTs are needed. While this particular example is pathological, it demonstrates scenarios where *lexicographic* ordering is asymptotically identical to no-gate cancellation, but TSP achieves an asymptotic advantage. This advantage is from $O(N^2)$ to $O(N \log N)$, which is a significant improvement.

C. Ordering the Cliques

Finally, the last step in *max-commute-tsp* is to select the order in which the cliques, produced via the minimum clique cover from Sec. IV-A, are simulated. Different orderings of the cliques can incur different Trotter errors by the same reasoning given above for the varying Pauli term orderings. Determining the optimal clique-clique ordering is intractable as it would require knowledge of the nested commutators between each of the cliques after the Baker–Campbell–Hausdorff expansion, i.e., an exponential amount of computation. Therefore, we use a heuristic, which runs in polynomial time, to decide upon a clique-clique

ordering that relies on the approximated commutator: the first-order approximated difference between U_{approx} and U_{exact} after the Taylor expansion. Note that, for two cliques $C_1 = \sum_{i=1}^{N_1} a_i A_i$ and $C_2 = \sum_{i=1}^{N_2} b_i B_i$, where $a_i, b_i \in \mathbb{R}$ and A_i, B_i are Pauli terms, the commutator between them is

$$[C_1, C_2] = C_1 C_2 - C_2 C_1 = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} a_i b_j (A_i B_j - B_j A_i). \tag{8}$$

Rather than compute the commutator between cliques exactly, the heuristic, shown in Fig. 5, exploits the information stored in the commutation graph that was used to group the Pauli terms into fully commuting cliques. Counting the number of edges between C_1 and C_2 in the commutator graph indicates the number of terms within Eq. (8)'s sum that evaluate to zero. The heuristic uses the number of inter-clique edges to greedily grow a tree (the nodes of the tree representing cliques) where each path through the tree corresponds to a separate clique-clique ordering. Intuitively, a permutation selected in this manner will produce commutators between the consecutive cliques that contain many zero terms. One can hope that this will reduce the overall magnitude of the commutator and therefore contribute very little to the overall Trotter error.

For each clique produced in the minimum clique cover, the heuristic constructs the tree described above with the current clique as the root. It then traverses each of these trees, producing a set of possible clique-clique permutations. Finally, for each permutation, compute $\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} |a_i b_j|$ over the non-zero terms and select the permutation with the smallest value, which one would expect to have the smallest contribution to the Trotter error. For a Hamiltonian which is partitioned into M cliques, this heuristic has a runtime of $O(M^4)$ and produces $O(M^2)$ different permutations.

V. Methodology

We compare the performance of different ordering strategies by measuring their ability to mitigate both physical and algorithmic errors via simulation and real device executions. A benchmark set of 79 molecular Hamiltonians was generated using the OpenFermion software package [30] and the NIST Chemistry WebBook [29].

For each of the ordering strategies we simulate the time dynamics of the benchmark Hamiltonians and at every time step we increase the Trotter number r until the HS achieves an error $\epsilon < 0.1$. The accuracy of a Hamiltonian simulation is measured using the diamond distance [52]–[54]. The diamond distance between two quantum processes $\mathcal{E}$ and $\mathcal{F}$ is defined as:

$$\begin{aligned} d_\diamond(\mathcal{E}, \mathcal{F}) := \epsilon &= \|\mathcal{E} - \mathcal{F}\|_\diamond \\ &= \max_\rho \|(\mathcal{E} \otimes \mathbb{1})\rho - (\mathcal{F} \otimes \mathbb{1})\rho\|_1, \end{aligned} \tag{9}$$

where ρ is the density matrix representation of a quantum state, $\mathbb{1}$ is the identity operator acting on the same size Hilbert space as $\mathcal{E}$ and $\mathcal{F}$, and $\|\ldots\|_1$ denotes the trace norm. The diamond distance is an important and commonly used metric for distinguishing between two quantum processes in the absence of noise (i.e., considering only algorithmic errors) [54], [55]. Once the error threshold is met we report the number of CNOT gates required in the final quantum circuit. The results of these noiseless simulations are presented in Sec. VI-A.

To capture the combined effects of algorithmic and physical errors on hardware execution we use noisy simulations with a depolarizing error model $\mathcal{E}(\rho) = \frac{pI}{2} + (1-p)\rho$ which adds a noise channel on the two-qubit gates [38]. Each entangling gate has probability p of depolarizing its control and target qubits (replacing the qubits with the completely mixed state $I/2$), and probability $1-p$ of leaving the qubits untouched.

We compare the output distributions of the noisy simulations and hardware executions using the Hellinger infidelity [56]–[58]. The Hellinger infidelity is defined as $1 - H_D(P,Q)$, where $H_D(P,Q)$ is the Hellinger distance, defined for two probability distributions P and Q as

$$H_D(P,Q) = \frac{1}{\sqrt{2}} ||\sqrt{P} - \sqrt{Q}||_2. \tag{10}$$

Eq. (10) is preferable for these experiments because they output probability distributions. Additionally, the diamond distance involves a maximization over all quantum states ρ to capture the worst-case difference between two quantum channels, but for experiments on real hardware a specific initial state must be chosen.

All of the code used to generate and test the benchmarks is available online in a Github repository [59].

VI. Benchmark Results

A. Noiseless Simulation

In both the NISQ and fault-tolerant regimes, circuit depth and total gate count are important metrics of comparison for quantum circuits. In the NISQ era, these metrics are closely related to the probability that a circuit will execute successfully or succumb to the effects of noise. In the fault-tolerant regime, circuit depth and gate count are directly linked to the runtime of quantum programs. In the case of Hamiltonian simulation, a higher Trotter number r provides a more accurate simulation, but the total gate cost scales proportionally with r.

In Fig. 6 we report the results of noiseless simulations of the time dynamics of the benchmark molecular Hamiltonians. At every time step we require that the HS achieve an error $\epsilon < 0.1$, increasing r until this constraint is satisfied, and then computing the final number of CNOT gates required. In Fig. 6a *max-commute-tsp* is able to simultaneously mitigate both algorithmic and physical errors.

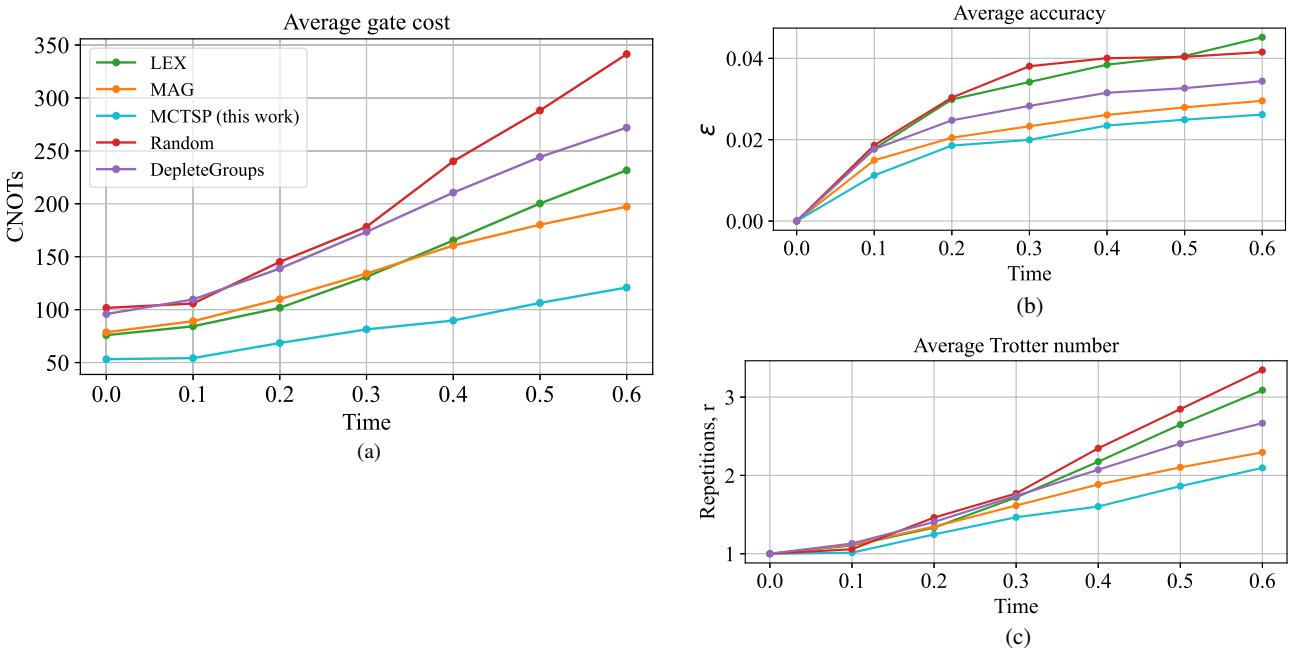

Figure 6: Noiseless simulations of the time dynamics of 79 molecular Hamiltonians with an error threshold $\epsilon < 0.1$. Each molecule is represented with 4 qubits, and at every time step the Trotter number r is increased until the diamond distance between the HS quantum circuit and the exact evolution unitary is below the error threshold.

Compared to the other ordering strategies, *max-commute-tsp* is able to produce accurate HS circuits using fewer entangling gates. On average, both the *max-commute-tsp* and *magnitude* orderings require similar Trotter numbers to reach the error threshold $\epsilon < 0.1$ as shown in Fig. 6c. However, the TSP ordering of the Pauli terms within cliques allows the *max-commute-tsp* strategy to cancel an average of 39% more gates than the *magnitude* ordering. The average 40% difference in gate count between the *lexicographic* and *max-commute-tsp* strategies is mostly due to the higher Trotter numbers needed for the *lexicographic* ordering to surpass the accuracy threshold.

The need to mitigate both sources of error is made apparent by comparing the results of the *lexicographic* and *magnitude* orderings. The *lexicographic* HS circuits cancel many gates and are able to mitigate physical errors quite well. The *magnitude* ordering is well suited to the regime of molecular Hamiltonians and is able to achieve high accuracy with low Trotter number. In Fig. 6a, at small t values the poor accuracy of the *lexicographic* ordering is compensated by its ability for gate cancellation, but for $t > 0.35$ the *magnitude* ordering is able to produce shorter HS circuits because it requires a smaller Trotter number.

B. Noisy Simulation and Hardware Evaluation

We conducted noisy simulations of each benchmark Hamiltonian under a depolarizing error model using the *lexicographic*, *magnitude*, and *max-commute-tsp* orderings with an initial state $\psi_{init} = \frac{1}{\sqrt{2}}(|0011\rangle + |1100\rangle)$. In addition, six small benchmark Hamiltonians were evaluated on trapped ion processors. For each simulation and hardware execution the time and Trotter number parameters were set to $t = 1$ and $r = 1$ due to the error rates of current NISQ hardware. The increased noise incurred by the deeper circuits for $r > 1$ quickly overwhelms the results of the computation.

Fig. 7 shows the measured distribution of Hellinger infidelities across four different error rates. The *max-commute-tsp* ordering produces HS circuits with lower infidelity on average and also attains a minimum infidelity that is 1.2, 2.4, 3.6, 6.7% lower than the minimum achieved by either the *lexicographic* or *magnitude* orderings for each of the 0.1, 0.5, 1.0, 2.0% error rates, respectively.

The results of the trapped ion experiments are contained in Table I. The ethene (C_2H_4) benchmark was run on a 7-qubit device [60] (4000 shots) while the remainder utilized an IonQ device with 11-qubits [3] (1000 shots each). Each benchmark circuit was prepared in the initial state ψ_{init} and consists of 4 data qubits and 1 ancilla qubit. Both processors have average entangling gate errors near 2%. The experimental results are in shown in Fig. 7 as the stars (7-qubit) and crosses (11-qubit). The hardware experiments confirm the results of the noisy-simulations and also highlight the impact of algorithmic errors — even for current NISQ processors. For the Cl_2, C_2H_2, F_2, and N_2 benchmark molecules the *magnitude* ordering, which produced deeper quantum

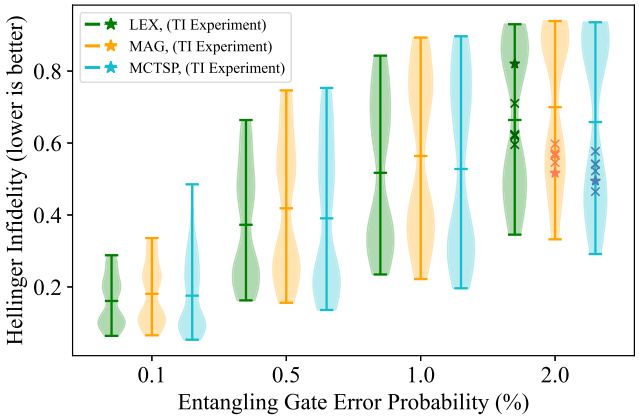

Figure 7: Distribution of Hellinger infidelities after evolving the entangled initial state ψ_{init}. The solid lines making up each violin plot denote the max, mean, and min of the distribution. The stars and crosses indicate experimental results obatained via a 7- and 11-qubit ion trap processor, respectively [3], [60].

Molecule	Diamond Dist. (ϵ) *lex*, *mag*, *mctsp*	2-qubit gates *lex*, *mag*, *mctsp*	Hellinger Inf. (%) *lex*, *mag*, *mctsp*
C_2H_4	2.9, 1.8e-3, 1.8e-3	55, 49, 41	75.9, 55.2, **53.8**
Cl_2	1.9, 1.4e-4, 1.4e-4	47, 53, 37	62.2, 56.6, **54.2**
C_2H_2	1.9, 1.3e-3, 1.3e-3	47, 53, 37	62.4, **57.2**, 57.7
F_2	1.1, 3.0e-3, 3.0e-3	47, 53, 37	71.0, 56.7, **52.2**
N_2	1.7, 1.2e-3, 1.2e-3	47, 53, 37	61.9, 54.6, **54.1**
O_2	3.1, 4.1e-3, 4.1e-3	41, 41, 27	59.6, 59.7, **46.5**

Table I: Diamond distances, gate counts, and Hellinger infidelities for the benchmarks evaluated on ion trap QPUs.

circuits, attained lower infidelities than the *lexicographic* ordering. The difference in performance can be attributed to the high (low) diamond distance between the *lexicographic* (*magnitude*) ordering and the ideal evolution. Interestingly, both strategies produced circuits of equal depth for the O_2 benchmark and achieved similar Hellinger infidelities despite a large difference in diamond distance. This may be due to the specific choice of initial state since the diamond distance is a measure of error over all initial states while the reported Hellinger infidelity is measured with respect to a single initial state.

VII. Conclusion & Future Directions

In this work, we introduced a new compilation method for Pauli term ordering, *max-commute-tsp*, which is able to **simultaneously mitigate algorithmic and physical errors** in the quantum circuits that perform Hamiltonian simulations. Our hardware-level insights into the nature of the algorithmic errors guided us toward a solution which exploits the commutativity between Pauli terms to mitigate these errors. Additionally, *max-commute-tsp* is able to mitigate physical errors by observing that the Pauli terms within each clique can be reordered to maximize gate cancellation via a TSP heuristic. While maintaining the same accuracy, *max-commute-tsp* is able to produce HS circuits that are 39% shorter than other compilation methods. Additionally, we use realistic noise models and real device executions to demonstrate the combined importance of algorithmic and physical error mitigation.

It is important to note that, in the case of Hamiltonian simulation, even small improvements in accuracy can have significant impact. We point out two specific reasons below:

- When concatenating many instances of the HS circuits together (i.e., for HS with Trotter number $r > 1$), errors will accumulate with each repetition of the circuit. Therefore, reducing the error of the HS subcircuit can have an exponential impact on the final accuracy. For variational algorithms (such as VQE and QAOA), the same circuit is executed many times with classical post-processing in between. Reducing the error within this circuit will at least linearly increase the accuracy of any estimated observables.
- For quantum chemistry applications, a small reduction in error is likely to have a substantial effect. An improvement in quantum circuit fidelity would similarly improve a quantum algorithm's ability to compute exact ground state energies. Most classical approaches utilize approximation algorithms to estimate the ground state energy (since an exact diagonalization requires exponential classical computing resources) and even single digit improvements in accuracy are highly desirable [61], [62].

Our open-source software repository [59] allows users to construct HS circuits compiled via any of the ordering strategies considered here. This general compilation framework can be easily adapted to specific problem instances. For example, one can perform *max-commute-tsp* on a partial set of the Pauli terms if the user has prior knowledge that the partial sets tend to share similar physical properties.

Future work will involve developing Hamiltonian simulations for regimes other than the molecular systems studied here (e.g., solid-state structures [63], high energy physics [64], protein folding [65], etc.). The varied physical properties of these systems may result in different correlations between Pauli terms. These regimes may by difficult for inflexible strategies such as *magnitude* or *lexicographic* because most of their development has been centered around molecular Hamiltonians. Recent work on QAOA [66] suggests the use of complicated mixing Hamiltonians that consist of many non-commuting Pauli terms, which would invoke Trotter errors during evolution. A flexible ordering strategy like *max-commute-tsp* would significantly reduce such Trotter errors, while also keeping the circuits short, improving the performance of QAOA and expanding the domain of tractable problems.

ACKNOWLEDGMENT

The authors would like to thank Nathan Wiebe for the introduction to HS and for helpful discussions on commuting Pauli terms, Pavel Lougovski for initial thoughts and helpful discussions, and Yipeng Huang, Xiaoliang Wu and Zain Saleem for helpful comments. We also want to thank Alaina Green, Norbert Linke and the rest of the UMD group for access and help with running the 7-qubit ion trap experiments.

This work was funded in part by EPiQC, an NSF Expedition in Computing, under grants CCF-1730082 / 1730449 / 1730082, by the NSF STAQ project under grant NSF Phy-1818914, and by DOE grants DE-SC0020289 and DE-SC0020331; and in part by NSF OMA-2016136 and the Q-NEXT DOE NQI Center. Y.S. was also funded in part by the NSF QISE-NET fellowship under grant number 1747426. The work of K.G. and M.S. is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Fusion Energy Sciences, under Award Number DE-SC0020249. This work was also funded by the US Department of Energy Office, Advanced Manufacturing Office (CRADA No. 2020-20099.) and by the NSF under Grant No. 2110860.

CONFLICTS OF INTEREST

Fred Chong is Chief Scientist at Super.tech and an advisor to Quantum Circuits, Inc.

REFERENCES

[1] F. Arute, K. Arya, R. Babbush, D. Bacon, Joseph C. Bardin, R. Barends, R. Biswas, S. Boixo, Fernando G. S. L. Brandão, David A. Buell, Brian J. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, M. Hartmann, A. Ho, M. Hoffmann, T. Huang, T. Humble, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. Klimov, S. Knysh, A. Korotkov, F. Kostritsa, D. Landhuis, Mike Lindmark, E. Lucero, D. Lyakh, Salvatore Mandrà, J. McClean, M. McEwen, A. Megrant, X. Mi, K. Michielsen, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Y. Niu, E. Ostby, A. Petukhov, John C. Platt, C. Quintana, E. Rieffel, P. Roushan, N. Rubin, D. Sank, K. Satzinger, V. Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, A. Vainsencher, Benjamin Villalonga, T. White, Z. Yao, P. Yeh, Adam Zalcman, H. Neven, and J. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574:505–510, 2019.

[2] Antonio D Córcoles, Abhinav Kandala, Ali Javadi-Abhari, Douglas T McClure, Andrew W Cross, Kristan Temme, Paul D Nation, Matthias Steffen, and JM Gambetta. Challenges and opportunities of near-term quantum computing systems. *arXiv preprint arXiv:1910.02894*, 2019.

[3] Kwame-Lante Wright, K. Beck, S. Debnath, J. Amini, Y. Nam, N. Grzesiak, Jiah-Shing Chen, N. C. Pisenti, M. Chmielewski, C. Collins, K. Hudek, J. Mizrahi, J. D. Wong-Campos, S. Allen, J. Apisdorf, P. Solomon, M. Williams, A. M. Ducore, A. Blinov, S. Kreikemeier, V. Chaplin, M. Keesan, C. Monroe, and J. Kim. Benchmarking an 11-qubit quantum computer. *Nature Communications*, 10, 2019.

[4] R. Barends, A. Shabani, L. Lamata, J. Kelly, A. Mezzacapo, U. Las Heras, R. Babbush, A. Fowler, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, E. Jeffrey, E. Lucero, A. Megrant, J. Mutus, M. Neeley, C. Neill, P. O'Malley, C. Quintana, P. Roushan, D. Sank, A. Vainsencher, J. Wenner, T. White, E. Solano, H. Neven, and J. Martinis. Digitized adiabatic quantum computing with a superconducting circuit. *Nature*, 534 7606:222–6, 2016.

[5] Cornelius Hempel, C. Maier, Jonathan Romero, J. McClean, T. Monz, H. Shen, P. Jurcevic, B. Lanyon, P. Love, R. Babbush, Alán Aspuru-Guzik, R. Blatt, and C. Roos. Quantum chemistry calculations on a trapped-ion quantum simulator. *Physical Review X*, 8, 2018.

[6] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5:4213, 2014.

[7] Teague Tomesh, Pranav Gokhale, Eric R Anschuetz, and Frederic T Chong. Coreset clustering on small quantum computers. *Electronics*, 10(14):1690, 2021.

[8] Seth Lloyd. Universal quantum simulators. *Science*, pages 1073–1078, 1996.

[9] Richard P Feynman. Simulating physics with computers. *Int. J. Theor. Phys*, 21(6/7), 1982.

[10] Andrew M Childs, Dmitri Maslov, Yunseong Nam, Neil J Ross, and Yuan Su. Toward the first quantum simulation with quantum speedup. *Proceedings of the National Academy of Sciences*, 115(38):9456–9461, 2018.

[11] Leonie Mueck. Quantum reform. *Nature chemistry*, 7(5):361–363, 2015.

[12] Alán Aspuru-Guzik, Anthony D Dutoi, Peter J Love, and Martin Head-Gordon. Simulated quantum computation of molecular energies. *Science*, 309(5741):1704–1707, 2005.

[13] Daniel A Lidar and Haobin Wang. Calculating the thermal rate constant with exponential speedup on a quantum computer. *Physical Review E*, 59(2):2429, 1999.

[14] Markus Reiher, Nathan Wiebe, Krysta M Svore, Dave Wecker, and Matthias Troyer. Elucidating reaction mechanisms on quantum computers. *Proceedings of the National Academy of Sciences*, 114(29):7555–7560, 2017.

[15] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[16] Andrew W Appel. *Modern compiler implementation in C*. Cambridge university press, 2004.

[17] Ryan Babbush, Jarrod McClean, Dave Wecker, Alán Aspuru-Guzik, and Nathan Wiebe. Chemical basis of trotter-suzuki errors in quantum chemistry simulation. *Physical Review A*, 91(2):022311, 2015.

[18] Andrew M Childs, Aaron Ostrander, and Yuan Su. Faster quantum simulation by randomization. *Quantum*, 3:182, 2019.

[19] Minh C Tran, Su-Kuan Chu, Yuan Su, Andrew M Childs, and Alexey V Gorshkov. Destructive error interference in product-formula lattice simulation. *Physical Review Letters*, 124(22):220502, 2020.

[20] Andrew Tranter, Peter J Love, Florian Mintert, Nathan Wiebe, and Peter V Coveney. Ordering of trotterization: Impact on errors in quantum simulation of electronic structure. *Entropy*, 21(12):1218, 2019.

[21] Andrew Tranter, Peter J Love, Florian Mintert, and Peter V Coveney. A comparison of the bravyi–kitaev and jordan–wigner transformations for the quantum simulation of quantum chemistry. *Journal of chemical theory and computation*, 14(11):5617–5630, 2018.

[22] Matthew B Hastings, Dave Wecker, Bela Bauer, and Matthias Troyer. Improving quantum algorithms for quantum chemistry. *arXiv preprint arXiv:1403.1539*, 2014.

[23] Naomichi Hatano and Masuo Suzuki. Finding exponential product formulas of higher orders. In Arnab Das and Bikas Chakrabarti, editors, *Quantum Annealing and Related Optimization Methods*, pages 37–67. Springer, Berlin, 2005.

[24] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, 2016.

[25] Ewout van den Berg and Kristan Temme. Circuit optimization of hamiltonian simulation by simultaneous diagonalization of pauli clusters. *Quantum*, 4:322, 2020.

[26] Pranav Gokhale, Olivia Angiuli, Yongshan Ding, Kaiwen Gui, Teague Tomesh, Martin Suchara, Margaret Martonosi, and Frederic T Chong. $O(N^3)$ Measurement Cost for Variational Quantum Eigensolver on Molecular Hamiltonians. *IEEE Transactions on Quantum Engineering*, 1:1–24, 2020.

[27] Vladyslav Verteletskyi, Tzu-Ching Yen, and Artur F Izmaylov. Measurement optimization in the variational quantum eigensolver using a minimum clique cover. *The Journal of Chemical Physics*, 152(12):124114, 2020.

[28] Tzu-Ching Yen, Vladyslav Verteletskyi, and Artur F Izmaylov. Measuring all compatible operators in one series of single-qubit measurements using unitary transformations. *Journal of Chemical Theory and Computation*, 16(4):2400–2409, 2020.

[29] Peter Linstrom. NIST Chemistry WebBook, NIST Standard Reference Database 69, 1997.

[30] Jarrod R McClean, Nicholas C Rubin, Kevin J Sung, Ian D Kivlichan, Xavier Bonet-Monroig, Yudong Cao, Chengyu Dai, E Schuyler Fried, Craig Gidney, Brendan Gimby, et al. Openfermion: the electronic structure package for quantum computers. *Quantum Science and Technology*, 5(3):034014, 2020.

[31] US Department of Energy. DOE INCITE program, 2020. https://www.doeleadershipcomputing.org/awardees/.

[32] Pascual Jordan and Eugene P Wigner. About the pauli exclusion principle. *Z. Phys.*, 47:631–651, 1928.

[33] Sergey B Bravyi and Alexei Yu Kitaev. Fermionic quantum computation. *Annals of Physics*, 298(1):210–226, 2002.

[34] Andrew M Childs, Yuan Su, Minh C Tran, Nathan Wiebe, and Shuchen Zhu. Theory of trotter error with commutator scaling. *Physical Review X*, 11(1):011020, 2021.

[35] David Poulin, Matthew B Hastings, Dave Wecker, Nathan Wiebe, Andrew C Doherty, and Matthias Troyer. The trotter step size required for accurate quantum simulation of quantum chemistry. *arXiv preprint arXiv:1406.4920*, 2014.

[36] Sadegh Raeisi, Nathan Wiebe, and Barry C Sanders. Quantum-circuit design for efficient simulations of many-body quantum dynamics. *New Journal of Physics*, 14(10):103017, 2012.

[37] James D Whitfield, Jacob Biamonte, and Alán Aspuru-Guzik. Simulation of electronic structure hamiltonians using quantum computers. *Molecular Physics*, 109(5):735–750, 2011.

[38] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, USA, 10th edition, 2011.

[39] Mahabubul Alam, Abdullah Ash-Saki, and Swaroop Ghosh. Circuit compilation methodologies for quantum approximate optimization algorithm. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 215–228. IEEE, 2020.

[40] Toshinari Itoko, Rudy Raymond, Takashi Imamichi, Atsushi Matsuo, and Andrew W Cross. Quantum circuit compilers using gate commutation rules. In *Proceedings of the 24th Asia and South Pacific Design Automation Conference*, pages 191–196, 2019.

[41] Anthony W Knapp. *Basic real analysis*. Springer Science & Business Media, 2005.

[42] John Day Dollard and Charles N. Friedman. *Product Integration with Application to Differential Equations*. Cambridge University Press, USA, 1st edition, 1979.

[43] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[44] Ravi Boppana and Magnús M Halldórsson. Approximating maximum independent sets by excluding subgraphs. *BIT Numerical Mathematics*, 32(2):180–196, 1992.

[45] Frank Thomson Leighton. A graph coloring algorithm for large scheduling problems. *Journal of research of the national bureau of standards*, 84(6):489–506, 1979.

[46] Michel Planat and Metod Saniga. On the pauli graphs of n-qudits. *arXiv preprint quant-ph/0701211*, 2007.

[47] Shantanu Debnath, Norbert M Linke, Caroline Figgatt, Kevin A Landsman, Kevin Wright, and Christopher Monroe. Demonstration of a small programmable quantum computer with atomic qubits. *Nature*, 536(7614):63, 2016.

[48] Norbert M Linke, Dmitri Maslov, Martin Roetteler, Shantanu Debnath, Caroline Figgatt, Kevin A Landsman, Kenneth Wright, and Christopher Monroe. Experimental comparison of two quantum computing architectures. *Proceedings of the National Academy of Sciences*, 114(13):3305–3310, 2017.

[49] Sartaj Sahni and Teofilo Gonzalez. P-complete approximation problems. *Journal of the ACM (JACM)*, 23(3):555–565, 1976.

[50] Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Carnegie-Mellon Univ Pittsburgh PA Management Sciences Research Group, 1976.

[51] Kyle Genova and David P Williamson. An experimental evaluation of the best-of-many Christofides' algorithm for the traveling salesman problem. *Algorithmica*, 78(4):1109–1130, 2017.

[52] Dorit Aharonov, Alexei Kitaev, and Noam Nisan. Quantum circuits with mixed states. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 20–30, 1998.

[53] Alexei Gilchrist, Nathan K Langford, and Michael A Nielsen. Distance measures to compare real and ideal quantum processes. *Physical Review A*, 71(6):062310, 2005.

[54] Earl Campbell. Random compiler for fast hamiltonian simulation. *Physical review letters*, 123(7):070503, 2019.

[55] John Watrous. *The theory of quantum information*, chapter 3.3. Cambridge university press, 2018.

[56] Robin Harper, Steven T Flammia, and Joel J Wallman. Efficient learning of quantum noise. *Nature Physics*, pages 1–5, 2020.

[57] Viacheslav P. Belavkin, Giacomo Mauro D'Ariano, and Maxim Raginsky. Operational distance and fidelity for quantum channels. *Journal of Mathematical Physics*, 46(6):062106, 2005.

[58] Qiskit hellinger fidelity, 2020. Available at https://qiskit.org/documentation/stubs/qiskit.quantum_info.hellinger_fidelity.html#qiskit.quantum_info.hellinger_fidelity.

[59] Teague Tomesh. dqs-term-grouping. https://github.com/teaguetomesh/dqs-term-grouping, 2021.

[60] C Huerta Alderete, Shivani Singh, Nhung H Nguyen, Daiwei Zhu, Radhakrishnan Balu, Christopher Monroe, CM Chandrashekar, and Norbert M Linke. Quantum walks and dirac cellular automata on a programmable trapped-ion quantum computer. *Nature Communications*, 11(1):1–7, 2020.

[61] Oscar A Douglas-Gallardo, David Adrian Saez, Stefan Vogt-Geisse, and Esteban Vöhringer-Martinez. Electronic structure benchmark calculations of inorganic and biochemical carboxylation reactions. *Journal of computational chemistry*, 40(13):1401–1413, 2019.

[62] Alexander J McCaskey, Zachary P Parks, Jacek Jakowski, Shirley V Moore, Titus D Morris, Travis S Humble, and Raphael C Pooser. Quantum chemistry as a benchmark for near-term quantum computers. *npj Quantum Information*, 5(1):1–8, 2019.

[63] Nicola Marzari, Arash A. Mostofi, Jonathan R. Yates, Ivo Souza, and David Vanderbilt. Maximally localized wannier functions: Theory and applications. *Rev. Mod. Phys.*, 84:1419–1475, Oct 2012.

[64] Benjamin Nachman, Davide Provasoli, Wibe A de Jong, and Christian W Bauer. Quantum algorithm for high energy physics simulations. *Physical Review Letters*, 126(6):062001, 2021.

[65] Anton Robert, Panagiotis Kl Barkoutsos, Stefan Woerner, and Ivano Tavernelli. Resource-efficient quantum algorithm for protein folding. *npj Quantum Information*, 7(1):1–5, 2021.

[66] Stuart Hadfield, Zhihui Wang, Bryan O'Gorman, Eleanor G Rieffel, Davide Venturelli, and Rupak Biswas. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms*, 12(2):34, 2019.